

# Oracle® Cloud

## Tworzenie serwisów za pomocą Oracle Content Management



F26930-21  
June 2021



Oracle Cloud Tworzenie serwisów za pomocą Oracle Content Management,

F26930-21

Copyright © 2018, 2021, Oracle i/lub jej spółki powiązane.

Główny autor: Bruce Silver

Współautorzy: Jean Wilson, Sarah Bernau, Kalpana N, Ron van de Crommert

Współtwórcy: Bonnie Vaughan

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software" or "commercial computer software documentation" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

# Spis treści

## Wstęp

---

Adresaci	xvi
Ułatwienia dostępu do dokumentacji	xvi
Różnorodność i inkluzja	xvi
Powiązane źródła informacji	xvii
Konwencje	xvii

## Część I Wprowadzenie

---

### 1 Oracle Content Management — omówienie

---

Dostęp do usługi Oracle Content Management	1-1
Omówienie ról	1-2
Zarządzanie zasobami	1-2
Zarządzanie zawartością	1-3
Praca zespołowa nad zawartością	1-3
Tworzenie serwisów	1-3
Integrowanie i rozszerzanie usługi Oracle Content Management	1-4
Wprowadzenie	1-4
Migracja do Oracle Cloud Infrastructure	1-5

### 2 Tworzenie serwisów — pierwsze kroki

---

Serwisy — pierwsze kroki	2-1
Omówienie procesu tworzenia serwisu	2-2
Informacje o tworzeniu serwisów	2-4
Omówienie funkcji zarządzania serwisem	2-5

### 3 Tworzenie pierwszego serwisu internetowego

---

Przed rozpoczęciem	3-2
Etap 1: Konfigurowanie środowiska	3-2

Importowanie składnika niestandardowego "Minimal-NavMenu"	3-2
Publikowanie składnika niestandardowego "Minimal-NavMenu"	3-3
Dodawanie przykładowych obrazów	3-4
Etap 2: Konfigurowanie serwisu internetowego	3-6
Tworzenie serwisu internetowego	3-6
Edytowanie serwisu internetowego	3-7
Etap 3: Publikowanie serwisu internetowego	3-27
Dalsze działania	3-30

## Część II Tworzenie i edytowanie serwisów

---

### 4 Tworzenie serwisów

---

Tworzenie serwisów	4-1
Kopiowanie serwisów	4-3
Zarządzanie wystąpieniami o serwis	4-5
Zmiana szczegółów wystąpienia o serwis	4-6
Wyświetlanie założeń systemowych dot. występowania o serwis	4-7

### 5 Edytowanie serwisów

---

Omówienie strony "Konstruktor serwisów"	5-1
Omówienie aktualizacji serwisu	5-4
Korzystanie z aktualizacji	5-6
Edytowanie — wskazówki i porady	5-6
Korzystanie ze stylów i formatowania	5-11
Dostosowywanie ustawień serwisów	5-12
Dodawanie niestandardowych właściwości serwisu	5-13
Praca z tabelami	5-15
Wysyłanie plików serwisu	5-17

### 6 Używanie szablonów i motywów w serwisach

---

Omówienie szablonów	6-1
Tworzenie szablonu z serwisu	6-4
Zmienianie szczegółów szablonu	6-6
Zmienianie założeń systemowych dot. szablonów	6-7
Zmienianie statusu lub odbiorców szablonu	6-8
Zarządzanie szablonami	6-9
Eksportowanie i importowanie szablonów	6-13
Omówienie motywów	6-15

Zarządzanie motywami	6-16
Publikowanie motywów	6-18
<b>7</b>	<b>Zarządzanie niestandardowymi składnikami i układami</b>
Omówienie składników niestandardowych	7-1
Omówienie układów	7-3
Używanie niestandardowych składników i układów	7-5
Rejestrowanie składników odległych	7-8
Tworzenie składników lokalnych, układów, edytorów pól zawartości lub niestandardowych formularzy	7-9
Eksportowanie bądź importowanie składników lub układów	7-11
<b>8</b>	<b>Praca ze stronami serwisu</b>
Nawigacja do strony	8-1
Wyświetlanie stron	8-1
Dodawanie stron	8-3
Przenoszenie stron	8-6
Usuwanie stron	8-7
Zmiana ustawień strony	8-7
Zmienianie układu strony	8-10
Zmienianie tła lub motywu	8-12
<b>9</b>	<b>Rozmieszczanie zawartości strony</b>
Dodawanie składników i układów sekcji	9-1
Praca z zasobami i elementami zawartości	9-3
Używanie wyzwalaczy i czynności	9-4
Używanie układów sekcji poziomej	9-6
Używanie układów z dwiema i trzema kolumnami	9-7
Używanie układów sekcji pionowej	9-9
Używanie układów sekcji z kartami	9-10
Używanie układów sekcji suwaka	9-10
<b>10</b>	<b>Używanie składników wbudowanych</b>
Składniki podstawowe	10-2
Tytuły	10-2
Akapity	10-4
Zwykły tekst	10-6
Przyciski	10-7

Składniki struktury	10-9
Separatory	10-9
Rozdzielacze	10-9
Składniki multimedialne	10-10
Obrazy	10-10
Galerie	10-13
Siatka galerii	10-16
Wideo z serwisu YouTube	10-20
Wideo	10-21
Składniki "dokumenty"	10-23
Dokumenty	10-23
Listy folderów	10-24
Listy plików	10-25
Menedżer dokumentów	10-27
Biblioteka projektu	10-29
Składniki społecznościowe	10-31
Pasek społecznościowy	10-31
Facebook — Lubię to/Rekomendacja	10-33
Twitter - Udostępnij/Obserwuj	10-33
Składnik "konwersacja"	10-34
Lista konwersacji	10-36
Składniki procesu	10-37
Formularz uruchamiania procesu	10-38
Lista zadań procesu	10-40
Formularz szczegółów zadania	10-42
Elementy zawartości	10-44
Składnik "element zawartości"	10-44
Element zastępczy zawartości	10-46
Lista zawartości	10-47
Wyszukiwanie zawartości	10-53
Rekomendacje	10-55
Inne składniki	10-56
Mapy	10-57
Nagłówki	10-58
Artykuły	10-59
Obrazy z tekstem	10-61
Grupy składników	10-63
Korzystanie z usługi Cobrowse na stronie	10-64
Oracle Intelligent Advisor	10-66
Oracle Visual Builder	10-67

### 11 Zarządzanie serwisami

---

Omówienie strony "Serwisy"	11-1
Zarządzanie serwisem i ustawienia serwisów	11-3
Zmienianie opisu, loga lub właściwości osadzania serwisu	11-8
Ustawienie właściwości dotyczących wyszukiwarek	11-9
Włączanie usługi Prerender Service na potrzeby optymalizacji pod kątem wyszukiwarek (SEO)	11-11
Ustawianie opcji dostarczania serwisu statycznego	11-12
Włączanie automatycznej kompilacji w trakcie publikowania	11-13
Przesyłanie domyślnych nagłówków Cache-Control skompilowanych serwisów	11-13
Określanie mobilnych użytkowników-agentów do obsługi skompilowanych adaptacyjnych układów	11-14
Określanie i konfigurowanie zindywidualizowanych adresów URL	11-14
Włączanie integracji z usługą Cobrowse	11-17
Dodawanie śledzenia analiz	11-19

### 12 Publikowanie serwisów

---

Przełączanie serwisu do trybu online lub offline	12-1
Publikowanie zmian serwisu	12-2

### 13 Serwisy zabezpieczone

---

Omówienie zabezpieczeń serwisu	13-1
Zmienianie zabezpieczeń serwisu	13-4

### 14 Praca z serwisami wielojęzycznymi

---

Serwisy wielojęzyczne — omówienie	14-1
Tłumaczenie serwisu	14-2
Zarządzanie zleceniami tłumaczeń serwisu	14-4
Ustawienia narodowe dla tłumaczeń	14-5
Niestandardowe ustawienia narodowe dla tłumaczeń	14-5
Ustawianie aliasów ustawień narodowych dla przekierowań adresu URL	14-6

### 15 Korzystanie z przekierowań lub odwzorowań adresów URL

---

Planowanie przekierowań	15-1
Proste uzgadnianie napisów	15-1

Uproszczone uzgadnianie wieloznaczne	15-2
Dodawanie przekierowań do serwisu	15-2
Określanie, w pliku JSON, reguł przekierowywania	15-3
Wysyłanie — do serwisu — pliku z regułami przekierowania	15-9
Odwzorowywanie adresu URL serwisu	15-10

## 16 Zwiększanie wydajności serwisu

---

Używanie pamięci podręcznej w celu zwiększenia wydajności	16-1
Buforowanie w trybie wykonawczym	16-2
Buforowanie dla konstruktora serwisów	16-3
Rendering ATF (Above the Fold)	16-3

## Część IV Tworzenie składników serwisów

---

### 17 Dostosowywanie projektów i stylów

---

Projekty — informacje podstawowe	17-1
Pliki projektu	17-1
Projekt tabeli responsywnej	17-4
Dostosowywanie stylów listy konwersacji	17-8
Dostosowywanie stylów listy folderów i listy plików	17-10
Dostosowywanie ikon paska społecznościowego	17-11
Konfigurowanie rozszerzeń stosujących stylistykę do wywiadów dla Oracle Intelligent Advisor	17-12

### 18 Omówienie użycia tła

---

Tła i motywy — informacje podstawowe	18-1
W jaki sposób są implementowane tła	18-1
Gdzie są przechowywane ustawienia	18-2

### 19 Ustawianie wyzwalaczy i czynności

---

Wyzwalacze i czynności — informacje podstawowe	19-1
Ustawianie wyzwalaczy	19-2
Ustawianie czynności	19-3



## 20 Tworzenie szablonów

---

Szablony — informacje podstawowe	20-1
Podstawowa struktura szablonu	20-2
Tworzenie szablonu	20-4
Eksportowanie szablonu	20-5
Importowanie szablonu	20-6
Praca z szablonem początkowym	20-7
Tworzenie szablonu serwisu z szablonu "Bootstrap" lub "Website Design"	20-10
Tworzenie szablonów za pomocą usługi Developer Cloud Service	20-16
Logowanie do konsoli Developer Cloud Service dla Oracle Content Management	20-17
Tworzenie projektu w usłudze Developer Cloud Service	20-18
Tworzenie szablonów w usłudze Developer Cloud Service	20-18
Kopiowanie szablonu w usłudze Developer Cloud Service	20-19
Importowanie szablonu do usługi Developer Cloud Service	20-19
Scalanie zmian	20-19
Eksportowanie szablonu z usługi Developer Cloud Service	20-20

## 21 Tworzenie motywów

---

Motywy — informacje podstawowe	21-1
Podstawowa struktura motywu	21-3
Nawigacja serwisu	21-4
Tworzenie motywu	21-7
Ukrywanie składników i układów sekcji motywu	21-12
Ukrywanie opcji wyrównania, szerokości lub odstępów dla motywu	21-14
Wiązanie składników z motywami	21-15
Sites Rendering API	21-18

## 22 Tworzenie układów

---

Układy — informacje podstawowe	22-1
Optymalizacja pod kątem wyszukiwarek (SEO)	22-3
Omówienie pliku components.json i jego formatu	22-5
Dostosowywanie grup narzędzi w konstruktorze serwisów	22-9
Ograniczanie składników w gniazdach	22-12
Czynienie zawartości układu edytowalną	22-14
Tworzenie układu sekcji	22-16
Tworzenie układu sekcji obsługującego skrypty Lazy Load	22-17
Tworzenie niestandardowych układów sekcji za pomocą API	22-18
Tworzenie układów zawartości	22-25
Tworzenie układów zawartości za pomocą Oracle Content Management	22-27

Przekazywanie widoku układu do układu zawartości	22-33
Generowanie adresu URL strony ze szczegółami serwisu przy użyciu API	22-34
Tworzenie układów zawartości lokalnie za pomocą usługi Developer Cloud Service	22-35
Tworzenie układu zawartości za pomocą usługi Developer Cloud Service	22-36
Definiowanie modułu RequireJS	22-37
Konfiguracja parametru funkcji Constructor	22-38
Renderowanie układu zawartości	22-39
Edytowanie układu zawartości w szablonie Mustache	22-39
Dodawanie funkcji dynamicznej manipulacji modelem DOM	22-39
Definiowanie stylów w pliku design.css	22-40
Uzyskiwanie elementów referencyjnych	22-40
Uzyskiwanie adresu URL zasobu multimedialnego	22-40
Uaktywnianie wyzwalaczy	22-41
Przechodzenie do strony wyszukiwania za pomocą zapytania wyszukiwania	22-41
Rozwijanie makra i renderowanie tekstu formatowanego	22-42
Łącze do strony "Szczegóły"	22-42
Rozwijanie makra w zapytaniach dotyczących listy zawartości	22-43
Tworzenie odpornych układów zawartości	22-46
Renderowanie elementów zawartości	22-47
Standaryzacja struktury danych dla układu zawartości	22-48
Tworzenie przykładowego szablonu bloga	22-52
Dodawanie odwzorowań układu zawartości do szablonów	22-52
Testowanie układów zawartości przy użyciu lokalnego zestawu testów	22-53
Testowanie przy użyciu lokalnego zestawu testów	22-53
Importowanie szablonów z układami zawartości do Oracle Content Management	22-53

## 23 Tworzenie składników

---

Składniki — informacje podstawowe	23-1
Tworzenie składników — informacje podstawowe	23-2
Tworzenie składnika	23-5
Tworzenie składników niestandardowych za pomocą usługi Developer Cloud Service	23-6
Tworzenie składnika niestandardowego dla Oracle Content Management	23-7
Tworzenie składnika niestandardowego	23-7
Pisanie i uruchamianie testów jednostkowych	23-9
Optymalizowanie (pomniejszanie) składników dla lepszej wydajności	23-10
Uruchamianie zleceń integracji ciągłej	23-11
Tworzenie możliwych do przetłumaczenia składników dla serwisów wielojęzycznych	23-11
Tworzenie składnika H1 za pomocą panelu "Ustawienia"	23-12
Tworzenie nowego składnika lokalnego	23-12
Tworzenie podstawowego składnika H1	23-13

Dodawanie stylów CSS dla swojego składnika	23-16
Dodawanie panelu ustawień umożliwiającego zmianę tekstu nagłówka	23-17
Aktualizacja motywu umożliwiająca użytkownikom wybór stylu składnika H1	23-19
Tworzenie prostego składnika z użyciem szablonów HTML	23-20
Tworzenie składnika HTML	23-20
Dodawanie danych, dostarczanych przez użytkownika, do składnika HTML	23-24
Porównanie składników lokalnych ze składnikami odległymi	23-28
Renderowanie ustawień składnika	23-29
Implementacja składnika lokalnego	23-30
Klasy stylów składników	23-31
Jak nadawać styl składnikom wbudowanym	23-34
Stosowanie stylów do składników — podstawy	23-34
Stosowanie stylów do określonych składników	23-36
Ustawianie właściwości składnika	23-42
Składniki renderowane w ramach wstawkowych	23-43
ID instancji i struktura składników renderowanych w ramach wstawkowych — informacje podstawowe	23-45
Zabezpieczenia dla składników odległych	23-46
Rejestrowanie składnika odległego	23-48
Usuwanie składnika	23-49
Sites SDK	23-50

## 24 Dostosowywanie pliku kontrolera

---

Plik kontrolera — informacje podstawowe	24-1
Plik domyślny kontrolera	24-1
Modyfikowanie domyślnego pliku controller.html	24-2
Obiekt SCS — informacje podstawowe	24-3
SCS.sitePrefix	24-3
SCS.preInitRendering	24-4
SCS.getDeviceInfo	24-4
Sekcje pliku kontrolera, których nie należy dostosowywać	24-4
Używanie tokenów do zapewnienia przenośności niestandardowego pliku kontrolera	24-5
Przykłady niestandardowych plików kontrolera	24-6
Zmienianie prefiksu serwisu	24-6
Dostosowywanie grafiki oczekiwania	24-7
Dostosowywanie ikon "favicon"	24-9
Dostosowywanie znaczników <noscript> i <meta> dla przeszukiwarek innych niż JavaScript	24-10
Pobieranie wstępne plików JavaScript	24-10
Weryfikowanie praw własności serwisu za pomocą dodatkowego kodu znacznikowego	24-11
Wspomaganie wykrywania urządzeń	24-11

## Część V Tworzenie aplikacji dla serwisów za pomocą innych narzędzi

---

### 25 Tworzenie za pomocą zestawu OCE Toolkit

---

Konfigurowanie zestawu OCE Toolkit na komputerze lokalnym	25-1
Instalowanie zależności za pomocą narzędzia npm	25-2
Korzystanie z narzędzia cec opartego na wierszu polecenia	25-2
Testowanie przy użyciu lokalnego zestawu testów	25-67
Uaktualnienie do jQuery 3.5.x	25-68
Tworzenie różnych elementów dla Oracle Content Management za pomocą usługi Developer Cloud Service	25-68
Korzystanie z Developer Cloud Service — informacje podstawowe	25-69
Logowanie do konsoli Developer Cloud Service dla Oracle Content Management	25-70
Tworzenie projektu w usłudze Developer Cloud Service	25-70
Tworzenie projektów Developer Cloud Service za pomocą szablonu usługi Oracle Content Management	25-71
Tworzenie projektu w usłudze Developer Cloud Service za pomocą zestawu OCE Toolkit pobranego z Oracle Content Management	25-71
Dodawanie zestawu OCE Toolkit do kodu projektu w nowym repozytorium Git	25-71
Testowanie niestandardowych składników, szablonów i układów zawartości przy użyciu lokalnego zestawu testów	25-72
Scalanie zmian	25-72
Propagacja zmian ze środowiska testowego do produkcyjnego za pomocą zestawu OCE Toolkit	25-72
Szyfrowanie hasła	25-79
Rejestrowanie serwera	25-79
Tworzenie raportu dot. użytkowania i uprawnień w zakresie serwisu	25-79
Pobieranie oraz wysyłanie dokumentów i folderów	25-80
Tworzenie serwisu z szablonu i zachowywanie tych samych identyfikatorów GUID zawartości	25-81
Tworzenie szablonu firmowego z serwisu standardowego	25-81
Importowanie i eksportowanie taksonomii	25-81
Importowanie i eksportowanie rekomendacji	25-83
Dodawanie lub usuwanie zawartości kolekcji	25-84
Tworzenie i rozwijanie niestandardowych edytorów pól za pomocą zestawu OCE Toolkit	25-84
Przenoszenie lub aktualizowanie serwisu z jednego serwera do innego	25-89
Transfer serwisu bez elementów zawartości	25-90
Grupowe pobieranie lub wysyłanie elementów zawartości serwisu	25-90
Indeksowanie stron serwisu za pomocą zestawu OCE Toolkit	25-91
Tworzenie typu zawartości dla tekstu na stronie serwisu	25-91

Tworzenie elementów zawartości indeksu strony za pomocą zestawu OCE Toolkit	25-93
Dodawanie wyszukiwania zawartości do serwisu w Oracle Content Management	25-93
Dodawanie do serwisu strony wyszukiwania	25-94
Dodawanie do motywu pola wyszukiwania	25-94
Indeksowanie serwisu wielojęzycznego za pomocą zestawu OCE Toolkit	25-95
Tworzenie uproszczonego składnika, łatwiejszego do dalszego rozwijania	25-98
Konfigurowanie usługi kompilacji serwisu	25-98
Kompilowanie serwisu w celu poprawy wydajności stron serwisu w trybie wykonawczym	25-99
Kompilowanie serwisu — omówienie	25-99
Interakcja z prerenderem	25-100
Renderowanie stron serwisu z użyciem kontrolera	25-100
Renderowanie skompilowanych stron serwisu	25-100
Kompilowanie szablonu	25-101
Konfigurowanie	25-101
Kompilowanie szablonu	25-102
Kompilowanie serwisu	25-107
Kompilowanie serwisu — wymagania wstępne	25-108
Kompilowanie serwisu	25-108
Kompilatory niestandardowe	25-109
Ograniczenia	25-110
Wykrywanie błędów kompilatorów niestandardowych	25-110
Kompilatory układów strony	25-111
Kompilatory składników	25-112
Kompilatory układów zawartości	25-113
SCSCompileAPI	25-115
Hydratacja składników	25-115
Publikowanie	25-118
Dostarczanie serwisu statycznego — kolejność	25-119
Buforowanie nagłówków	25-119
Strony szczegółów	25-120
Dodawanie elementów zawartości do kanału	25-120
Kompilowanie serwisu dla urządzeń mobilnych	25-122
Cykl życia serwisu oraz skompilowane strony	25-123
Tworzenie na serwerze Oracle Content Management nowego serwisu lub nowego zlecenia tłumaczenia zasobu	25-123
Tłumaczenie serwisu przez dostawcę usług tłumaczeniowych (LSP)	25-126
Tworzenie zlecenia tłumaczenia za pomocą zestawu OCE Toolkit	25-127
Wyświetlanie zleceń tłumaczeń	25-127
Tworzenie łącznika z tłumaczeniem	25-128
Generowanie mapy serwisu dla serwisów wielojęzycznych	25-129
Przesyłanie zlecenia tłumaczenia do dostawcy usług tłumaczeniowych (LSP)	25-130

## Część VI Dodatki

---

### 26 Samouczek: Tworzenie składników za pomocą technologii Knockout

---

Tworzenie składników za pomocą technologii Knockout — wprowadzenie i wymagania wstępne	26-1
Etap 1: Tworzenie składnika	26-2
Etap 2: Sprawdzanie struktury renderowania składnika lokalnego	26-4
Etap 3: Sprawdzanie struktury ustawień składnika lokalnego	26-7
Etap 4: Wyświetlanie nowej właściwości w składniku	26-11
Etap 5: Rejestrowanie wyzwalaczy	26-12
Etap 6: Uaktywnianie wyzwalaczy	26-14
Etap 7: Rejestrowanie czynności	26-16
Etap 8: Wykonywanie czynności	26-17
Etap 9: Tworzenie osobnego tytułu dla każdego wystąpienia składnika	26-19
Etap 10: Używanie zagnieżdżonych składników z funkcjami edycji miejscowej	26-20
Etap 11: Obsługa różnych układów	26-22
Etap 12: Definiowanie stylów niestandardowych	26-24
Etap 13: Renderowanie składnika w ramce wstawkowej	26-25
Etap 14: Używanie stylów niestandardowych, gdy składnik jest renderowany w ramce wstawkowej	26-29
Etap 15: Integracja z funkcjami "Cofnij" i "Ponów" strony	26-32
Etap 16: Zarządzanie zasobami	26-33
Samouczek — przegląd	26-38

### 27 Sites SDK — opis

---

SitesSDK.getProperty('componentAssets', callback)	27-3
SitesSDK.getProperty(propertyName, callback)	27-4
SitesSDK.getSiteProperty(propertyName, callback)	27-6
SitesSDK.setProperty('componentAssets', [assets])	27-6
SitesSDK.setProperty(propertyName, propertyValue)	27-7
SitesSDK.filePicker({options}, callback)	27-9
SitesSDK.openDocumentPicker(options)	27-10
SitesSDK.openAssetPicker(options)	27-10
SitesSDK.publish(messageType, payload)	27-11
SitesSDK.subscribe(messageType, callback)	27-11
SitesSDK.Utils.addSiteThemeDesign(cssUrl)	27-14

## 28 Rozwiązywanie problemów

---

Podczas próby utworzenia serwisu nie widać żadnych szablonów	28-1
Nie można usunąć serwisu	28-1
Nie można otworzyć drzewa serwisu ani edytować strony	28-2
Został dodany składnik, lecz nie pojawia się on na stronie	28-2
Składniki "foldery", "pliki" i "konwersacje" nie działają	28-2
Po zmianie układu strony znikła część jej zawartości	28-2
Została wysłana nowa wersja obrazu, lecz nie pokazuje się on na stronie	28-3
Dodano składnik oparty na innej usłudze, lecz ten składnik nie działa	28-3
W moim firmowym serwisie jest wyświetlane ostrzeżenie	28-3

# Wstęp

W niniejszym dokumencie opisano, jak — korzystając z usługi Oracle Content Management — tworzyć zasoby Experience, w tym serwisy, zasoby i wszystkie powiązane z nimi struktury oraz założenia systemowe, a także jak nimi zarządzać.

## Adresaci

Publikacja ta jest przeznaczona dla użytkowników systemu Oracle Cloud, którzy chcą pracować z plikami za pomocą przeglądarki internetowej, klienta synchronizacji typu Desktop lub urządzenia mobilnego.

## Ułatwienia dostępu do dokumentacji

W celu uzyskania informacji na temat zobowiązań firmy Oracle dotyczących ułatwień dostępu należy odwiedzić stronę Oracle Accessibility Program: <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

### Dostęp do Asysty Technicznej Oracle

Klienci firmy Oracle, którzy nabyli opcję asysty technicznej, posiadają dostęp do elektronicznej Asysty Technicznej w portalu "My Oracle Support". W celu uzyskania informacji należy odwiedzić stronę: <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> lub stronę dla osób z upośledzeniem słuchu: <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs>.

## Różnorodność i inkluzja

Oracle w pełni się angażuje w działania na rzecz różnorodności i inkluzji. Oracle szanuje i docenia możliwość zatrudniania różnorodnych pracowników, pozwalającą zwiększać zdolności przywódcze i innowacyjność. W ramach naszej inicjatywy tworzenia bardziej otwartej, niewykluczającej kultury, pozytywnie oddziałującej na naszych pracowników, klientów i partnerów pracujemy nad usunięciem — z naszych produktów i naszej dokumentacji — terminów nieuwzględniających wrażliwości różnych ludzi. Zdajemy sobie także sprawę z konieczności utrzymania zgodności z istniejącymi technologiami naszych klientów oraz zapewnienia ciągłości usług w miarę rozwoju zarówno oferty Oracle, jak i standardów branżowych. Ze względu na wspomniane techniczne ograniczenia nasze działania mające na celu usunięcie takich terminów są realizowane na bieżąco i ich ukończenie będzie wymagać czasu oraz współpracy z podmiotami zewnętrznymi.



## Powiązane źródła informacji

Więcej informacji można znaleźć w następujących podręcznikach i serwisach firmy Oracle:

- Oracle Public Cloud:  
<http://cloud.oracle.com>
- *What's New for Oracle Content Management*
- *Collaborating on Documents with Oracle Content Management*
- *Administering Oracle Content Management*
- *Developing with Oracle Content Management As a Headless CMS*
- *Integrating and Extending Oracle Content Management*
- *Known Issues for Oracle Content Management*
- *Getting Started with Oracle Cloud*

## Konwencje

W niniejszym dokumencie przyjęto następujące formy wyróżniania tekstu.

Konwencja	Znaczenie
<b>pogrubienie</b>	Czcionką pogrubioną są wyróżnione elementy graficznego interfejsu użytkownika powiązane z daną czynnością oraz terminy zdefiniowane w tekście lub w słowniczku.
<i>kursywa</i>	Kursywą są wyróżnione tytuły książek oraz zmienne, dla których trzeba podać konkretne wartości.
czcionka o stałej szerokości	Czcionką o stałej szerokości są wyróżnione polecenia (występujące w obrębie akapitu), adresy URL, przykładowe kody, teksty wyświetlane na ekranie oraz teksty wpisywane przez użytkownika.

# Część I

## Wprowadzenie

W tej części zamieszczono informacje koncepcyjne dotyczące tworzenia serwisów za pomocą Oracle Content Management oraz zawarto łatwy w użyciu samouczek pomagający użytkownikowi utworzyć swój pierwszy serwis. Część ta zawiera następujące rozdziały:

- [Tworzenie serwisów — pierwsze kroki](#)
- [Tworzenie pierwszego serwisu internetowego](#)

# 1

## Oracle Content Management — omówienie

Oracle Content Management to oparte na chmurze środowisko do wielokanałowego zarządzania treściami i zawartością, przyspieszające procesy obsługi. Zapewnia wydajne funkcje pracy zespołowej i zarządzania procesami Workflow, usprawniające i przyspieszające tworzenie oraz dostarczanie zawartości, zwiększając przy tym zaangażowanie się pracowników i klientów.

Korzystając z Oracle Content Management, można błyskawicznie współpracować z zespołami wewnętrznymi i zewnętrznymi, używając dowolnego urządzenia do zatwierdzania zawartości oraz tworzenia kontekstowych zasobów Experience. Gotowe narzędzia, przyjazne dla użytkowników biznesowych, sprawiają, że tworzenie nowych, internetowych zasobów Experience z interesującą zawartością jest niezwykle proste. Korzystając z tej samej platformy zawartości i tych samych procesów, można zwiększyć cyfrową współpracę wszystkich osób zainteresowanych. Techniczne i organizacyjne zatory są przeszłością, nie ma już żadnych barier w tworzeniu atrakcyjnych zasobów.

### Uwaga:

Oracle Content Management w edycji Starter zawiera ograniczony zestaw funkcji. Aby można było korzystać z pełnego zestawu funkcji, należy uaktualnić do edycji Premium.

## Dostęp do usługi Oracle Content Management

Po uzyskaniu prawa dostępu do usługi Oracle Content Management, do użytkownika jest wysyłany powitalny e-mail z adresem URL instancji i nazwą użytkownika. Informacje te są potrzebne do zalogowania się do usługi, dlatego warto zachować ten e-mail na przyszłość.

Istnieją różne sposoby interakcji z Oracle Content Management:

- Interfejs internetowy zapewnia użytkownikowi łatwy dostęp przy użyciu preferowanej przeglądarki internetowej. Za pomocą tego interfejsu można m.in. zarządzać swoją zawartością w chmurze, udostępniać pliki i foldery innym osobom, rozpoczynać konwersacje i w nich uczestniczyć oraz tworzyć serwisy (jeśli jest to dozwolone).
- Aplikacja typu Desktop zapewnia synchronizację plików i folderów, znajdujących się w chmurze, z plikami i folderami z komputera. Można synchronizować własne pliki oraz udostępnione, dzięki czemu użytkownik zawsze ma dostęp do najnowszej wersji.
- Dodatek dla pakietu Microsoft Office zapewnia dostęp do funkcji Oracle Content Management bezpośrednio z programów Microsoft Word, Microsoft Excel, Microsoft PowerPoint i Microsoft Outlook.
- Mobilne aplikacje dla systemów Android i iOS ułatwiają uzyskiwanie dostępu przy użyciu telefonu komórkowego lub innych urządzeń mobilnych. Aplikacje mobilne są dobrze znane, ponieważ wyglądają i działają, tak jak usługa w przeglądarce internetowej. Można uzyskiwać dostęp do swojej zawartości w chmurze, wyszukiwać oraz sortować pliki i foldery, udostępniać zawartość oraz korzystać z konwersacji.

- Interfejsy REST API i SDK udostępniają programistom zaawansowane narzędzia do programistycznego wdrażania funkcji Oracle Content Management w aplikacjach internetowych i aplikacjach mobilnych.

## Omówienie ról

Funkcje Oracle Content Management, do których można uzyskiwać dostęp, zależą od przypisanej roli. W zależności od posiadanej roli użytkownika są dostępne różne opcje. Użytkownicy standardowi mogą pracować z dokumentami, konwersacjami i serwisami. Użytkownicy firmowi mogą dodatkowo uzyskiwać dostęp do zasobów. Dla programistów są dostępne opcje używane do tworzenia i dostosowywania elementów serwisów internetowych, takich jak szablony, motywy, składniki i układy. Dla administratorów są widoczne opcje umożliwiające konfigurowanie usług, integrowanie usług z innymi aplikacjami biznesowymi oraz konfigurowanie repozytoriów zasobów.

W Oracle Content Management istnieją różne typy ról.

- **Role poziomu organizacji** — Role przypisywane w organizacji, decydujące o wykonywanych zadaniach oraz określające możliwość korzystania z różnych funkcji.
- **Role użytkownika** — Decydują o tym, które funkcje użytkownik widzi w Oracle Content Management.
- **Role (uprawnienia) poziomu zasobów** — To, co można zobaczyć i zrobić z zasobem (takim jak dokument, element zawartości, serwis czy szablon) zależy od roli przypisanej użytkownikowi podczas udostępniania mu tego zasobu.

Więcej informacji...

## Zarządzanie zasobami

Oracle Content Management oferuje zaawansowane funkcje do zarządzania wszystkimi zasobami, używanymi w różnych kanałach, w tym w serwisach internetowych, materiałach marketingowych, kampaniach e-mailowych, sklepach online i blogach. Udostępnia centrum zawartości dla wszystkich zasobów, w którym można organizować zasoby w repozytoriach i kolekcjach oraz tworzyć reguły definiujące sposób używania tych zasobów.

Dostępne są także wszechstronne funkcje zarządzania oraz procesy Workflow, prowadzące przez proces tworzenia i zatwierdzania zasobów oraz zapewniające dostępność tylko autoryzowanych wersji.

Z łatwością można oznakować i filtrować zasoby, co przyspiesza ich późniejsze wyszukiwanie i używanie tam, gdzie jest to możliwe. Funkcje inteligentnej zawartości automatycznie znakują i sugerują zasoby, których należałoby użyć.

W celu określenia, jakie informacje mają być gromadzone, gdy użytkownicy tworzą zasoby, można określić typy zasobów. *Typy zasobów cyfrowych* definiują atrybuty niestandardowe, wymagane dla zasobów cyfrowych (pliki, obrazy i filmy wideo). *Typy zawartości* pogrupują różne składniki zawartości w jednostki wielokrotnego użytku. Użytkownicy — na podstawie tych typów zasobów — mogą następnie tworzyć elementy zawartości i zasoby cyfrowe, zapewniając w ten sposób spójne ich użytkowanie i prezentację w kanałach.

Więcej informacji...

## Zarządzanie zawartością

Korzystając z Oracle Content Management, można zarządzać swoją zawartością w chmurze — wszystko w jednym miejscu i dostępne z dowolnego miejsca.

Można grupować pliki w foldery i przeprowadzać operacje zarządzania plikami (kopiowanie, przenoszenie, usuwanie itp.) niemal tak samo jak w komputerze lokalnym. Ponieważ wszystkie pliki znajdują się w chmurze, użytkownik ma do nich dostęp z każdego miejsca, także przy użyciu urządzeń mobilnych. Jeśli zostanie zainstalowana aplikacja typu Desktop, cała zawartość będzie automatycznie synchronizowana z komputerem lokalnym, dzięki czemu użytkownik zawsze ma dostęp do najnowszej wersji.

Więcej informacji...

## Praca zespołowa nad zawartością

Korzystając z zawartości w chmurze, można z łatwością udostępniać pliki lub foldery do współpracy z innymi osobami — wewnątrz organizacji lub poza nią. Osoba, której zostanie udostępniona zawartość, uzyskuje dostęp do najnowszych informacji. Można nadać prawa dostępu w odniesieniu do całych folderów lub udostępnić łącza do określonych elementów. Każdy dostęp do udostępnionych elementów jest rejestrowany, dzięki czemu można monitorować, kiedy i w jaki sposób uzyskano dostęp do udostępnionego elementu.

W Oracle Content Management konwersacje są jednym ze środków współpracy z innymi osobami — pozwalają dyskutować w czasie rzeczywistym i przysyłać komentarze. Można rozpocząć autonomiczną konwersację na dowolny temat, dodając do niej (w razie potrzeby) pliki. Chcąc szybko uzyskać informacje zwrotne, można także rozpocząć konwersację dotyczącą określonego pliku, folderu, zasobu lub serwisu.

Wszystkie wiadomości, pliki i adnotacje powiązane z konwersacją zostają zachowane, dzięki czemu można z łatwością śledzić i przeglądać dyskusję. Konwersacje są przechowywane w chmurze; użytkownik może je wyświetlać i w nich uczestniczyć będąc poza biurem, korzystając ze swojego urządzenia mobilnego.

Więcej informacji...

## Tworzenie serwisów

Korzystając z Oracle Content Management, można błyskawicznie tworzyć, a następnie publikować marketingowe i społecznościowe serwisy internetowe (od koncepcji do uruchomienia), zapewniając interesujące zasoby Experience online. Proces ten jest całkowicie zintegrowany: zawartość, praca zespołowa i kreatywność łączą się wzajemnie w jednym środowisku opracowywania i publikowania.

Aby szybko rozpocząć pracę, można użyć elementów, takich jak gotowe szablony, składniki obsługujące technikę "przeciągnij i upuść", przykładowe układy stron oraz motywy stron, tworząc serwis z predefiniowanych bloków konstrukcyjnych. Programiści mogą także tworzyć niestandardowe szablony, motywy niestandardowe i składniki niestandardowe w celu zaspokojenia unikatowych potrzeb firmy.

Z łatwością można dodawać filmy wideo (z serwisu YouTube), filmy wideo wykorzystujące strumieniowanie, obrazy, nagłówki, akapity, łącza do mediów społecznościowych i inne obiekty serwisów, przeciągając składniki do odpowiednich zaprojektowanych gniazd na

stronie. Błyskawicznie można przełączać motywy serwisu i zmieniać wizerunek marki w serwisie, zapewniając zoptymalizowany, spójny wygląd.

Można pracować nad jedną lub kilkoma aktualizacjami, wyświetlać podgląd aktualizacji w serwisie, a następnie — gdy jest już gotowa — opublikować ją jednym kliknięciem.

Oracle Content Management, poza tworzeniem i publikowaniem serwisów i konstruktorze serwisów, obsługuje także tworzenie serwisów "head-less" za pomocą REST API, React JS, Node JS i innych technologii internetowych.

Więcej informacji...

## Integrowanie i rozszerzanie usługi Oracle Content Management

Oracle Content Management, będąca jedną z ofert Oracle Platform-as-a-Service (PaaS), współpracuje w sposób przezroczysty z innymi usługami Oracle Cloud.

Internetowy interfejs użytkownika można osadzić w swoich aplikacjach internetowych, dzięki czemu użytkownicy mogą bezpośrednio wchodzić w interakcję z zawartością. Za pomocą środowiska AIF (Application Integration Framework) można integrować — poprzez czynności niestandardowe — usługi i aplikacje innych podmiotów z interfejsem Oracle Content Management. Można także tworzyć łączniki z zawartością, służące do przenoszenia już utworzonej w innym miejscu zawartości do usługi Oracle Content Management, zarządzania zawartością centralnie oraz używania jej w nowych zasobach Experience w wielu kanałach.

Korzystając z bogatego zestawu narzędzi REST API i SDK, obejmujących zarządzanie zawartością i serwisami, a także dostarczanie treści i pracę zespołową, można wbudować funkcje Oracle Content Management w swoje aplikacje internetowe.

Można tworzyć aplikacje klienckie, wchodzące w interakcję z Content SDK i zasobami w chmurze. Można tworzyć integracje niestandardowe z obiektami pracy zespołowej oraz pobierać zasoby do użycia, zawsze gdy są potrzebne. Można wykorzystywać i dostarczać zawartość i zasoby zoptymalizowane pod kątem każdego z kanałów, bez względu na to, czy jest to serwis internetowy, sieć CDN (Content Delivery Network) czy aplikacja mobilna.

Więcej informacji...

## Wprowadzenie

Chcąc szybko rozpocząć pracę z usługą Oracle Content Management, warto odwiedzić [Centrum Pomocy Oracle](#), zawierające takie zasoby, jak [dokumentacja](#), [video](#), [przewodniki](#) i [informacje dla programistów](#).

Jeśli będzie potrzebna pomoc, można ją uzyskać od [Asysty Technicznej](#) i [społeczności](#).

## Migracja do Oracle Cloud Infrastructure

Jeśli posiadana subskrypcja Oracle Content Management nie działa jeszcze w OCI (Oracle Cloud Infrastructure) z konsolą Infrastructure, Oracle zaleca migrację do tego natywnego środowiska OCI. Pozwoli to w przyszłości odnosić jeszcze więcej korzyści z platformy chmurowej Oracle.

Migracja nie jest procesem automatycznym; aby go zainicjować, trzeba przesłać zlecenie SR.

Więcej informacji...

# 2

## Tworzenie serwisów — pierwsze kroki

Oracle Content Management to oparte na chmurze środowisko do wielokanałowego zarządzania treściami i zawartością, przyspieszające procesy obsługi. Korzystając z Oracle Content Management, można błyskawicznie współpracować z zespołami wewnętrznymi i zewnętrznymi, używając dowolnego urządzenia do zatwierdzania zawartości, zarządzania zasobami cyfrowymi oraz tworzenia kontekstowych zasobów Experience za pomocą narzędzi przyjaznych dla użytkownika biznesowego.

- [Oracle Content Management — omówienie](#)
- [Serwisy — pierwsze kroki](#)
- [Omówienie procesu tworzenia serwisu](#)
- [Informacje o tworzeniu serwisów](#)
- [Omówienie funkcji zarządzania serwisem](#)
- [Tworzenie pierwszego serwisu internetowego](#)



## Serwisy — pierwsze kroki

Każdy użytkownik z odpowiednimi uprawnieniami może tworzyć serwis internetowy, korzystając z usługi Oracle Content Management. Nie jest wymagane używanie żadnych innych narzędzi ani oprogramowania czy elementów kodu. Interfejs użytkownika jest interfejsem graficznym, intuicyjnym i przyjaznym.

### Uwaga:

Jeśli jest używana usługa Oracle Content Management w edycji Starter, istnieje ograniczenie tylko do jednego serwisu (brak funkcji zarządzania serwisami). Aby uzyskać dostęp do pełnego zestawu funkcji, należy uaktualnić usługę do Oracle Content Management edycji Premium.

To, kto może tworzyć serwis, zależy od kilku czynników:

- Czy administrator usługi włączył możliwość tworzenia serwisów.
- Czy administrator usługi włączył funkcję zarządzania serwisami. Zob. [Omówienie funkcji zarządzania serwisem](#)
- Czy tworzenie serwisów zostało ograniczone do administratorów serwisów (możliwość ta jest dostępna, gdy funkcja zarządzania serwisem jest wyłączona).
- Czy administrator serwisów uczynił szablony dostępnymi.

Tworząc serwis, zaczynamy od szablonu. Szablon zawiera wszystko, co jest potrzebne do rozpoczęcia pracy nad serwisem, w tym strukturę kodu serwisu, serwis domyślny



z przykładowymi stronami i zawartością, motyw ze stylami oraz zasoby (takie jak obrazy), czy nawet składniki niestandardowe. Zob. [Omówienie procesu tworzenia serwisu](#).

Jeśli będzie tworzony serwis firmowy, serwis ten zostanie powiązany z repozytorium. Ponadto dla tego serwisu trzeba zdefiniować założenie systemowe, określające język domyślny. Repozytorium i założenie systemowe dot. lokalizacji muszą zostać utworzone przed utworzeniem serwisu. W repozytorium są przechowywane zasoby i dokumenty potrzebne dla serwisu. Założenia systemowe dot. repozytorium określają, co można robić z zasobami. Szczegółowe informacje, jak korzystać z repozytoriów, są dostępne pod hasłem Omówienie repozytoriów zasobów.

Edytując serwis, tworzymy nową aktualizację albo używamy aktualizacji już istniejącej. W ramach tej aktualizacji można edytować i dodawać zawartość, dostosowywać ustawienia stylów, a także dodawać i usuwać strony, zmieniać ich układ oraz je organizować. Informacje, jak posługiwać się edytorem, są dostępne pod hasłem [Omówienie strony "Konstruuj"](#).

Nie wszystkie aktualizacje muszą zostać przygotowane jednocześnie. Wszyscy członkowie zespołu mogą pracować równoległe i niezależnie nad kilkoma aktualizacjami. Na przykład my możemy się zająć aktualizacją obejmującą cotygodniowe wiadomości, podczas gdy inny członek zespołu może dodawać strony związane ze zbliżającą się konferencją na tematy handlowe. Aktualizacje można edytować, sprawdzać i zapisywać tak często, jak jest to potrzebne; aktualizację można w dowolnej chwili scalić z serwisem podstawowym.

Gdy już jesteśmy gotowi, uruchamiamy serwis. Można go opublikować w Internecie jednym kliknięciem. Tak pokrótce wygląda cały proces od koncepcji do uruchomienia.

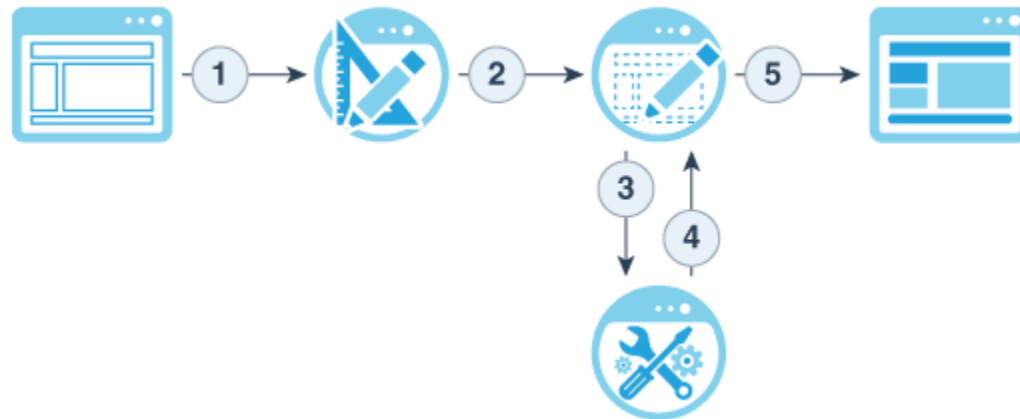
## Omówienie procesu tworzenia serwisu

Przyjrzyjmy się ogólnemu procesowi tworzenia i publikowania serwisu internetowego.

Przed rozpoczęciem pracy administrator serwisu musi włączyć możliwość tworzenia serwisów, skonfigurować, kto może tworzyć serwisy, oraz zainstalować i uczynić dostępnymi szablony, których użytkownicy będą używać do tworzenia serwisów. Administrator usługi musi przypisać bieżącemu użytkownikowi przynajmniej uprawnienie "Pobierający" w odniesieniu do szablonów. Zob. Informacje o rolach użytkownika, [Serwisy — pierwsze kroki](#) i [Omówienie funkcji zarządzania serwisem](#).

Jeśli będzie tworzony serwis firmowy, umożliwiający używanie zasobów i serwisów wielojęzycznych, serwis ten zostanie powiązany z repozytorium. W repozytorium są przechowywane zasoby i dokumenty potrzebne dla serwisu. Założenia systemowe dot. repozytorium określają, co można robić z zasobami. Repozytorium musi zostać utworzone, zanim przystąpimy do tworzenia serwisu firmowego. Zob. [Omówienie repozytoriów zasobów](#).

Poniżej został przedstawiony proces tworzenia serwisu oraz zostały podane łącza do bardziej szczegółowych informacji.



1. **Wybrać szablon i nazwę serwisu:** Jeśli funkcja zarządzania serwisem jest włączona, a serwis wymaga zatwierdzenia, trzeba — zanim będzie można wykonać następną czynność — poczekać na zatwierdzenie serwisu. Serwis ten, aż do jego opublikowania, pozostaje w trybie offline.
2. **Utworzyć serwis:** Zostanie utworzona pusta struktura, którą można dostosować.
3. **Utworzyć nową aktualizację lub użyć istniejącej:** Aktualizacja to nazwany zbiór zmian dokonywanych w bieżącym serwisie podstawowym. Za każdym razem, gdy serwis jest wyświetlany lub edytowany w edytorze, jest używana aktualizacja. Zob. [Omówienie aktualizacji serwisu](#).
4. **Otworzyć serwis w edytorze:** Korzystając z edytora obsługującego technikę "przeciągnij i upuść", można dodawać strony i zawartość do swojego serwisu. Zob. [Omówienie aktualizacji serwisu](#).
  - **Dodać strony do serwisu:** Można wybrać predefiniowany układ, aby szybko zdefiniować typ strony. Zob. [Dodawanie stron](#).
  - **Dodać składniki do strony:** Na stronę można przeciągać teksty, obrazy, dokumenty i inne elementy. Zob. [Używanie składników wbudowanych](#).
  - **Zmienić zawartość lub właściwości składnika:** Można zmodyfikować odstępy, wyrównanie i inne właściwości składnika. Zob. [Korzystanie ze stylów i formatowania](#).
  - **Opcjonalnie przełączyć układ stosowany dla strony:** Projekt strony można zmieniać na bieżąco. Zob. [Zmienianie układu strony](#).
  - **Opcjonalnie przełączyć motyw stosowany dla serwisu:** Można szybko zmienić wygląd i styl całego serwisu. Zob. [Używanie szablonów i motywów w serwisach](#).
  - **Ustawić właściwości dla wyszukiwarek:** Można ułatwić wyszukiwarkom internetowym odnajdywanie serwisu w celu zwiększenia liczby odwiedzin. Zob. [Ustawienie właściwości dotyczących wyszukiwarek](#)
5. **Zastosować aktualizację serwisu:** Można zaktualizować serwis, używając zmian zawartych w aktualizacji. Zob. [Publikowanie zmian serwisu](#).
  - **Udostępnić serwis innym członkom zespołu:** Można udostępnić serwis innym użytkownikom i przypisać każdemu z nich rolę określającą, co może z tym serwisem robić. Więcej informacji o udostępnianiu można znaleźć pod hasłem [Omówienie zabezpieczeń serwisu](#).
  - **Ustawić zabezpieczenia serwisu:** Można wybrać, którzy użytkownicy będą mieli dostęp do opublikowanego serwisu. Zob. [Zmienianie zabezpieczeń serwisu](#).

- **Opublikować serwis:** Można opublikować zmiany w serwisie. Zob. [Publikowanie zmian serwisu](#).
- **Przełączyć serwis do trybu online:** Można uczynić serwis dostępnym dla użytkowników zgodnie z jego zabezpieczeniami. Zob. [Przełączanie serwisu do trybu online lub offline](#).

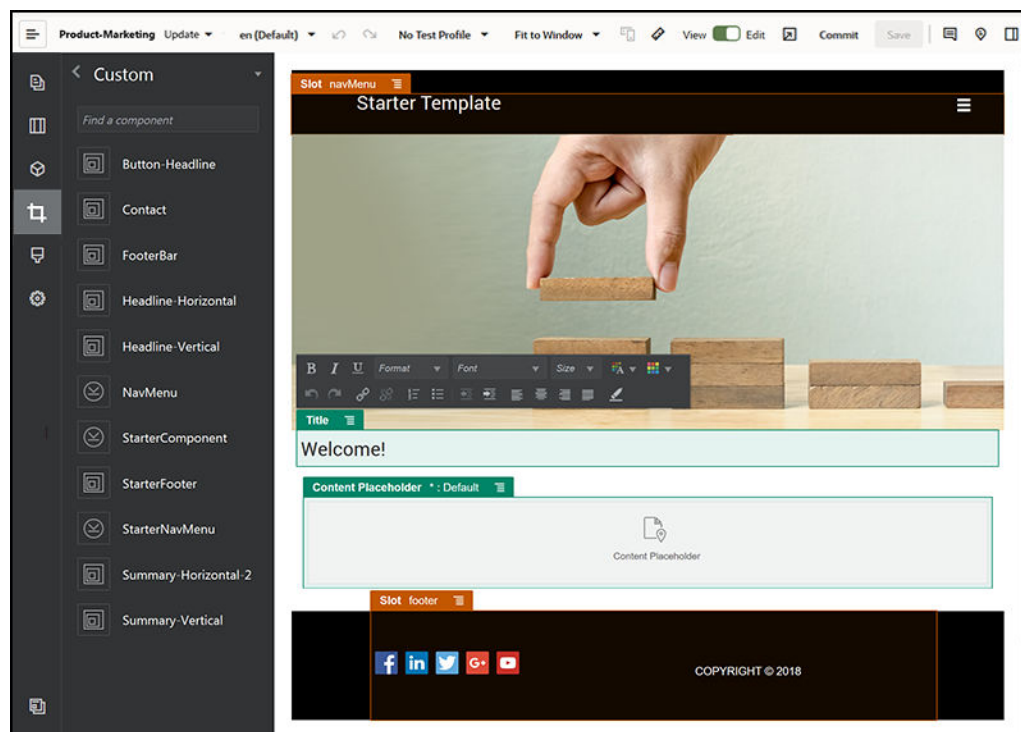
## Informacje o tworzeniu serwisów

Będąc programistą serwisu, definiujemy strukturę, która może być używana przez twórców serwisów do ich konstruowania, obejmującą:

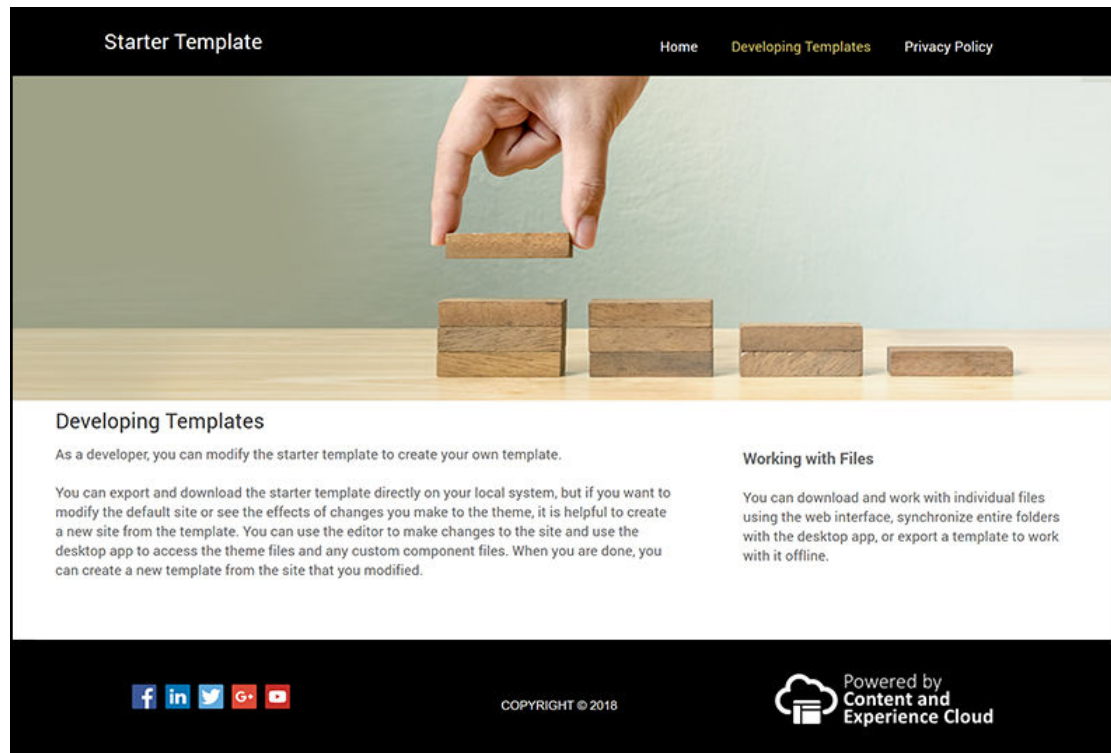
- **szablony standardowe** służące jako podstawa serwisu, oparte na motywie, wyposażone w przykładowe strony i zawartość, składniki niestandardowe oraz inne zasoby niezbędne do rozpoczęcia konstruowania serwisu
- **motywy** służące do definiowania ogólnego wyglądu i stylu serwisu, zawierające logo, znaki graficzne, arkusze stylów, pliki konfiguracyjne oraz działający w tle kod definiujący nawigację po serwisie
- **składniki niestandardowe** służące do dodawania określonych typów zawartości do stron serwisu, umożliwiające tworzenie elementów złożonych, które można osadzać na stronach serwisu przy użyciu dowolnej technologii

Separując prezentację serwisu od jego zawartości, uzyskuje się pewność, że każdy utworzony serwis jest zgodny ze standardami i elementami promującymi markę danej organizacji, zaś twórcy zawartości mogą się koncentrować na treści, co czyni proces tworzenia serwisu o wiele bardziej efektywnym.

Poniżej jest przedstawiony przykład prostego szablonu, który może zostać użyty przez twórcę zawartości podczas dodawania zawartości do serwisu. Warto zwrócić uwagę na ogólne elementy zastępcze rozmieszczone na stronie:



Poniżej jest pokazany przykład, jak po dodaniu zawartości mógłby wyglądać podgląd serwisu skonstruowanego za pomocą prostego szablonu. Elementy zastępcze zostały tu dostosowane z użyciem konkretnej zawartości i elementów nawigacyjnych:



Ponadto programiści serwisów mogą tworzyć i definiować [arkusze stylów](#), [konstruować serwisy](#) za pomocą interfejsu internetowego lub zestawu OCE Toolkit oraz konfigurować integracje między Oracle Content Management i innymi usługami. Podobnie jak inni pracownicy, mogą także współpracować z innymi, udostępniając zawartość, zaczynając konwersacje lub uczestnicząc w już prowadzonych albo używając aplikacji typu Desktop lub mobilnych.

Programiści, aby móc korzystać z Oracle Content Management, muszą mieć przypisaną rolę "użytkownik standardowy" lub "użytkownik firmowy". Programiści mający przypisaną rolę "użytkownik standardowy" mogą tworzyć składniki, motywy i szablony standardowe. Programiści mający przypisaną rolę "użytkownik firmowy" mogą także tworzyć [układy](#) oraz zapisywać serwis jako szablon standardowy lub firmowy.

Oracle Content Management — poza rozwojem i tworzeniem serwisów — może być także używana w środowisku "headless" jako zaawansowany, elastyczny system zarządzania zawartością (CMS), wykorzystywany jako system zaplecza w chmurze.

## Omówienie funkcji zarządzania serwisem

Zarządzanie serwisem ułatwia użytkownikom biznesowym tworzenie serwisów, zgodnych z zasadami obowiązującymi w firmie, oraz zapewnia administratorom serwisów łatwy sposób kontrolowania i śledzenia serwisów z centralnej lokalizacji.

 **Uwaga:**

Jeśli jest używana usługa Oracle Content Management w edycji Starter, istnieje ograniczenie tylko do jednego serwisu (brak funkcji zarządzania serwisami). Aby uzyskać dostęp do pełnego zestawu funkcji, należy uaktualnić usługę do Oracle Content Management edycji Premium.

### Problem

Współcześnie pojawił się istotny problem. Firmy muszą zapewnić odpowiednią obsługę, z użyciem wielu zasobów Experience. Zasoby te powinny być szybko tworzone i wprowadzane do środowiska wykonawczego oraz być zgodne z wizerunkiem marki i przekazywanymi wiadomościami. Firmy często stwierdzają, że tworzenie i dostarczanie nowych zasobów Experience trwa zbyt długo. Każda firma, aby odnieść sukces, potrzebuje zaawansowanej, adaptowalnej, skalowalnej i szybkiej technologii, za pomocą której będzie mogła tworzyć wiele spójnych zasobów Experience oraz zarządzać nimi, a także rozwijać inicjatywy związane z produktami i samą marką. Trzeba szybko i skutecznie udostępniać nowe zasoby Experience. W przeciwnym bowiem razie można przeoczyć wiele możliwości rynkowych oraz nie przyciągnąć w optymalny sposób klientów.

Gwałtowny przyrost zasobów Experience jest często wynikiem wielu lat nieskoordynowanego postępowania, w którego wyniku każdy dział, każda marka i każdy zespół nabywały osobno — bez centralnego nadzoru — systemy zarządzania zawartością internetową (systemy WCM). W rezultacie powstaje skomplikowane środowisko, utrudniające komercyjne działania, podatne na zagrożenia oraz pociągające za sobą spiralę nieprzewidywalnych kosztów.

Brak spójnych procesów tworzenia zasobów Experience, w powiązaniu z użyciem różnych technologii WCM, stanowi poważne zagrożenie bezpieczeństwa. Gdy różne jednostki mają swoje własne problemy z serwisem internetowym, główne obowiązki, takie jak zapewnienie bezpieczeństwa, ulegają rozdrobnieniu, co dodatkowo przyczynia się do pogorszenia odporności na zagrożenia. Wszystkie te zasoby Experience, pochodzące z różnych systemów WCM z różnymi aplikacjami internetowymi i różnym stopniem nadzoru nad infrastrukturą IT, utrudniają (a wręcz uniemożliwiają) zapewnienie właściwego poziomu bezpieczeństwa.

Dodatkowo utrzymywanie i konserwacja wielu drogich, komercyjnych systemów WCM przyczynia się do powielania kosztów — na przykład corocznych opłat za odnowienie licencji czy kosztów obsługi przez partnerów wewnętrznych lub zewnętrznych.

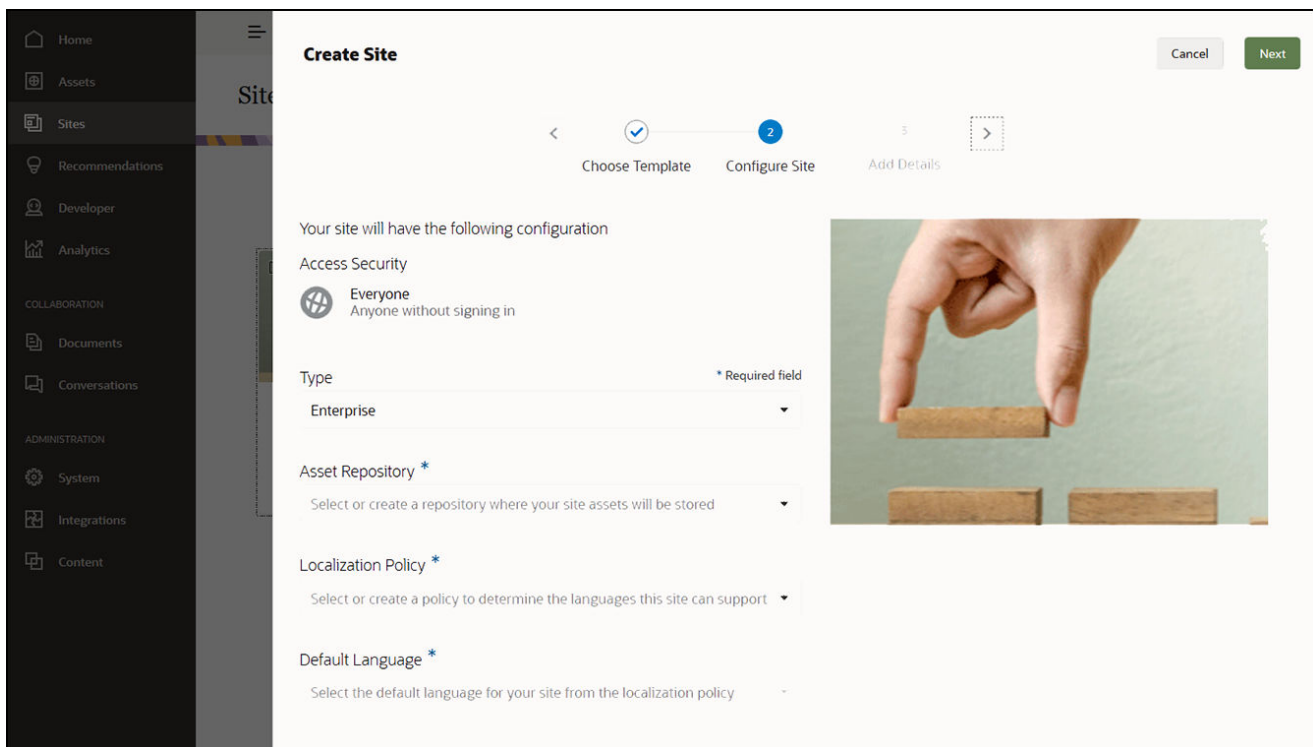
To zaś prowadzi do ograniczeń na wielu poziomach:

- Ograniczenia firmy:
  - Brak samoobsługi; zależność od działu IT lub kosztowne zlecenie zadań na zewnątrz
  - Niemożność przeprowadzania aktualizacji bez pomocy działu technicznego
  - Brak przyjaznych dla biznesu narzędzi do zarządzania pracą
  - Brak scentralizowanego wglądu we wszystkie aspekty obsługi i działania
- Ograniczenia działu IT:
  - Wymagane angażowanie działu IT do każdego zasobu Experience

- Brak funkcji zarządzania zasobami Experience utworzonymi przez firmę, w tym niemonitorowanymi, zlecanymi na zewnątrz
- Rozdrobnione stopy technologiczne, wykorzystywane do obsługi użytkowników
- Konieczność wdrażania zmian (i zarządzania nimi) w zawartości i układach setek zasobów Experience
- Ograniczenia użytkowników:
  - Niespójne przesyłanie wiadomości i komunikatów poprzez różne kanały
  - Nieaktualne informacje
  - Niska wydajność, prowadząca do porzucenia danego kanału

### Rozwiązanie

Oferowana przez Oracle Content Management funkcja zarządzania upraszcza oraz przyspiesza proces dostarczania zasobów Experience użytkownikom biznesowym, a jednocześnie ułatwia działom IT kontrolę i śledzenie tych zasobów z centralnej lokalizacji oraz pełne zarządzanie całym ich cyklem życia, zmniejszając przy tym koszty tworzenia i utrzymania każdego nowego zasobu Experience, niezbędnego dla firmy. Funkcja zarządzania jest wbudowana w rdzeń usługi Oracle Content Management — musi tylko zostać włączona przez administratora. Zob. Konfigurowanie ustawień serwisów i zasobów w podręczniku *Administering Oracle Content Management*.



### Funkcje

- **Katalog zatwierdzonych szablonów** — Programiści IT mogą dodawać do katalogu szablonów zbiory szablonów serwisów odpowiednich dla różnych linii działalności. Mogą także określić, które szablony są dostępne i dla kogo są dostępne. Mogą stosować założenia systemowe dot. szablonów w celu określenia typu zabezpieczeń, z którym

nowe serwisy muszą być zgodne, oraz ustalenia, czy nowe serwisy wymagają zatwierdzenia. Przyspiesza to i upraszcza procedurę występowania o nowe serwisy, a jednocześnie zapewnia użytkownikom biznesowym możliwość postępowania zgodnie z obowiązującymi w firmie wytycznymi dotyczącymi marki i bezpieczeństwa. Zob. [Zmianianie statusu lub odbiorców szablonu](#) i [Zmianianie założeń systemowych dot. szablonów](#).

- **Usprawnione występowanie, zatwierdzenie i wyposażenie** — Użytkownicy biznesowi zyskują możliwość błyskawicznego występowania o nowe serwisy z wymaganymi zatwierdzeniami i automatycznym wyposażeniem. Zob. [Tworzenie serwisów](#) i [Zarządzanie wystąpieniami o serwis](#).
- **Zarządzanie serwisami** — Pracownicy działu IT mogą zarządzać wszystkimi serwisami z jednego miejsca niezależnie od tego, kto utworzył serwis i wprowadził go do środowiska wykonawczego. Ponadto mogą monitorować status serwisu oraz zmieniać status każdego serwisu wprowadzonego do środowiska wykonawczego. Zob. [Zarządzanie serwisem i ustawienia serwisów](#).

### Korzyści

Funkcja zarządzania pozwala ograniczyć koszty tworzenia i utrzymywania nowych zasobów Experience.

- Korzyści dla przedsiębiorstwa:
  - Błyskawiczne wyposażenie i udostępnianie w sposób niezależny od działu IT
  - Oddzielenie zawartości od projektu; możliwość wielokrotnego użycia
  - Globalne zarządzanie zasobami Experience, użytkownikami i uprawnieniami za pomocą jednej konsoli
  - Wgląd (poprzez raporty) w operacje związane z zasobami Experience
- Korzyści dla działu IT:
  - Brak złożonych implementacji (rozwiązanie chmurowe)
  - Automatyczne skalowanie w okresie rozwoju i szczytu sezonowego
  - Funkcja zarządzania zasobami Experience, zapewniająca ich bezpieczeństwo i zgodność ze standardami oraz promocję marki
- Korzyści dla użytkowników:
  - Spójne przesyłanie wiadomości i komunikatów poprzez różne kanały
  - Aktualne informacje
  - Optymalna wydajność
  - Bezpieczne środowisko obsługi

### Proces

Poniżej są przedstawione czynności związane z tworzeniem serwisów (i zarządzaniem nimi) z użyciem funkcji zarządzania serwisem:

1. Administrator włącza funkcję zarządzania. Zob. Konfigurowanie ustawień serwisów i zasobów w podręczniku *Administering Oracle Content Management*.
2. Administrator serwisów czyni zatwierdzone szablony dostępnymi do tworzenia serwisów. Zob. [Zmianianie statusu lub odbiorców szablonu](#) i [Zmianianie założeń systemowych dot. szablonów](#).

3. Użytkownik tworzy wystąpienie o serwis z zatwierdzonego szablonu. Zob. [Tworzenie serwisów](#).
4. Administrator serwisów zatwierdza wystąpienie o serwis; serwis zostaje automatycznie utworzony. Administrator serwisów może również odrzucić wystąpienie o serwis, wyjaśniając powód odrzucenia wystąpienia. Twórca serwisu może rozwiązać problem i ponownie przesłać wystąpienie. Zob. [Zarządzanie wystąpieniami o serwis](#)
5. Twórca serwisu kontynuuje proces tworzenia serwisu, edytując i publikując serwis. Zob. [Omówienie procesu tworzenia serwisu](#).
6. Na stronie "Serwisy" można na bieżąco wprowadzać aktualizacje oraz zarządzać serwisami. Na stronie "Serwisy" administrator serwisów może zobaczyć *wszystkie* serwisy; pozostali użytkownicy mogą zobaczyć serwisy, które utworzyli lub których są współużytkownikami. Zob. [Zarządzanie serwisem i ustawienia serwisów](#).



# 3

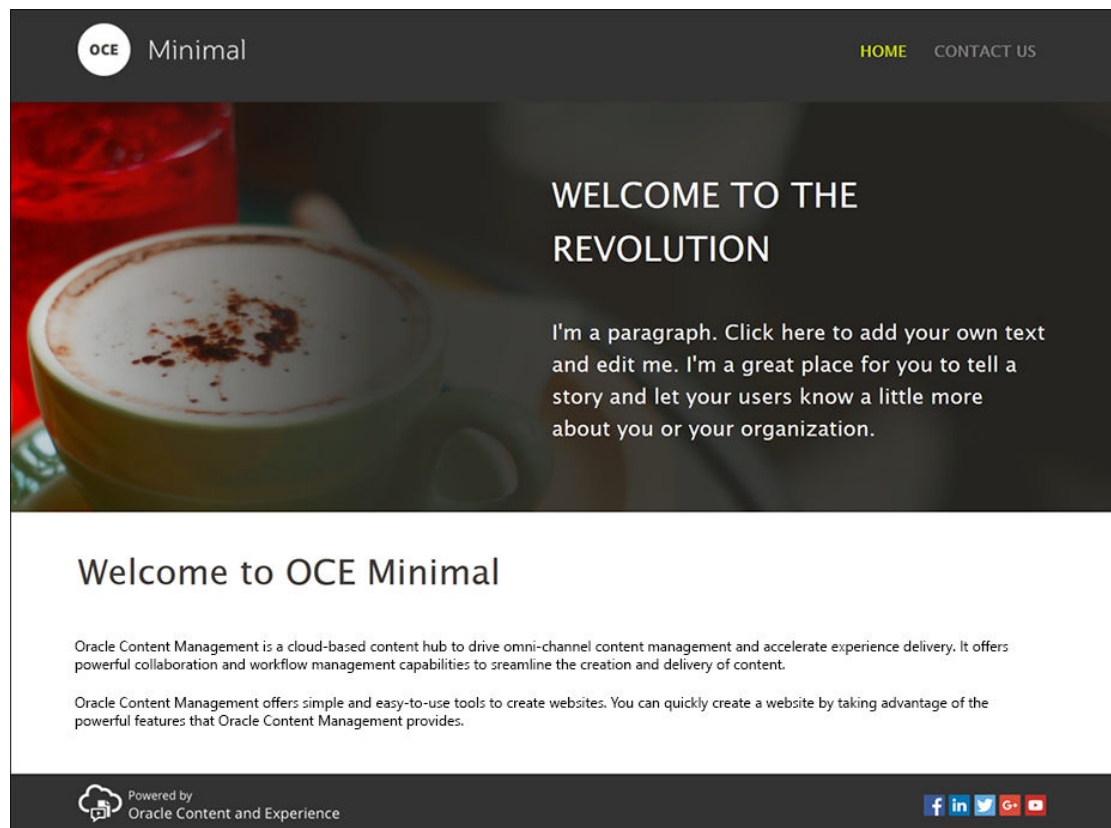
## Tworzenie pierwszego serwisu internetowego

Korzystając z Oracle Content Management, można szybko i z łatwością utworzyć swój pierwszy serwis internetowy.

W tym samouczku są omówione sposoby konfigurowania i publikowania serwisu internetowego przy użyciu gotowego szablonu o nazwie **Blank-Template**, dostępnego w usłudze Oracle Content Management. Można się przekonać, jak łatwo jest zbudować swój pierwszy serwis internetowy za pomocą konstruktora serwisów, bez konieczności programowania.

Samouczek składa się z trzech podstawowych etapów:

1. [Konfigurowanie środowiska](#)
2. [Konfigurowanie serwisu internetowego](#)
3. [Publikowanie serwisu internetowego](#)



Zanim jednak zaczniemy, sprawdźmy najpierw [kilka rzeczy](#).

## Przed rozpoczęciem

Aby można było rozpocząć pracę, wymagany jest dostęp do instancji Oracle Content Management z następującymi **rolami** poziomu aplikacji:

- **CECContentAdministrator**
- **CECDeveloperUser**

Jeśli bieżący użytkownik nie ma przypisanych powyższych ról, należy zwrócić się do administratora usługi o ich przypisanie.

Po uzyskaniu dostępu do instancji Oracle Content Management, należy się zalogować jako administrator zawartości.

Zaczynamy.

## Etap 1: Konfigurowanie środowiska

Najpierw trzeba skonfigurować środowisko, importując składnik niestandardowy "Minimal-NavMenu", a następnie wysyłając przykładowe obrazy.

W tym celu należy pobrać pakiet zasobów **OCESamplesAssetPack.zip**, dostępny na stronie <https://www.oracle.com/middleware/technologies/content-experience-downloads.html>. Następnie w pobranym pakiecie zasobów należy odszukać plik **OCECreateYourFirstSite\_**, zawierający następujące dwa pliki zip:

- **Minimal-NavMenu.zip**, który zawiera niestandardowy składnik "Minimal-NavMenu"
- **Minimal-Images.zip**, który zawiera przykładowe obrazy, używane w serwisie internetowym.

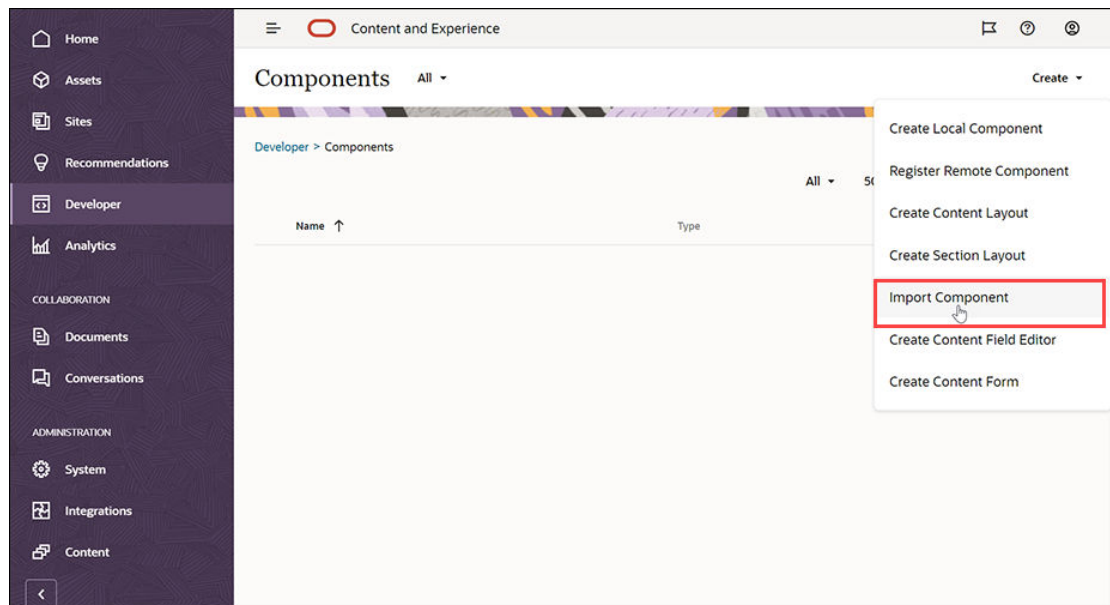
Teraz można rozpocząć konfigurowanie środowiska:

1. [Importowanie składnika niestandardowego "Minimal-NavMenu"](#)
2. [Publikowanie składnika niestandardowego "Minimal-NavMenu"](#)
3. [Dodawanie przykładowych obrazów](#)

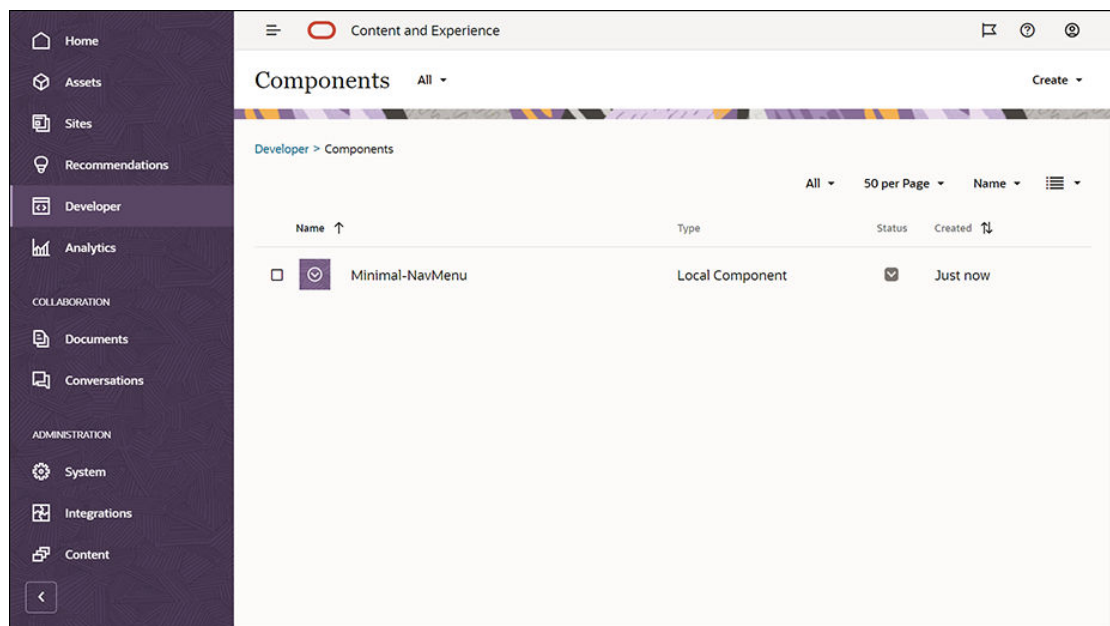
## Importowanie składnika niestandardowego "Minimal-NavMenu"

Po zalogowaniu się do interfejsu internetowego Oracle Content Management należy wybrać kolejno opcje **Programista** (lewe menu nawigacyjne) i **Wyświetl wszystkie składniki**. Jeśli nie jest widoczna opcja **Programista**, oznacza to, że bieżący użytkownik nie wymaganej **roli** użytkownika.

Na stronie "Składniki" należy nacisnąć przycisk **Utwórz**, po czym wybrać opcję **Importuj składnik**. Następnie należy wysłać plik **Minimal-NavMenu.zip** (dostępny w pliku **OCECreateYourFirstSite\_data.zip**), po czym go wybrać.



Składnik "Minimal-NavMenu" jest już wyszczególniany na stronie "Składniki".

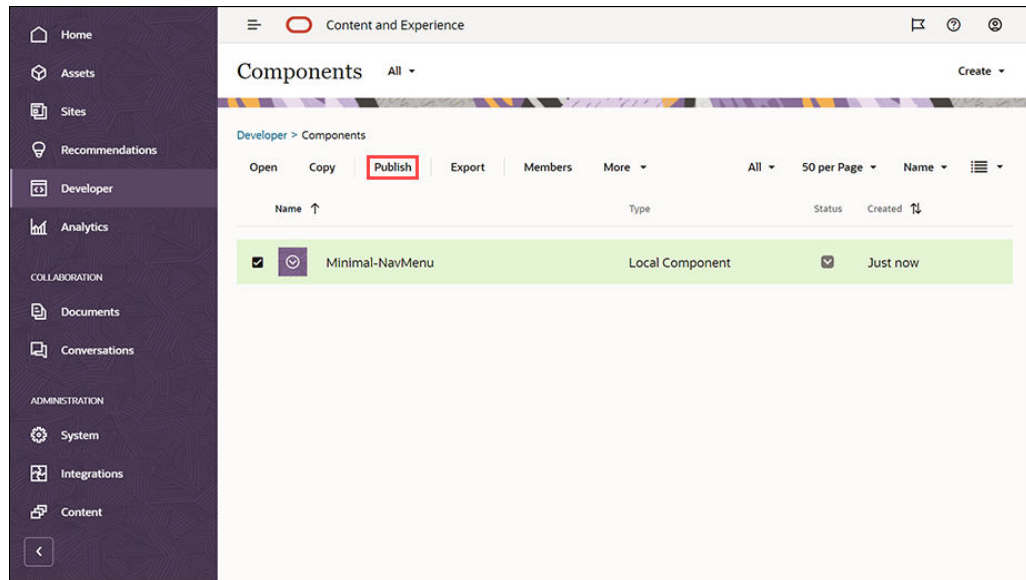


Następny etap: opublikowanie składnika niestandardowego "Minimal-NavMenu"

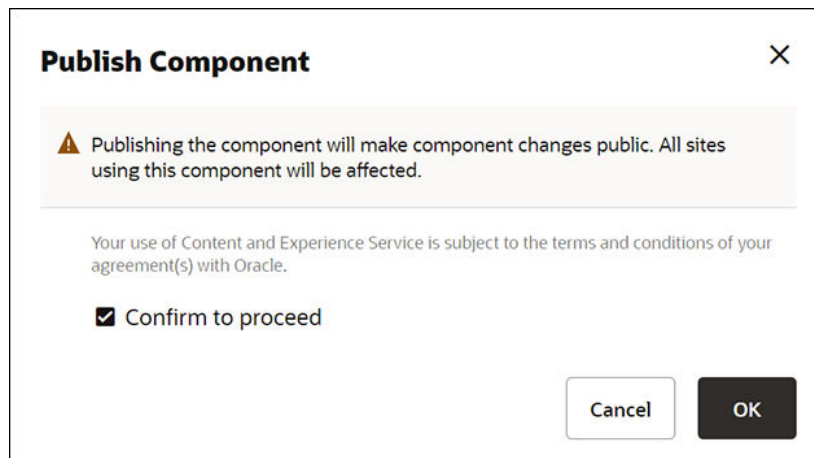
## Publikowanie składnika niestandardowego "Minimal-NavMenu"

Kolejnym etapem jest publikacja zaimportowanego składnika "Minimal-NavMenu".

Na stronie "Składniki" należy wybrać składnik "Minimal-NavMenu", po czym wybrać z paska menu lub z menu podręcznego (kliknięcie prawym przyciskiem myszy) opcję **Publikuj**.



W oknie dialogowym "Publikowanie składnika" należy wybrać opcję **Potwierdzenie kontynuowania**, po czym nacisnąć przycisk **OK**.



Po opublikowaniu składnika "Minimal-NavMenu" zostanie wyświetlone na górze strony powiadomienie informujące o publikacji składnika.

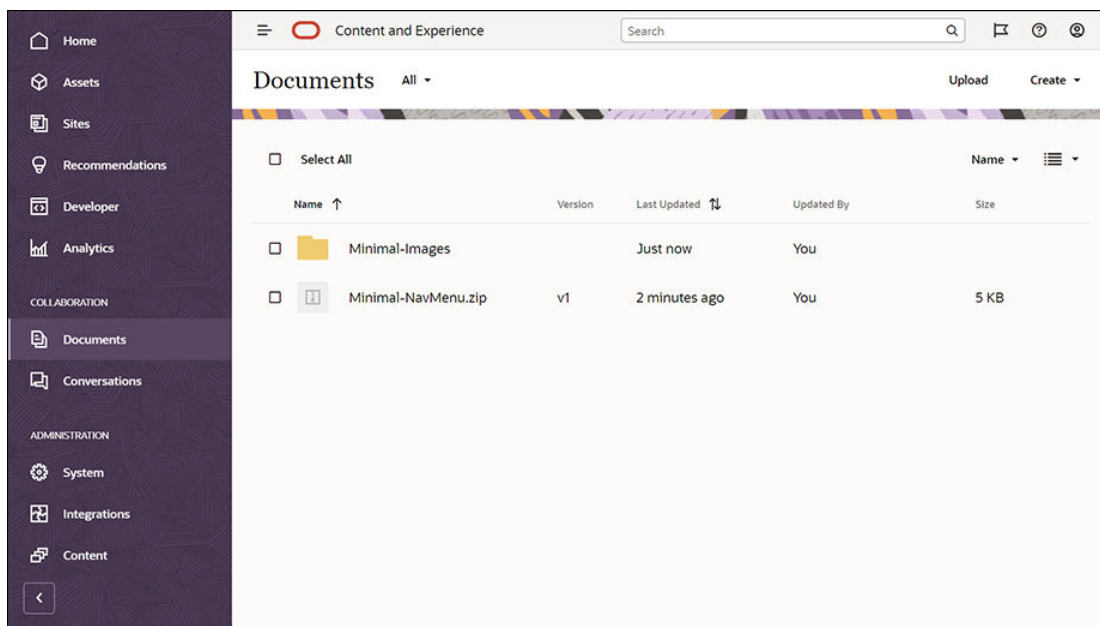
Następny etap: [dodanie przykładowych obrazów](#)

## Dodawanie przykładowych obrazów

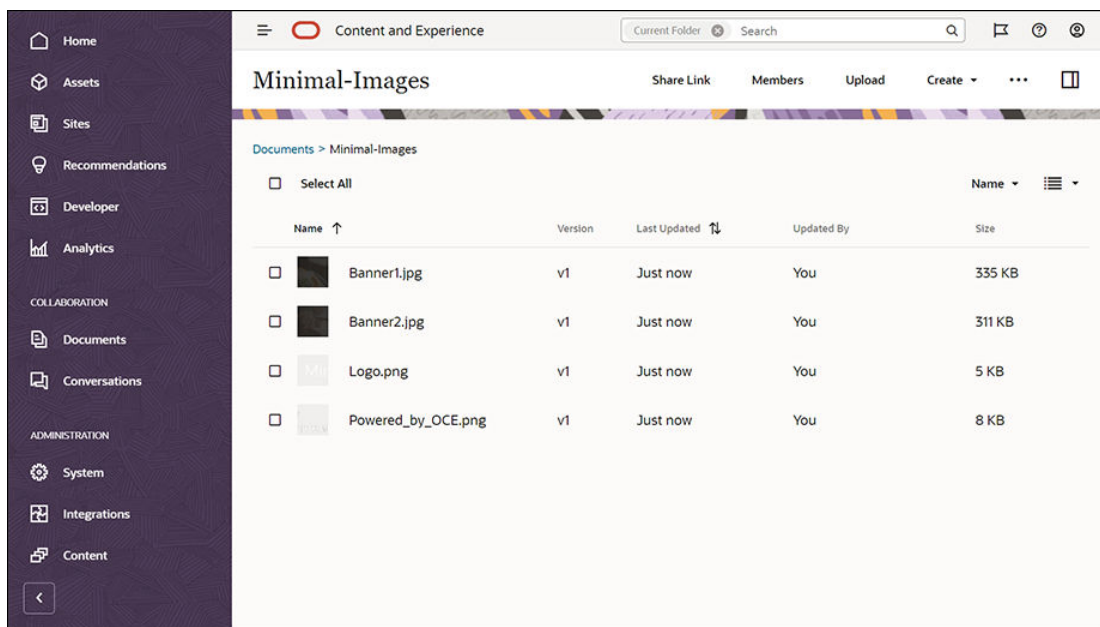
Na tym etapie trzeba dodać do folderu **Dokumenty** przykładowe obrazy (które będą używane w tworzonym serwisie internetowym).

Należy wypakować zawartość pliku **Minimal-Images.zip** (dostępny w pliku [OCECreateYourFirstSite\\_data](#)) do folderu o nazwie **Minimal-Images**, znajdującego się w komputerze lokalnym.

Aby utworzyć folder o nazwie **Minimal-Images**, należy z lewego menu nawigacyjnego wybrać kolejno opcje **Dokumenty** i **Utwórz**.



Następnie należy wysłać wszystkie obrazy z folderu **Minimal-Images** (znajdującego się w komputerze lokalnym) do nowo utworzonego folderu **Minimal-Images** w usłudze Oracle Content Management.



**Uwaga:**

Obrazy używane w szablonie i serwisie internetowym są obrazami od dostawcy Shutterstock. Jeśli obrazy te będą wielokrotnie używane, trzeba uzyskać na nie licencję od Shutterstock. Obrazy te można także zastąpić własnymi.

Następny etap: skonfigurowanie serwisu internetowego

## Etap 2: Konfigurowanie serwisu internetowego

Mając wszystkie elementy potrzebne do utworzenia serwisu internetowego, można rozpocząć jego konfigurowanie.

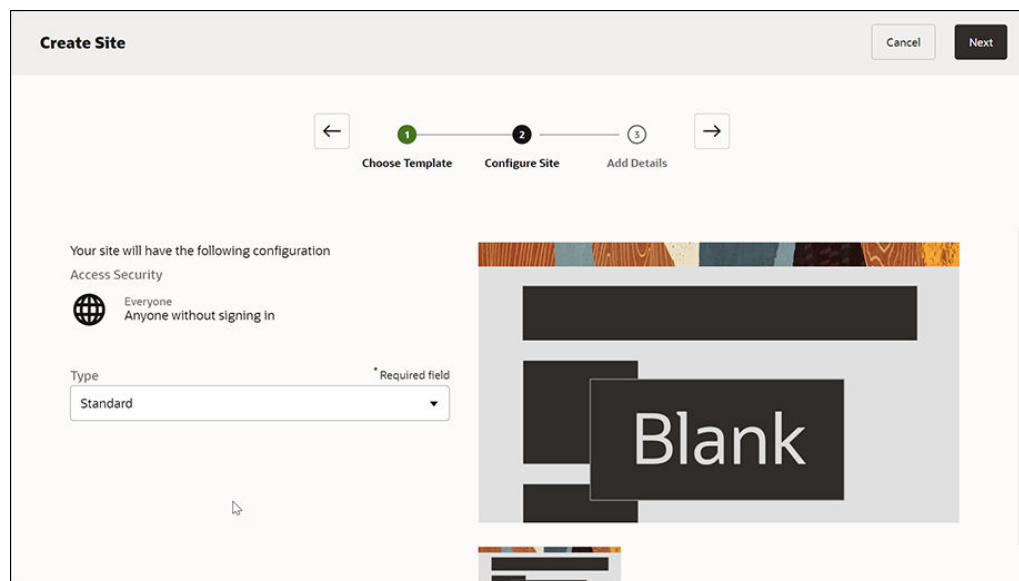
1. Tworzenie serwisu internetowego
2. Edytowanie serwisu internetowego

### Tworzenie serwisu internetowego

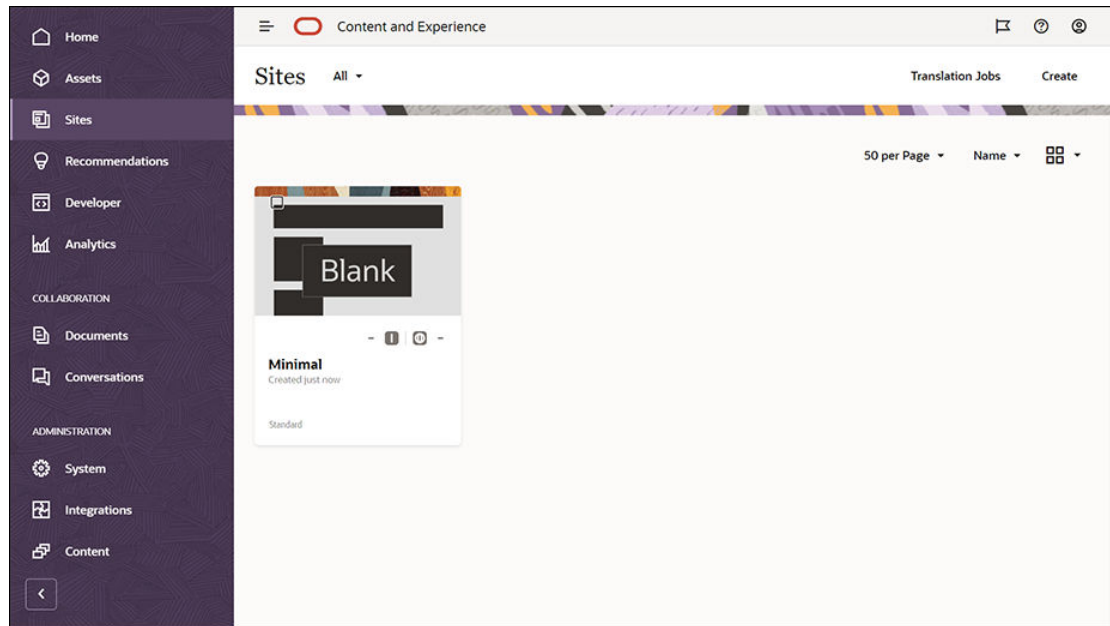
Z lewego menu nawigacyjnego należy wybrać kolejno opcje **Serwisy** i **Utwórz**. Następnie należy wybrać szablon **Blank-Template**, po czym nacisnąć przycisk **Dalej**.

Jeśli szablon **Blank-Template** nie jest widoczny, należy się skontaktować z administratorem usługi. Administrator usługi musi uczynić gotowe szablony dostępnymi do użycia. Zazwyczaj administrator usługi [instaluje gotowe szablony](#), które usługa Oracle Content Management udostępnia podczas udostępniania usługi.

Należy się upewnić, że wybrano tworzenie standardowego serwisu internetowego, a następnie naciśnięto przycisk **Dalej**. Następnie należy podać nazwę serwisu internetowego (na przykład **Minimal**). Nacisnąć przycisk **Zakończ**.



Nowy serwis internetowy został utworzony i wyszczególniony na stronie "Serwisy".

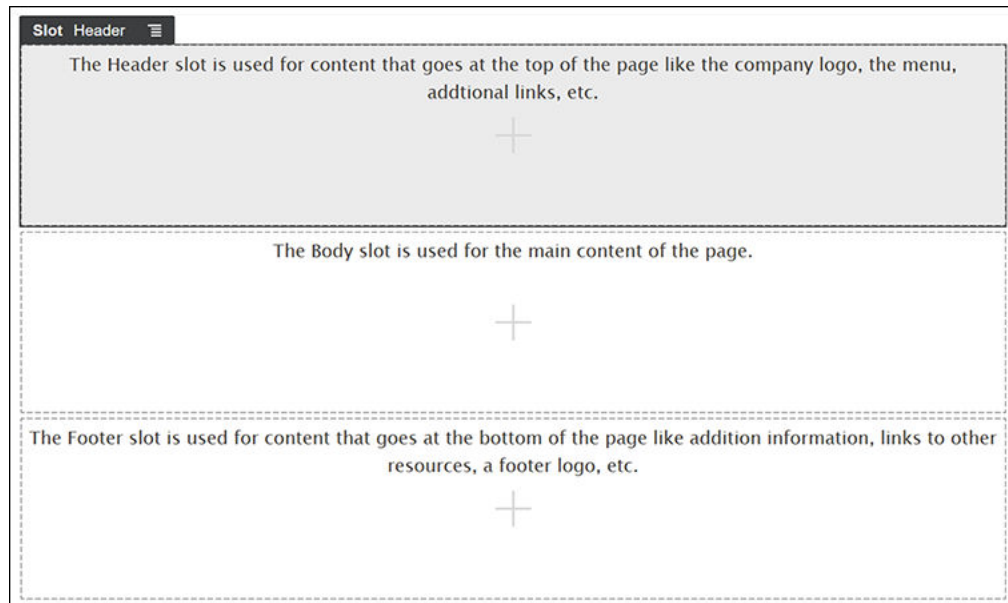


Następny etap: edycja serwisu internetowego

## Edytowanie serwisu internetowego

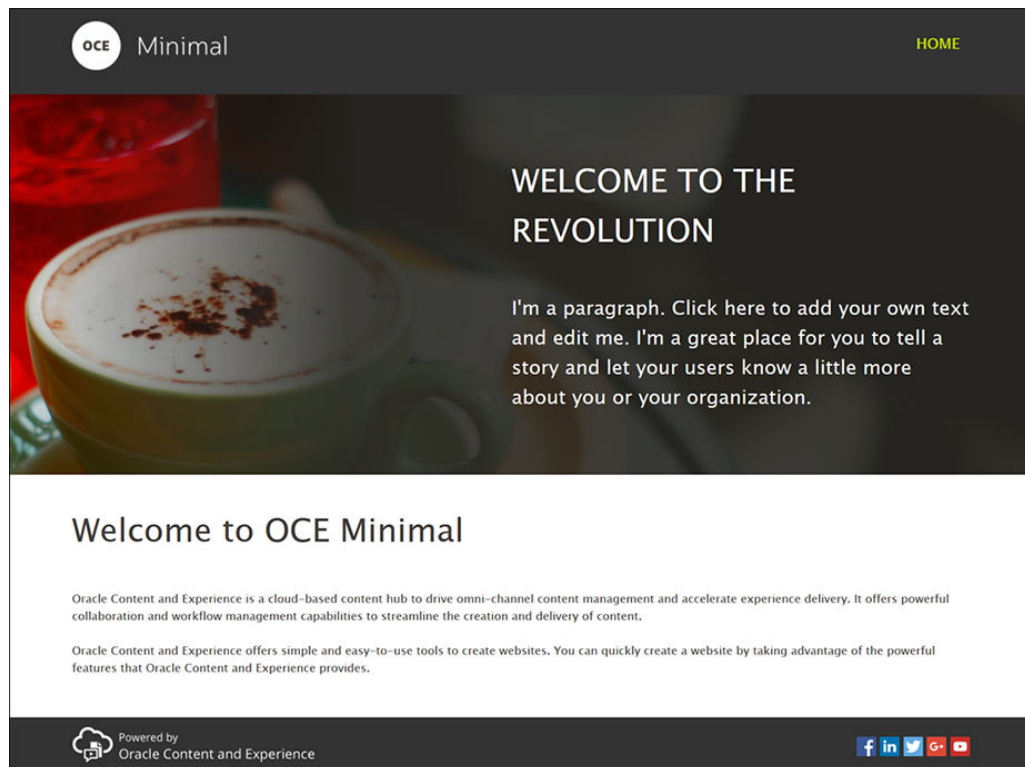
Nowo utworzony serwis internetowy należy otworzyć w konstruktorze serwisów. W tym celu należy wybrać serwis, a następnie wybrać — z paska menu lub z menu podręcznego (kliknięcie prawym przyciskiem myszy) — opcję **Otwórz**. Następnie w konstruktorze serwisów ustawić przełącznik () w pozycji **Edytuj**. Na koniec wprowadzić nazwę aktualizacji, po czym nacisnąć przycisk **OK**.

W trybie edycji można zauważyć, że serwis internetowy zawiera trzy gniazda będące obszarami dostępnymi na stronie (w zależności od układu strony). Umieszczając wskaźnik myszy na poszczególnych znakach "+" na stronie, można wyświetlić gniazda, takie jak "Nagłówek", "Treść" i "Stopka".




Gniazdo "Nagłówek" jest zazwyczaj używane do wyświetlania m.in. loga firmy i menu nawigacyjnego. Gniazdo "Treść" jest używane dla głównej zawartości strony, a gniazdo "Stopka" — dla informacji o prawach autorskich, łączy do mediów społecznościowych i innych dodatkowych informacji.

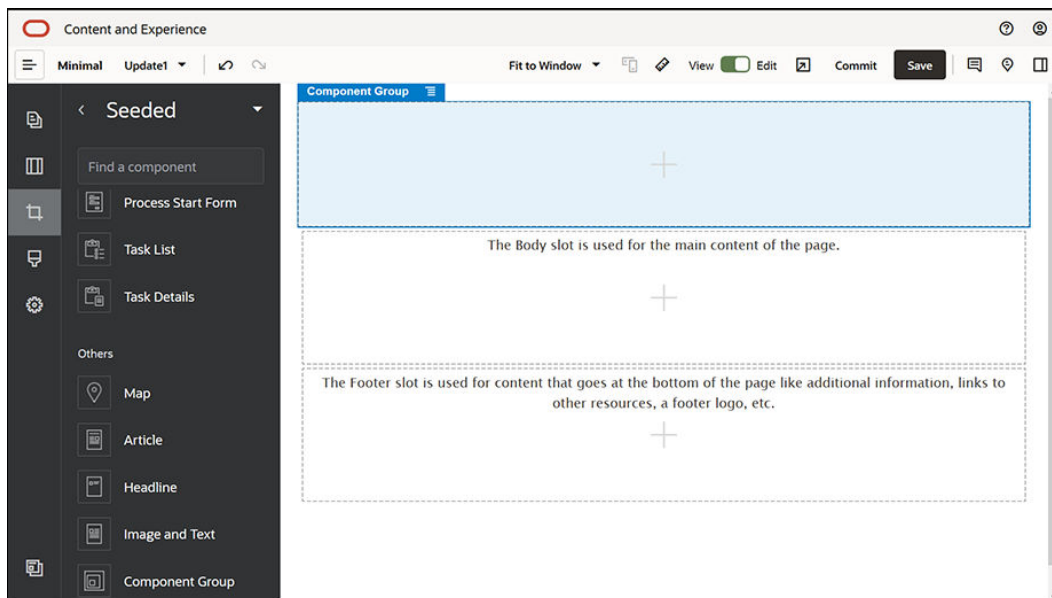
Najpierw utwórzmy stronę startową. Na poniższym rysunku pokazano, jak będzie wyglądała ukończona strona startowa:




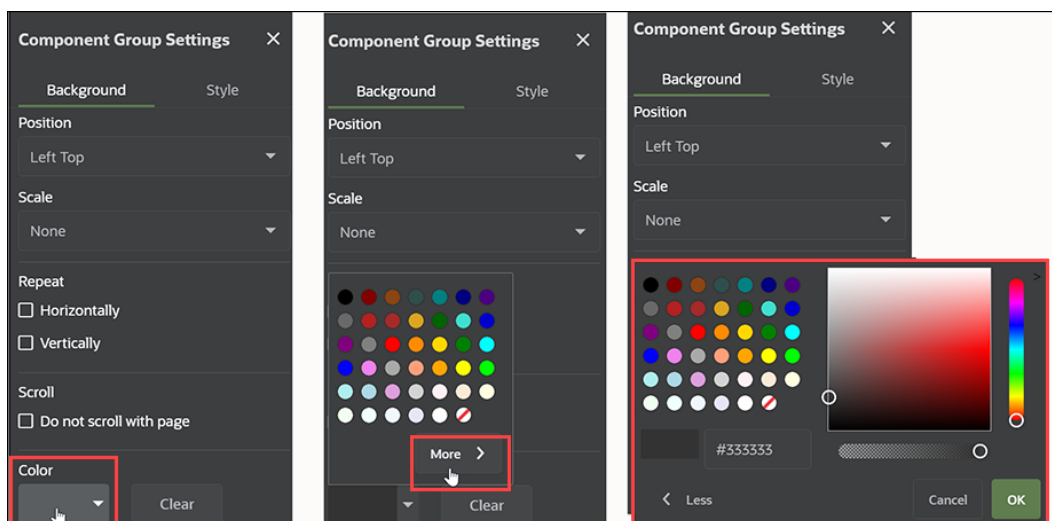
Należy użyć gotowych składników do wypełnienia gniazda "Nagłówek":



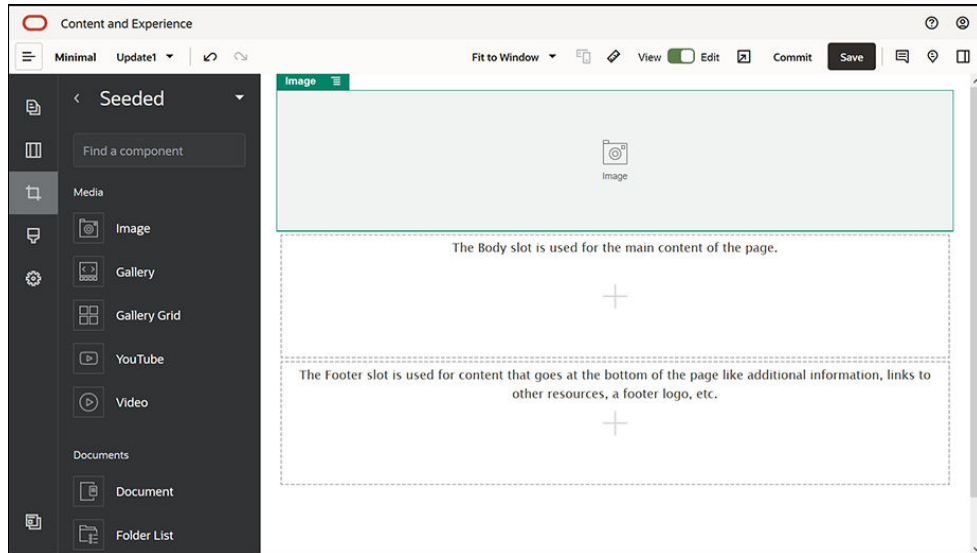
1. Aby wyświetlić listę gotowych składników, dostępnych w Oracle Content Management, kliknąć na ikonie  na lewym pasku bocznym, po czym wybrać opcję **Standardowe**.
2. Na lewym pasku bocznym wyszukać gotowy składnik o nazwie "grupa składników". Przeciągnąć go do gniazda "Nagłówek".




3. Kliknąć na ikonie menu  grupy składników, po czym wybrać opcję **Ustawienia**. W ustawieniach wybrać z rozwijanej listy opcję **Kolor** (dostępnej na dole listy ustawień), po czym nacisnąć przycisk **Więcej**. Wprowadzić wartość **#333333**, po czym nacisnąć przycisk **OK**.

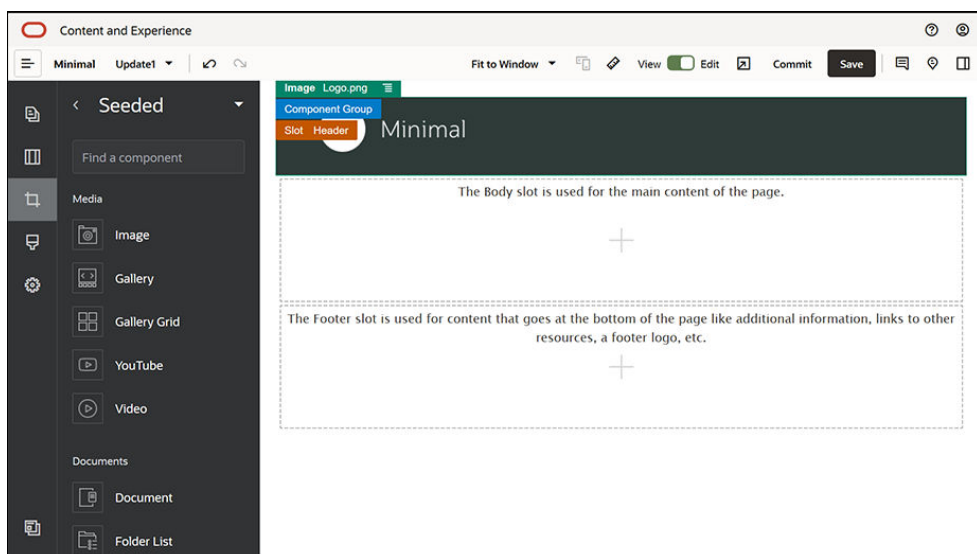


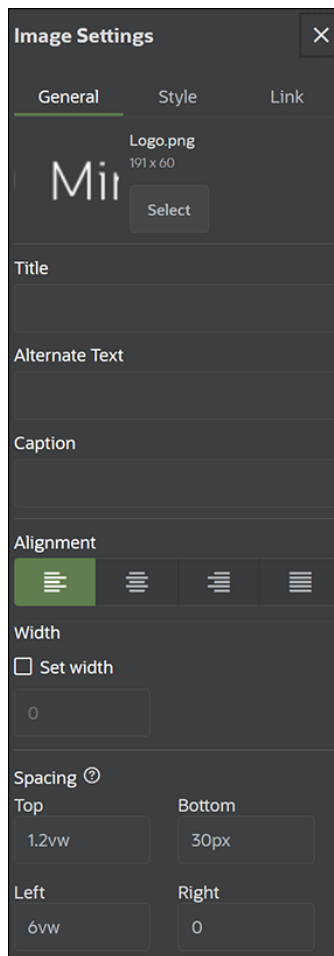
4. Przeciągnąć składnik "obraz" do grupy składników.



5. Kliknąć na ikonie menu  składnika "obraz", po czym wybrać opcję **Ustawienia**. Dokonać ustawień na karcie **Ogólne**.


Właściwość	Wartość
Wybór	Logo.png z folderu "Minimal-Images"
Wyrównanie	Z lewej
Szerokość	Anulować wybór opcji <b>Ustaw szerokość</b>
Na górze	1.2vw
Na dole	30px
Z lewej	6vw
Z prawej	0

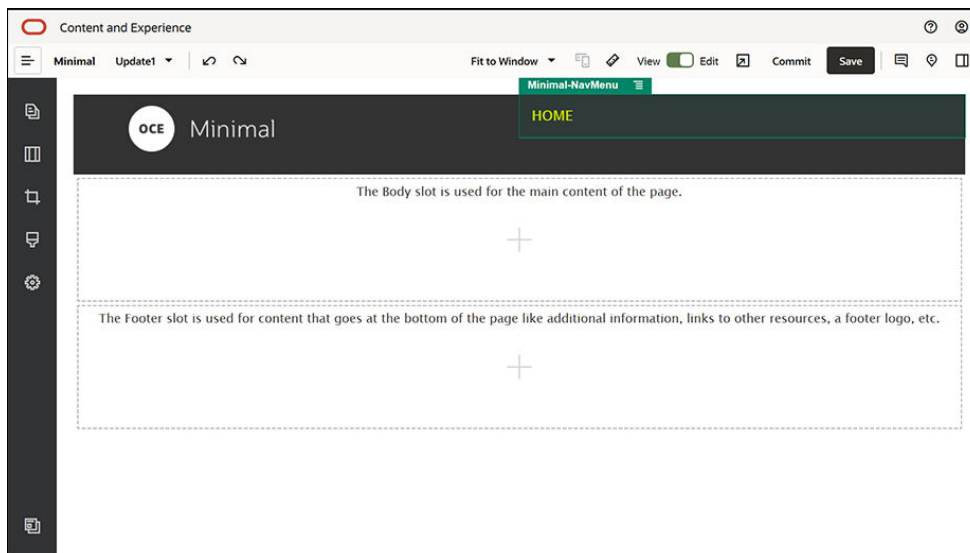




- Powiązanie obrazu logo ze stroną startową. Dokonać ustawień składnika "obraz" na karcie **Łącze**.


Właściwość	Wartość
Wybór typu łącza	Strona serwisu
Strona	HOME
Miejsce docelowe	Otwórz w tym samym oknie

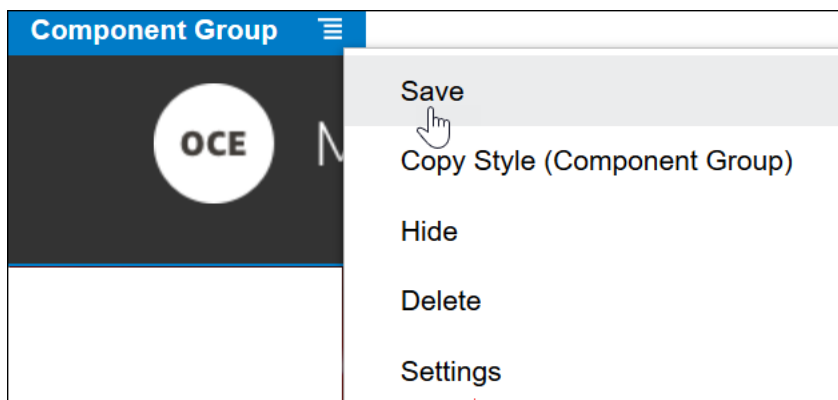
- Aby wyświetlić listę składników niestandardowych, kliknąć na ikonie  na lewym pasku bocznym, po czym nacisnąć przycisk **Niestandardowe**.
- Teraz trzeba dodać menu nawigacyjne do strony startowej przy użyciu niestandardowego składnika "Minimal-NavMenu". Przeciągnąć składnik "Minimal-NavMenu" do grupy składników i umieścić go po prawej stronie składnika "obraz". Kliknąć na tytule składnika "Minimal-NavMenu", aby upewnić się, że jego elementem nadrzędnym jest wcześniej dodana grupa składników. W ten sposób można określać, gdzie dowolny składnik znajduje się w strukturze strony internetowej.



Dokonać ustawień składnika "Minimal-NavMenu" na karcie **Ogólne**.

Właściwość	Wartość
Wyrównanie	Z prawej
Na górze	1.2vw
Na dole	0
Z lewej	0
Z prawej	6vw

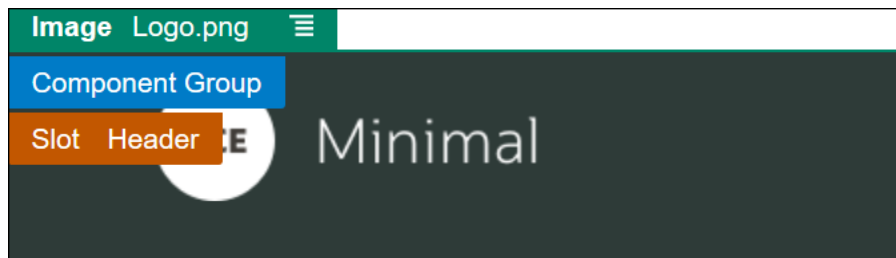
- Nagłówek jest już gotowy. Teraz trzeba zapisać tę grupę składników jako niestandardową grupę składników, dzięki czemu będzie można jej używać później w innych stronach internetowych. Kliknąć na tytule grupy składników, następnie kliknąć na jej ikonie menu , po czym nacisnąć przycisk **Zapisz**. W polu **Nazwa** okna dialogowego "Zapisz".



 **Wskazówka:**

Jeśli wskutek kliknięcia na grupie składników zostanie wyświetlony tytuł składnika "obraz" lub "Minimal-NavMenu", a nie tytuł grupy składników, należy ponownie kliknąć na tytule składnika "obraz" lub "Minimal-NavMenu". Powinien zostać wyświetlony tytuł grupy składników. Teraz można kliknąć na tytule grupy składników, a następnie wykonać powyższe etapy.



Na poniższym rysunku jest pokazywana struktura nadrzędna składnika "obraz" w gnieździe "Nagłówek".

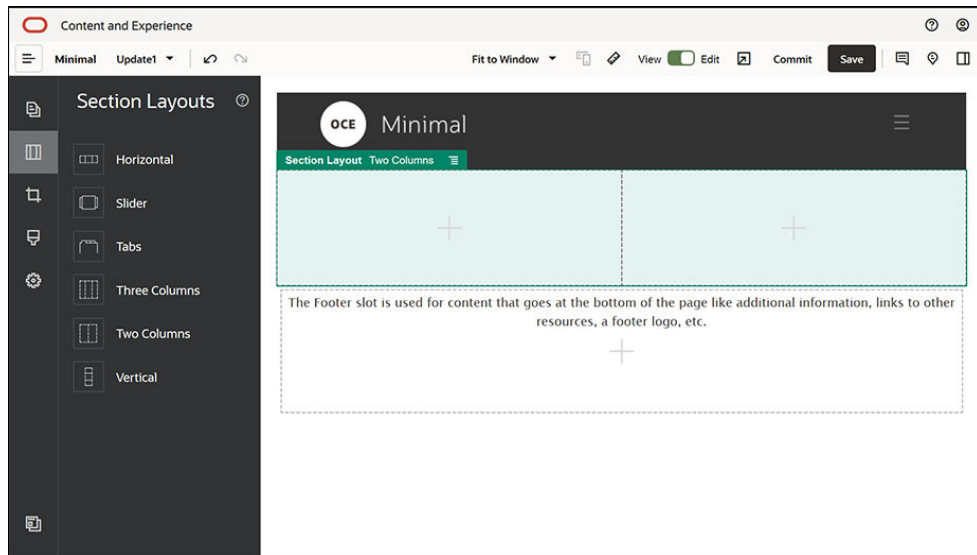


10. Aby zapisać zmiany, w prawym górnym rogu konstruktora serwisów nacisnąć przycisk **Zapisz**. Gniazdo "Nagłówek" powinno mieć teraz następującą postać:



Teraz zajmijmy się gniazdem "Treść":

1. Kliknąć na ikonie  na lewym pasku bocznym, po czym nacisnąć przycisk **Standardowe**.
2. Z lewego paska bocznego przeciągnąć składnik "grupa składników" do gniazda "Treść". Przy użyciu tej grupy składników oraz składników (które zostaną dodane do tej grupy) będzie tworzony baner.
3. Kliknąć na ikonie  na lewym pasku bocznym.
4. Przeciągnąć układ sekcji dwukolumnowy do grupy składników.




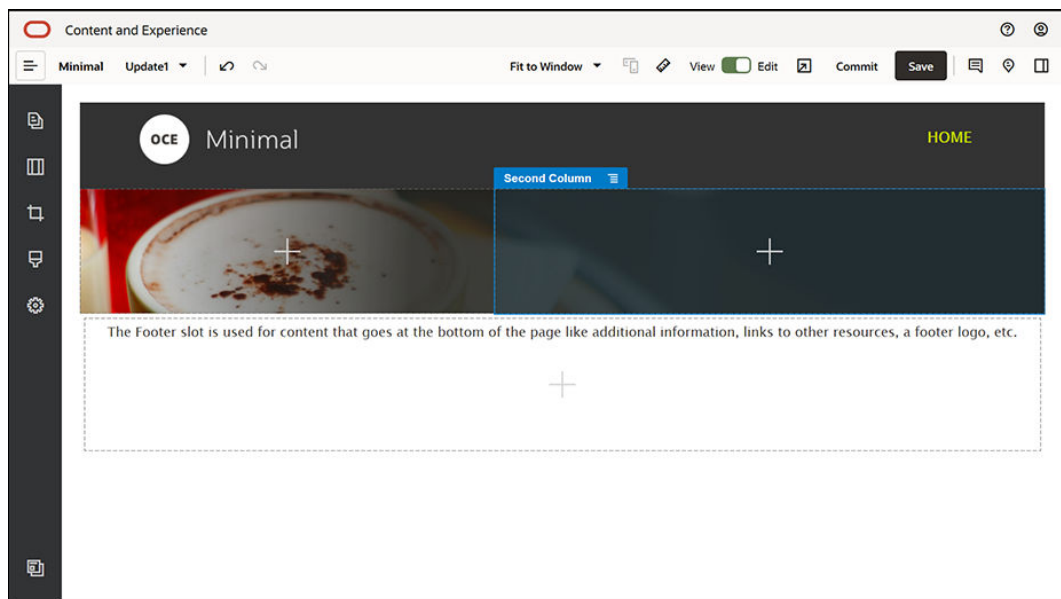
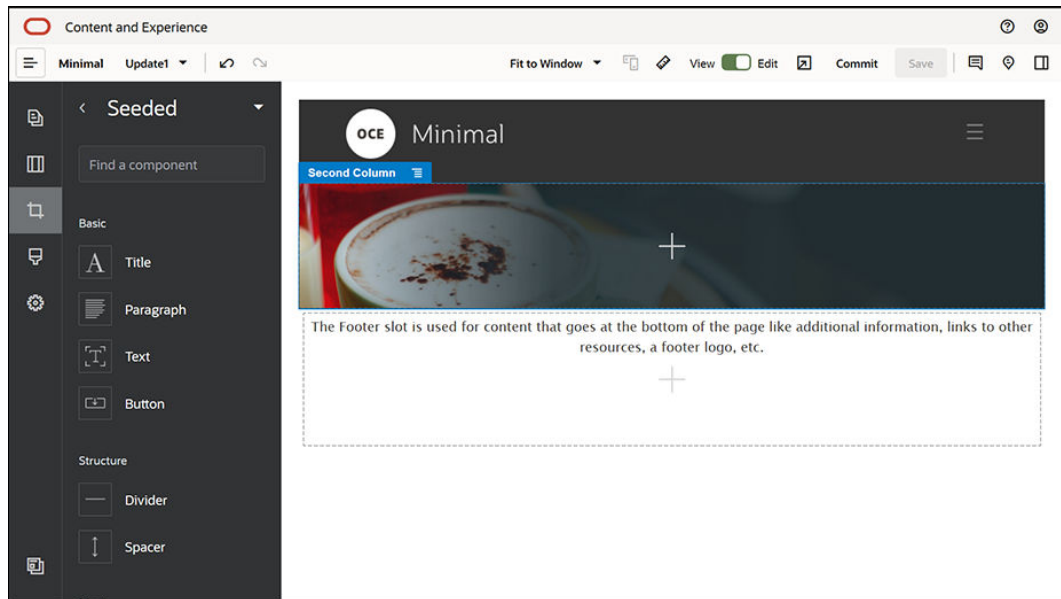
5. Dokonać ustawień układu sekcji na karcie **Ogólne**. Naciśnąć przycisk **Ustawienia niestandardowe**, po czym określić poniższe ustawienia.

Właściwość	Wartość
Szerokość pierwszej kolumny (%)	43
Szerokość drugiej kolumny (%)	57
Responsywny punkt zatrzymania (piksele)	1 023
Funkcjonowanie responsywne	Ukryj pierwszą kolumnę

Dokonać ustawień na karcie **Tło**.

Właściwość	Wartość
Obraz	Banner1.jpg z folderu "Minimal-Images"
Pozycja	Wyśrodkuj, wyśrodkuj
Skala	Rozciągnij

6. Aby wyświetlić listę składników standardowych, kliknąć na ikonie  na lewym pasku bocznym.
7. Z listy składników standardowych przeciągnąć składnik "tytuł" do drugiej kolumny układu dwukolumnowego.



8. Kliknąć na składniku "tytuł", po czym wpisać tekst "WELCOME TO THE REVOLUTION". Wybrać tekst, po czym w edytorze tekstów zmienić kolor tekstu na **biały**. Dokonać ustawień składnika "tytuł" na karcie **Ogólne**.

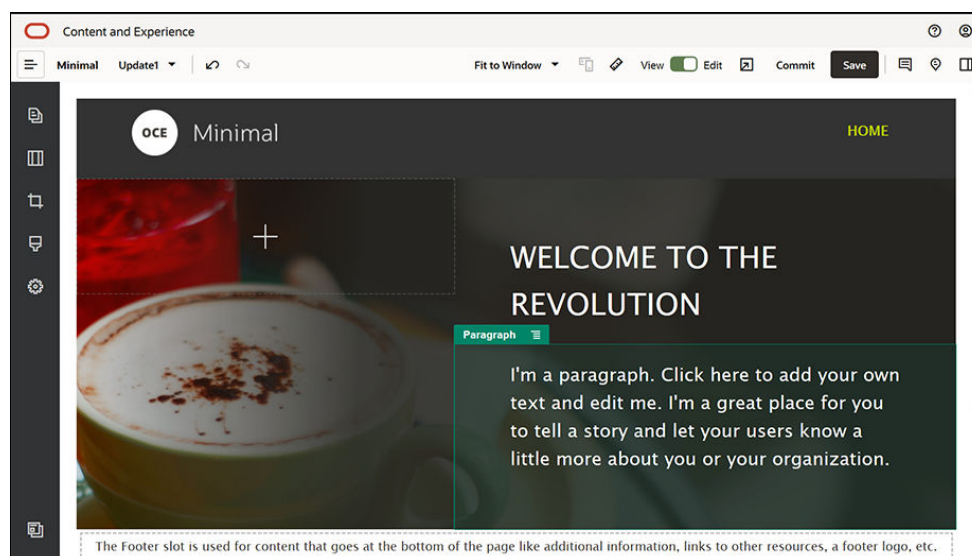
Właściwość	Wartość
Na górze	6vw
Na dole	1.8vw
Z lewej	6vw
Z prawej	6vw


9. Z listy składników standardowych (z lewego paska boczego) przeciągnąć składnik "akapit" poniżej składnika "tytuł", w obrębie drugiej kolumny układu dwukolumnowego. Dokonać ustawień składnika "akapit" na karcie **Ogólne**.

Właściwość	Wartość
Na górze	1.8vw
Na dole	6vw
Z lewej	6vw
Z prawej	6vw

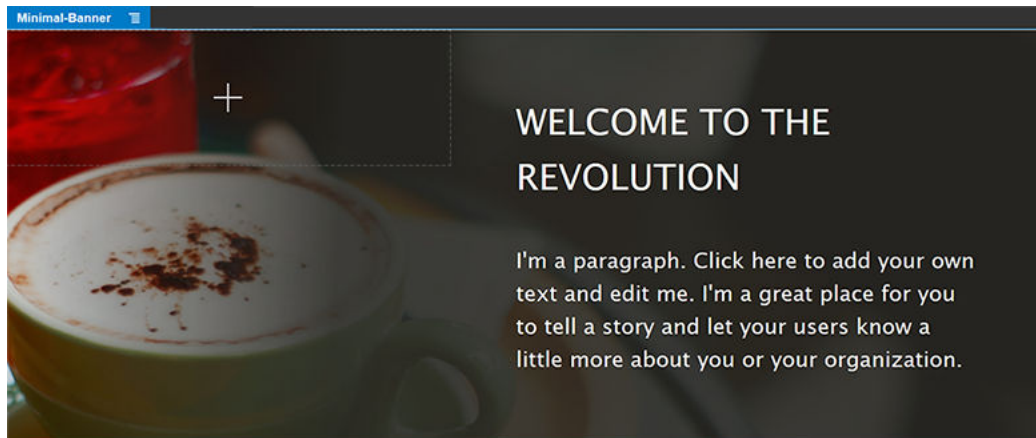
10. Kliknąć na składniku "akapit", po czym wpisać następujący tekst: "Akapit. Proszę tu kliknąć, aby dodać własny tekst i dokonać edycji. To doskonałe miejsce na opowiedzenie historii oraz umożliwienie użytkownikom poznania firmy."

Wybrać tekst, po czym w edytorze tekstów ustawić rozmiar tekstu na 24. W edytorze tekstów ustawić także kolor tekstu na **biały**.

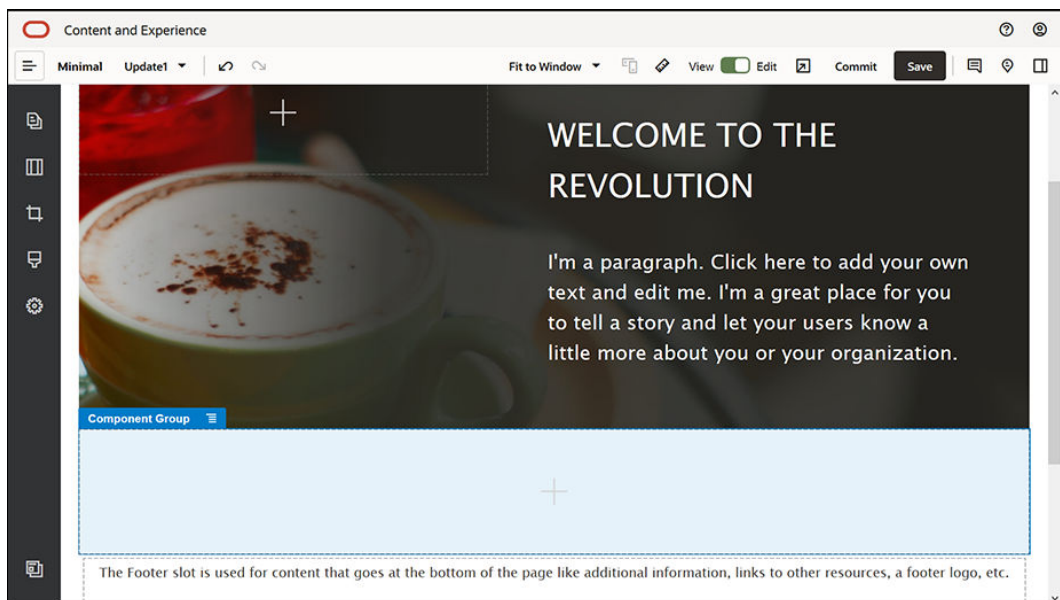


11. Baner jest już gotowy. Teraz trzeba zapisać tę grupę składników jako niestandardową grupę składników, dzięki czemu będzie można jej używać później w innych stronach internetowych. Kliknąć na ikonie menu  grupy składników, po czym nacisnąć przycisk **Zapisz**. W polu **Nazwa** okna dialogowego "Zapisywanie grupy składników" wpisać "Minimal-Banner", po czym nacisnąć przycisk **Zapisz**. Od teraz nazwa (**Minimal-Banner**) będzie wyświetlana dla grupy składników.



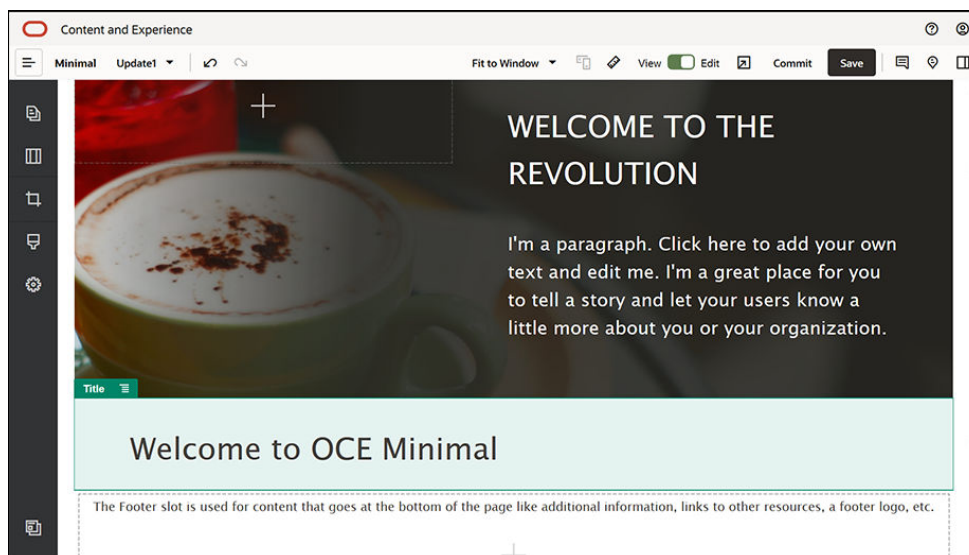


12. Z lewego paska bocznego przeciągnąć kolejny składnik "grupa składników" do gniazda "Treść", umieszczając go poniżej już dodanej grupy składników "Minimal-Banner".



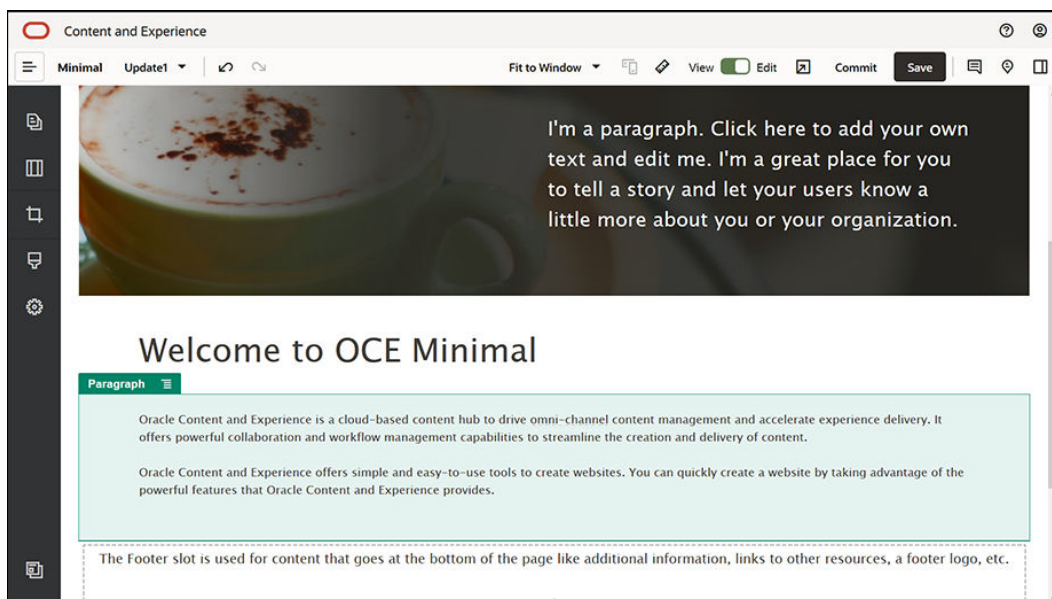
13. Przeciągnąć składnik "tytuł" do grupy składników.
14. Kliknąć na składniku "tytuł", po czym wpisać tekst "Welcome to OCE Minimal".
15. Dokonać ustawień składnika "tytuł" na karcie **Ogólne**.


Właściwość	Wartość
Na górze	3vw
Na dole	1.8vw
Z lewej	6vw
Z prawej	6vw

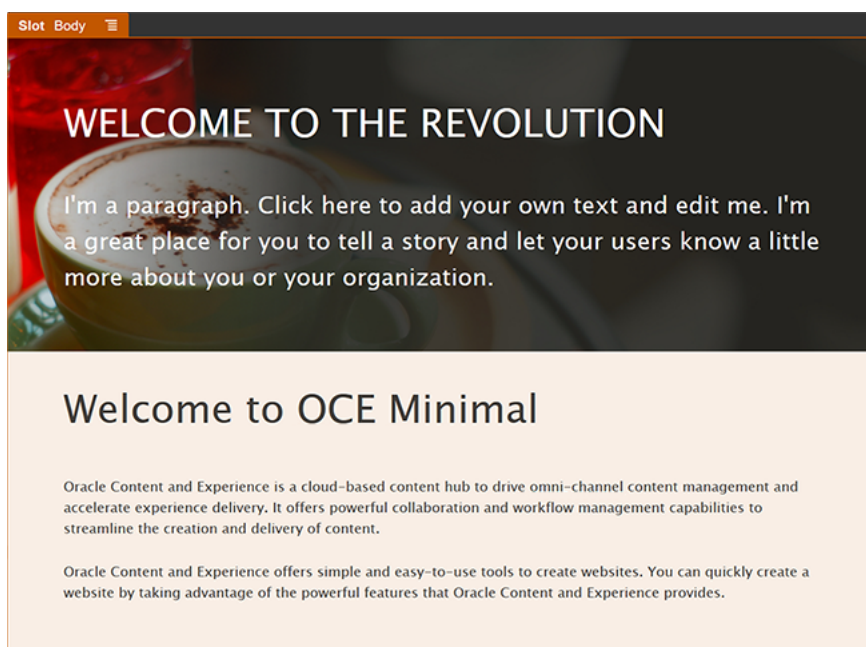


16. Z lewego paska bocznego przeciągnąć składnik "akapit" do grupy składników, umieszczając go poniżej składnika "tytuł". Kliknąć na składniku "akapit", po czym wpisać następujący tekst:
- "Oracle Content Management to oparte na chmurze środowisko do wielokanałowego zarządzania treściami i zawartością, przyspieszające procesy obsługi. Zapewnia wydajne funkcje pracy zespołowej i zarządzania procesami Workflow, usprawniające i przyspieszające tworzenie oraz dostarczanie zawartości."
- "Oracle Content Management oferuje łatwe w użyciu narzędzia do tworzenia serwisów internetowych. Można szybko utworzyć serwis internetowy, korzystając z zaawansowanych funkcji, udostępnianych przez Oracle Content Management."
17. Dokonać ustawień składnika "akapit" na karcie **Ogólne**.

Właściwość	Wartość
Na górze	20px
Na dole	50px
Z lewej	6vw
Z prawej	6vw

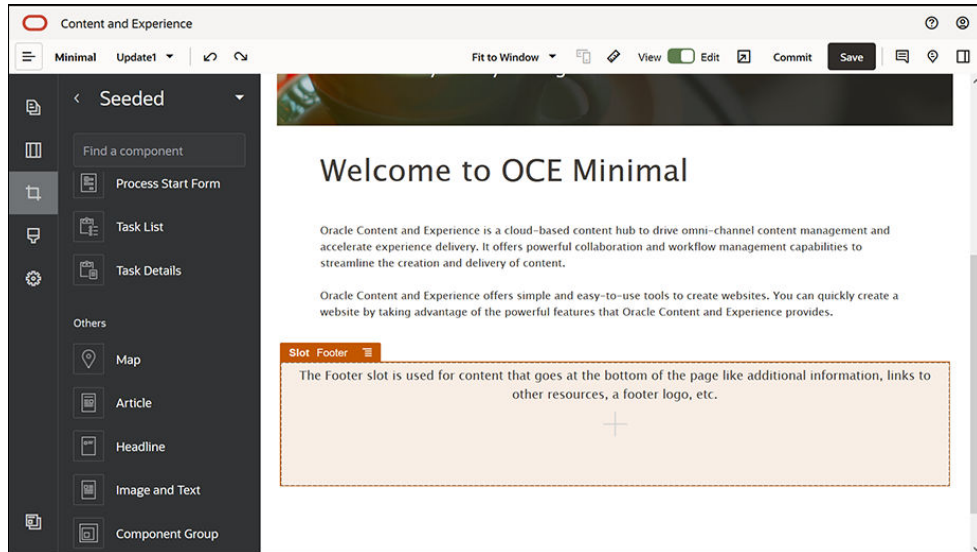


18. Gniazdo "Treść" jest już ukończone. Teraz trzeba zapisać grupę składników jako niestandardową grupę składników, dzięki czemu będzie można jej używać później w innych stronach internetowych. Kliknąć na ikonie menu  grupy składników, po czym nacisnąć przycisk **Zapisz**. W polu **Nazwa** okna dialogowego "Zapisywanie grupy składników" wpisać "Minimal-Body", po czym nacisnąć przycisk **Zapisz**.
19. Aby zapisać zmiany, w prawym górnym rogu konstruktora serwisów nacisnąć przycisk **Zapisz**. Gniazdo "Treść" powinno mieć teraz następującą postać:



Teraz zajmijmy się gniazdem "Stopka":

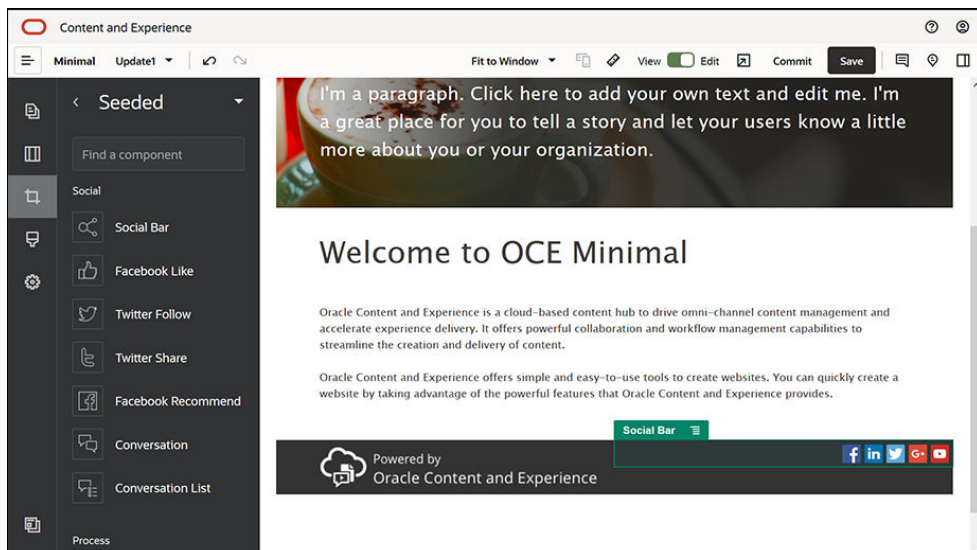
1. Z lewego paska bocznego przeciągnąć składnik "grupa składników" do gniazda "Stopka". W ustawieniach grupy składników ustawić pole **Kolor** na wartość **#333333**.



- Przeciagnąć składnik "obraz" do grupy składników, po czym dokonać ustawień tego składnika n karcie **Ogólne**.

Właściwość	Wartość
Wybór	Powered_by_OCE.png z folderu "Minimal-Images"
Wyrównanie	Z lewej
Szerokość	Anulować wybór opcji <b>Ustaw szerokość</b>
Na górze	0.9vw
Na dole	0.9vw
Z lewej	6vw
Z prawej	0

- Z lewego paska bocznego przeciagnąć składnik "pasek społecznościowy" do grupy składników i umieścić go po prawej stronie składnika "obraz".




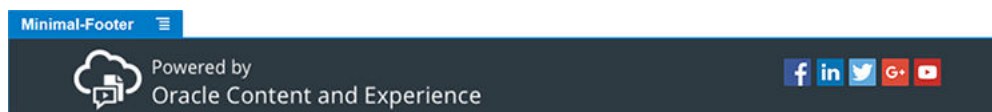
Na karcie **Ogólne** dokonać ustawień dla składnika "pasek społecznościowy".

Właściwość	Wartość
Na górze	1.8vw
Na dole	1.8vw
Z lewej	0.3vw
Z prawej	6vw


Aby dokonać ustawień, nacisnąć na kracie **Ogólne** przycisk **Ikony**, po czym kliknąć na nazwie ikony.

Właściwość	Wartość
URL	<ul style="list-style-type: none"> <li>• <a href="https://www.facebook.com/Oracle/">https://www.facebook.com/Oracle/</a> (dla serwisu Facebook)</li> <li>• <a href="https://www.linkedin.com/company/oracle/">https://www.linkedin.com/company/oracle/</a> (dla serwisu LinkedIn)</li> <li>• <a href="https://twitter.com/Oracle">https://twitter.com/Oracle</a> (dla serwisu Twitter)</li> <li>• <a href="https://www.youtube.com/oracle/">https://www.youtube.com/oracle/</a> (dla serwisu YouTube)</li> </ul>
Miejsce docelowe	Otwórz w nowym oknie

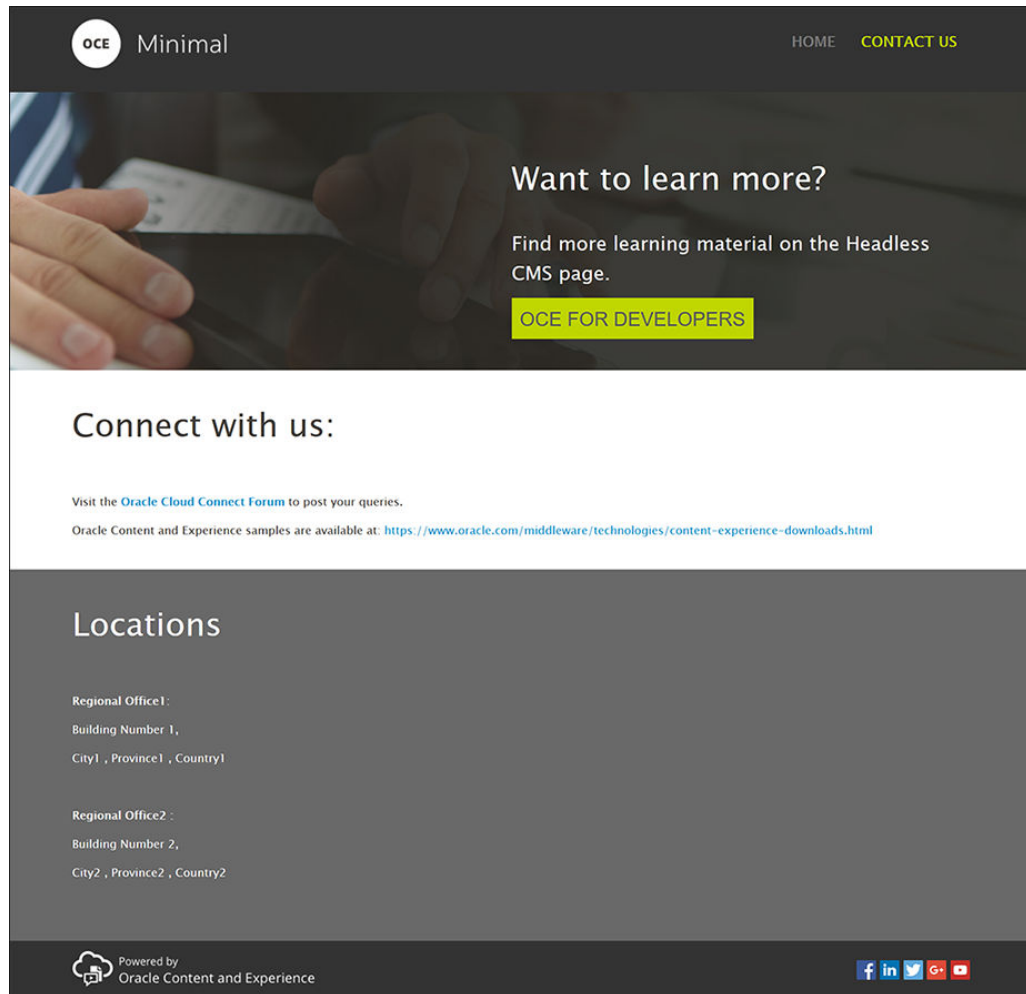
4. Stopka jest już gotowa. Teraz trzeba zapisać tę grupę składników jako niestandardową grupę składników, dzięki czemu będzie można jej używać później w innych stronach internetowych. Kliknąć na ikonie menu  grupy składników, po czym nacisnąć przycisk **Zapisz**. W polu **Nazwa** okna dialogowego "Zapisywanie grupy składników" wpisać "Minimal-Footer", po czym nacisnąć przycisk **Zapisz**. Grupa składników powinna mieć następującą postać:





5. Aby zapisać zmiany, w prawym górnym rogu konstruktora serwisów nacisnąć przycisk **Zapisz**.

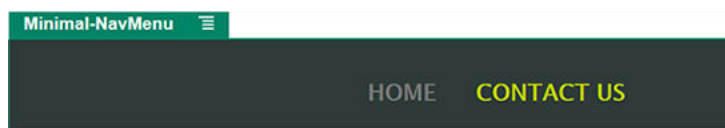
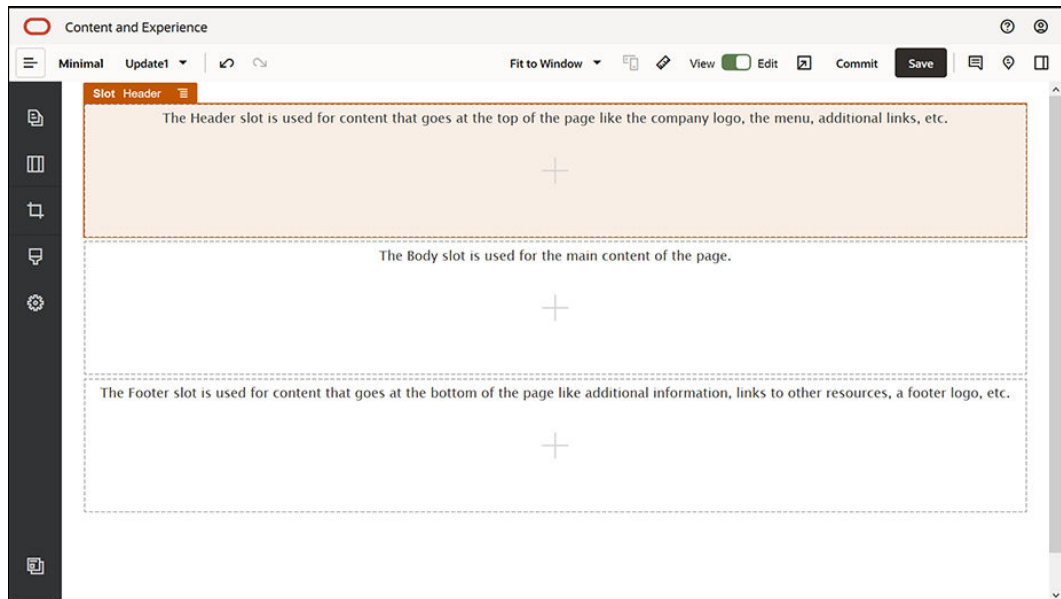
Klikając na ikonie , dostępnej w prawym górnym rogu konstruktora serwisów, można wyświetlić podgląd pierwszej strony serwisu internetowego. Serwis internetowy nadal nie jest opublikowany i nie może zostać wyświetlony przez inne osoby.

Ukończyliśmy tworzenie strony "HOME". Utwórzmy stronę "CONTACT US". Na poniższym rysunku pokazano, jak będzie wyglądała ukończona strona kontaktowa:

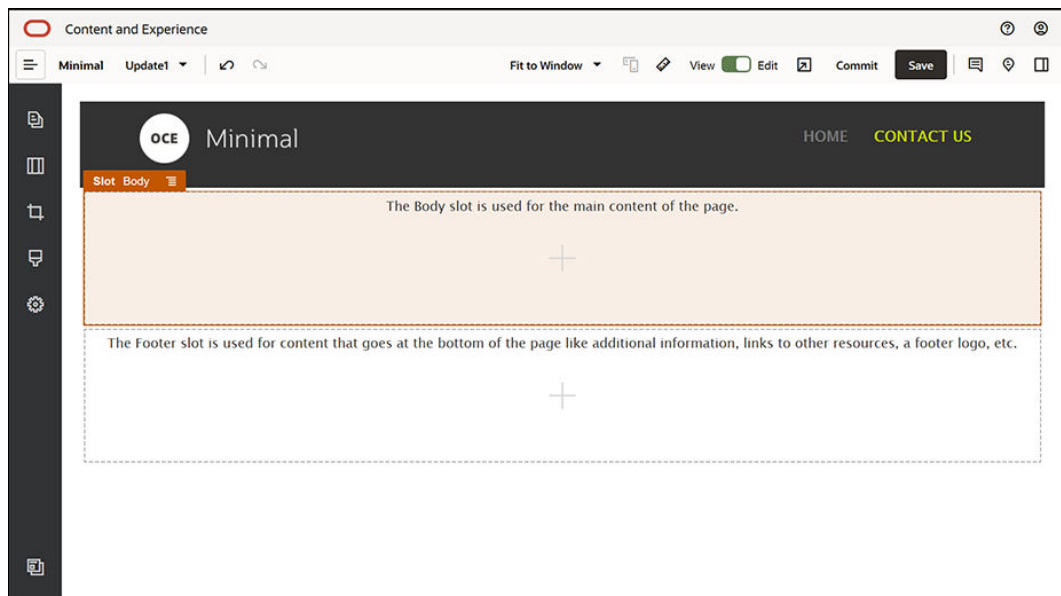


Teraz dodajmy składniki do różnych gniazd:

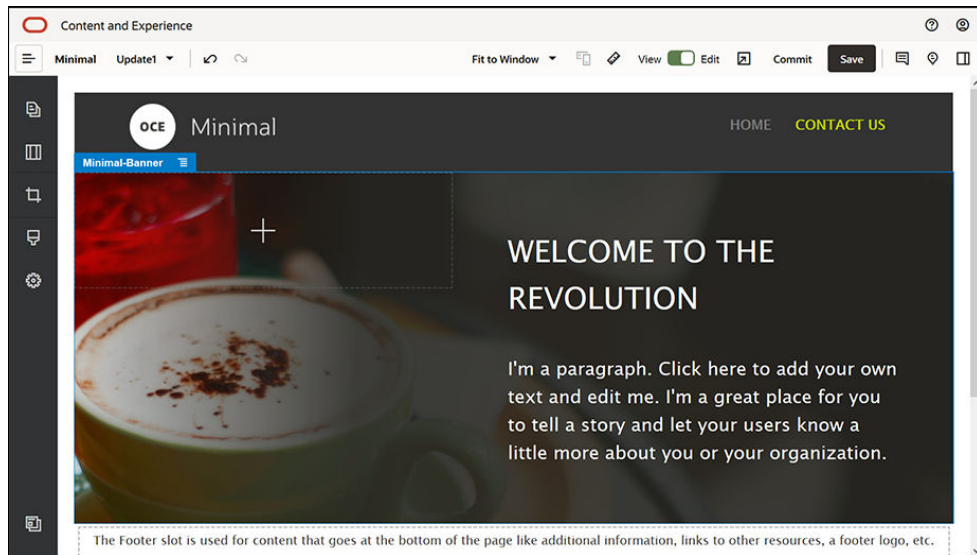
1. Kliknąć na ikonie  na lewym pasku bocznym, po czym nacisnąć przycisk **Dodaj stronę**.
2. W polu **Nazwa strony** wpisać "CONTACT US", po czym nacisnąć przycisk **Zamknij**. Została dodana nowa strona do serwisu internetowego.
3. Kliknąć na ikonie  na lewym pasku bocznym, po czym nacisnąć przycisk **Niestandardowe**.
4. Przeciągnąć składnik "Minimal-Header" (który został wcześniej utworzony i zapisany) do gniazda "Nagłówek".



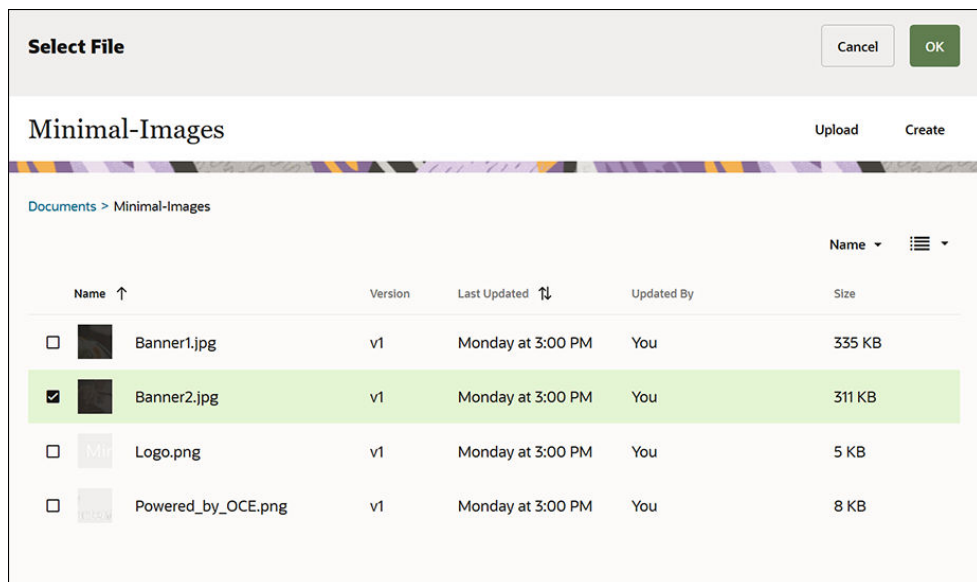
Należy zwrócić uwagę, że składnik "Minimal-NavMenu" automatycznie wybrał nową stronę "CONTACT US", która właśnie została utworzona.



5. Z lewego paska bocznego przeciągnąć składnik "Minimal-Banner" do gniazda "Treść".




6. Teraz trzeba zmodyfikować baner, tak aby jego wygląd różnił się od banera strony "HOME" i pasował do strony "CONTACT US". W obrębie grupy składników zmodyfikować ustawienia układu sekcji dwukolumnowego:  
W polu **Obraz** karty **Tło** nacisnąć przycisk **Wybierz obraz**, po czym wybrać — z wcześniej utworzonego folderu **Minimal-Images** w folderze **Dokumenty** — obraz **Banner2.jpg**.



7. W obrębie układu sekcji dwukolumnowego zmodyfikować w składnikach "tytuł" i "akapit" teksty.
  - W składniku "tytuł" wpisać tekst "Want to learn more?".
  - W składniku "tytuł" wpisać tekst "Find more learning material on the Headless CMS page".



8. Kliknąć na ikonie  na lewym pasku bocznym, po czym nacisnąć przycisk **Standardowe**.
9. Obraz strony "CONTACT US" zawiera przycisk o nazwie **OCE FOR DEVELOPERS** jako część banera. Trzeba dodać ten przycisk do banera. W obrębie grupy składników "Minimal-Banner" przeciągnąć składnik "przycisk" do drugiej kolumny układu sekcji dwukolumnowego (poniżej ostatnio dodanego składnika "akapit"). Dokonać ustawień składnika "przycisk" na karcie **Ogólne**.

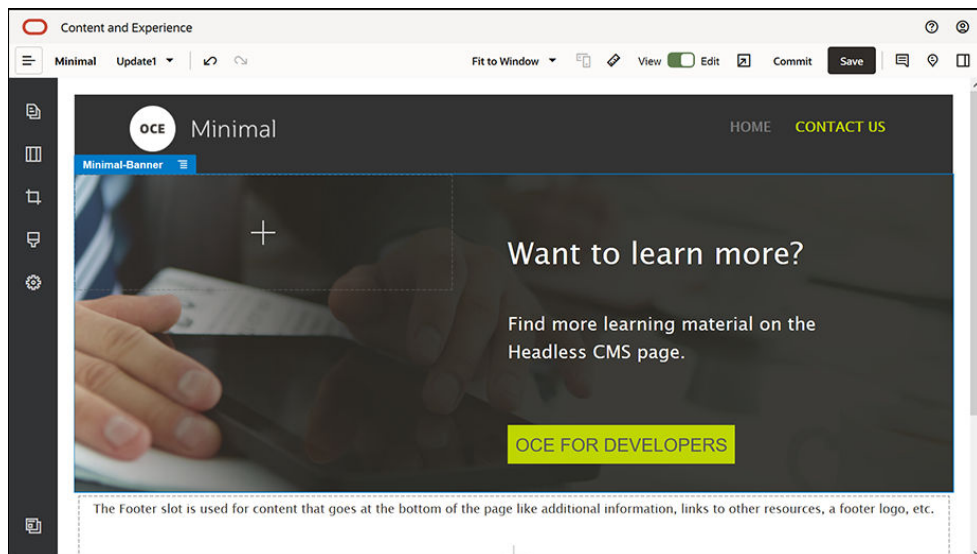
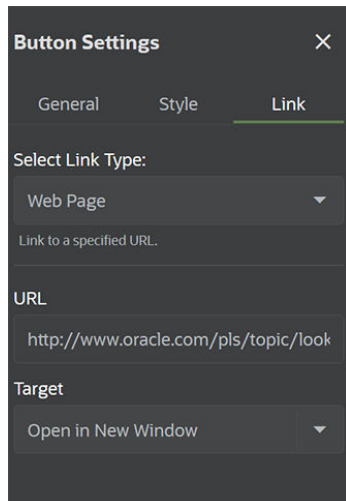
Właściwość	Wartość
Etykieta	OCE FOR DEVELOPERS
Na górze	0.3vw
Na dole	3vw
Z lewej	6vw
Z prawej	0.3vw


Na karcie **Styl** wybrać opcję **Dostosuj**, a następnie uzupełnić ustawienia.

Właściwość	Wartość
Kolor tła	#c0d600
Czcionka	<ul style="list-style-type: none"> <li>• Wprowadzić 24 jako rozmiar.</li> <li>• Wprowadzić kolor #58595b.</li> </ul>
Obramowanie	Brak
Kolor po wskazaniu	<ul style="list-style-type: none"> <li>• Dla pola <b>TŁO</b> ustawić wartość #e1fa00.</li> <li>• Dla pola <b>CZCIONKA</b> ustawić wartość #58595b.</li> <li>• Dla pola <b>OBRAMOWANIE</b> ustawić wartość #2222dd.</li> </ul>
Narożniki	0


Dokonać ustawień na karcie **Łącze**.

Właściwość	Wartość
Wybór typu łącza	Strona internetowa
URL	<a href="http://www.oracle.com/pls/topic/lookup?ctx=cloud&amp;id=content-cloud-headless">http://www.oracle.com/pls/topic/lookup?ctx=cloud&amp;id=content-cloud-headless</a>
Miejsce docelowe	Otwórz w nowym oknie



10. Kliknąć na ikonie  na lewym pasku bocznym, po czym nacisnąć przycisk **Niestandardowe**.
11. Przeciągnąć składnik "Minimal-Body" poniżej składnika "Minimal-Banner" do gniazda "Nagłówek".
12. W obrębie składnika "Minimal-Body" zmodyfikować w składnikach "tytuł" i "akapit" teksty.
  - a. W składniku "tytuł" wpisać tekst "Connect with us:".
  - b. Wprowadzić do składnika "akapit" szczegóły, takie jak adres e-mail i inne łącza związane z Asystą Techniczną:  
"Visit the [Forum Oracle Cloud Connect](#) to post your queries."  
"Przykłady związane z usługą Oracle Content Management są dostępne na stronie: <https://www.oracle.com/middleware/technologies/content-experience-downloads.html>"

13. Z lewego paska bocznego przeciągnąć kolejny składnik "Minimal-Body" do gniazda "Treść", umieszczając go poniżej już wcześniej dodanego składnika "Minimal-Body".
14. W ustawieniach grupy składników ustawić pole **Kolor** karty **Tło** na wartość **#696969**.
15. W obrębie składnika "Minimal-Body" zmodyfikować składniki "tytuł" i "akapit".
  - a. W składniku "tytuł" wpisać tekst "Locations". Wybrać tekst, po czym w edytorze tekstów zmienić kolor tekstu na **biały**.
  - b. W składniku "akapit" wpisać poniższy tekst.  
"Regionalne biuro1:  
Budynek nr 1,  
Miasto1, województwo1, kraj1  
Regionalne biuro2:  
Budynek nr 2,  
Miasto2, województwo2, kraj2"  
Wybrać tekst, po czym w edytorze tekstów zmienić kolor tekstu na **biały**.
16. Teraz trzeba dodać stopkę do strony "CONTACT US". Przeciągnąć składnik "Minimal-Footer" do gniazda "Stopka".
17. Aby zapisać zmiany, w prawym górnym rogu konstruktora serwisów nacisnąć przycisk **Zapisz**.

Klikając na ikonie , dostępnej w prawym górnym rogu konstruktora serwisów, można wyświetlić podgląd pierwszej strony serwisu internetowego i upewnić się, czy strona wygląda dobrze. Należy się upewnić, że po kliknięciu na obrazie loga na stronie "CONTACT US" użytkownik jest przenoszony z powrotem do strony startowej. Następnie należy przetestować menu i upewnić się, że nawigacja między stroną serwisu internetowego działa poprawnie.

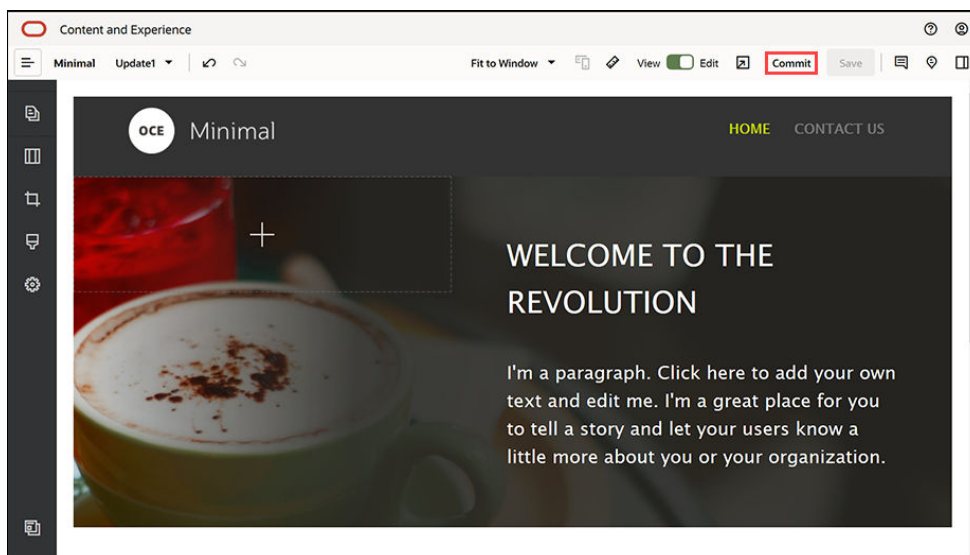
Serwis internetowy jest gotowy do opublikowania.

[Następny etap: opublikowanie serwisu internetowego](#)

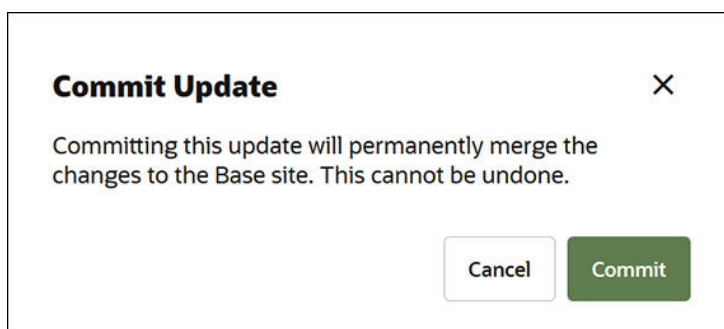
## Etap 3: Publikowanie serwisu internetowego

Teraz po pomyślnym utworzeniu serwisu internetowego można go opublikować, dzięki czemu stanie się on dostępny i widoczny w trybie online dla użytkowników.

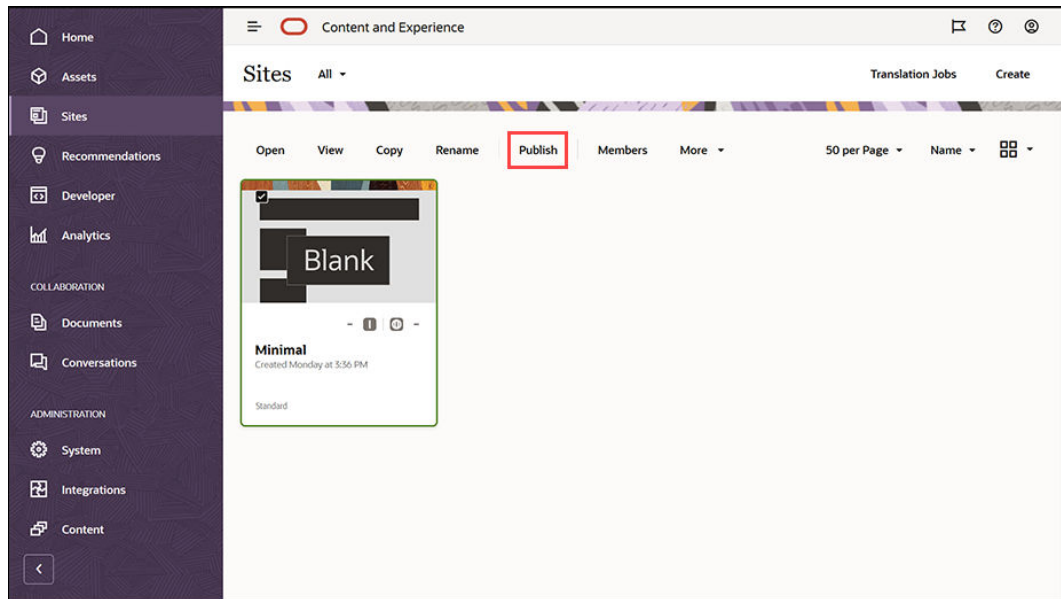
1. Gdy wszystko wygląda dobrze, można zatwierdzić zmiany w podstawowym serwisie internetowym, wybierając opcję **Zatwierdź** (przycisk w prawym górnym rogu konstruktora serwisów).



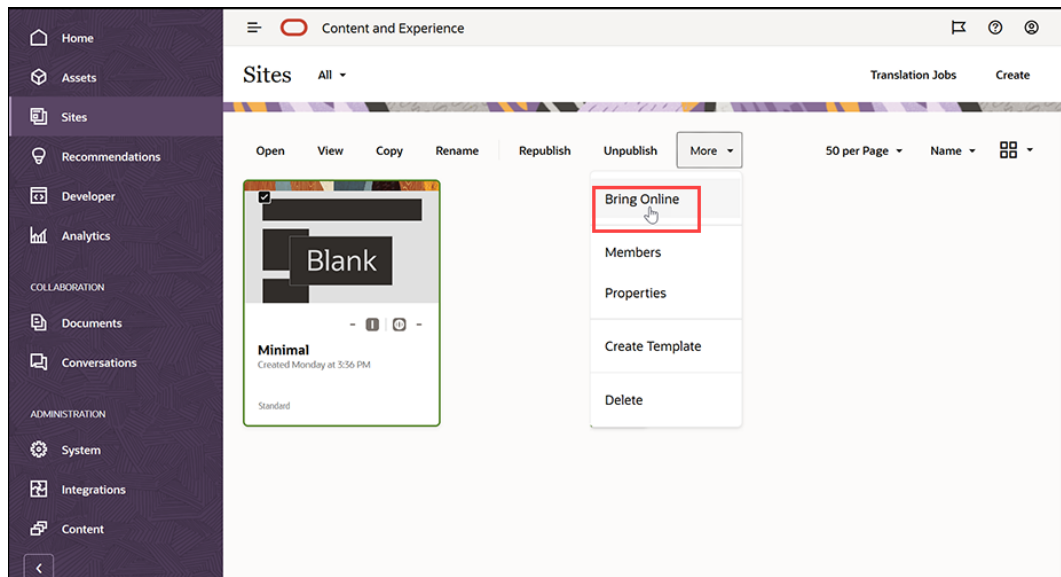
2. W oknie dialogowym "Zatwierdzanie aktualizacji" nacisnąć przycisk **Zatwierdź**.

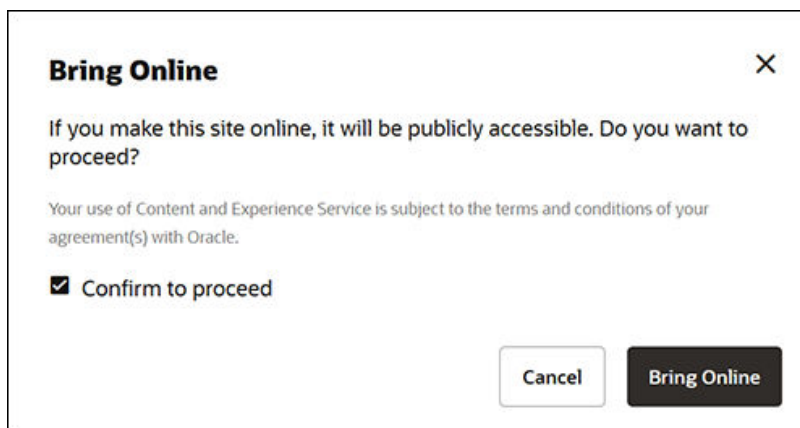


3. Po zatwierdzeniu zmian serwis internetowy jest gotowy do opublikowania.
4. Zamknąć konstruktor serwisów.
5. W lewym menu nawigacyjnym, w interfejsie internetowym Oracle Content Management, nacisnąć przycisk **Serwisy**, po czym wybrać serwis internetowy.
6. Z paska menu lub menu podręcznego wybrać opcję **Publikuj**



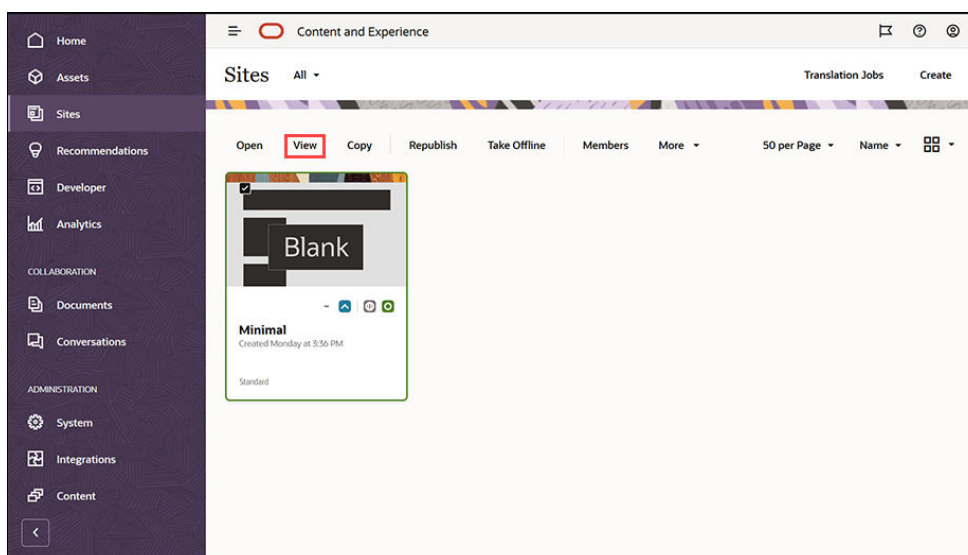
7. Po opublikowaniu serwisu internetowego zostanie wyświetlone na górze strony powiadomienie informujące o publikacji serwisu.
8. Na stronie "Serwisy" wybrać ponownie serwis internetowy, po czym wybrać z paska menu lub z menu podręcznego (kliknięcie prawym przyciskiem myszy) opcję **Przełącz do trybu online**.  
W oknie dialogowym "Przełączanie do trybu online" wybrać kolejno opcje **Potwierdzenie kontynuowania** i **Przełącz do trybu online**.





9. To wszystko. Serwis internetowy działa w trybie online, dzięki czemu inne osoby mogą go wyświetlić.

Aby wyświetlić publiczny serwis internetowy, wybrać z paska menu lub menu podręcznego opcję **Wyświetl**.



## Dalsze działania

Można dostosować serwis internetowy odpowiednio do potrzeb firmy. Poniżej zamieszczono kilka przydatnych łączy, które mogą pomóc rozpocząć pracę.

- [Korzystanie ze stylów i formatowania](#)
- [Edytowanie — wskazówki i porady](#)
- [Zarządzanie serwisem i ustawienia serwisów](#)
- [Wysyłanie plików serwisu](#)
- [Dodawanie stron](#)
- [Przenoszenie stron](#)

- [Rozmieszczanie zawartości strony](#)
- [Zmiana ustawień strony](#)
- [Zmienianie układu strony](#)
- [Praca z tabelami](#)
- [Ustawienie właściwości dotyczących wyszukiwarek](#)

# Część II

## Tworzenie i edytowanie serwisów

W tej części przedstawiono, jak zacząć tworzyć serwisy przy użyciu szablonów i motywów oraz jak edytować serwisy, używając składników i układów w celu porządkowania i dodawania zawartości. Część ta zawiera następujące rozdziały:

- [Tworzenie serwisów](#)
- [Edytowanie serwisów](#)
- [Używanie szablonów i motywów w serwisach](#)
- [Zarządzanie niestandardowymi składnikami i układami](#)
- [Praca ze stronami serwisu](#)
- [Rozmieszczanie zawartości strony](#)
- [Używanie składników wbudowanych](#)



# 4

## Tworzenie serwisów

Każdy użytkownik z odpowiednimi uprawnieniami może tworzyć serwis internetowy, korzystając z usługi Oracle Content Management. Nie jest wymagane używanie żadnych innych narzędzi ani oprogramowania czy elementów kodu. Interfejs użytkownika jest interfejsem graficznym, intuicyjnym i przyjaznym.

Tworząc serwis, zaczynamy od szablonu. Szablon zawiera wszystko, co jest potrzebne do rozpoczęcia pracy nad serwisem, w tym strukturę kodu serwisu, serwis domyślny z przykładowymi stronami i zawartością, motyw ze stylami oraz zasoby (takie jak obrazy), czy nawet składniki niestandardowe. Zob. [Omówienie procesu tworzenia serwisu](#).

Oferowana przez Oracle Content Management funkcja zarządzania serwisami upraszcza oraz przyspiesza proces dostarczania zasobów Experience użytkownikom biznesowym, a jednocześnie ułatwia działom IT kontrolę i śledzenie tych zasobów z centralnej lokalizacji oraz pełne zarządzanie całym ich cyklem życia, zmniejszając przy tym koszty tworzenia i utrzymania każdego nowego zasobu Experience, niezbędnego dla firmy. Funkcja zarządzania jest wbudowana w rdzeń usługi Oracle Content Management — musi tylko zostać włączona przez administratora. Zob. Konfigurowanie ustawień serwisów i zasobów w podręczniku *Administering Oracle Content Management*.

- [Omówienie procesu tworzenia serwisu](#)
- [Tworzenie serwisów](#)
- [Kopiowanie serwisów](#)
- [Zarządzanie wystąpieniami o serwis](#)
- [Zmiana szczegółów wystąpienia o serwis](#)
- [Wyświetlanie założeń systemowych dot. występowania o serwis](#)

## Tworzenie serwisów

Aby utworzyć serwis, należy wybrać szablon, nazwać serwis, po czym dodać zawartość. Jeśli jest włączona funkcja zarządzania, trzeba przesłać (przed dodaniem zawartości) wystąpienie o serwis, a następnie (po zatwierdzeniu wystąpienia) dodać zawartość do serwisu.

Aby można było utworzyć serwis, administrator musi umożliwić tworzenie serwisów oraz udostępnić szablony bieżącemu użytkownikowi. Jeśli opcja **Utwórz** na stronie "Serwisy" nie jest widoczna lub jeśli nie są dostępne żadne szablony, należy się skontaktować z administratorem. Zob. [Serwisy — pierwsze kroki](#) i [Omówienie funkcji zarządzania serwisem](#).

Jeśli będzie tworzony serwis firmowy, umożliwiający używanie zasobów i serwisów wielojęzycznych, serwis ten zostanie powiązany z repozytorium, założeniem systemowym dot. lokalizacji i (prawdopodobnie) językiem domyślnym. Jeśli nie jest dostępne repozytorium ani założenie systemowe dot. lokalizacji, należy je utworzyć lub zwrócić się do administratora zawartości o ich utworzenie.

 **Uwaga:**

Korzystając z usługi Oracle Content Management w edycji Starter, nie można włączyć funkcji zarządzania i można utworzyć tylko jeden serwis. Ponadto tłumaczenia są niedostępne. Aby uzyskać dostęp do pełnego zestawu funkcji i nieograniczonej liczby serwisów, należy uaktualnić usługę Oracle Content Management do edycji Premium.

1. Kliknąć na karcie **Serwisy**.
2. Nacisnąć przycisk **Utwórz**.
3. Na stronie "Wybór szablonu" wybrać szablon dla serwisu.
  - Jeśli funkcja zarządzania jest włączona, szablon określi, czy tworzony serwis będzie serwisem standardowym (nie używa założenia systemowego dot. repozytorium zasobów ani założenia systemowego dot. lokalizacji) czy serwisem firmowym (używa założenia systemowego dot. repozytorium zasobów i założenia systemowego dot. lokalizacji).
4. Na stronie "Konfigurowanie serwisu" są wyświetlane zatwierdzenia wymagane przed utworzeniem serwisu, wymagany dla niego minimalny poziom zabezpieczeń oraz obrazy stron serwisu. Jeśli zatwierdzanie jest ograniczone do określonych osób, nacisnąć przycisk **Pokaż zatwierdzających**, aby zobaczyć, kto może zatwierdzić wystąpienie o serwis.
  - Jeśli funkcja zarządzania jest włączona, wybrać typ tworzonego serwisu: **Standardowy** lub **Firmowy**.
  - Jeśli jest tworzony serwis standardowy, nacisnąć — aby przejść do następnej strony — przycisk **Dalej**.
  - Jeśli jest tworzony serwis firmowy, należy:
    - a. Wybrać repozytorium domyślne, które będzie używane do zarządzania zawartością tego serwisu. Jeśli bieżący użytkownik jest administratorem repozytorium i nie są dostępne żadne repozytoria lub istnieje potrzeba użycia nowego, wybrać opcję **Utwórz nowe repozytorium**, a następnie wykonać etapy, pozwalające utworzyć repozytorium. Aby uzyskać więcej informacji o używaniu różnych repozytoriów w serwisie, zob. Zapewnianie serwisowi dostępu do więcej niż jednego repozytorium w podręczniku *Zarządzanie zasobami przy użyciu Oracle Content Management*.
    - b. Wybrać dla serwisu założenie systemowe dot. lokalizacji. Jeśli bieżący użytkownik jest administratorem repozytorium i nie są dostępne żadne założenia systemowe lub istnieje potrzeba użycia nowego, wybrać opcję **Utwórz nowe założenie systemowe**, a następnie wykonać etapy, pozwalające utworzyć założenie systemowe.
    - c. Jeśli dla założenia systemowego szablonu dot. lokalizacji nie został ustawiony język domyślny, trzeba zrobić to teraz. Widoczne są tylko języki wymagane przez założenie systemowe dot. lokalizacji.
    - d. Aby przejść do następnej strony, nacisnąć przycisk **Dalej**.
5. Na stronie "Dodawanie szczegółów" podać poniższe informacje.
  - Wpisać nazwę serwisu. Nazwa ta będzie używana w adresie URL serwisu. Może się ona składać tylko z liter, cyfr, znaków podkreślenia (   ) i łączników (-). W adresie URL jest uwzględniana wielkość liter. Użyta spacja zostanie

automatycznie zastąpiona łącznikiem. Nie należy używać nazwy już istniejącego serwisu, różniącej się tylko wielkością liter. Na przykład, jeśli istniejący serwis nazywa się *ABC*, to nie należy tworzyć serwisu o nazwie *Abc*.

 **Uwaga:**

Dla szablonów, motywów, składników, serwisów ani stron serwisów nie wolno używać następujących nazw: authsite, content, pages, scsTEMPLATE\_\*, \_comps, \_components, \_compsdelivery, \_idcservice, \_sitescloud, \_sitesclouddelivery, \_themes, \_themesdelivery. Następujących nazw można używać dla stron serwisów, ale nie można używać dla szablonów, motywów, składników ani serwisów: documents, sites.

- Jeśli jest tworzony serwis firmowy, założenie systemowe dot. szablonu może zezwolić na edytowanie domyślnego prefiksu serwisu dla wartości przyjaznego adresu URL. Prefiks ten będzie dołączany do krótkiego opisu elementu zawartości (część adresu URL właściwa dla strony lub zasobu).
- Opcjonalnie: podać opis serwisu.
- Opcjonalnie: podać uzasadnienie tego wystąpienia o serwis. Pomoże to administratorowi serwisów podjąć decyzję o zatwierdzeniu.

#### 6. Nacisnąć przycisk **Zakończ**.

Jeśli wystąpienie o serwis wymaga zatwierdzenia, jest wysyłany e-mail do osoby, która musi to wystąpienie zatwierdzić. Jeśli zostanie zatwierdzone, użytkownik otrzyma e-mail informujący o zatwierdzeniu, a serwis zostanie utworzony. Jeśli wystąpienie zostanie odrzucone, użytkownik otrzyma e-mail informujący o odrzuceniu. Oczekujące wystąpienia można zobaczyć na stronie "Serwisy", wybierając z menu filtrowania opcję **Żądania**. Jeśli wystąpienie zostało odrzucone, pojawi się komunikat informujący o przyczynie odrzucenia, dzięki czemu można rozwiązać problem i ponownie przesłać wystąpienie.

Jeśli wystąpienie o serwis zostało ustawione na automatyczne zatwierdzenie, serwis zostanie automatycznie utworzony. Na pasku postępu będzie pokazywana nazwa nowego serwisu i status operacji tworzenia. Gdy serwis zostanie utworzony, jego nazwa pojawi się na liście serwisów; statusem serwisu będzie "offline".

Jeśli zostanie utworzony serwis firmowy, to w wybranym repozytorium zostanie utworzona odpowiednia kolekcja serwisu. Jeśli ten serwis zostanie udostępniony innemu użytkownikowi, to będzie miał on te same uprawnienia do powiązanej kolekcji.

Po utworzeniu serwisu można go udostępniać, publikować oraz przełączać do trybu online, a także zmieniać zabezpieczenia oraz dodawać i edytować zawartość. Zob. [Zarządzanie serwisem i ustawienia serwisów](#) i [Edytowanie serwisów](#).


## Kopiowanie serwisów

Aby zacząć szybko pracę, można skopiować serwis już istniejący. Do nowego serwisu, o podanej nowej nazwie, zostanie skopiowane wszystko z oryginalnego serwisu, w tym:

motyw, wszystkie oczekujące aktualizacje, strony, zawartość stron, rekomendacje, wszystkie inne zasoby (takie jak obrazy) oraz założenia systemowe.

 **Uwaga:**

Jeśli jest kopiowany serwis, który używa zawartości z więcej niż jednego repozytorium, serwis ten trzeba skopiować przy użyciu narzędzia OCE Toolkit. Zob. [Tworzenie za pomocą zestawu OCE Toolkit](#) i [Korzystanie z narzędzia cec opartego na wierszu polecenia](#)

1. Kliknąć na karcie **Serwisy**.
2. Wybrać serwis, który ma zostać skopiowany, po czym wybrać z menu podręcznego opcję **Kopiuj** albo kliknąć na ikonie  (na pasku czynności).
3. Na stronie "Konfigurowanie serwisu" są wyświetlane zatwierdzenia wymagane przed utworzeniem serwisu, wymagany dla niego minimalny poziom zabezpieczeń oraz obrazy stron serwisu. Wykonać następujące etapy:
  - a. Z rozwijanej listy **Kopiowanie** wybrać, czy skopiować tylko serwis podstawowy czy z uwzględnieniem aktualizacji. Jeśli zostaną dołączone aktualizacje, skopiowany serwis uwzględni wszelkie oczekujące aktualizacje oryginalnego serwisu. Jeśli jest kopiowany serwis standardowy, kliknąć — aby przejść do następnej strony — na strzałce, po czym pominąć etap 4.
  - b. Jeśli jest kopiowany serwis firmowy, to domyślne repozytorium zasobów, założenie systemowe dot. lokalizacji, wymagane języki oraz język domyślny są ustawiane na podstawie oryginalnego serwisu. Opcjonalnie: wybrać inne repozytorium do zarządzania zawartością serwisu.
  - c. Aby przejść do następnej strony, nacisnąć przycisk **Dalej**.
4. Na stronie "Dodawanie szczegółów" podać poniższe informacje.
  - Wpisać nazwę serwisu. Nazwa ta będzie używana w adresie URL serwisu. Może się ona składać tylko z liter, cyfr, znaków podkreślenia (   ) i łączników (-). W adresie URL jest uwzględniana wielkość liter. Użyta spacja zostanie automatycznie zastąpiona łącznikiem. Nie należy używać nazwy już istniejącego serwisu, różniącej się tylko wielkością liter. Na przykład, jeśli istniejący serwis nazywa się *ABC*, to nie należy tworzyć serwisu o nazwie *Abc*.

 **Uwaga:**

Dla szablonów, motywów, składników, serwisów ani stron serwisów nie wolno używać następujących nazw: authsite, content, pages, scstemplate\_\*, \_comps, \_components, \_compsdelivery, \_idcservice, \_sitescloud, \_sitesclouddelivery, \_themes, \_themesdelivery. Następujących nazw można używać dla stron serwisów, ale nie można używać dla szablonów, motywów, składników ani serwisów: documents, sites.

- Jeśli jest kopiowany serwis firmowy, opcjonalnie można edytować domyślny prefiks serwisu dla wartości przyjaznego adresu URL. Prefiks ten będzie

dołączany do krótkiego opisu elementu zawartości (część adresu URL właściwa dla strony lub zasobu).

- Podać opcjonalny opis serwisu.
- Opcjonalnie podać uzasadnienie tego wystąpienia o serwis. Pomoże to administratorowi serwisów podjąć decyzję o zatwierdzeniu.

#### 5. Nacisnąć przycisk **Zakończ**.

Jeśli wystąpienie o serwis wymaga zatwierdzenia, serwis zostanie utworzony, gdy administrator serwisów to wystąpienie zatwierdzi. Oczekujące wystąpienia można zobaczyć na stronie "Serwisy", wybierając z menu filtrowania opcję **Żądania**. Jeśli wystąpienie zostanie odrzucone, pojawi się komunikat informujący o przyczynie odrzucenia, dzięki czemu można rozwiązać problem i ponownie przesłać wystąpienie.

Jeśli wystąpienie o serwis zostało ustawione na automatyczne zatwierdzenie, serwis zostanie automatycznie utworzony. Na pasku postępu będzie pokazywana nazwa nowego serwisu i status operacji tworzenia. Gdy serwis zostanie utworzony, jego nazwa pojawi się na liście serwisów; statusem serwisu będzie "offline".

Jeśli zostanie skopiowany serwis firmowy, to w wybranym repozytorium zostanie utworzona odpowiednia kolekcja serwisu. Jeśli ten serwis zostanie udostępniony innemu użytkownikowi, to będzie miał on te same uprawnienia do powiązanej kolekcji.

Po utworzeniu serwisu można go udostępniać, publikować oraz przełączać do trybu online, a także zmieniać zabezpieczenia oraz dodawać i edytować zawartość. Jeśli jest udostępniany serwis, który ma dostęp do więcej niż jednego repozytorium, to udostępniane jest tylko repozytorium domyślne. Pozostałe repozytoria trzeba udostępnić osobno. Zob.

[Zarządzanie serwisem i ustawienia serwisów](#) i [Edytowanie serwisów](#).

## Zarządzanie wystąpieniami o serwis

Jeśli funkcja zarządzania serwisem jest włączona, serwisy — zanim zostaną utworzone — mogą wymagać zatwierdzenia. Oczekujące wystąpienia o serwis można zobaczyć na stronie "Serwisy", korzystając z filtru "Żądania".

### Uwaga:

Korzystając z usługi Oracle Content Management w edycji Starter, nie można włączyć funkcji zarządzania. Aby uzyskać dostęp do pełnego zestawu funkcji i nieograniczonej liczby serwisów, należy uaktualnić usługę Oracle Content Management do edycji Premium.

Zob. [Omówienie funkcji zarządzania serwisem](#).

Aby wyświetlić oczekujące wystąpienia o serwis, należy na stronie "Serwisy", wybrać z menu filtrowania opcję **Żądania**. Będąc *administratorem serwisów*, można zobaczyć w swoim systemie wszystkie wystąpienia o serwis, w tym wystąpienia, które wymagają zatwierdzenia przez inne osoby. Będąc *zatwierdzającym*, można zobaczyć wszystkie wystąpienia o serwis, dla których jest się osobą zatwierdzającą. *Wszyscy użytkownicy* mogą zobaczyć wystąpienia o serwis, które sami przesłali.

Na liście wystąpień jest wyświetlana miniatura serwisu, nazwa serwisu, występujący o serwis, data wystąpienia o serwis, opis serwisu, status wystąpienia oraz ikona pokazująca, czy serwis wymaga czy nie wymaga logowania się.

W zależności od przypisanej roli można wykonywać następujące czynności:

- Aby wyświetlić więcej szczegółów wystąpienia, takich jak wymagane minimalne zabezpieczenia, używany szablon, opcjonalne uzasadnienie serwisu oraz miniatury, należy kliknąć na nazwie serwisu lub wybrać wystąpienie, po czym nacisnąć przycisk **Wyświetl**.
- Będąc administratorem serwisów lub zatwierdzającym, można zatwierdzić wystąpienie: należy je wybrać, a następnie nacisnąć przycisk **Zatwierdź**. Serwis zostanie automatycznie utworzony po jego zatwierdzeniu.
- Będąc administratorem serwisów lub zatwierdzającym, można odrzucić wystąpienie: należy wyświetlić szczegóły, po czym wybrać opcję **Odrzuć**. Następnie należy podać przyczynę odrzucenia wystąpienia, po czym nacisnąć przycisk **Odrzuć**.  
Mimo że alternatywnie można odrzucić wystąpienie, wybierając wystąpienie z listy wystąpień, a następnie naciskając przycisk **Odrzuć**, to przy tej metodzie nie można podać przyczyny odrzucenia wystąpienia.
- Jeśli przesłane wystąpienie zakończyło się niepowodzeniem lub zostało odrzucone, można wyświetlić szczegóły, odpowiednio edytować wystąpienie, po czym **ponownie je przesłać**.
- Będąc twórcą serwisu, można usunąć wystąpienie, wybierając je, a następnie naciskając przycisk **Usuń**.

 **Uwaga:**

Jeśli użytkownik występujący o serwis został usunięty, administrator serwisu może usunąć wystąpienie o serwis.

Po utworzeniu serwisu można go udostępniać, publikować oraz przełączać do trybu online, a także zmieniać zabezpieczenia oraz dodawać i edytować zawartość. Zob. [Zarządzanie serwisem i ustawienia serwisów](#) i [Edytowanie serwisów](#).

## Zmiana szczegółów wystąpienia o serwis

Na karcie szczegółów wystąpienia o serwis są pokazywane: nazwa serwisu, opis, uzasadnienie, używany szablon, właściciel (osoba występująca o serwis) oraz obrazy serwisu.

Będąc właścicielem wystąpienia, można aktualizować jego szczegóły.

Aby wyświetlić lub zmienić szczegóły wystąpienia o serwis, należy:

1. Nacisnąć przycisk **Serwisy**, po czym wybrać z menu "Serwisy" opcję **Zlecenia**.
2. Kliknąć na wystąpieniu, aby je przejrzeć lub edytować.
3. Będąc zatwierdzającym, można **zatwierdzić** lub **odrzucić** wystąpienie. Będąc osobą występującą, można edytować odpowiednio informacje, po czym **ponownie przesłać** swoje wystąpienie.

## Wyświetlanie założeń systemowych dot. występowania o serwis

Na karcie szczegółów założeń systemowych dot. występowania o serwis są pokazywane: typ serwisu (standardowy lub firmowy), kto może uzyskać dostęp do opublikowanego serwisu oraz wszelkie zatwierdzenia, które trzeba uzyskać przed utworzeniem serwisu. W przypadku serwisów firmowych są także pokazywane: repozytorium zasobów, założenie systemowe dot. lokalizacji, język wymagany oraz język domyślny dla serwisu.

Aby wyświetlić założenia systemowe dot. występowania o serwis, należy:

1. Nacisnąć przycisk **Serwisy**, po czym wybrać z menu "Serwisy" opcję **Zlecenia**.
2. Kliknąć na wystąpieniu, które będzie przeglądane lub edytowane, po czym kliknąć na karcie **Założenia systemowe**.
3. Będąc zatwierdzającym, można **zatwierdzić** lub **odrzucić** wystąpienie. Będąc osobą występującą, można zmienić repozytorium zasobów, po czym **ponownie przesłać** swoje wystąpienie.

# 5

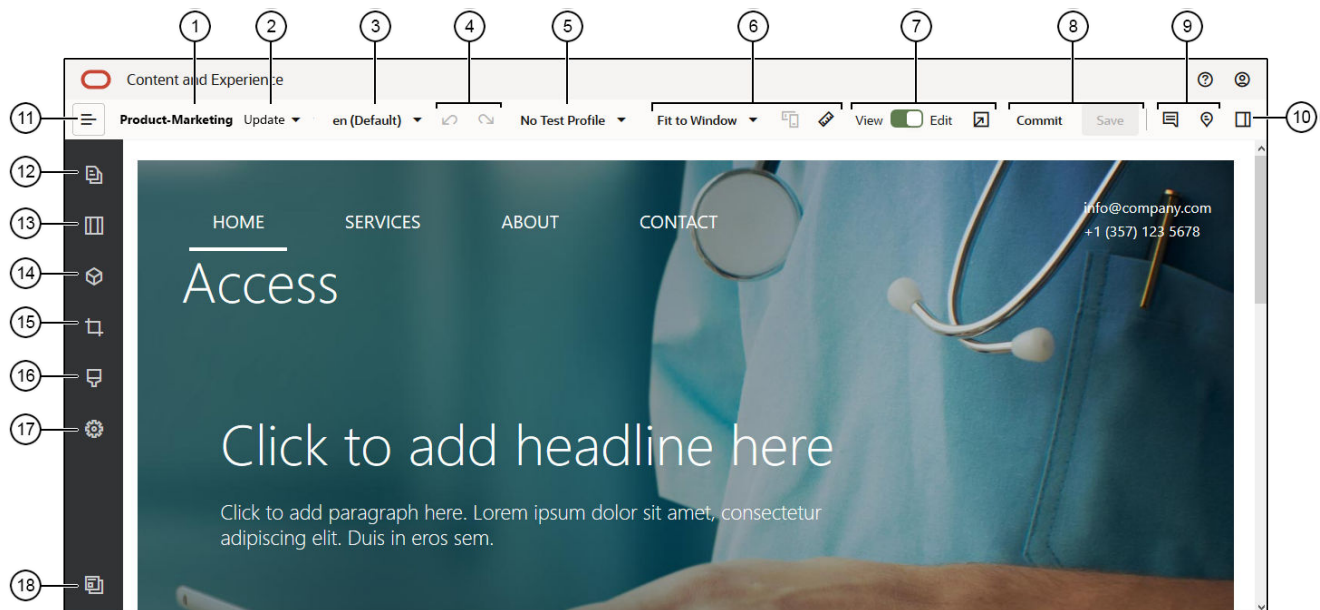
## Edytowanie serwisów

Omówiono tu pokrótce zarówno sam edytor, jak i jego możliwości.

- [Omówienie strony "Konstruktor serwisów"](#)
- [Omówienie aktualizacji serwisu](#)
- [Korzystanie z aktualizacji](#)
- [Edytowanie — wskazówki i porady](#)
- [Korzystanie ze stylów i formatowania](#)
- [Dodawanie niestandardowych właściwości serwisu](#)
- [Dostosowywanie ustawień serwisów](#)
- [Praca z tabelami](#)
- [Wysyłanie plików serwisu](#)







### Omówienie strony "Konstruktor serwisów"





Gdy edytujemy istniejącą aktualizację lub tworzymy aktualizację serwisu, otwiera się ona w konstruktorze serwisów. Dobrze jest zatem zapoznać się z układem strony i dostępnymi narzędziami.




Warto zapamiętać:









Numer	Opis
1	Jest wyświetlana <b>nazwa</b> serwisu.
2	Można sprawdzić informację <b>Aktualizacja</b> , aby się upewnić co do nazwy aktualizacji, z którą obecnie pracujemy. Jeśli jest dostępna więcej niż jedna aktualizacja, można się przełączyć do innej.
3	Aby zobaczyć, która wersja jest obecnie używana, należy sprawdzić <b>język</b> .
4	Za pomocą ikony <b>Cofnij</b> (  ) można cofnąć ostatnią edycję lub zmianę dokonaną w edytorze. Za pomocą ikony <b>Ponów</b> (  ) można ponownie zastosować ostatnią wycofaną zmianę. W bieżącej aktualizacji można używać wielokrotnie opcji cofania w celu cofnięcia serii zmian, w tym zmiany w zawartości, stylu i organizacji strony.
5	<b>Profile testowe</b> zapewniają możliwość tworzenia profili z różnymi atrybutami odbiorców oraz używania profili reprezentujących różnych odwiedzających serwis w celu przetestowania rekomendacji. Profile testowe są konfigurowane podczas tworzenia rekomendacji przez administratora serwisu.
6	Używając <b>opcji podglądu</b> , można zobaczyć, jak będzie wyglądała zawartość w różnych sytuacjach. Można przechodzić do dowolnej strony i przeglądać serwis podstawowy wraz ze zmianami zastosowanymi w bieżącej aktualizacji. Wybierając opcję <b>Dopasuj do okna</b> , można wybrać odpowiedni wymiar, aby zobaczyć stronę w takiej postaci, w jakiej będzie wyświetlana na ekranie o wybranym rozmiarze. Udostępnionych jest kilka rozmiarów. Można również tworzyć własne rozmiary urządzenia. Aby zobaczyć oznaczenia, należy kliknąć na ikonie  . Klikając na przedziale na linijce, można szybko zobaczyć, jak będzie wyglądać serwis w różnych rozmiarach. Aby zobaczyć, jak serwis będzie wyglądał w urządzeniu mobilnym (w zależności od orientacji), można również kliknąć na ikonie  . Motywy z reagującym projektem automatycznie porządkują zawartość strony pod kątem najlepszego jej przedstawienia na wybranym rozmiarze ekranu.
<div style="border: 1px solid #0070C0; padding: 10px; background-color: #E6F2FF; margin: 10px auto; width: fit-content;">  <b>Uwaga:</b>  <p>Łącza do stron serwisu nie działają w trybie podglądu.</p> </div>	
7	Za pomocą <b>przełącznika</b> na górze strony można zmieniać tryb z widoku na edycję. Aby uzyskać podgląd serwisu, należy kliknąć na ikonie  . Strona jest pokazywana w takiej postaci, w jakiej będą widzieć ją odwiedzający, tj. bez granic gniazd i składników oraz bez innych pomocy wizualnych używanych w trybie edycji. Do poruszania się po serwisie można używać — znajdujących się na stronach — łączy, w tym łączy z menu nawigacyjnego, łączy w tekście itd. Łącza do innych serwisów są ze względów bezpieczeństwa otwierane w nowych oknach. Łącza do stron w bieżącym serwisie są — w zależności od określonego celu łączy — otwierane w tym samym lub w nowym oknie.
8	Dokonane zmiany można przetwarzać przy użyciu dostępnych <b>opcji</b> . Naciskając przycisk <b>Zatwierdź</b> , można scalić zmiany w <b>serwisie podstawowym</b> . Naciskając przycisk <b>Zapisz</b> , można zapisać zmiany wprowadzone do <b>aktualizacji</b> .


Numer	Opis
9	<p>Za pomocą <b>dodatkowych opcji adnotacji</b> można dodawać adnotacje lub wyświetlać już istniejące.</p> <ul style="list-style-type: none"> <li>Aby dodać adnotację, należy kliknąć na ikonie , kliknąć na elemencie, do którego ma zostać dodana adnotacja, następnie w polu tekstowym wpisać komentarz, po czym nacisnąć przycisk <b>Prześlij</b>. Zob. Dodawanie adnotacji w podręczniku <i>Collaborating on Documents with Oracle Content Management</i>.</li> <li>Aby wyświetlić adnotację, należy kliknąć na ikonie . Gdy użytkownik kliknie na adnotacji, jej treść zostanie wyróżniona w konwersacji.</li> </ul>
10	<p>Klikając na ikonie , pokazuje się (lub ukrywa) <b>panel konwersacji</b>, za pomocą którego można tworzyć dyskusje dotyczące tego serwisu lub w nich uczestniczyć. Zob. Korzystanie z konwersacji w podręczniku <i>Collaborating on Documents with Oracle Content Management</i>.</p>
11	<p>Klikając na ikonie , ukrywa się lub pokazuje narzędzia na pasku bocznym.</p>

 **Uwaga:**

Aby można było korzystać z narzędzi z paska bocznego, trzeba ustawić przełącznik  w pozycji **Edytuj**.

Po kliknięciu na ikonie (12–18) (na basku bocznym) można zarządzać stronami, dodawać składniki, zmieniać motyw itp.

Numer	Ikona...	Przeznaczenie...
12		Edytowanie i dodawanie stron, zarządzanie strukturami stron zagnieżdżonych, reorganizacja stron, zmiana ustawień stron.
13		Dodawanie układów sekcji, takich jak układy poziome i pionowe.
14		Dodawanie zasobów do serwisu. Można wybrać dowolne zasoby, które są przechowywane w repozytorium wybranym podczas tworzenia serwisu.
15		Wstawianie składników do serwisu. Można wybrać różne typy składników, takie jak używane w motywie, niestandardowe lub standardowe, albo wyświetlić wszystkie dostępne składniki.
16		Wybór innego motywu dla serwisu albo zmiana tła strony lub gniazda
17		Dostosowywanie ustawień serwisu, takich jak słowa kluczowe na potrzeby optymalizacji pod kątem wyszukiwarek (SEO) czy przekierowania do serwisu.

Numer	Ikona...	Przeznaczenie...
18		Powrót do strony "Serwisy", umożliwiającej zarządzanie serwisami.

## Omówienie aktualizacji serwisu

Aktualizacja to nazwany zbiór zmian dokonywanych w bieżącym serwisie podstawowym. Zmiany pozostają zawarte w aktualizacji do chwili, gdy je zatwierdzimy i trwale zaktualizujemy serwis podstawowy.

Za każdym razem, gdy serwis jest wyświetlany lub edytowany w edytorze, jest używana aktualizacja. Wszelkie zmiany, dokonane przy użyciu edytora, stają się częścią tej aktualizacji. Można mieć jedną lub więcej aktualizacji i można w różnych chwilach kontynuować dodawanie zmian do aktualizacji. Aktualizacje zapewniają elastyczne zarządzanie edycjami serwisu. Na przykład kilka osób może pracować nad swoimi aktualizacjami odnoszącymi się do różnych części serwisu. Można sprawdzać i modyfikować poszczególne aktualizacje i — gdy będą gotowe — zatwierdzić je dla serwisu podstawowego. Aby aktualizacje stały się dostępne online, nadal trzeba opublikować serwis.

Aktualizacje można organizować na kilka sposobów:


- Zawierając zmiany, odnoszące się do strony, w aktualizacji nazwanej zgodnie ze stroną
- Zawierając zmiany, dokonane przez określonego użytkownika, w aktualizacji nazwanej zgodnie z nazwą użytkownika
- Prowadząc zmiany dokonane na określony dzień lub przypadające na ważny etap projektu







### Uwaga:

Aktualizacja pokazuje zawarte w niej zmiany w odniesieniu do serwisu podstawowego. Mimo że w edytorze łatwo jest przełączać się między różnymi aktualizacjami, to jednak w danej chwili można wyświetlić tylko jedną aktualizację. Jeśli istnieje kilka aktualizacji dla danej strony, może być trudno dowiedzieć się, czy nie istnieją różne zmiany tego samego obszaru zawartości. Aby uniknąć kolidujących zmian, należy ukierunkowywać poszczególne aktualizacje na określone strony lub obszary serwisu.

Zmiany zawarte w bieżącej aktualizacji są — po ich zatwierdzeniu — stosowane w serwisie podstawowym, a aktualizacja zostaje wówczas usunięta. Aby zmiany te zostały wyświetlone w trybie online dla każdego z dostępem do serwisu, trzeba serwis opublikować (przełączając go do trybu online lub ponownie go publikując).

Aby edytować serwis, należy:

1. Wybrać serwis z listy, po czym wybrać z menu podręcznego opcję **Otwórz** albo kliknąć na ikonie  (na pasku czynności).

2. Edytor zostaje otwarty w trybie podglądu. Aby można było dokonywać zmian lub używać opcji nawigacji z paska bocznego, upewnić się, że przełącznik  jest ustawiony w pozycji **Edytuj**.
3. Jeśli jest to pierwsza aktualizacja serwisu, wprowadzić jej nazwę i opcjonalny opis, po czym nacisnąć przycisk **OK**. W nazwie można używać liter, cyfr, znaków podkreślenia (   ) i łączników (-). Użyta spacja zostanie automatycznie zastąpiona łącznikiem. Jeśli już istnieją aktualizacje dla serwisu, wybrać odpowiednią aktualizację z listy, po czym kliknąć na ikonie .
4. Aby edytować określoną stronę, odszukać ją, posługując się drzewem serwisu (na lewym pasku bocznym) lub własną nawigacją serwisu bądź wyszukując stronę. Aby wyświetlić drzewo serwisu, kliknąć po lewej stronie na ikonie , a następnie na ikonie .
5. Dodać i odpowiednio zmienić zawartość strony. Wybrać opcję **Dopasuj do okna**, aby można było korzystać z opcji układu i zobaczyć, jak strona będzie wyglądała na różnych urządzeniach oraz w różnych rozmiarach. Udostępnionych jest kilka rozmiarów. Można również tworzyć własne rozmiary urządzenia. Aby zobaczyć oznaczenia, należy kliknąć na ikonie . Klikając na przedziale na linijce, można szybko zobaczyć, jak będzie wyglądać serwis w różnych rozmiarach. Aby zobaczyć, jak serwis będzie wyglądał w urządzeniu mobilnym (w zależności od orientacji), można również kliknąć na ikonie .
6. Po ukończeniu edycji serwisu zapisać dokonane zmiany w jeden z następujących sposobów:
  - Nacisnąć przycisk **Zapisz**, aby zmiany zostały zapisane w bieżącej aktualizacji. Można od razu kontynuować pracę nad bieżącą aktualizacją lub wrócić do niej później.
  - Nacisnąć przycisk **Zatwierdź**, aby zastosować zmiany do serwisu podstawowego.Aby użytkownicy mogli zobaczyć zmiany, trzeba opublikować serwis. Zob. [Przełączanie serwisu do trybu online lub offline](#) lub [Publikowanie zmian serwisu](#).

Zmiany zawarte w bieżącej aktualizacji są — po ich zatwierdzeniu — stosowane w serwisie podstawowym, a aktualizacja zostaje wówczas usunięta. Ponieważ przy każdym wyświetlaniu lub edytowaniu serwisu w edytorze trzeba używać aktywnej aktualizacji, następuje powrót do listy serwisów i tam można utworzyć nową aktualizację.

Przy pracy z aktualizacjami proces jest liniowy:

- Istnieje oryginalny serwis (serwis podstawowy). Nazwijmy go "Wersja 1".
- Tworzymy aktualizację. Gdy zlecimy scalenie aktualizacji ("Zatwierdź"), aktualizacja zostanie trwale wpisana do istniejącego serwisu. Mamy teraz nową wersję "Wersja 2" serwisu podstawowego.
- Jeśli utworzymy i scalimy kolejną aktualizację, zostanie ona trwale wpisana do istniejącego serwisu. Mamy teraz nową wersję "Wersja 3" serwisu podstawowego.



Pracując z aktualizacjami, należy pamiętać, że:

Bieżący serwis podstawowy + Scalona aktualizacja = Nowa wersja serwisu podstawowego

## Korzystanie z aktualizacji

Za każdym razem, gdy serwis jest edytowany w edytorze, jest używana aktualizacja. Wszelkie zmiany, dokonane przy użyciu edytora, stają się częścią tej aktualizacji.

Aby utworzyć aktualizację i jej użyć, należy:


1. Na stronie "Serwisy" wybrać serwis, po czym wybrać z menu podręcznego opcję **Otwórz** albo kliknąć na ikonie  (na pasku czynności).
2. Ustawić przełącznik edycji (  ) w pozycji "Edytuj".
3. Jeśli jest to pierwsza aktualizacja serwisu, wprowadzić jej nazwę i opcjonalny opis, po czym nacisnąć przycisk **OK**. Może się ona składać tylko z liter, cyfr, znaków podkreślenia ( \_ ) i łączników ( - ). Użyta spacja zostanie automatycznie zastąpiona łącznikiem. Aby użyć istniejącej aktualizacji, kliknąć na niej na liście.
4. Nazwa serwisu i bieżąca aktualizacja są w edytorze wyświetlane na pasku górnym. Jeśli istnieje więcej niż jedna aktualizacja, można przełączać się między nimi, wybierając odpowiednią z listy aktualizacji.
5. Dodać i odpowiednio zmienić zawartość strony. Korzystając z dostępnych opcji rozmiaru, zobaczyć, jak strona będzie wyglądała na różnych urządzeniach.
6. Po ukończeniu edytowania nacisnąć przycisk **Zapisz**. Gdy zmiany będą gotowe do scalenia ich z serwisem podstawowym, można aktualizację zatwierdzić. Aby użytkownicy mogli zobaczyć zmiany, trzeba opublikować serwis. Zob. [Publikowanie zmian serwisu](#).



## Edytowanie — wskazówki i porady

Poniżej zamieszczono kilka informacji o edytorze, które mogą pomóc rozpocząć pracę.


- [Ukrywanie lub pokazywanie paska bocznego](#)
- [Poruszanie się](#)
- [Zarządzanie stronami](#)
- [Edytowanie z użyciem techniki "przeciągnij i upuść"](#)
- [Składniki](#)
- [Dostosowywanie rozmiaru i odstępów składników](#)
- [Edytowalna zawartość układu](#)
- [Style i formatowanie](#)
- [Cofanie zmian](#)

### Ukrywanie lub pokazywanie paska bocznego

Aby wyświetlić opcje zarządzania stronami i ich zawartością oraz opcje edycji, należy kliknąć na ikonie .













Aby pasek boczny był dostępny, trzeba ustawić przełącznik  w pozycji **Edytuj**. Klikając na ikonie , można ukryć pasek boczny i tym samym zwiększyć obszar podglądu strony.

### Poruszanie się


Aby wybrać inną stronę, korzystając z drzewa serwisu, należy kliknąć na ikonie  na pasku bocznym. Za pomocą pola wyszukiwania (na górze listy stron) można odszukać właściwą stronę. Można także korzystać z nawigacji serwisu oraz z łączy znajdujących się na stronach.

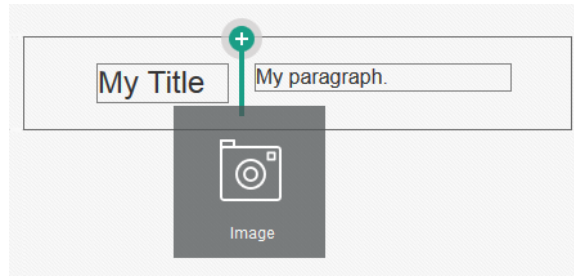
Wszelkie zmiany, które zostały dokonane na stronie, są zapisywane z chwilą przejścia do innej strony. Można także nacisnąć przycisk **Zapisz**, aby zapisać zmiany w bieżącej aktualizacji.

### Zarządzanie stronami

- Aby dodać stronę, należy wybrać poziom (lub gałąź), na którym ma zostać dodana strona, po czym nacisnąć przycisk **Dodaj stronę** albo — aby dodać stronę podrzędną — kliknąć kolejno na ikonach  i . Na dole drzewa serwisu jest dodawana "Nowa strona"; pojawia się wezwanie do nadania jej nazwy i określenia innych ustawień.
- Aby usunąć stronę, należy ją zaznaczyć, po czym kliknąć na ikonie .
- Aby przenieść stronę, należy ją zaznaczyć, po czym przenieść w nowe miejsce. Alternatywnie można wyciąć stronę, po czym wkleić ją w nowym miejscu. Aby wyciąć stronę, należy kolejno kliknąć na ikonach  i . Aby wkleić stronę, należy wybrać gałąź, w której ma zostać wklejona strona, po czym kliknąć kolejno na ikonach  i .
- Aby skopiować stronę, należy ją zaznaczyć, po czym kolejno kliknąć na ikonach  i . Aby wkleić stronę, należy wybrać gałąź, w której ma zostać wklejona strona, po czym kliknąć kolejno na ikonach  i .
- Aby zmienić ustawienia strony (takie jak metadane, nagłówek i stopka), należy kliknąć na ikonie . Zob. [Zmiana ustawień strony](#).

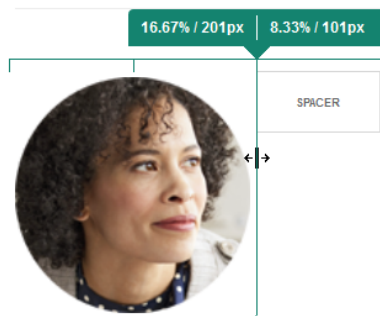
### Edytowanie z użyciem techniki "przeciągnij i upuść"

Aby dodać składnik z paska bocznego albo aby przenieść składnik na stronie, należy kliknąć na danym elemencie, po czym przeciągnąć go we właściwe miejsce na stronie. Gdy element jest przeciągany na stronę, są pokazywane granice dostępnych gniazd oraz wszystkich istniejących elementów. Pasek umieszczania  sygnalizuje, gdzie znajdzie się nowa zawartość (na górze, na dole, z lewej, z prawej):




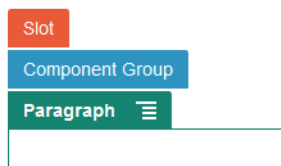
W gnieździe może się znajdować kilka elementów; elementy można przemieszczać na stronie, przeciągając je w nowe miejsca. Można także dostosować względną szerokość dwóch składników w gnieździe, klikając na granicy między składnikami i odpowiednio ją przeciągając. Składnik jest przyciągany do najbliższej linii siatki sygnalizowanej na linijce wyświetlanej nad składnikami.

Rozmiar każdego ze składników jest przedstawiany zarówno w pikselach, jak i jako procent miejsca dostępnego w gnieździe. Aby dostosować szerokość do wartości innych niż zdefiniowane przez siatkę, należy podczas klikania i przeciągania granicy przytrzymać naciśnięty klawisz **Ctrl**.




## Składniki

Po umieszczeniu składnika na stronie można dostosować wyrównanie, odstępy i inne właściwości; w tym celu należy wybrać element, klikając na jego ikonie menu , po czym wybrać opcję **Ustawienia**. Po kliknięciu na nazwie składnika (zamiast na nazwie składnika ("Menu")), będzie można zobaczyć i wybrać ikonę "Menu" dla gniazda i grupy składników (jeśli bieżący składnik jest częścią grupy składników). Jeśli zostanie wybrana jedna z innych kart, będzie można zobaczyć ikonę "Menu" i ustawić właściwości tego elementu:



Projektanci motywów mogą określać, które składniki mają zostać zawarte w motywie; dlatego niektóre składniki dostępne w jednym serwisie mogą w innym być niedostępne. Projektanci motywów mogą także określać, które składniki są dozwolone w danym gnieździe w danym układzie strony. Jeśli składnik nie jest dozwolony

w określonym gnieździe, pasek umieszczania zmienia kolor i znak (na minus)  oraz zostaje wyświetlony komunikat podobny do następującego:



### Dostosowywanie rozmiaru i odstępów składników


Można dostosowywać rozmiary różnych składników (takich jak galerie i obrazy) w różnych jednostkach standardu CSS oraz dostosowywać odstępy między składnikami. Na przykład:

- px (piksele): jednostka domyślna. Jeśli została podana tylko wartość liczbowa, domyślnie przyjmuje się, że jest ona wyrażona w pikselach.
- % (procent): ustawia element na procent rozmiaru nadrzędnego elementu HTML. Przykład: 25%
- em: ustawia rozmiar elementu wyrażony w powiększonych spacjach (em-space). Przykład: 25em
- vw: ustawia rozmiar jako procent szerokości wyświetlanego obszaru i reaguje na zmianę jego rozmiaru. Przykład: 10vw

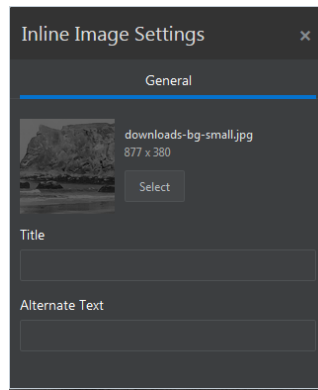
### Edytowalna zawartość układu

Projektanci motywów mogą dodawać do układów strony wbudowaną zawartość (taką jak oświadczenie dotyczące praw autorskich), której nie można zmieniać za pomocą edytora. Mogą także określać zawartość (w tym zasoby cyfrowe), mającą postać zwykłego tekstu lub obrazu, jako edytowalną. Dzięki temu współtwórca może zmienić tekst lub obraz, lecz nie może zmodyfikować lokalizacji ani innych atrybutów układu. Elementy edytowalne mają menu z opcjami, umożliwiającymi pogrubienie, pochylenie i/lub podkreślenie tekstu oraz zmianę lub usunięcie łącza:



Edytowalne elementy graficzne mają ikonę ustawień  oraz panel, w którym można określić obraz, tytuł i tekst alternatywny.





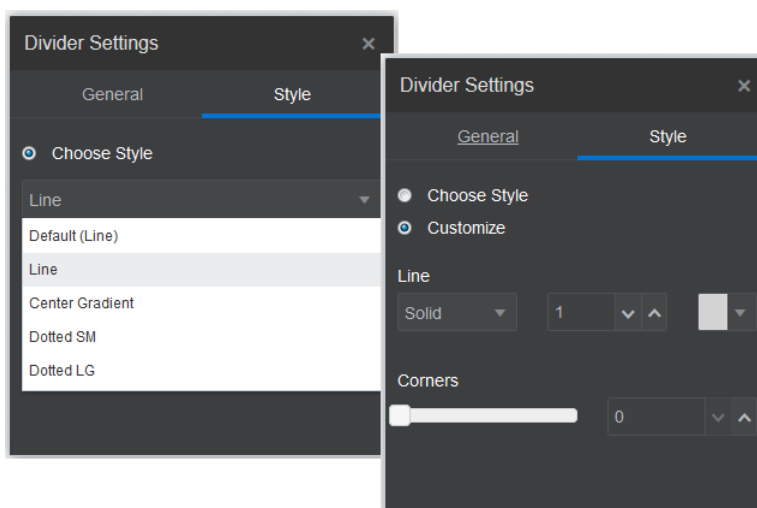
Rozmiar ramki zawierającej tekst lub obraz dostosowuje się do długości tekstu lub rozmiaru obrazu.

#### Uwaga:


Wprowadzane zmiany są stosowane tylko na bieżącej stronie. Pierwotna zawartość jest przechowywana (wraz z układem) w motywie i jest ustawiana jako domyślna przy pierwszym zastosowaniu na stronie.


### Style i formatowanie

Większość składników ma zdefiniowane przez motyw style podstawowe (jeden lub większą ich liczbę) określające niektóre aspekty wyglądu składnika. Style można z łatwością przełączać lub przesłaniać. Aby wybrać jeden z dostępnych stylów, należy otworzyć panel "Ustawienia" składnika, kliknąć na karcie "Ustawienia", kliknąć na karcie **Styl**, rozwinąć menu **Wybór stylu**, po czym wybrać odpowiedni styl. Aby podać własne wartości właściwości określonych w definicji stylu, należy nacisnąć przycisk **Dostosuj**, a następnie ustawić opcje formatowania.



## Cofanie zmian

Za pomocą ikony  można cofnąć ostatnią edycję lub zmianę dokonaną w edytorze. Cofanie można przeprowadzać więcej niż raz w celu cofnięcia serii zmian.






Za pomocą ikony  można ponownie zastosować ostatnią wycofaną zmianę. Jeśli cofanie było przeprowadzone kolejno kilka razy, można także kilka razy użyć ponawiania.

W bieżącej aktualizacji można cofać zmiany dokonane w zawartości, stylu i organizacji strony. Niektóre czynności nie są uwzględniane w łańcuchu cofania:

- Jeśli przełączymy się do innej aktualizacji, łańcuch cofania zostanie zresetowany i nie będzie można cofnąć zmian w aktualizacji, nad którą wcześniej pracowaliśmy. W obrębie danej aktualizacji można cofać zmiany, nawet po ich zapisaniu.
- Jeśli w edytorze zmienimy widok, na przykład przełączając strony lub zmieniając rozmiar danej strony, to tego typu zmiany musimy wycofać ręcznie.
- Jeśli edytujemy składniki tekstowe, takie jak tytuły lub akapity, to edytor tekstu ma swój własny łańcuch cofania. Po wyjściu z edytora tekstu nie będzie można tych zmian cofnąć.

## Korzystanie ze stylów i formatowania

Większość składników ma zdefiniowane przez motyw style podstawowe (jeden lub większą ich liczbę) określające niektóre aspekty wyglądu składnika. Style można z łatwością przełączać, a także można przesłaniać ustawienia ze stylu, używając wybranych opcji.



1. Przejść do strony, która ma być edytowana, i upewnić się, że przełącznik  jest ustawiony w pozycji **Edytuj**.
2. Aby ustawić podstawowy styl składnika, kliknąć na ikonie  menu składnika, po czym wybrać opcję **Ustawienia**. Kliknąć na karcie **Styl**.
  - Aby użyć stylu z motywu serwisu, wybrać opcję **Wybór stylu**, po czym wybrać z menu odpowiedni styl. Style są definiowane dla poszczególnych składników, dlatego lista stylów może się różnić. Na przykład style dla obrazu różnią się od stylów dla akapitu.
  - Aby określić własne opcje formatowania podstawowego, należy nacisnąć przycisk **Dostosuj**, a następnie ustawić opcje formatowania.
3. Aby skopiować i wkleić podstawowy styl do innych podobnych składników, kliknąć na ikonie  menu składnika, po czym wybrać opcję **Kopiuj styl składnika**. Kliknąć na ikonie menu  podobnego składnika, po czym wybrać opcję **Wklej styl: nazwa składnika**.
4. Aby sformatować tekst w składniku "tytuł" lub "akapit", kliknąć na składniku "tekst". Pojawi się pasek narzędzi wraz z opcjami formatowania. Zaznaczyć tekst, który ma zostać sformatowany, po czym użyć dowolnych opcji, np. określających czcionkę, kolor lub wyrównanie. Dokonywane zmiany są od razu stosowane. Aby usunąć formatowanie, zaznaczyć tekst, po czym kliknąć na ikonie . Zmiany formatowania są stosowane w odniesieniu do stylu podstawowego. Jeśli zmienimy styl podstawowy, zmiany te pozostaną na stałe.

## Dostosowywanie ustawień serwisów

Ustawienia serwisu można dostosować w zakresie dodawania ikon serwisu, pliku kontrolera, mapy serwisu, plików robotów i dodatkowych plików oraz określenia dostawcy odwzorowań.

Ustawienia te są przechowywane w aktualizacji aż do chwili jej zatwierdzenia. Gdy aktualizacja zostanie opublikowana, pliki będą przechowywane w folderze głównym motywu, tak aby miały do nich dostęp wszystkie serwisy używające tego motywu.

Aby zmienić dla całego serwisu sposób działania ikon i łączy, należy:

1. Otworzyć serwis do edycji.
2. Kliknąć na ikonie  na pasku bocznym, po czym kliknąć na ikonie **Serwis** (.
3. W sekcji **Ulubione ikony** wybrać obraz, który ma zostać użyty dla serwisu, gdy serwis jest minimalizowany w przeglądarce lub na innej platformie, na przykład w urządzeniu mobilnym. Ikona musi być przechowywana jako zasób cyfrowy, do którego można uzyskać dostęp. Kliknąć na opcji **Wybór pliku do wysłania**, przejść do ikony, wybrać ją, po czym nacisnąć przycisk **OK**. Aby ulubiona ikona działała we wszystkich przeglądarkach, trzeba ikonę dostosować:
  - **Dla przeglądarek Chrome i Safari** musi w pliku kontrolera występować odwołanie do ulubionej ikony (z prefiksem serwisu), jak pokazano w przykładzie:

```
<link rel="shortcut icon" href="/mySitePrefix/favicon.ico" />
```

**Dla przeglądarek Internet Explorer 11 i Firefox** trzeba zawrzeć podobny wpis w szablonach strony. Szablon strony może być częścią motywu i może być używany w wielu serwisach. Dlatego nie może używać ustalonego prefiksu serwisu — musi używać tokenu. Przykład:

```
<link rel="shortcut icon" href="<!--$SCS_SITE_PATH-->/  
favicon.ico" />
```

Gdy strona będzie pobierana, token zostanie zastąpiony prefiksem serwisu.

Należy pamiętać, że przeglądarki Internet Explorer i Firefox ładują ulubioną ikonę z kontrolera i ponownie z szablonu strony. Dlatego ikona będzie migająca, chyba że będzie określona zarówno w kontrolerze, jak i w szablonie strony. Przeglądarki Chrome i Safari ładują tylko ikonę, która jest określana przez kontroler.

4. W sekcji **Plik kontrolera** można dodać plik zmieniający sposób przetwarzania łączy przez przeglądarki. Poza odwoływaniem się do ulubionych ikon, można również dodawać tagi OpenGraph lub tagi metadanych, służące do weryfikacji serwisu webmastera bądź udostępniania serwisu w mediach społecznościowych. Można pobrać domyślny plik kontrolera i edytować go albo wysłać swój własny plik. Plik musi być przechowywany jako zasób cyfrowy, do którego można uzyskać dostęp. Kliknąć na opcji **Wybór pliku do wysłania**, przejść do pliku, wybrać go, po czym nacisnąć przycisk **OK**.

Na przykład, jeśli serwis zawiera przycisk "Facebook - Udostępnij", można udostępnić metadane, których serwis Facebook używa do wyświetlania

szczegółów dotyczących serwisu użytkownika w serwisie Facebook, tak jak pokazano w poniższym przykładzie:

```
<meta property="og:image" content="https://my.domain.com/fb-image.jpg" />
<meta property="og:title" content="My Site Title on FB!" />
<meta property="og:url" content="https://my.domain.com" />
<meta property="og:site_name" content="My Site Name on FB" />
```

Można dodać tag weryfikacji webmastera Google, podobny do poniższego przykładu:

```
<meta name="google-site-verification"
content="GCVURS9d2fP6jev5upt0Yt1AIp71C9D__ALqS8pg" />
```



5. W sekcji **Mapa serwisu i pliki robotów** można przesłać niestandardową mapę serwisu i pliki robotów. *Mapa serwisu* jest to plik XML, którego można użyć do wyświetlenia adresów URL serwisu i informacji o każdym adresie URL, takich jak data ostatniej aktualizacji. *Plik robota* jest to plik tekstowy, który można utworzyć w celu poinstruowania wyszukiwarek, w jaki sposób mają indeksować strony występujące w serwisie. Pliki muszą być przechowywane jako zasoby cyfrowe, do których można uzyskać dostęp. Kliknąć na opcji **Wybór pliku do wysłania**, przejść do pliku, wybrać go, po czym nacisnąć przycisk **OK**.
6. W sekcji **Dodatkowe pliki** można (jeśli trzeba) wysłać większą liczbę plików, na przykład potrzebnych do sprawdzenia praw własności serwisu. Plik musi być przechowywany jako zasób cyfrowy, do którego można uzyskać dostęp. Kliknąć na opcji **Wybór pliku do wysłania**, przejść do pliku, wybrać go, po czym nacisnąć przycisk **OK**.
7. Można wybrać dostawcę składnika "mapa" i używanych łączy. Należy wybrać **Oracle Maps** lub **Google Maps**.
8. Gdy aktualizacja zostanie opublikowana, zmiany zostaną opublikowane i udostępnione do użycia.

## Dodawanie niestandardowych właściwości serwisu

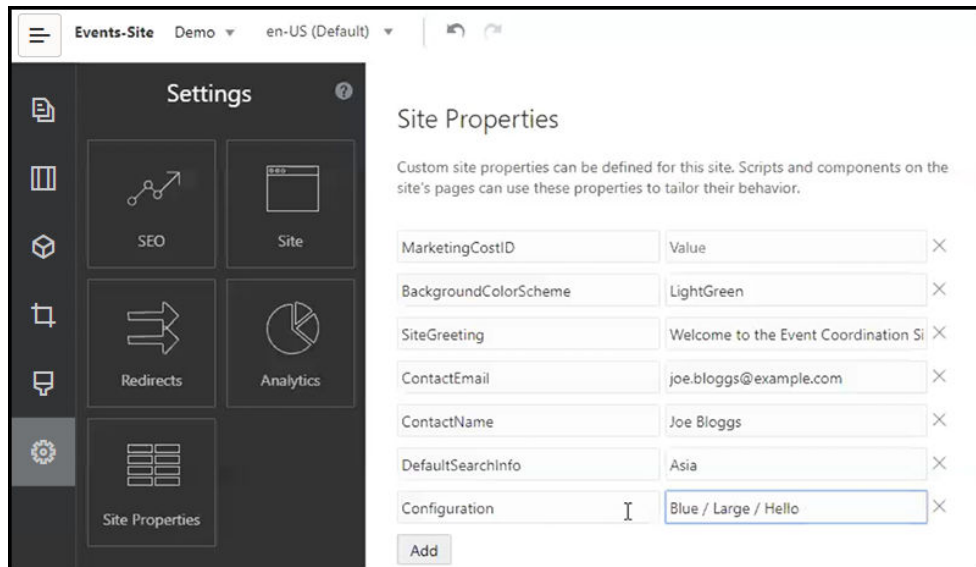
Do serwisu można dodawać niestandardowe właściwości w formie par "nazwa-wartość". Właściwości te są przechowywane wraz z serwisem i są udostępniane skryptom oraz składnikom na stronach serwisu. Za pomocą tych właściwości można parametryzować lub dostosowywać serwis bez konieczności wprowadzania zmian w kodzie używanych skryptów i składników.

Na przykład właściwości niestandardowe mogą być używane do zmiany koloru tła strony, precyzowania wyników wyszukiwania, wypełniania wykazów i (ogólnie) określania zmiennych zależnych od serwisu.

Niestandardowe właściwości serwisu są dodawane podczas edytowania serwisu przy użyciu panelu "Ustawienia".

1. Otworzyć serwis do edycji.
2. Kliknąć na ikonie  na pasku bocznym, po czym kliknąć na ikonie **Właściwości serwisu** (.
3. Nacisnąć przycisk **Dodaj**.

4. Podać nazwę i wartość niestandardowej właściwości serwisu. Można dodać maksymalnie do 50 niestandardowych właściwości serwisu. Dla pola "Nazwa" istnieje limit 200 znaków, a pola wartość — 2000 znaków.



5. Kliknąć na ikonie X obok pary "nazwa-wartość", aby usunąć ją.
6. Po ukończeniu dodawania lub usuwania niestandardowych właściwości serwisu, nacisnąć przycisk **Zatwierdź**.

 **Uwaga:**

Zmiany nie są scalane z już istniejącymi niestandardowymi właściwościami serwisu. Wskutek zatwierdzenia zmian w niestandardowych właściwościach serwisu, zostają zastąpione wszelkie już istniejące niestandardowe właściwości serwisu w serwisie podstawowym.

Po zdefiniowaniu niestandardowych właściwości serwisu można ich używać w serwisie i skryptach serwisu, na przykład w stopce lub w polu "Dodatkowy napis-zapytanie" składników obsługujących rozszerzenie SCSSMacro (takich jak listy zawartości) bądź poprzez tokeny w składnikach "tytuł" i "akapit".

Ten skrypt działa z SCSSRenderAPI w trybie wykonawczym i podczas projektowania. Wartości niestandardowych właściwości serwisu są także dostępne — dla kodu układów i składników w kompilatorze szablonów — poprzez SCSSCompileAPI. Ten zestaw API ma nową funkcję `getCustomSiteProperty`, analogiczną do `SCSSRenderAPI`. Umożliwia ona odczytywanie wartości niestandardowej właściwości serwisu przez kod układu i kod składnika.

Na przykład jest definiowany niestandardowy układ sekcji, który będzie używał `SCSSRenderAPI` do wywoływania i pobierania niestandardowych właściwości serwisu dla nagłówka oraz nazwy i e-maila osoby kontaktowej. Poniższy skrypt buduje napis

HTML przy użyciu niestandardowych właściwości serwisu, który jest dołączany do modelu DOM.

```
define([
    'jquery'
], function( $ ){
    'use strict';



    function SectionLayout( params ) {
    }
    SectionLayout.prototype = {
        render: function( parentObj ) {
            var html = '';

            try {
                html += '<div>';
                html += '<h1>' +
SCSRenderAPI.getCustomSiteProperty('SiteGreeting') + '</h1>';
                html += '<div>For more information, contact <a
href="mailto:' + SCSRenderAPI.getCustomSiteProperty('ContactEmail')
'"> +
SCSRenderAPI.getCustomSiteProperty('ContactName') + '</a></div>';
                html += '</div>';
                $(parentObj).append( html );
            } catch( e ) {
                console.error( e );
            }
        },
    };
};
```

Właściwości niestandardowe są zachowywane podczas tworzenia szablonu z serwisu wraz z dodanymi niestandardowymi właściwościami serwisu oraz podczas tworzenia serwisu z szablonu zawierającego niestandardowe właściwości serwisu.

## Praca z tabelami

W składniku "akapit" można zawierać tabele utworzone bądź wklejone z istniejącego źródła HTML.

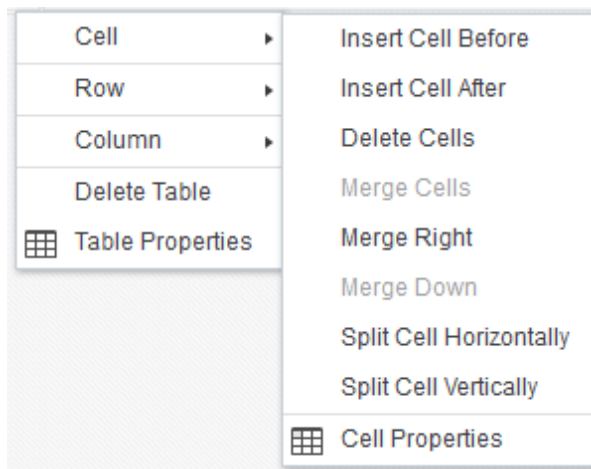
1. Przejsz do strony, która ma być edytowana, i upewnij się, że przełącznik  jest ustawiony w pozycji **Edytuj**.
2. Dodać składnik do strony albo kliknąć w istniejącym składniku "akapit".
3. Aby dodać tabelę z innej strony HTML, kliknąć i przeciągnąć kursorem myszy (zaznaczając w ten sposób tabelę), po czym skopiować zaznaczenie i wkleić je do składnika "akapit". Aby utworzyć nową tabelę w miejscu, w którym znajduje się kursor, kliknąć na ikonie . Określić liczbę wierszy i kolumn, szerokość oraz inne opcje formatowania. Można dodać opcjonalną **nazwę napisową** (wyśrodkowaną nad tabelą) i opcjonalne **podsumowanie** stanowiące dodatkowy kontekst dla technologii wspomagających, takich jak automatyczne czytniki ekranu.
4. Po ukończeniu, aby zamknąć okno, nacisnąć przycisk **OK**. Aby zmienić te ustawienia tabeli, kliknąć prawym przyciskiem myszy na tabeli, po czym wybrać opcję **Właściwości**

**tabeli.** Aby usunąć tabelę, kliknąć prawym przyciskiem myszy na tabeli, po czym wybrać opcję **Usuń tabelę**.

 **Uwaga:**

W przypadku opracowywania dla różnych rozmiarów wyświetlania (na przykład dla ekranów urządzeń mobilnych), można użyć kodu, dostępnego w pliku design.css z motywu StarterTheme, do [utworzenia tabeli responsywnej](#), umożliwiającej przedstawianie danych (danych z poszczególnych wierszy) w postaci stosu, gdy jest wyświetlana w urządzeniach mobilnych.

5. Aby dodać, usunąć lub zmodyfikować określone wiersze, kolumny lub komórki, kliknąć prawym przyciskiem myszy na wierszu, kolumnie lub komórce, po czym wybrać z menu odpowiednią opcję. Na przykład:



 **Uwaga:**

W danej chwili można zmieniać właściwości tylko jednej komórki.

6. Można scalać i rozdzielać komórki, tworząc złożone układy tabel. Na przykład:

		Split Cells	
Merged cells			

- Aby podzielić komórkę na dwie komórki, kliknąć prawym przyciskiem myszy w komórce, wybrać opcję **Komórka**, po czym wybrać opcję **Podziel komórkę poziomo** lub **Podziel komórkę pionowo**.

- Aby scalić dwie poziome komórki, kliknąć prawym przyciskiem myszy w lewej komórce, wybrać opcję **Komórka**, po czym wybrać opcję **Scal w prawo**.
  - Aby scalić dwie pionowe komórki, kliknąć prawym przyciskiem myszy w górnej komórce, wybrać opcję **Komórka**, po czym wybrać opcję **Scal w dół**.
7. Za pomocą opcji z menu formatowania ogólnego zmienić format i wyrównanie tekstu w komórkach.

## Wysyłanie plików serwisu

Obrazy i dokumenty, przeznaczone do użycia w serwisie, można wysłać w dowolnej chwili za pomocą interfejsu Oracle Content Management. Pliki można także wysłać bezpośrednio z konstruktora serwisów podczas pracy z obrazami tła i składnikami "obraz", "galeria" lub "dokument".

Jeśli w serwisie mają być używane pliki obrazów i dokumentów, możemy używać obrazów przechowywanych wraz z serwisem albo z innej lokalizacji, do której mamy dostęp. Możemy także używać obrazów, które zostały nam udostępnione lub które wysłaliśmy z lokalizacji lokalnej lub sieciowej.

### Wysyłanie plików


Aby wysłać pliki z lokalizacji lokalnej lub sieciowej, należy:

1. Wybrać opcję **Dokumenty**, po czym przejść do lokalizacji, w której ten plik ma być przechowywany. Aby dodać nowy folder w bieżącej lokalizacji, nacisnąć przycisk **Utwórz**.
2. Nacisnąć przycisk **Wyślij**.
3. Odszukać i wybrać pliki, po czym nacisnąć przycisk **Otwórz**.

### Wysyłanie plików bezpośrednio z edytora

Pracując z obrazami tła lub składnikami (takimi jak dokumenty lub obrazy), możemy wysłać pliki bezpośrednio, jeśli nie ma ich w lokalizacji Oracle Content Management.

Na przykład, aby wysłać pliki z lokalizacji lokalnej lub sieciowej w celu użycia ich ze składnikiem "galeria", należy:

1. Aby dodać obrazy do galerii, kliknąć na jej ikonie , wybrać pozycję **Ustawienia**, po czym kliknąć na opcji **Obrazy** na karcie **Ogólne**.
2. Nacisnąć przycisk **Dodaj obrazy**.
3. Przejść do miejsca, w którym plik będzie przechowywany, albo nacisnąć przycisk **Utwórz**, aby dodać nowy folder w bieżącej lokalizacji.
4. Nacisnąć przycisk **Wyślij**.
5. Odszukać i wybrać pliki, po czym nacisnąć przycisk **Otwórz**.
6. Wybrać jeden lub większą liczbę obrazów w repozytorium, po czym nacisnąć przycisk **OK**.



 **Uwaga:**

W oknie są wyświetlane wszystkie dostępne pliki. Trzeba wybrać typ pliku właściwy w danym kontekście. Na przykład, jeśli jest wybierany plik obrazu, należy wybrać plik w formacie graficznym (GIF, JPG, JPEG, PNG lub SVG). Aby utworzyć łącze do pliku, wybrać opcję **Zamiast kopiować plik do serwisu, użyj odwołania do pliku oryginalnego**. Jeśli ta opcja nie zostanie wybrana, w serwisie będzie składowana kopia pliku, do której będą się odnosiły odwołania z serwisu.

# 6

## Używanie szablonów i motywów w serwisach

Szablon zawiera wszystko, co jest potrzebne do rozpoczęcia pracy nad serwisem, w tym strukturę kodu serwisu, serwis domyślny z przykładowymi stronami i zawartością, motyw ze stylami oraz zasoby (takie jak obrazy), czy nawet składniki niestandardowe. *Motyw* definiuje wygląd i ogólny styl serwisu, w tym schemat kolorów, rozmiar czcionek, typ czcionek i tła stron.

### Szablony

- [Omówienie szablonów](#)
- [Tworzenie szablonu z serwisu](#)
- [Zmienianie szczegółów szablonu](#)
- [Zmienianie założeń systemowych dot. szablonów](#)
- [Zmienianie statusu lub odbiorców szablonu](#)
- [Zarządzanie szablonami](#)
- [Eksportowanie i importowanie szablonów](#)

### Motywy

- [Omówienie motywów](#)
- [Zarządzanie motywami](#)
- [Publikowanie motywów](#)

## Omówienie szablonów

Szablon zawiera wszystko, co jest potrzebne do rozpoczęcia pracy nad serwisem, w tym strukturę kodu serwisu, serwis domyślny z przykładowymi stronami i zawartością, motyw ze stylami oraz zasoby (takie jak obrazy), czy nawet składniki niestandardowe.

### Szablony domyślne

Oracle Content Management udostępnia różne szablony, których można używać do tworzenia serwisów. Wystarczy wybrać szablon, nadać nazwę serwisowi i już można zacząć dodawać zawartość. Szablony te są zazwyczaj instalowane przez administratora usługi podczas jej inicjalizowania.

Szablon	Opis
<b>Pusty szablon</b>	Pusty szablon to jedna strona zawierająca gniazda "Nagłówek", "Treść" i "Stopka", zapewniające pełną swobodę przy korzystaniu z własnego projektu.

Szablon	Opis
<b>Prezentacja materiałów towarzyszących</b>	Szablon "Prezentacja materiałów towarzyszących" zawiera styl menu poziomego wyrównanego do prawej strony, proste układy stron oraz strony poziomego głównego, na których są wyszczególniane powiązane zasoby wideo, a następnie zagnieżdżone strony szczegółów prezentujące poszczególne filmy wideo. Zawiera składnik "menu standardowe" oraz wyświetla obecnie zalogowanego użytkownika.
<b>Wprowadzenie nowego produktu</b>	Szablon "Wprowadzenie nowego produktu" zawiera menu poziome wyrównane do prawej strony oraz układy służące do wyświetlania analizy przypadków, szczegółów funkcji, wycen, danych firmy oraz danych kontaktowych. Strona startowa zawiera obracający się obraz banera oraz tekst.
<b>Produkty i usługi - przegląd</b>	Szablon "Produkty i usługi - przegląd" zawiera układy służące do wyświetlania m.in. analizy przypadków, szczegółów dot. oferowanych produktów, danych firmy i zasad ochrony prywatności. Strona startowa zawiera obracający się obraz banera oraz tekst. Szablon jest szablonem w pełni reagującym.
<b>Szablon początkowy</b>	Szablon początkowy służy do tworzenia własnych, gotowych do użycia serwisów. Dostarcza prosty, w pełni funkcjonalny przykład, który można eksplorować i rozszerzać o składniki i interakcje. Zawiera strukturę kodową serwisu, serwis domyślny z przykładowymi stronami i zawartością, motyw ze stylami oraz zasoby (takie jak obrazy) oraz składniki niestandardowe z funkcjonalnymi wyzwalaczami i czynnościami. Przykładowe strony zawierają informacje o tworzeniu szablonów oraz łączy do zasobów dostarczających więcej szczegółowych informacji.
<b>Szkolenie Prezentacja Pokaz</b>	Szablony te różnią się kolorystyką. Są szablonami responsywnymi i w każdej grupie mają składnik menu "Niestandardowy kod JavaScript". Są wyposażone w niestandardowe grupy składników standardowych. Wyświetlają również zalogowanego użytkownika.
<b>Dostęp Wiedza Relacja</b>	Szablony te różnią się kolorystyką. Są szablonami responsywnymi i w każdej grupie mają składnik menu "Niestandardowy kod JavaScript". Są wyposażone w niestandardowe grupy składników standardowych.
<b>Udostępnianie</b>	Szablon ten zawiera serwis z jedną, długą stroną. Jest szablonem responsywnym, wyposażonym w składnik menu "Niestandardowy kod JavaScript", prowadzący do różnych miejsc na stronie (nie do innych stron).
<b>Szablon początkowy JET</b>	Oracle JET (JavaScript Extension Toolkit) to modułowy zestaw narzędzi typu "open source" wykorzystujący biblioteki JavaScript typu "open source". Szczegółowe informacje są dostępne na stronie <a href="#">Oracle Jet</a> .

Można także utworzyć szablon z istniejącego serwisu bądź wyeksportować istniejący szablon, zmodyfikować go w trybie offline i zaimportować jako nowy szablon. Firma lub organizacja może mieć własne szablony przeznaczone do użycia przez jej pracowników.

#### Jak działają szablony

Podczas tworzenia serwisu szablon jest używany w następujący sposób:

- Jeśli funkcja zarządzania serwisem jest włączona, serwisy — zanim zostaną utworzone — mogą wymagać zatwierdzenia. Zob. [Omówienie funkcji zarządzania serwisem](#).

- Serwis domyślny (z szablonu) jest kopiowany do nowego serwisu i stanowi punkt startowy dla tworzonych stron.
- Wszystkie potrzebne pliki wspomagające są kopiowane do nowego serwisu.
- Jeśli w folderze motywów nie ma motywu szablonu, motyw jest kopiowany do folderu motywów. Serwis odwołuje się do motywu znajdującego się w folderze motywów. Jeśli motyw istnieje, to nowy serwis odwołuje się do niego.
- Jeśli istnieją składniki niestandardowe, których jeszcze nie ma w folderze składników, to są one kopiowane do tego folderu. Serwis odwołuje się do składników znajdujących się w folderze składników. Jeśli jakikolwiek składnik istnieje, to nowy serwis odwołuje się do niego.

### Tworzenie szablonów niestandardowych

Szablon zawiera w jednym pakiecie wszystkie elementy (serwis, układ, nawigacja, przykładowa zawartość itd.) potrzebne programiście do konstruowania serwisu internetowego. Można dodawać składniki i interakcje do serwisu, udostępniając w ten sposób gotowe rozwiązania spełniające określone potrzeby biznesowe.

Szablon jest reprezentowany przez strukturę folderów, z którą można pracować jak z innymi folderami. W Oracle Sites Cloud Service do niektórych elementów szablonu (takich jak motyw i składniki niestandardowe) występują odwołania z powiązanych lokalizacji w Oracle Content Management. Na przykład szablon odwołuje się do powiązanego motywu (z listy dostępnych motywów) w taki sam sposób, w jaki serwis odwołuje się do tego motywu.

Jeśli szablon jest tworzony z istniejącego serwisu, to w nowym szablonie — jako serwis domyślny — jest używana kopia tego serwisu. Szablon odwołuje się do motywu używanego przez serwis i do wszystkich składników niestandardowych, które są używane na stronach serwisu. Motyw i składniki niestandardowe nie są kopiowane do szablonu, lecz występują do nich odwołania (tak samo jak w przypadku serwisu).

#### Uwaga:

Szablon odzwierciedla serwis, który został użyty podczas tworzenia szablonu. Dalsze zmiany, dokonywane w serwisie użytym do utworzenia szablonu, nie są odzwierciedlane w serwisie zdefiniowanym w szablonie.

Można tworzyć szablony zawartości, które mogą być używane do udostępniania modeli zawartości, zawierających układy zawartości, elementy zawartości i zasoby cyfrowe, potrzebne do obsługi modelu zawartości (takiego jak przykładowa zawartość). Szablony zawartości powinny być tworzone na podstawie serwisów z opublikowanymi elementami zawartości i zasobami cyfrowymi. Należy zwrócić uwagę, że typy zawartości nie są tworzone, gdy szablon serwisu jest importowany z pakietu z zawartością. Typy zawartości są tworzone, gdy serwis jest tworzony z szablonu zawartości. Dlatego właścicielem tych typów nie jest użytkownik, który zaimportował szablon serwisu. Właścicielem będzie użytkownik, który utworzy pierwszy serwis z tego szablonu. Jeśli trzeba, użytkownik ten może udostępnić szablon innym użytkownikom. W zależności od typu subskrypcji usługi Content Management Cloud oraz od daty rozpoczęcia usługi, funkcja ta może być niedostępna.

Gdy szablon jest eksportowany, wszystkie jego elementy (wraz z kopią motywu i wszystkimi składnikami) są gromadzone w pakiecie, który można pobrać i z którym można pracować w trybie offline.

Jeśli jest importowany szablon, który został zmodyfikowany w trybie offline, i okaże się, że nazwa lub ID szablonu, motywu lub składnika niestandardowego już istnieje, to pojawi się wezwanie do rozwiązania konfliktu. Może zostać udostępniona opcja utworzenia nowego szablonu, motywu lub składnika niestandardowego albo — w niektórych przypadkach — opcja zastąpienia istniejącego szablonu, motywu lub składnika niestandardowego importowaną wersją. Zob. [Tworzenie szablonów](#).

Szablon można także zaimportować do określonego repozytorium. Podczas importu użytkownik ma możliwość wyboru, czy już istniejące zasoby mają zostać zaktualizowane czy zduplikowane. Zob. [Importowanie szablonów do określonego repozytorium](#).

#### Udostępnianie szablonów

Po utworzeniu szablonu (poprzez importowanie, kopiowanie lub utworzenie z serwisu), nikt nie może z niego korzystać, dopóki szablon ten nie zostanie udostępniony.

#### Uwaga:

Dotyczy to również szablonów oferowanych w usłudze Oracle Content Management i zainstalowanych przez administratora. Jeśli nie są widoczne żadne szablony, należy się skontaktować z administratorem. Prawdopodobnie nie zostały one udostępnione.

Jeśli szablon jest udostępniany użytkownikowi po raz pierwszy, są mu automatycznie udostępniane powiązany motyw i wszystkie powiązane składniki niestandardowe oraz jest mu nadawana rola "Pobierający", dzięki czemu są one dostępne, gdy użytkownik tworzy serwis z szablonu. Kolejne zmiany w szablonie, odnoszące się do roli użytkownika, nie aktualizują informacji dotyczących udostępniania, związanych z powiązaniem motywem ani składnikami niestandardowymi.

Jeśli funkcja zarządzania serwisem jest włączona, szablony można uczynić dostępnymi za pomocą założeń systemowych dot. szablonów. Zob. [Omówienie funkcji zarządzania serwisem](#) i [Zmianie założeń systemowych dot. szablonów](#).

## Tworzenie szablonu z serwisu

Mając serwis, który może zostać użyty jako punkt startowy dla innych serwisów, można z tego serwisu utworzyć szablon.

#### Uwaga:


Szablon z serwisu może utworzyć tylko osoba mająca (w odniesieniu do tego serwisu) przypisaną rolę "Pobierający", "Współtwórca" lub "Menedżer". Ponadto administrator musi włączyć odpowiednie opcje w menu **Utwórz**. Jeśli na stronie "Szablony" nie jest widoczne menu **Utwórz**, należy się skontaktować z administratorem.

Przedstawiono tu, jak utworzyć szablon z serwisu. Można także zaimportować pakiet szablonu utworzony lub zmodyfikowany w trybie offline. Zob. [Eksportowanie i importowanie szablonów](#).

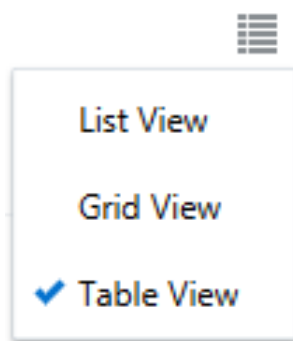
 **Uwaga:**



Jeśli szablon jest tworzony z serwisu, który używa zawartości z więcej niż jednego repozytorium, szablon ten trzeba utworzyć przy użyciu narzędzia OCE Toolkit. Zob. [Tworzenie za pomocą zestawu OCE Toolkit](#) i [Korzystanie z narzędzia cec opartego na wierszu polecenia](#).

Aby utworzyć szablon z serwisu za pomocą interfejsu internetowego Oracle Content Management, należy:

1. Wybrać opcję **Serwisy**, po czym wybrać serwis, który ma zostać użyty.
2. Wybrać z menu podręcznego opcję **Utwórz szablon** albo kliknąć na ikonie  (na pasku czynności).
3. Podać nazwę szablonu. Może się ona składać tylko z liter, cyfr, znaków podkreślenia ( \_ ) i łączników (-). Użyta spacja zostanie automatycznie zastąpiona łącznikiem.  
  
Dla szablonów, motywów, składników, serwisów ani stron serwisów nie wolno używać następujących nazw: authsite, content, pages, scstemplate\_\*, \_comps, \_components, \_compsdelivery, \_idcservice, \_sitescloud, \_sitesclouddelivery, \_themes, \_themesdelivery. Następujących nazw można używać dla stron serwisów, ale nie można używać dla szablonów, motywów, składników ani serwisów: documents, sites.
4. Podać dodatkowy opis szablonu.
5. Wybrać, czy w szablonie będą uwzględniane nieopublikowane elementy zawartości i zasoby cyfrowe.
6. Po ukończeniu nacisnąć przycisk **Utwórz szablon**.

Gdy szablon zostanie utworzony, jego nazwa pojawi się na liście szablonów. Aby wyświetlić wszystkie szablony, należy wybrać kolejno opcje **Programista** i **Wyświetl wszystkie szablony**. Można określić, jak szablony będą wyświetlane, klikając na ikonie "Widok", po czym wybierając odpowiednią opcję z listy.



Aby wyświetlić foldery i pliki powiązane z szablonem, wybrać z menu podręcznego (kliknięcie prawym przyciskiem myszy) opcję **Otwórz** albo kliknąć na ikonie  (na pasku czynności). Aby wyświetlić lub zmienić nazwę, opis i inne szczegóły dotyczące szablonu, kliknąć na nazwie szablonu lub wybrać z menu podręcznego opcję **Szczegóły** albo kliknąć na ikonie  (na pasku czynności).

W nowym szablonie — jako serwis domyślny — jest używana kopia tego serwisu. Szablon odwołuje się do motywu używanego przez serwis i do wszystkich składników

niestandardowych, które są używane na stronach serwisu. Motyw i składniki niestandardowe nie są kopiowane do szablonu, lecz występują do nich odwołania (tak samo jak w przypadku serwisu).

Jeśli szablon został utworzony z serwisu firmowego, założenie systemowe dot. lokalizacji i język domyślny (używane przez serwis pierwotny) będą domyślnie wybierane dla wszystkich nowych serwisów utworzonych z tego szablonu.

Szablon odzwierciedla serwis, który został użyty do utworzenia szablonu w chwili jego tworzenia. Wszelkie przyszłe zmiany, dokonywane w serwisie użytym do utworzenia szablonu, nie są odzwierciedlane w serwisie zdefiniowanym w szablonie.

## Zmianianie szczegółów szablonu



Na karcie "Szczegóły szablonu" są pokazywane następujące informacje: nazwa szablonu, autor, opis, motyw szablonu, składniki niestandardowe dołączone do szablonu oraz obrazy podglądu szablonu. Jeśli funkcja zarządzania serwisem jest włączona, jest prezentowany także status szablonu (czy szablon jest dostępny do użycia podczas tworzenia serwisów).

Mając odpowiednie uprawnienia, można zmieniać lub aktualizować właściwości szablonu, takie jak nazwa i opis. Można także dodawać lub usuwać obrazy podglądu szablonu.


Szczegóły szablonu mogą być aktualizowane przez jego twórcę (właściciela szablonu), administratora serwisów lub osobę, której ten szablon został udostępniony i której została przypisana rola "Współtwórca" lub "Menedżer".

Jeśli zostaną dodane obrazy podglądu, będą one przechowywane w folderze zasobów szablonu. Pliki te nie pojawiają się w żadnym powiązanim serwisie internetowym, lecz umożliwiają właścicielowi szablonu udostępnianie informacji o samym szablonie.

Aby wyświetlić lub zmienić szczegóły szablonu, należy:

1. Wybrać kolejno opcje **Programista i Wyświetl wszystkie szablony**.
2. Wybrać szablon, po czym wybrać z menu podręcznego opcję **Szczegóły** albo kliknąć na ikonie  (na pasku czynności).
3. Odpowiednio zmodyfikować dane. Jeśli nie można edytować informacji, oznacza to, że bieżący użytkownik nie ma przypisanej roli "Pobierający", "Współtwórca" ani "Menedżer".
4. Aby dodać obraz podglądu, kliknąć na ikonie . Odszukać obraz, który ma zostać użyty, albo nacisnąć przycisk **Wyślij**, aby wysłać obraz przechowywany lokalnie. Rozmiar obrazu powinien wynosić przynajmniej 180 na 135 pikseli (proporcje 4:3). Mniejsze obrazy mogą nie być optymalnie wyświetlane, a większe mogą mieć wpływ na wydajność.

Jeśli wszystkie miejsca dla obrazów podglądu zostaną wypełnione, to — aby można było dodać nowy obraz — trzeba usunąć jeden z obrazów. Aby usunąć

obraz podglądu, kliknąć na ikonie  obrazu.


5. Nacisnąć przycisk **Zapisz**.

## Zmianianie założeń systemowych dot. szablonów

Na karcie "Założenia systemowe dot. szablonów" są pokazywane typ szablonu (standardowy lub firmowy), wymagane zatwierdzenie oraz minimalne zabezpieczenia dla serwisów utworzonych przy użyciu szablonu. Jest prezentowany także status szablonu (czy szablon jest dostępny do użycia podczas tworzenia serwisów).

Karta "Założenia systemowe dot. szablonów" jest widoczna tylko wtedy, gdy funkcja zarządzania serwisem jest włączona, a bieżący użytkownik jest administratorem serwisów. Zob. [Omówienie funkcji zarządzania serwisem](#).

Aby wyświetlić lub zmienić założenia systemowe dot. szablonów, należy:

1. Wybrać kolejno opcje **Programista** i **Wyświetl wszystkie szablony**.
2. Wybrać szablon, po czym wybrać z menu podręcznego opcję **Szczegóły** albo kliknąć na ikonie  (na pasku czynności).
3. Wybrać kartę **Założenia systemowe**.
4. Wybrać, czy wystąpienia o serwis, utworzone przy użyciu tego szablonu, wymagają zatwierdzenia przez administratora serwisów, czy są automatycznie zatwierdzane lub czy wymagają zatwierdzenia przez określone osoby. Aby ograniczyć zatwierdzanie do określonych osób, zacząć wpisywać nazwę lub adres e-mail osoby bądź grupy, która ma zostać dodana jako zatwierdzający, po czym wybrać z wyników wyszukiwania odpowiednią osobę lub grupę. Aby usunąć osobę zatwierdzającą, kliknąć na znaku **X** obok jej nazwy.
5. Wybrać minimalne zabezpieczenia wymagane dla serwisów utworzonych przy użyciu tego szablonu. Jeśli wskazane, twórcy serwisów mogą wybrać wyższy poziom zabezpieczeń dla swoich serwisów.
  - **Określeni użytkownicy usługi** — Dostęp do serwisu mogą uzyskać tylko wybrani użytkownicy, którzy mogą się zalogować do tej instancji Oracle Content Management. Po utworzeniu serwisu jego twórca wybiera użytkowników Oracle Content Management. Zob. [Zmianianie zabezpieczeń serwisu](#).
  - **Określeni użytkownicy chmury** — Dostęp do serwisu mogą uzyskać tylko wybrani użytkownicy, którzy mogą się zalogować do domeny. Po utworzeniu serwisu jego twórca wybiera użytkowników chmury. Zob. [Zmianianie zabezpieczeń serwisu](#).
  - **Użytkownicy usługi** — Dostęp do serwisu może uzyskać każdy użytkownik, który może się zalogować do tej instancji Oracle Content Management.
  - **Użytkownicy chmury** — Dostęp do serwisu może uzyskać każdy użytkownik, który może się zalogować do domeny.
  - **Każdy** — Dostęp do serwisu może uzyskać dowolna osoba bez logowania się.
6. Aby określić, kiedy używający szablonu serwis ma wygasnąć, wybrać założenie systemowe dot. wygasania. Administratorzy serwisów mogą określić, czy wygasłe serwisy mają być przełączane do trybu offline czy usuwane. Właściciele i menedżerowie serwisów przed wygaśnięciem serwisu otrzymują powiadomienie e-mailowe oraz uzyskują możliwość przesunięcia daty wygasania. Jeśli nie przedłużą czasu wygasania, otrzymają powiadomienie e-mailowe, gdy serwis zostanie przełączony do trybu offline lub usunięty. W tym czasie mogą przedłużyć termin wygaśnięcia, a następnie przełączyć serwis do trybu online lub — jeśli trzeba — przywrócić go z Kosza. Są dostępne następujące opcje założenia systemowego dot. wygasania:



- **Nigdy**
- **rok**
- **2 lata**
- **Niestandardowe**  
Wybór opcji **Niestandardowe** umożliwia ustawienie okresu wygasania krótszego niż rok lub dłuższego niż 2 lata.

 **Uwaga:**

Po utworzeniu serwisu administratorzy serwisów mogą w oknie dialogowym **Właściwości serwisu** zmienić założenie systemowe serwisu dot. wygasania.


7. Edytując szablon firmowy, trzeba wybrać, jak będzie tworzony prefiks serwisu dla wartości przyjaznego adresu URL. Prefiks ten będzie dołączany do krótkiego opisu elementu zawartości (część adresu URL właściwa dla strony lub zasobu). Prefiks może być automatycznie generowany na podstawie nazwy serwisu; można także umożliwić użytkownikom wpisywanie prefiksu.
8. Jeśli jest edytowany szablon firmowy, wybrać założenie systemowe dot. lokalizacji używane do określenia, które języki są wymagane dla serwisu.
9. Po ukończeniu nacisnąć przycisk **Zapisz**.

## Zmianianie statusu lub odbiorców szablonu

Na karcie "Odbiorcy szablonu" są prezentowane status szablonu (czy szablon jest dostępny do użycia podczas tworzenia serwisów) oraz informacja, kto może używać szablonu do tworzenia serwisów.

Karta "Odbiorcy szablonu" jest widoczna tylko wtedy, gdy funkcja zarządzania serwisem jest włączona, a bieżący użytkownik jest administratorem serwisów. Zob. [Omówienie funkcji zarządzania serwisem](#).

Aby wyświetlić lub zmienić odbiorców szablonu, należy:

1. Wybrać kolejno opcje **Programista** i **Wyświetl wszystkie szablony**.
2. Wybrać szablon, po czym wybrać z menu podręcznego opcję **Szczegóły** albo kliknąć na ikonie  (na pasku czynności).
3. Wybrać kartę **Odbiorcy**.
4. Aby zmienić status szablonu, kliknąć na przełączniku statusu. Aby można było tworzyć serwisy przy użyciu szablonu, trzeba uczynić go aktywnym dla użytkowników. Domyślnie, gdy szablon zostanie aktywowany, stanie się on dostępny dla wszystkich twórców serwisów. Chcąc zmienić osoby, które mogą używać szablonu, trzeba zmienić ustawienia na tej karcie.
5. Wybrać, kto może używać tego szablonu do tworzenia serwisów.
  - **Określone osoby** — Tylko wybrane osoby mogą używać tego szablonu do tworzenia serwisów. Należy rozpocząć wprowadzać nazwę (lub adres e-mail) osoby bądź grupy, która będzie mogła używać tego szablonu, po czym wybrać

z wyników wyszukiwania odpowiednią osobę lub grupę. Aby usunąć osobę lub grupę, kliknąć na znaku **X** obok jej nazwy.

- **Każdy** — Każdy, kto może tworzyć serwisy, będzie mógł używać tego szablonu.

6. Po ukończeniu nacisnąć przycisk **Zapisz**.

## Zarządzanie szablonami

Folder szablonu można skopiować lub usunąć, a także można zmienić jego nazwę (podobnie jak w przypadku każdego innego folderu), lecz gdy jest importowany lub udostępniany, wymaga specjalnego traktowania.




Należy otworzyć stronę szablonu, wybierając kolejno opcje **Programista i Wyświetl wszystkie szablony**. Do wykonania przedstawionych poniżej zadań służą opcje z paska czynności lub z menu podręcznego ze strony "Szablony".

Zadanie	Opis
Tworzenie szablonu	Mając serwis, który może zostać użyty jako punkt startowy dla innych serwisów, można z tego serwisu utworzyć szablon.

### Uwaga:


Trzeba mieć, w odniesieniu do serwisu, przypisaną rolę "Pobierający", "Współtwórca" lub "Menedżer", a administrator musi tę opcję włączyć. Jeśli na stronie "Szablony" nie jest widoczne menu **Utwórz**, należy się skontaktować z administratorem. Ponadto, jeśli szablon jest tworzony z serwisu, który używa zawartości z więcej niż jednego repozytorium, szablon ten trzeba utworzyć przy użyciu narzędzia OCE Toolkit. Zob. [Tworzenie za pomocą zestawu OCE Toolkit](#) i [Korzystanie z narzędzia cec opartego na wierszu polecenia](#).

Na stronie "Szablony" nacisnąć przycisk **Utwórz**, po czym wybrać opcję **Utwórz z istniejącym serwisem**.  
 . Wybrać serwis, nadać nazwę szablonowi, po czym nacisnąć przycisk **Utwórz**.  
 . Zob. [Tworzenie szablonu z serwisu](#).



Zadanie	Opis
Wyświetlanie i edytowanie plików szablonów	<p>Foldery i pliki, wchodzące w skład szablonu, można eksplorować, wybierając szablon, a następnie wybierając z menu podręcznego opcję <b>Otwórz</b> albo klikając na pasku czynności na ikonie .</p> <p>Będąc twórcą oprogramowania internetowego, można pobierać i modyfikować poszczególne pliki szablonu; można także, korzystając z aplikacji typu Desktop, synchronizować wszystkie zmiany dokonywane w systemie lokalnym. Można także wyeksportować pakiet szablonu zawierający powiązany motyw i wszystkie niestandardowe składniki, po czym pracować nad nimi w trybie offline w preferowanym środowisku programistycznym.</p> <p>Szablon jest reprezentowany przez strukturę folderów, z którą można pracować jak z innymi folderami.</p> <p>Gdy szablon jest eksportowany, wszystkie jego elementy (wraz z kopią motywu i wszystkimi składnikami) są gromadzone w pakiecie, który można pobrać i z którym można pracować w trybie offline. Zob. <a href="#">Tworzenie szablonów</a>.</p>
Tworzenie serwisu	<p>Aby utworzyć serwis z szablonu, należy wybrać szablon, po czym wybrać z menu podręcznego opcję <b>Utwórz serwis</b> albo kliknąć na ikonie  (na pasku czynności). Zob. <a href="#">Tworzenie serwisów</a>.</p>
Zmianie nazwy szablonu	<p>Wybrać szablon, którego nazwa ma zostać zmieniona, po czym wybrać z menu podręcznego opcję <b>Zmień nazwę</b> albo kliknąć na ikonie  (na pasku czynności). Następnie wpisać nazwę szablonu, różniącą się od każdej nazwy szablonu na tym samym serwerze. Wytyczne dotyczące nazw są dostępne pod hasłem <a href="#">Tworzenie szablonu z serwisu</a>.</p>
Kopiowanie szablonu	<p>Można utworzyć szablon, kopiując istniejący szablon i dokonując zmian w kopii.</p>



 **Uwaga:**



Gdy szablon jest kopiowany, nie są kopiowane informacje dotyczące jego udostępniania.

Wybrać szablon, który ma zostać skopiowany, po czym wybrać z menu podręcznego opcję **Kopiuj** albo kliknąć na ikonie  (na pasku czynności). Wpisać nazwę szablonu, różniącą się od każdej nazwy szablonu na tym samym serwerze. Inne wytyczne dotyczące nazw są dostępne pod hasłem [Tworzenie szablonu z serwisu](#)

Na pasku postępu są pokazywane nowa nazwa szablonu i status operacji kopiowania.

Zadanie	Opis
Usuwanie lub przywracanie szablonu	<p>Mając odpowiednie uprawnienia, można usunąć folder szablonu i jego zawartość. Gdy szablon zostanie usunięty, jego folder oraz wszystkie powiązane foldery i pliki są przenoszone do Kosza.</p> <p>Szablon może zostać usunięty lub przywrócony przez jego twórcę (właściciela szablonu) lub osobę, której ten szablon został udostępniony i której została przypisana rola współtwórcy lub menedżera.</p>
	<div style="border: 1px solid #0070C0; padding: 10px; background-color: #E6F2FF;"> <p> <b>Uwaga:</b></p> <p>Usuwanie szablonu nie usuwa się powiązanych z nim motywów i składników niestandardowych.</p> </div>
	<p>Aby usunąć szablon, należy wybrać szablon, po czym wybrać z menu podręcznego opcję <b>Usuń</b> albo kliknąć na ikonie  (na pasku czynności). Zostanie wyświetlone pytanie, czy szablon ma zostać przeniesiony do Kosza. Usunięty szablon pozostaje w Koszu, aż:</p> <ul style="list-style-type: none"> <li>• Szablon zostanie przywrócony.</li> <li>• Szablon zostanie trwale usunięty.</li> <li>• Zostanie osiągnięty przydział miejsca na Kosz.</li> <li>• Kosz zostanie automatycznie opróżniony zgodnie z interwałem ustawionym przez administratora serwisu. Wartością domyślną jest 90 dni.</li> </ul> <p>Aby przywrócić szablon, należy wybrać z rozwijanej listy "Szablony" opcję <b>Kosz</b>, następnie wybrać szablon, po czym wybrać z paska menu lub z menu podręcznego (kliknięcie prawym przyciskiem myszy) opcję <b>Przywróć</b>.</p>
Eksportowanie lub importowanie szablonu	<p>Można wyeksportować szablon i zmodyfikować go offline, a następnie zaimportować go jako nowy szablon albo jako zamiennik istniejącego szablonu. Można także wyeksportować szablon, aby przenieść go (zaimportować) do innej instancji.</p> <p>Eksportując szablon, kopiuje się go do folderu jako plik .zip. Pakiet szablonu można pobrać bezpośrednio z tego folderu, a następnie rozpakować pobrany pakiet i pracować nad poszczególnymi plikami. Po ukończeniu pracy nad plikami szablonu można utworzyć plik .zip zawierający pakiet szablonu i zaimportować go, zastępując szablon pierwotny lub tworząc nowy.</p> <p>Zob. <a href="#">Eksportowanie i importowanie szablonów</a></p>

Zadanie	Opis
Udostępnianie szablonu i zarządzanie rolami współużytkowników	<p>Będąc właścicielem lub menedżerem szablonu, można udostępnić szablon innym użytkownikom oraz przypisać im rolę określającą, co użytkownik może z tym szablonem robić.</p> <p>Jeśli funkcja zarządzania <i>nie jest</i> włączona, to przypisanie dowolnej roli użytkownikowi daje mu uprawnienie do tworzenia serwisu z szablonu. Jeśli funkcja zarządzania <i>jest</i> włączona, trzeba — aby można było nadać użytkownikowi uprawnienie do tworzenia serwisu z szablonu — aktywować szablon i upewnić się, że użytkownik został uwzględniony w odbiorcach szablonu. Zob. <a href="#">Zmienianie statusu lub odbiorców szablonu</a>.</p> <p>Motywy i składniki niestandardowe szablonu są przechowywane jako osobne obiekty — w szablonie występują odwołania do nich. Dlatego współużytkowanie motywów i składników niestandardowych jest zarządzane osobno. Jeśli bieżący użytkownik udostępnia szablon innemu użytkownikowi, to — niezależnie od roli, jaką ten użytkownik ma przypisaną w odniesieniu do tego szablonu — jest mu automatycznie nadawana rola "Pobierający" w odniesieniu do wszelkich powiązanych motywów, tak że są one dostępne, gdy ten użytkownik tworzy serwis z szablonu.</p> <p>Wybrać szablon, który ma zostać udostępniony, po czym wybrać z menu podręcznego lub z paska czynności opcję <b>Współużytkownicy</b>.</p> <p>Podać adres e-mail lub adresy e-mail oraz przypisać jedną z następujących ról:</p> <ul style="list-style-type: none"> <li>• <b>Oglądający:</b> Oglądający mogą wyświetlać foldery i pliki wchodzące w skład szablonu, lecz nie mogą niczego zmieniać.</li> <li>• <b>Pobierający:</b> Pobierający może dodatkowo pobierać pliki i zapisywać je w swoim komputerze.</li> <li>• <b>Współtwórca:</b> Współtwórcy mogą dodatkowo edytować szczegóły i pliki szablonu, wysyłać nowe pliki, usuwać pliki, a także usunąć cały szablon.</li> <li>• <b>Menedżer:</b> Menedżerowie mogą także dodawać użytkowników i przypisywać im role. Właścicielowi (twórcy) szablonu jest automatycznie przypisywana rola "Menedżer".</li> </ul>
Wyświetlanie lub edytowanie szczegółów szablonu	<p>Aby wyświetlić szczegóły szablonu, należy go wybrać, po czym wybrać z menu podręcznego opcję <b>Szczegóły</b> albo kliknąć na ikonie  (na pasku czynności). Szczegóły szablonu obejmują nazwę, autora, pliki pomocnicze, opis, motyw, składniki niestandardowe i miniatury szablonu. Mając przypisaną odpowiednią rolę w odniesieniu do szablonu, można edytować szczegóły. Zob. <a href="#">Zmienianie szczegółów szablonu</a>.</p>
Wyświetlanie lub edytowanie założeń systemowych dot. szablonów (z włączoną funkcją zarządzania)	<p>Jeśli funkcja zarządzania serwisem jest włączona, a bieżący użytkownik jest administratorem serwisów, można wyświetlić założenia systemowe dot. szablonów (takie jak wymagane zatwierdzenie, minimalne zabezpieczenia, wygaśnięcie serwisu i lokalizacja). Należy wybrać szablon, po czym wybrać z menu podręcznego opcję <b>Szczegóły</b> albo kliknąć na ikonie  (na pasku czynności), a następnie kliknąć na karcie <b>Założenia systemowe</b>. Zob. <a href="#">Zmienianie założeń systemowych dot. szablonów</a>.</p>

Zadanie	Opis
Czynienie szablonu dostępnym przy tworzeniu serwisu oraz zarządzanie osobami, które mogą używać szablonu do tworzenia serwisów (z włączoną funkcją zarządzania)	<p>Jeśli funkcja zarządzania jest włączona, użytkownik — o ile jest administratorem serwisów — może uczynić szablon dostępnym przy tworzeniu serwisów oraz zarządzać osobami, które mogą używać szablonu do tworzenia serwisów. Obok aktywnych szablonów (dostępnych przy tworzeniu serwisu) pojawia się ikona .</p> <p>Aby zmienić status lub odbiorcę, należy wybrać szablon, po czym wybrać z menu podręcznego opcję <b>Szczegóły</b> albo kliknąć na ikonie  (na pasku czynności).</p> <ul style="list-style-type: none"> <li>Aby uczynić szablon dostępnym przy tworzeniu serwisu, kliknąć — w celu aktywowania szablonu — na przełączniku statusu.</li> <li>Aby zmienić osoby, które mogą używać szablonu do tworzenia serwisów, kliknąć na karcie <b>Odbiorcy</b>. Zob. <a href="#">Zmianie statusu lub odbiorców szablonu</a>.</li> </ul>

## Eksportowanie i importowanie szablonów

Można wyeksportować szablon i zmodyfikować go offline, a następnie zaimportować go jako nowy szablon albo jako zamiennik istniejącego szablonu. Można także wyeksportować szablon, aby przenieść go (zaimportować) do innej instancji Oracle Content Management.

### Eksportowanie szablonów

Eksportując szablon, kopiuje się go do folderu usługi Oracle Content Management jako plik .zip. Pakiet szablonu można pobrać bezpośrednio z tego folderu, po czym rozpakować pakiet i pracować z poszczególnymi plikami. Po ukończeniu pracy z plikami szablonu można utworzyć plik .zip zawierający pakiet szablonu i zaimportować go do swojego serwisu, zastępując szablon pierwotny lub tworząc nowy.

#### Uwaga:

Gdy szablon jest eksportowany, nie są dołączane informacje dotyczące jego udostępniania.

Aby wyeksportować szablon, należy:

- Wybrać kolejno opcje **Programista** i **Wyświetl wszystkie szablony**.
- Wybrać szablon, po czym wybrać z paska menu lub z menu podręcznego (kliknięcie prawym przyciskiem myszy) opcję **Eksportuj**.
- Przejsć do istniejącego folderu albo utworzyć nowy folder: nacisnąć przycisk **Utwórz**, a następnie podać nazwę i opcjonalny opis.
- Wybrać folder: zaznaczyć jego pole wyboru, po czym nacisnąć przycisk **OK**.

## Importowanie szablonów

### Uwaga:

Jeśli szablon zostanie zaimportowany do innego serwera, niektóre łącza domyślnego serwisu mogą być niepoprawne w kontekście nowego serwera. Jeśli serwis zawiera łącza referencyjne do obrazów lub innej zawartości, a nie kopiuje zawartości bezpośrednio do serwisu, to zawartość ta nie jest dostępna w nowym serwerze. Zawartość ta, nawet jeśli zostanie skopiowana do nowego serwera, będzie miała inny wewnętrzny ID (łącza nie będzie poprawne). Podczas importowania szablonu użytkownik zostanie powiadomiony o stronach zawierających niepoprawne łącza referencyjne.

Aby zaimportować pakiet szablonu:

1. Wybrać kolejno opcje **Programista** i **Wyświetl wszystkie szablony**.
2. Nacisnąć przycisk **Utwórz**, po czym wybrać opcję **Importuj pakiet szablonu**.
3. Mając wysłany pakiet szablonu, przejść do folderu (zawierającego ten pakiet szablonu) i otworzyć go. Jeśli pakiet szablonu nie został jeszcze wysłany, przejść do folderu, do którego ma zostać wysłany szablon lub utworzyć nowy folder. Nacisnąć przycisk **Wyślij**, a następnie odszukać i wybrać pakiet szablonu, po czym nacisnąć przycisk **Otwórz**.
4. Aby można było używać szablonu, zaznaczyć pole wyboru obok nazwy pliku szablonu, po czym nacisnąć przycisk **OK**. Dla szablonu zostają utworzone nowe foldery, wraz z powiązaniem motywem i innymi składnikami niestandardowymi. Jeśli okaże się, że nazwa lub ID szablonu, motywu lub składnika niestandardowego już istnieje, pojawi się wezwanie do rozwiązania konfliktu. Może pojawić się potrzeba utworzenia nowego szablonu, motywu lub składnika albo zastąpienia istniejących elementów importowaną wersją.

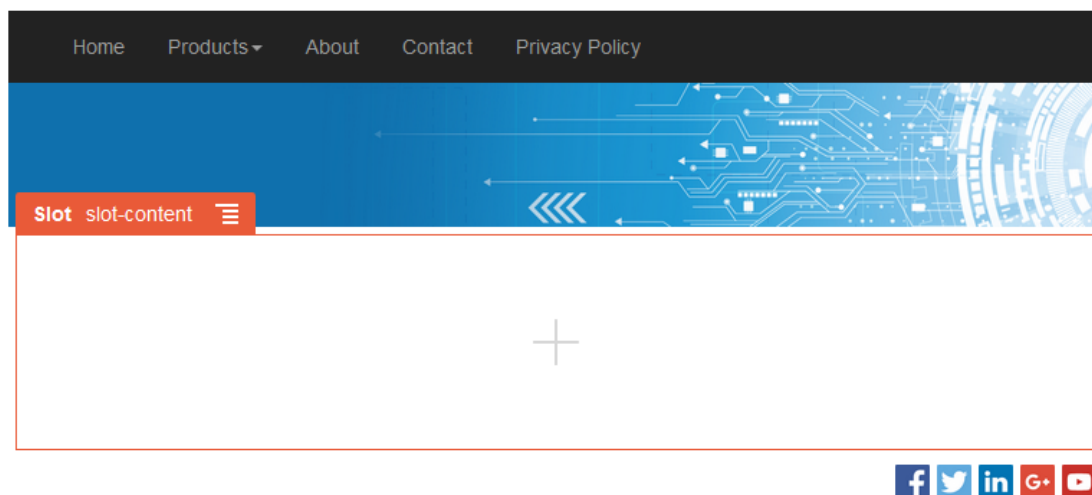
### Importowanie szablonów do określonego repozytorium

Można wybrać repozytorium, a następnie zaimportować do niego pakiet szablonu.

1. Nacisnąć przycisk **Zawartość**, po czym z menu wybrać opcję **Repozytoria**.
2. Wybrać repozytorium, do którego ma zostać zaimportowany pakiet szablonu, po czym nacisnąć przycisk **Importuj zawartość**.
3. Mając wysłany pakiet szablonu, przejść do folderu (zawierającego ten pakiet szablonu) i otworzyć go. Jeśli pakiet szablonu nie został jeszcze wysłany, przejść do folderu, do którego ma zostać wysłany szablon lub utworzyć nowy folder. Nacisnąć przycisk **Wyślij**, a następnie odszukać i wybrać pakiet szablonu, po czym nacisnąć przycisk **OK**.
4. Określić, czy zasoby, które już istnieją w repozytorium, mają być aktualizowane przy użyciu nowych wersji bądź czy mają być duplikowane, po czym nacisnąć przycisk **Importuj**.
5. Aby wyświetlić listę zasobów i typów zawartości, które zostały zaimportowane, nacisnąć przycisk **Szczegóły**. Po ukończeniu nacisnąć przycisk **OK**.

## Omówienie motywów

Motyw definiuje wygląd i ogólny styl serwisu, w tym schemat kolorów, rozmiar czcionek, typ czcionek i tła stron. Motywy zapewniają wizualną spójność stron serwisu. Możemy dostosować projekt i dodać zawartość, tworząc serwis prezentujący nasz styl, naszą markę i naszą wizję.



Motyw zawiera:

- Układy strony
- Arkusz stylów kaskadowych (CSS)
- Warianty arkusza stylów
- Pliki konfiguracyjne
- Kod tła definiujący nawigację serwisu

Każdy serwis internetowy korzysta z motywu. Gdy serwis jest tworzony z szablonu, z szablonu tego jest dziedziczony motyw. Używany dla serwisu motyw można w dowolnej chwili zmienić.

Oracle Content Management udostępnia różne, gotowe do użycia szablony, zawierające motywy. Aby utworzyć nowy motyw, wystarczy skopiować już istniejący. Można pobierać i modyfikować pliki z motywami bądź używać aplikacji typu Desktop i synchronizować zmiany dokonane w systemie lokalnym. Informacje o innych sposobach tworzenia motywów są dostępne pod hasłem [Tworzenie motywów](#).

Programiści mogą tworzyć motywy, w których jest używany określony podzbiór składników. Gdy serwis jest tworzony przy użyciu jednego z tych motywów, będą widoczne tylko te składniki, które są powiązane z tym motywem. Dzięki temu nie trzeba się zastanawiać, czy dany składnik jest czy nie jest poprawny dla wybranego motywu. Nadal można używać wszelkich standardowych składników oraz składników "Używane w motywie".

Jeśli serwis używa nowego, nieopublikowanego motywu, to zostanie on automatycznie opublikowany przy pierwszym przełączeniu serwisu do trybu online. Jeśli dokonamy zmian w motywie i zechcemy zaktualizować serwisy online, tak aby te zmiany zostały



uwzględnione, musimy motyw opublikować jawnie. Może to zrobić tylko właściciel motywu lub użytkownik z uprawnieniami menedżera.

 **Uwaga:**

Gdy motyw zostanie opublikowany, dokonane zmiany zostaną odzwierciedlone we wszystkich korzystających z niego serwisach online. Na przykład, jeśli w motywie zmienimy czcionkę domyślną, a następnie ten motyw opublikujemy, we wszystkich serwisach korzystających z tego motywu będzie używana nowa czcionka domyślna.

## Zarządzanie motywami

Folder motywu można skopiować lub usunąć, a także można zmienić jego nazwę (podobnie jak w przypadku każdego innego folderu), lecz motyw wymaga specjalnego traktowania, ponieważ może być współużytkowany przez więcej niż jeden serwis.

Można wyświetlić wszystkie motywy, wybierając kolejno opcje **Programista** i **Wyświetl wszystkie motywy**. Do wykonania przedstawionych poniżej zadań służą opcje z paska menu lub z menu podręcznego ze strony "Motywy".



Zadanie	Opis
Tworzenie nowego motywu	Aby utworzyć nowy motyw, wystarczy skopiować już istniejący. Informacje o innych sposobach tworzenia motywów są dostępne w podręczniku <a href="#">Tworzenie motywów</a> .
Kopiowanie motywu	Można utworzyć nowy motyw, kopiując istniejący motyw i dokonując zmian w kopii.

 **Uwaga:**

Gdy motyw jest kopiowany, nie są kopiowane informacje dotyczące jego udostępniania.

Wybrać motyw, który ma zostać skopiowany, po czym wybrać z paska menu lub z menu podręcznego opcję **Kopiuj**. Wprowadzić nazwę, która nie została użyta dla innego motywu. Może się ona składać tylko z liter, cyfr, znaków podkreślenia ( \_ ) i łączników (-). Użyta spacja zostanie automatycznie zastąpiona łącznikiem. Nacisnąć przycisk **Kopiuj**.

Dla szablonów, motywów, składników, serwisów ani stron serwisów nie wolno używać następujących nazw: authsite, content, pages, scstemplate\_\*, \_comps, \_components, \_compsdelivery, \_idcservice, \_sitescloud, \_sitesclouddelivery, \_themes, \_themesdelivery. Następujących nazw można używać dla stron serwisów, ale nie można używać dla szablonów, motywów, składników ani serwisów: documents, sites.

Zadanie	Opis
Udostępnianie motywu	<p>Jeśli administrator włączył udostępnianie, a bieżący użytkownik jest właścicielem lub menedżerem motywu, można udostępnić swój motyw innym użytkownikom. Udostępniając motyw, trzeba przypisać użytkownikowi rolę określającą, co można z tym motywem robić.</p> <div style="border: 1px solid #0070C0; padding: 10px; margin-top: 10px;"> <p> <b>Uwaga:</b></p> <p>Jeśli ktoś udostępnia szablon użytkownikowi, to — niezależnie od roli, jaką ma on przypisaną w odniesieniu do tego szablonu — jest mu automatycznie nadawana rola "Pobierający" w odniesieniu do wszelkich powiązanych motywów, tak że są one dostępne, gdy ten użytkownik tworzy serwis z szablonu.</p> </div> <p>Kliknąć prawym przyciskiem myszy na motywie, który ma zostać udostępniony, po czym wybrać opcję <b>Udostępniij</b>, a następnie nacisnąć przycisk <b>Dodaj współużytkowników</b>. Podać adres e-mail lub adresy e-mail oraz przypisać jedną z następujących ról:</p> <ul style="list-style-type: none"> <li>• <b>Oglądający:</b> Oglądający mogą wyświetlać foldery i pliki wchodzące w skład motywu, lecz nie mogą niczego zmieniać.</li> <li>• <b>Pobierający:</b> Pobierający może dodatkowo pobierać pliki i zapisywać je w swoim komputerze.</li> <li>• <b>Współtwórca:</b> Współtwórcy mogą dodatkowo edytować motyw, wysłać nowe pliki, usuwać pliki, a także usunąć cały motyw.</li> <li>• <b>Menedżer:</b> Menedżerowie mogą także dodawać użytkowników i przypisywać im role. Twórcy (właścicielowi) motywu jest automatycznie przypisywana rola "Menedżer".</li> </ul>
Edytowanie motywu	<p>Będąc twórcą oprogramowania internetowego, można pobierać i modyfikować poszczególne pliki motywu; można także, korzystając z aplikacji typu Desktop, synchronizować wszystkie zmiany dokonywane w systemie lokalnym.</p> <p>Można także wyeksportować pakiet szablonu zawierający powiązany motyw i wszystkie niestandardowe składniki, po czym pracować nad nimi w trybie offline w preferowanym środowisku programistycznym. Zob. <a href="#">Tworzenie motywów</a>.</p> <p>Jeśli dokonamy zmian w motywie, to — aby te zmiany zostały uwzględnione w serwisach korzystających z tego motywu — musimy go opublikować.</p>
Publikowanie motywu	<p>Jeśli serwis używa nowego, nieopublikowanego motywu, to zostanie on automatycznie opublikowany przy pierwszym przełączeniu serwisu do trybu online. Jeśli dokonamy zmian w motywie i zechcemy zaktualizować serwisy online, tak aby te zmiany zostały uwzględnione, musimy motyw opublikować jawnie. Motyw można opublikować, będąc jego właścicielem lub mając przypisaną rolę "Menedżer".</p> <p>Wybrać motyw na liście, po czym nacisnąć na pasku menu przycisk <b>Publikuj</b> (lub <b>Publikuj ponownie</b> dla motywu wcześniej opublikowanego). Obok motywu na liście jest dodawana ikona "Opublikowane" (). Zob. <a href="#">Publikowanie motywów</a>.</p>

Zadanie	Opis
Usuwanie lub przywracanie motywu	Mając odpowiednie uprawnienia, można usunąć folder motywu i jego zawartość. Gdy motyw zostanie usunięty, jego folder oraz wszystkie powiązane foldery i pliki są przenoszone do Kosza. Motyw może zostać usunięty lub przywrócony przez jego twórcę (właściciela motywu) lub osobę, której ten motyw został udostępniony i której została przypisana rola "Współtwórca" lub "Menedżer".

 **Uwaga:**

Nie można usunąć motywu, jeśli jest on używany przez jakiś serwis.

Aby usunąć motyw, kliknąć prawym przyciskiem myszy na odpowiednim motywie, po czym wybrać opcję **Usuń**. Zostanie wyświetlone pytanie, czy motyw ma zostać przeniesiony do Kosza. Usunięty motyw pozostaje w Koszu, aż:

- Motyw zostanie przywrócony.
- Motyw zostanie trwale usunięty.
- Zostanie osiągnięty przydział miejsca na Kosz.
- Kosz zostanie automatycznie opróżniony zgodnie z interwałem ustawionym przez administratora serwisu. Wartością domyślną jest 90 dni.

Aby przywrócić motyw, na pasku menu strony motywu kliknąć na ikonie **Kosz**, następnie kliknąć prawym przyciskiem myszy na motywie na liście, po czym wybrać opcję **Przywróć**.

## Publikowanie motywów

Motyw wyznacza ogólny wygląd i styl serwisu. Motyw można zaktualizować, aby zmienić wygląd korzystających z niego serwisów.


Jeśli serwis używa nowego, nieopublikowanego motywu, to zostanie on automatycznie opublikowany przy pierwszym przełączeniu serwisu do trybu online. Jeśli dokonamy zmian w motywie i zechcemy zaktualizować serwisy online, tak aby te zmiany zostały uwzględnione, musimy motyw opublikować jawnie. Zmiany, dokonane w motywie, może opublikować tylko jego właściciel lub osoba mająca przypisaną rolę "Menedżer".

 **Uwaga:**

Gdy motyw zostanie opublikowany, dokonane zmiany zostaną odzwierciedlone we wszystkich korzystających z niego serwisach online. Przed opublikowaniem motywu należy przetestować zmiany w trybie offline, a także trzeba rozumieć ich wpływ na powiązane serwisy.

Aby opublikować motyw, należy:

1. Wybrać kolejno opcje **Programista** i **Wyświetl wszystkie motywy**.
2. Wybrać z listy motywów odpowiedni motyw.

3. Z paska menu lub menu podręcznego wybrać opcję **Publikuj**
4. Wybrać opcję **Potwierdzenie kontynuowania**, po czym nacisnąć przycisk **OK**. Obok motywu na liście jest dodawana ikona "Opublikowane" ().

## Zarządzanie niestandardowymi składnikami i układami

Będąc programistą, można tworzyć niestandardowe składniki i układy oraz nimi zarządzać. Będąc współtwórcą serwisu, można rejestrować zarówno składniki (aplikacje) pochodzące od innych podmiotów, jak i grupy składników.

Do składników niestandardowych zaliczają się grupy składników tworzone w edytorze oraz lokalne i odległe składniki tworzone za pomocą opisanych tu opcji. Do układów niestandardowych zaliczają się *układy sekcji*, służące do rozmieszczania składników w gnieździe na stronie, oraz *układy zawartości*, służące do rozmieszczania pól w elemencie zawartości.

- [Omówienie składników niestandardowych](#)
- [Omówienie układów](#)
- [Używanie niestandardowych składników i układów](#)
- [Rejestrowanie składników odległych](#)
- [Tworzenie składników lokalnych, układów, edytorów pól zawartości lub niestandardowych formularzy](#)
- [Eksportowanie bądź importowanie składników lub układów](#)



Informacje, jak używać składników w swoim serwisie, są dostępne pod hasłem [Rozmieszczanie zawartości](#).

Informacje, jak korzystać z poszczególnych składników, są zawarte pod hasłem [Używanie składników wbudowanych](#).

Szczegółowe informacje dotyczące tworzenia składników można znaleźć pod hasłem [Tworzenie składników](#).

## Omówienie składników niestandardowych

*Składniki* są indywidualnymi komponentami strony internetowej. Co można zobaczyć, przeglądając się stronie internetowej? Zazwyczaj jest to kilka tytułów, parę akapitów tekstu i kilka łączy do innych stron w serwisie. Dodatkowo pojawiają się obrazy, przyciski, separatory, mapy i galerie. Każdy z tych elementów jest składnikiem.

Aby dodać składnik do strony, należy się upewnić, że przełącznik  jest ustawiony w pozycji **Edytuj**, kliknąć na ikonie , po czym wybrać typ składnika, który ma być używany.

Następnie należy przeciągnąć składnik z panelu do odpowiedniego gniazda na stronie. To wszystko. Na stronę można przeciągać składniki "tytuł", "akapit" i inne dostępne.

Można z łatwością rejestrować i włączać składniki odległe (aplikacje), a także tworzyć własne składniki, używając opcji z menedżera składników. Wybrać kolejno opcje **Programista** i **Wyświetl wszystko**.

Następnie należy nacisnąć przycisk **Utwórz**, po czym wybrać powiązaną opcję w celu utworzenia nowego składnika lokalnego lub zarejestrowania składnika odległego. Składniki, tworzone i udostępniane w ten sposób, są w edytorze wyświetlane w panelu składników niestandardowych.


Szczegółowe informacje dotyczące tworzenia składników można znaleźć pod hasłem [Tworzenie składników](#).

### Składniki odległe

Mając składnik (aplikację), który ma zostać użyty i pochodzący od innego dostawcy, wystarczy ten składnik zarejestrować, po czym można używać go w swoim serwisie. Jest to łatwe.

Rejestrując składnik odległy, należy podać jego adres URL oraz drugi adres URL przeznaczony dla dowolnych ustawień, które autor serwisu może określać dla składnika odległego.

Po zarejestrowaniu składnika odległego można udostępnić go innym użytkownikom. Wszystkie zarejestrowane składniki odległe, których właścicielem jest bieżący użytkownik lub które zostały mu udostępnione, są w konstruktorze serwisów wyświetlane w panelu "Składniki niestandardowe".

Właściwości składnika odległego można zmieniać przy użyciu edytora, postępując jak w przypadku każdego innego składnika. Wystarczy kliknąć na ikonie menu  składnika, po czym wybrać opcję **Ustawienia**. Naciskając przycisk **Niestandardowe**, otwiera się adres URL ustawień, podany podczas rejestrowania składnika odległego.

#### Uwaga:

Ponieważ składniki odległe znajdują się na serwerze innym niż serwer, na którym działa Oracle Content Management, są — w celu zapewnienia bezpieczeństwa — ujmowane we wstawkowej ramce (przy użyciu elementu iFrame). Nie wszystkie składniki odległe mogą być ujmowane we wstawkowej ramce. O tym, czy jest to możliwe, trzeba się dowiedzieć od dostawcy.

### Składniki lokalne

Programiści mogą tworzyć składniki mające dostęp do tych samych funkcji i możliwości, co składniki udostępniane przez Oracle Content Management. Można wstawić składnik bezpośrednio na stronie albo ująć go we wstawkowej ramce (przy użyciu elementu iFrame).

Gdy użytkownik naciśnie przycisk **Utwórz** w celu utworzenia składnika lokalnego, do listy składników zostanie dodany w pełni funkcjonalny składnik przykładowy z nazwą określoną przez użytkownika i z unikatowym identyfikatorem. Będąc programistą, można modyfikować przykład w celu utworzenia własnych rozwiązań.

Szczegółowe informacje dotyczące tworzenia składników można znaleźć pod hasłem [Tworzenie składników](#).

## Omówienie układów

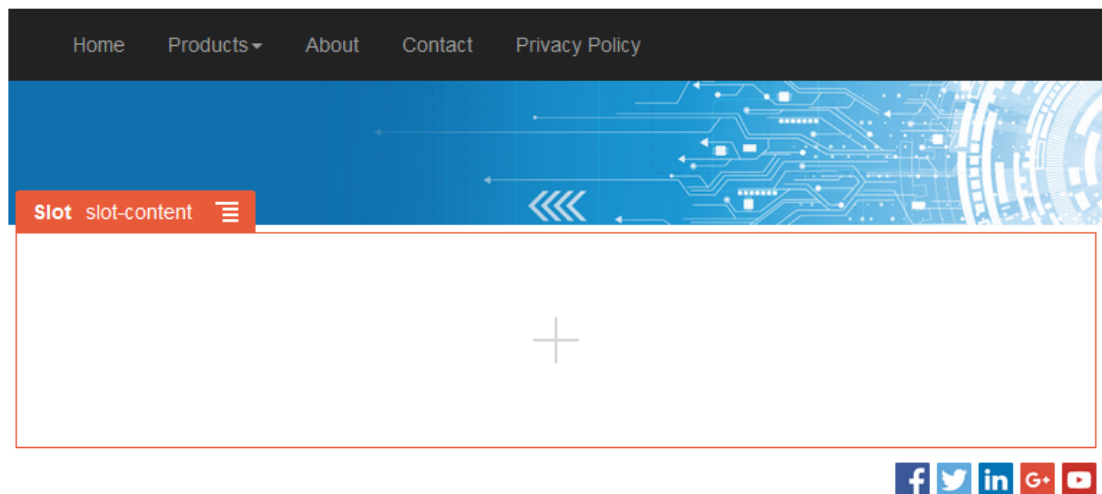
Układ strony służy do rozmieszczania gniazd i zawartości na stronie. Układ sekcji służy do rozmieszczania zawartości w gniazdach. Układ zawartości służy do rozmieszczania pól w elementach zawartości.

Ogólnie rzecz biorąc, za pomocą układu określa się sposób prezentacji zawartości, lecz nie jej treść. Oddzielenie zawartości od jej prezentacji ułatwia przedstawianie tej samej zawartości na różne sposoby lub zmianę prezentacji bez ingerowania w jej zawartość.

### Układ strony

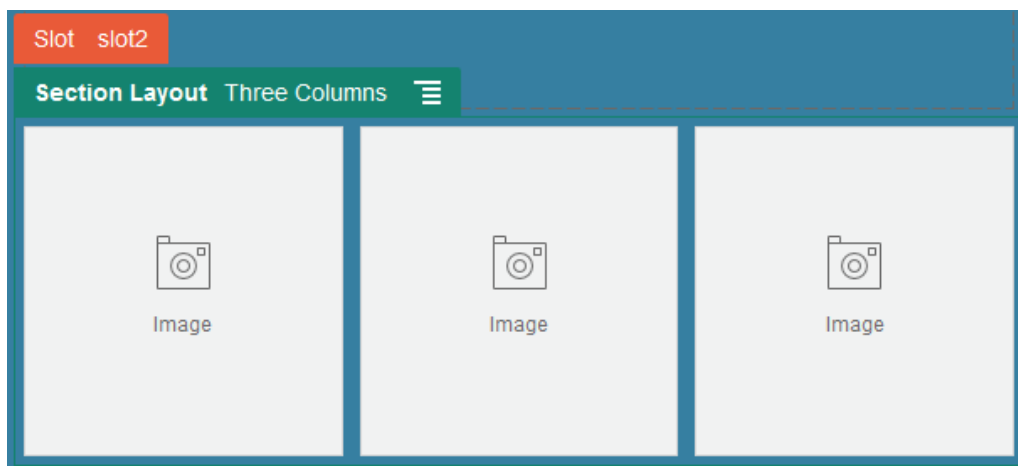
Dodając stronę do serwisu, wybiera się jej układ. Każdy układ wyznacza obszary na stronie — określane terminem "gniazdo" — do których można przeciągać zawartość. Układ strony określa liczbę gniazd na stronie i ich położenie. Układ może zawierać również zawartość, która jest predefiniowana i umieszczana na stronie. Zawartość ta może być statyczna i nieedytowalna (taka jak logo firmy) lub w niewielkim zakresie edytowalna, taka jak tekst nagłówka, który może zostać zmieniony przez współtwórcę, lecz nie można zmienić położenia ani wyglądu nagłówka.

Układy strony są przechowywane w motywie. Motywy mogą zawierać jeden lub większą liczbę układów. Twórca oprogramowania może kopiować i modyfikować już istniejący motyw w celu utworzenia nowego. Zob. [Tworzenie układów](#).



### Układ sekcji

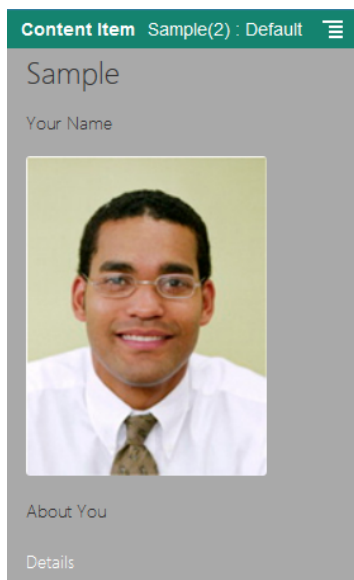
Układ sekcji automatycznie organizuje dodawaną do niej zawartość, ułatwiając współtwórcom dodawanie zawartości bez poświęcania czasu na jej formatowanie. Na przykład układ sekcji może automatycznie uporządkować zawartość w kolumnach lub na pionowej liście. Współtwórca serwisu może, w celu porządkowania zawartości, dodać do gniazda jeden lub większą liczbę układów sekcji.



Programista — aby rozwiązać pewne problemy z układem lub uprościć pracę współtwórców — może utworzyć dodatkowe układy sekcji, jako uzupełnienie już istniejących. Projektant motywu może nawet wbudować układy sekcji w gniazdo w układzie strony. Zob. [Tworzenie układów](#).

#### Układ zawartości

Będąc użytkownikiem firmowym, można tworzyć i korzystać z elementów zawartości na podstawie udostępnionych typów zawartości i układów. Taka struktura zawartości umożliwia użytkownikowi, będącemu współtwórcą, gromadzenie zawartości elementu zawartości poza konstruktorem serwisów. Wiele układów zawartości powiązanych z określonym typem zawartości umożliwia projektantowi serwisów wyświetlanie elementu zawartości w różnych kontekstach, bez ingerowania w zgromadzoną zawartość. Będąc twórcą oprogramowania, można tworzyć nowe układy z udostępnionego domyślnego układu. Zob. [Tworzenie układów](#).





## Używanie niestandardowych składników i układów


Będąc programistą, można tworzyć niestandardowe składniki i układy oraz nimi zarządzać. Będąc współtwórcą serwisu, można rejestrować zarówno składniki (aplikacje) pochodzące od innych podmiotów, jak i grupy składników.


Do składników niestandardowych zaliczają się grupy składników tworzone w edytorze oraz lokalne i odległe składniki tworzone za pomocą opisanych poniżej opcji. Mając aplikację pochodzącą od innego dostawcy (składnik odległy), wystarczy ją zarejestrować, po czym można jej używać w swoim serwisie. Twórcy oprogramowania mogą tworzyć składniki lokalne mające dostęp do tych samych funkcji i możliwości, co składniki udostępniane przez Oracle Content Management.

Do układów niestandardowych zaliczają się *układy sekcji*, służące do rozmieszczania składników w gnieździe na stronie, oraz *układy zawartości*, służące do rozmieszczania pól w elemencie zawartości.

Do wykonania przedstawionych poniżej zadań służą opcje z paska menu lub z menu podręcznego ze strony "Składniki".

Zadanie	Opis
Rejestrowanie składnika odległego	<p>Rejestrując składnik odległy, należy podać jego adres URL oraz drugi adres URL przeznaczony dla dowolnych ustawień, które autor serwisu może określać dla składnika odległego.</p> <p>Aby można było utworzyć składnik, administrator musi włączyć odpowiednie opcje w menu <b>Utwórz</b>. Jeśli na stronie "Składniki" nie jest widoczne menu <b>Utwórz</b>, należy się skontaktować z administratorem.</p> <p>Aby zarejestrować składnik odległy, należy nacisnąć przycisk <b>Utwórz</b>, po czym wybrać powiązaną opcję rejestrowania składnika odległego. Zob. <a href="#">Rejestrowanie składników odległych</a>.</p>
Tworzenie składnika lub układu lokalnego	<p>Gdy jest tworzony składnik lub układ lokalny, jest udostępniany w pełni funkcjonalny składnik lub układ przykładowy, który jest używany jako podstawa nowego składnika lub układu.</p> <p>Aby można było utworzyć składnik lub układ, administrator musi włączyć odpowiednie opcje w menu <b>Utwórz</b>. Jeśli na stronie "Składniki" nie jest widoczne menu <b>Utwórz</b>, należy się skontaktować z administratorem.</p> <p>Aby zarejestrować składnik odległy, należy nacisnąć przycisk <b>Utwórz</b>, po czym wybrać powiązaną opcję tworzenia składnika lub układu lokalnego. Zob. <a href="#">Tworzenie składników lokalnych, układów, edytorów pól zawartości lub niestandardowych formularzy</a>.</p> <p>Informacje o innych sposobach tworzenia składników są dostępne pod hasłem <a href="#">Tworzenie składników</a>.</p>

Zadanie	Opis
Kopiowanie składnika lub układu	<p>Można utworzyć nowy składnik lub układ, kopiując istniejący składnik lub układ i dokonując zmian w kopii.</p>
	<div style="border-left: 2px solid #0070C0; border-right: 2px solid #0070C0; border-bottom: 2px solid #0070C0; padding: 10px; background-color: #E6F2FF;"> <p> <b>Uwaga:</b></p> <p>Po utworzeniu bądź zarejestrowaniu składnika lub układu nie można zmienić jego nazwy. Można skopiować składnik lub układ i nadać kopii nową nazwę. Wszystkie inne informacje rejestracyjne, w tym wartości kluczy dla składników odległych, są zachowywane. Informacje dotyczące udostępniania są niezależne od informacji rejestracyjnych i nie są kopiowane.</p> </div>
	<p>Kliknąć prawym przyciskiem myszy na składniku lub układzie, który ma zostać skopiowany, po czym wybrać opcję <b>Kopiuj</b>. Wpisać nazwę, a następnie nacisnąć przycisk <b>Kopiuj</b>. Może się ona składać tylko z liter, cyfr, znaków podkreślenia ( ) i łączników (-). Użyta spacja zostanie automatycznie zastąpiona łącznikiem.</p>
Tworzenie edytora pól zawartości	<p>Można utworzyć edytor pól zawartości, po czym go wypromować i używać do kontrolowania wyglądu pól danych podczas tworzenia typu zawartości. Wszystkie typy danych są obsługiwane z wyjątkiem typów <i>Multimedia</i> i <i>Odniesienie</i>. Składnik — zanim stanie się dostępny do użycia podczas tworzenia typu zawartości — musi zostać wypromowany.</p>
Udostępnianie składnika lub układu	<p>Składnik lub układ można udostępnić innym użytkownikom Oracle Content Management. Można to zrobić, będąc jego właścicielem lub mając przypisaną rolę "Menedżer". Udostępniając składnik lub układ, trzeba przypisać użytkownikowi rolę określającą, co można z tym składnikiem lub układem robić.</p> <p>Kliknąć prawym przyciskiem myszy na składniku lub układzie, który ma zostać udostępniony, po czym wybrać opcję <b>Udostępnij</b>, a następnie nacisnąć przycisk <b>Dodaj współużytkowników</b>.</p> <p>Podać adres e-mail lub adresy e-mail oraz przypisać jedną z następujących ról:</p> <ul style="list-style-type: none"> <li>• <b>Oglądający:</b> Oglądający mogą zobaczyć składnik lub układ (na liście składników i układów niestandardowych w edytorze) i użyć go. Mogą także wyświetlać foldery i pliki wchodzące w skład motywu, lecz nie mogą niczego zmieniać.</li> <li>• <b>Pobierający:</b> Pobierający może dodatkowo pobierać pliki i zapisywać je w swoim komputerze.</li> <li>• <b>Współtwórca:</b> Współtwórcy mogą dodatkowo edytować składnik lub układ, wysłać nowe pliki, usuwać pliki, a także usunąć cały składnik lub układ, o ile nie jest on używany w serwisie.</li> <li>• <b>Menedżer:</b> Menedżerowie mogą także dodawać użytkowników i przypisywać im role. Twórcy (właścicielowi) składnika lub układu jest automatycznie przypisywana rola "Menedżer".</li> </ul>

Zadanie	Opis
Edytowanie składnika lub układu	<p>Będąc twórcą oprogramowania internetowego, można pobierać i modyfikować poszczególne pliki składników lub układów; można także, korzystając z aplikacji typu Desktop, synchronizować wszystkie zmiany dokonywane w systemie lokalnym.</p> <p>Można także wyeksportować składnik lub układ indywidualnie albo w ramach pakietu szablonu, zawierającego dowolne składniki niestandardowe i układy sekcji, po czym pracować nad składnikiem w trybie offline w preferowanym środowisku programistycznym.</p> <p>Jeśli dokonamy zmian w składniku lub układzie, to — aby te zmiany zostały uwzględnione w serwisach korzystających z tego składnika lub układu — musimy go opublikować.</p>
Eksportowanie bądź importowanie składnika lub układu	<p>Składnik lub układ można wyeksportować i zmodyfikować offline, a następnie zaimportować go jako nowy składnik lub układ albo jako zamiennik istniejącego składnika lub układu. Można także wyeksportować składnik lub układ, aby przenieść go (zaimportować) do innej instancji. Składnik lub układ można wyeksportować indywidualnie bądź w ramach pakietu szablonu, zawierającego dowolne składniki niestandardowe i układy.</p> <p>Eksportując składnik lub układ, kopiuje się go do folderu jako plik .zip. Pakiet składnika lub układu można pobrać bezpośrednio z tego folderu, a następnie rozpakować pobrany pakiet i pracować nad poszczególnymi plikami. Po ukończeniu pracy nad plikami składnika lub układu można utworzyć plik .zip zawierający pakiet składników lub układów i zaimportować go, zastępując składnik lub układ pierwotny bądź tworząc nowy.</p> <p>Zob. <a href="#">Eksportowanie bądź importowanie składników lub układów</a>.</p>
Publikowanie składnika lub układu	<p>Jeśli serwis używa nowego, nieopublikowanego składnika lub układu, to zostanie on automatycznie opublikowany przy pierwszym przełączeniu serwisu do trybu online. Jeśli dokonamy zmian w składniku lub układzie i zechcemy zaktualizować serwisy online, tak aby te zmiany zostały uwzględnione, musimy składnik lub układ opublikować jawnie. Składnik lub układ można opublikować, będąc jego właścicielem lub mając przypisaną rolę "Menedżer".</p> <p>Wybrać składnik lub układ na liście, po czym nacisnąć na pasku menu przycisk <b>Publikuj</b> (bądź <b>Publikuj ponownie</b> dla składnika lub układu wcześniej opublikowanego). Obok składnika lub układu na liście jest wyświetlana ikona "Opublikowane" ()</p>

Zadanie	Opis
Usuwanie bądź przywracanie składnika lub układu	Mając odpowiednie uprawnienia, można usunąć folder składnika lub układu i jego zawartość. Gdy składnik lub układ zostanie usunięty, jego folder oraz wszystkie powiązane foldery i pliki są przenoszone do Kosza. Składnik lub układ może zostać usunięty bądź przywrócony przez jego twórcę (właściciela składnika lub układu) lub osobę, której ten składnik lub układ został udostępniony i której została przypisana rola "Współtwórca" lub "Menedżer".

 **Uwaga:**

Nie można usunąć składnika lub układu, który jest używany przez dowolny serwis bądź dowolną aktualizację, w tym serwisy i aktualizacje znajdujące się w Koszu.

Aby usunąć składnik lub układ, kliknąć prawym przyciskiem myszy na odpowiednim składniku lub układzie, po czym wybrać opcję **Usuń**. Zostanie wyświetlone pytanie, czy składnik lub układ ma zostać przeniesiony do Kosza. Usunięty składnik lub układ pozostaje w Koszu, aż:

- Składnik lub układ zostanie przywrócony.
- Składnik lub układ zostanie trwale usunięty.
- Zostanie osiągnięty przydział miejsca na Kosz.
- Kosz zostanie automatycznie opróżniony zgodnie z interwałem ustawionym przez administratora serwisu. Wartością domyślną jest 90 dni.

Aby przywrócić składnik lub układ, kliknąć na **Koszu**, następnie kliknąć prawym przyciskiem myszy na składniku lub układzie na liście, po czym wybrać opcję **Przywróć**.

## Rejestrowanie składników odległych

Aby można było użyć w serwisie składnika odległego, trzeba najpierw zarejestrować go w Oracle Content Management.

Można rejestrować składniki odległe pochodzące od innych dostawców, a także te, które sami utworzyliśmy.

 **Uwaga:**

Aby można było zarejestrować składnik odległy, administrator musi włączyć odpowiednie opcje w menu **Utwórz**. Jeśli na stronie "Składniki" nie jest widoczne menu **Utwórz**, należy się skontaktować z administratorem.

Aby zarejestrować składnik odległy, przeznaczony do użycia w Oracle Content Management, należy:

1. Wybrać kolejno opcje **Programista** i **Wyświetl wszystkie składniki**.
2. Nacisnąć przycisk **Utwórz**, po czym wybrać opcję **Zarejestruj składnik odległy**.

3. W oknie rejestracji składnika odległego wpisać lub wybrać następujące informacje:
  - **Nazwa:** Nazwa składnika widziana przez użytkowników.
  - **Opis:** Opis składnika widziany przez użytkowników.
  - **URL składnika:** Punkt końcowy, używany w ramce iFrame do renderowania zawartości składnika na stronie. Musi być oparty na protokole HTTPS.
  - **URL ustawień:** Punkt końcowy, używany w ramce iFrame do renderowania ustawień składnika odległego, dodawanego do strony. Musi być oparty na protokole HTTPS.
  - **Szerokość ustawień:** Określa szerokość domyślną (w pikselach) panelu ustawień składnika.
  - **Wysokość ustawień:** Określa wysokość domyślną (w pikselach) panelu ustawień składnika.
  - **Klucz:** 192-bitowy klucz AES, powiązany ze składnikiem odległym, używany — gdy składnik jest udostępniany — do utworzenia podpisanego tokenu haszowanego skrótu. Służy do szyfrowania i zapewnienia bezpiecznego odczytu i zapisu ustawień składnika.
4. Nacisnąć przycisk **Zarejestruj**.

Gdy składnik odległy zostanie utworzony, jego nazwa pojawi się na liście składników. Pliki, użyte do zarejestrowania składnika, można eksplorować, klikając na nazwie składnika na liście składników.

Informacje dotyczące rejestracji składnika są przechowywane w katalogu używanym przez serwisy utworzone w tej samej instancji Oracle Content Management, lecz składnik pozostaje usługą odległą.

Ikona składnika jest dodawana w edytorze do panelu "Składniki niestandardowe", z nazwą przypisaną do składnika. Składnik można udostępnić innym użytkownikom; będą go w edytorze widzieć w panelu "Składniki niestandardowe".

## Tworzenie składników lokalnych, układów, edytorów pól zawartości lub niestandardowych formularzy

Gdy twórca oprogramowania przystępuje do tworzenia składnika lub układu lokalnego, jest mu udostępniany w pełni funkcjonalny przykład, który może modyfikować pod kątem nowego składnika lub układu.

Aby można było utworzyć składnik lub układ, administrator musi włączyć odpowiednie opcje w menu **Utwórz**. Jeśli na stronie "Składniki" nie jest widoczne menu **Utwórz**, należy się skontaktować z administratorem.

Programiści mogą tworzyć składniki mające dostęp do tych samych funkcji i możliwości, co składniki udostępniane przez Oracle Content Management.

Aby utworzyć przykładowy składnik lub układ lokalny, należy:

1. Wybrać kolejno opcje **Programista** i **Wyświetl wszystkie składniki**.
2. Nacisnąć przycisk **Utwórz**, po czym wybrać typ składnika lub układu, który ma zostać utworzony.
3. Wpisać nazwę składnika lub układu. Nie może to być nazwa już używana dla innego składnika lub układu.

Może się ona składać tylko z liter, cyfr, znaków podkreślenia ( \_ ) i łączników (-). Użyta spacja zostanie automatycznie zastąpiona znakiem podkreślenia.

Dla szablonów, motywów, składników, serwisów ani stron serwisów nie wolno używać następujących nazw: authsite, content, pages, scstemplate\_\*, \_comps, \_components, \_compsdelivery, \_idcservice, \_sitescloud, \_sitesclouddelivery, \_themes, \_themesdelivery. Następujących nazw można używać dla stron serwisów, ale nie można używać dla szablonów, motywów, składników ani serwisów: documents, sites.

4. Opcjonalnie: podać opis składnika lub układu.
5. Dla typu składnika wybrać opcję **Domyślne** (aby wstawić składnik bezpośrednio na stronie), **W środowisku Sandbox** (aby ująć składnik w ramce wstawkowej przy użyciu elementu iFrame) lub **Szablon** (aby renderowanie odbywało się tylko przy użyciu szablonu).

 **Uwaga:**

Jeśli wybrano opcję **Szablon**, składnik lokalny nie będzie oparty na technologii Knockout i dlatego, gdy zostanie umieszczony na stronie, nie będzie mógł używać zagnieżdżonych składników. Na przykład może to oznaczać, że edycja tytułu lub treści musi się odbywać poprzez ustawienia niestandardowe w panelu ustawień składnika, a nie poprzez wybór tekstu składnika wyświetlanego na stronie.

6. Nacisnąć przycisk **Utwórz**.  
Na pasku postępu będzie pokazywany status operacji tworzenia. Gdy składnik, układ lub formularz zostanie utworzony, jego nazwa pojawi się na liście składników. Foldery i pliki, zawarte w składniku lub układzie, można eksplorować po kliknięciu na nazwie składnika (lub układu) wyświetlanej na liście składników.
7. Aby wybrać ikonę inną niż ta, która jest domyślnie przypisywana do składnika lub układu, należy:
  - a. Wybrać z listy składnik lub układ.
  - b. Wybrać opcję **Właściwości**.
  - c. Kliknąć na karcie **Logo składnika**.
  - d. Kliknąć na obrazie loga, wyświetlanym w galerii, po czym nacisnąć przycisk **Gotowe**.

Szczegółowe informacje dotyczące tworzenia składników można znaleźć pod hasłem [Tworzenie składników](#).

Po dostosowaniu składnika, układu, edytora pól zawartości lub formularza zawartości, można go udostępnić innym lub wypromować, tak aby mogli użyć go w następujący sposób:

- Składnik niestandardowy: gdy jest używany konstruktor serwisów, ikona składnika jest dodawana w edytorze do panelu "Składniki niestandardowe", z nazwą przypisaną do składnika.
- Układ sekcji: gdy jest używany konstruktor serwisów, ikona układu sekcji jest dodawana w edytorze do panelu "Układy sekcji", z nazwą przypisaną do układu.
- Układ zawartości: administrator zawartości może przypisać układ do jednego lub kilku typów zawartości jako widok domyślny albo dodawany do listy układów,

dzięki czemu projektant może wybrać ten układ w konstruktorze serwisów i określić, w jaki sposób element zawartości tego typu będzie wyświetlany na stronie.

- **Edytor pól zawartości:** Aby uczynić edytor pól zawartości dostępnym do użycia podczas tworzenia typów zawartości, należy go wybrać, nacisnąć przycisk **Promuj**, po czym potwierdzić decyzję i nacisnąć przycisk **OK**. Wypromowany edytor pól zawartości staje się dostępny i za jego pomocą można podczas tworzenia typów zawartości kontrolować wygląd wszystkich pól danych z wyjątkiem typów *Multimedia* i *Odniesienie*.
- **Formularz zawartości:** Aby uczynić formularz zawartości dostępnym do użycia podczas tworzenia typów zawartości, należy go wybrać, nacisnąć przycisk **Promuj**, po czym potwierdzić decyzję i nacisnąć przycisk **OK**. Po wypromowaniu formularze zawartości są dostępne do używania z wszystkimi typami zawartości i typami zasobów.

## Eksportowanie bądź importowanie składników lub układów

Składnik można wyeksportować i zmodyfikować offline, a następnie zaimportować go jako nowy składnik albo jako zamiennik istniejącego składnika. Można także wyeksportować składnik, aby przenieść go (zaimportować) do innej instancji Oracle Content Management.

### Eksportowanie

Eksportując składnik, kopiuje się go do folderu usługi Oracle Content Management jako plik .zip. Składnik można pobrać bezpośrednio z tego folderu, po czym rozpakować pakiet i pracować z poszczególnymi plikami. Po ukończeniu pracy z plikami składnika można utworzyć plik .zip zawierający foldery i pliki składnika, po czym zaimportować ten plik, zastępując składnik pierwotny lub tworząc nowy.

#### Uwaga:

Gdy składnik jest eksportowany, nie są dołączane informacje dotyczące jego udostępniania.

Aby wyeksportować składnik, należy:

1. Wybrać kolejno opcje **Programista** i **Wyświetl wszystkie składniki**.
2. Wybrać składnik lub układ, po czym wybrać z paska menu lub z menu podręcznego (kliknięcie prawym przyciskiem myszy) opcję **Eksportuj**.
3. Przejść do folderu albo utworzyć nowy folder: nacisnąć przycisk **Nowy**, podać nazwę i opcjonalny opis, po czym nacisnąć przycisk **Utwórz**. Aby utworzyć folder, kliknąć na jego ikonie lub nazwie.
4. Wybrać folder, zaznaczając pole wyboru obok powiązanego folderu, po czym nacisnąć przycisk **OK**.

W wybranym folderze zostanie utworzony plik pakietu składnika lub układu, mający nazwę składnika i rozszerzenie .zip.

### Importowanie

Aby można było zaimportować lub utworzyć składnik, układ lub edytor pól zawartości, administrator musi włączyć odpowiednie opcje w menu **Utwórz**. Jeśli na stronie "Składniki" nie jest widoczne menu **Utwórz**, należy się skontaktować z administratorem.

Aby zaimportować składnik, układ lub edytor pól zawartości, należy:

1. Wybrać kolejno opcje **Programista** i **Wyświetl wszystkie składniki**.
2. Nacisnąć przycisk **Utwórz**, po czym wybrać opcję **Importuj składnik**. Wybrać tę opcję, aby zaimportować składnik, układ sekcji lub układ zawartości.
3. Mając wysłany składnik lub układ, przejść do folderu (zawierającego ten składnik lub układ) i otworzyć go. Jeśli pakiet nie został jeszcze wysłany, przejść do folderu, do którego ma zostać wysłany składnik lub układ albo utworzyć nowy folder. Nacisnąć przycisk **Wyślij**, następnie odszukać pakiet ze składnikiem lub układem, po czym nacisnąć przycisk **Otwórz**.
4. Zaznaczyć pole wyboru obok nazwy pliku składnika lub układu, po czym nacisnąć przycisk **OK**. Nowy składnik lub układ został utworzony i dodany do listy składników. Jeśli okaże się, że już istnieje składnik lub układ o tej samej nazwie lub tym samym ID, pojawi się wezwanie do rozwiązania konfliktu. Może pojawić się potrzeba utworzenia nowego składnika lub układu albo zastąpienia istniejącego składnika lub układu importowaną wersją.



# 8





## Praca ze stronami serwisu

Utworzymy stronę, wybierzemy układ i zdecydujemy o miejscu strony w serwisie.

- [Nawigacja do strony](#)
- [Wyświetlanie stron](#)
- [Dodawanie stron](#)
- [Przenoszenie stron](#)
- [Usuwanie stron](#)
- [Zmiana ustawień strony](#)
- [Zmienianie układu strony](#)
- [Zmienianie tła lub motywu](#)

### Nawigacja do strony

Aby wyświetlić hierarchiczną strukturę stron i nawigować po niej, należy:



1. Otworzyć serwis do edycji. Upewnić się, że przełącznik  jest ustawiony w pozycji "Edytuj".
2. Aby wyświetlić opcje zarządzania stronami i ich zawartością oraz opcje edycji, kliknąć na ikonie .
3. Kliknąć na ikonie , aby wyświetlić pierwszy poziom stron.
4. Kliknąć na stronie, aby ją wyświetlić.  
Strony, przy których jest wyświetlana strzałka (>), mają zagnieżdżoną warstwę stron. Klikając na takiej stronie, wyświetla się strony zagnieżdżone. Właściwą stronę można także odszukać, używając pola wyszukiwania (na górze listy stron).
5. Aby wyświetlić lub zmienić ustawienia określonej strony, kliknąć na jej ikonie . Zob. [Zmiana ustawień strony](#).

### Wyświetlanie stron

Aby wyświetlić strony serwisu, należy otworzyć go w edytorze, a następnie używać różnych opcji podglądu, rozmiaru i orientacji.

Aby zmienić sposób wyświetlania stron podczas pracy z edytorem, należy:

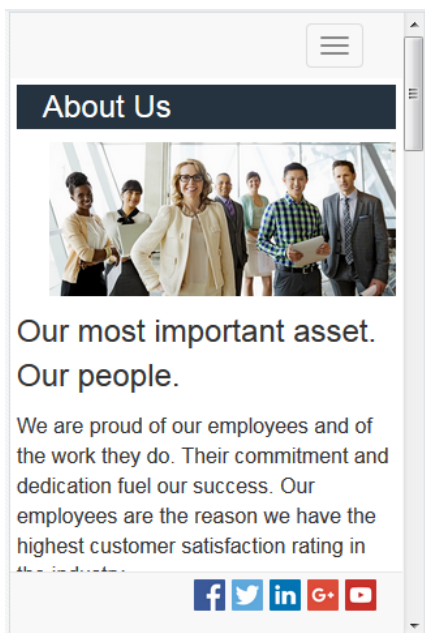
1. Otworzyć serwis do edycji.
2. Przejść do strony.




3. Aby uzyskać podgląd strony w edytorze, upewnić się, że  jest ustawiony w pozycji "Podgląd". Strona jest wówczas pokazywana w takiej postaci, w jakiej będą widzieć ją odwiedzający, tj. bez pomocy wizualnych używanych w trybie edycji.
4. Aby uzyskać w oddzielnym oknie podgląd bieżącej aktualizacji zastosowanej do serwisu podstawowego, kliknąć na ikonie .

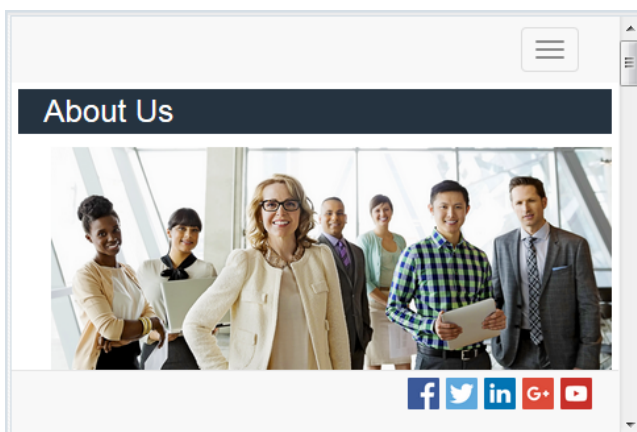
 **Uwaga:**

Łącza do stron serwisu nie działają w trybie podglądu.

5. Aby zobaczyć stronę, tak jak będzie ona wyglądała w urządzeniu z określonym rozmiarem ekranu, kliknąć na opcji **Dopasuj do ekranu** na górnym pasku menu, po czym wybrać z listy odpowiedni zestaw wymiarów ekranu. Motywy z reagującym projektem automatycznie porządkują zawartość strony pod kątem najlepszego jej przedstawienia na wybranym rozmiarze ekranu.




6. Aby utworzyć niestandardowy rozmiar, wybrać kolejno opcję **Dopasuj do okna i Utwórz ustawienia wstępne urządzenia...** Wypełnić wymagane pola. Aby usunąć niestandardowy rozmiar, kliknąć na wyświetlanej obok niego ikonie . Można także włączyć linijkę , po czym — aby szybko zobaczyć różne rozmiary — wybrać dowolny przedział na linijce.
7. Aby zmienić orientację z poziomej na pionową lub odwrotnie, kliknąć na ikonie .



## Dodawanie stron



Strony w serwisie są uporządkowane w hierarchiczną strukturę, przypominającą strukturę folderów, zwaną *drzewem serwisu*. Można dodać stronę, zdefiniować jej ustawienia oraz określić jej docelowe miejsce w serwisie.

Stronę można dodać bezpośrednio, a można też dodać łącze prowadzące do strony zewnętrznej.

- **Strona internetowa:** Strona i jej zawartość rezydują w serwisie. Można nadać nazwę stronie i zdefiniować jej zawartość, określić miejsce strony w serwisie oraz ustalić, jak ma się ona zachowywać w kontekście bieżącego serwisu.
- **Łącze zewnętrzne:** Strona, pochodząca z innej lokalizacji, jest określana przez adres URL. Można nadać nazwę stronie, określić jej miejsce w serwisie oraz ustalić, jak ma się ona zachowywać w kontekście bieżącego serwisu. Ponieważ jest używana strona z aktywnego serwisu, nie można zmienić jej zawartości. W drzewie serwisu, przed nazwą strony, która jest powiązana za pomocą łącza z zewnętrznym adresem URL, jest wyświetlana ikona .

### Dodawanie strony internetowej

Aby dodać do serwisu stronę internetową, należy:

1. Otworzyć serwis do edycji.
2. Wybrać poziom (lub gałąź), na którym ma zostać dodana strona, po czym nacisnąć przycisk **Dodaj stronę** albo — aby dodać stronę podrzędną — kliknąć kolejno na ikonach  i . Na dole drzewa serwisu jest dodawana "Nowa strona"; pojawia się wezwanie do nadania jej nazwy i określenia innych ustawień.
3. Jako typ strony wybrać **Strona internetowa**. Opcji tej należy użyć, jeśli chcemy zarządzać zawartością strony, a nie tylko używać strony pochodzącej z innego serwisu.
4. Nadać stronie nazwę. Może się ona składać tylko z liter, cyfr, znaków podkreślenia ( \_ ) i łączników (-).

Dla szablonów, motywów, składników, serwisów ani stron serwisów nie wolno używać następujących nazw: authsite, content, pages, scsTEMPLATE\_\*, \_comps, \_components, \_compsdelivery, \_idcSERVICE, \_sitesCLOUD, \_sitesCLOUDdelivery, \_themes, \_themesdelivery. Następujących nazw można używać dla stron serwisów, ale nie można używać dla szablonów, motywów, składników ani serwisów: documents, sites.

5. Określić URL strony. Domyślnie nazwa ta będzie używana w adresie URL strony. Spacje są automatycznie zastępowane łącznikami. Aby użyć adresu URL innego niż domyślny, należy nacisnąć przycisk **Przesłoń**, po czym dodać nazwę pliku używaną w adresie URL. Może się ona składać tylko z liter, cyfr, znaków podkreślenia ( \_ ) i łączników ( - ). Należy pamiętać, aby dołączyć rozszerzenie nazwy pliku. Rozszerzeniem domyślnym jest .html.
6. Wybrać układ strony. Definiuje on ogólną strukturę strony, lecz nie jej zawartość. Układ zawiera nazwane *gniazda* (jedno lub większą ich liczbę), w których można umieszczać zawartość. Liczba i typ układów strony zależy od powiązanego z serwisem motywu.
7. Wybrać układ strony dla urządzenia mobilnego. Układ ten definiuje ogólną strukturę strony, gdy jest ona wyświetlana na urządzeniu mobilnym.
8. Podać tytuł strony. Nie pojawia się on na samej stronie, lecz na pasku tytułu w przeglądarce, gdy strona zostanie wyświetlona. Aby dodać tytuł pojawiający się na stronie, należy za pomocą edytora dodać składnik "tytuł".
9. Dodać opcjonalny opis. Opis ten nie pojawia się na samej stronie, lecz dostarcza informacji o stronie, z których — oprócz nas — mogą skorzystać inni współtwórcy.
10. Opcjonalnie: określić rozdzielone przecinkiem słowa kluczowe pomagające wyszukiwarkom rozpoznawać zawartość strony. Słowa kluczowe są przydatne do identyfikowania terminów i pojęć, które nie występują w tekście strony lub które pojawiają się na obrazach. Nie pojawiają się na stronie, lecz są używane przez wyszukiwarki do odnajdywania i rozpoznawania serwisu. Dobry opis, słowa kluczowe i synonimy mogą zwiększyć liczbę odwiedzin serwisu.
11. Opcjonalnie: dodać do nagłówka strony skrypty lub tagi na potrzeby analiz lub śledzenia. Każdy kod, użyty w nagłówku lub stopce, trzeba dokładnie sprawdzić, aby się upewnić, że działa poprawnie i nie stwarza zagrożenia dla serwisu.
12. Opcjonalnie: dodać do stopki strony skrypty lub tagi na potrzeby analiz lub śledzenia. Domyślnie stopka zawiera tekst pokazywany w wyskakującym oknie, dotyczący użycia plików cookie w serwisie, oraz łącze do oświadczenia o ochronie prywatności. Będąc twórcą oprogramowania, można ten tekst zmienić. Każdy kod, użyty w nagłówku lub stopce, trzeba dokładnie sprawdzić, aby się upewnić, że działa poprawnie i nie stwarza zagrożenia dla serwisu.
13. Opcjonalnie: użyć dowolnych z dostępnych opcji dotyczących strony:
  - **Strona błędu:** Jeśli jakiś błąd uniemożliwi wyświetlenie żądanej strony, zamiast niej zostanie wyświetlona domyślna strona błędu. Domyślnie, gdy ta opcja zostanie wybrana, zostają również wybrane opcje **Ukryj stronę w nawigacji** i wszystkie opcje wyszukiwarki. Jeśli strona błędów ma nie być uwzględniana w nawigacji lub w wynikach wyszukiwarki, można wyczyścić pole wyboru tej opcji.
  - **Ukryj stronę w nawigacji:** Strona nie będzie uwzględniana w nawigacji automatycznie generowanej dla serwisu.
  - **Strona szczegółów:** Strona ta służy do wyświetlania szczegółowych informacji o elemencie zawartości wybranym na innej stronie. Zawartość strukturalna jest dostępna tylko dla użytkowników firmowych. Domyślnie, po wybraniu tej opcji, zostaje również wybrana opcja **Ukryj stronę w nawigacji**. Jeśli strona szczegółów ma nie być uwzględniana w nawigacji, można wyczyścić pole wyboru tej opcji. Jeśli elementy zawartości strukturalnej zostały skonfigurowane do używania strony szczegółów, użytkownik może kliknąć na łączu w celu uzyskania szczegółowych informacji o elemencie zawartości.

- **Strona wyszukiwania:** Strona ta służy do wyświetlania wyników wyszukiwania. Można wybrać tę stronę w ustawieniach "Łącze" składnika "wyszukiwanie zawartości". Zob. [Wyszukiwanie zawartości](#). Domyślnie, po wybraniu tej opcji, zostaje również wybrana opcja **Ukryj stronę w nawigacji**. Jeśli strona wyszukiwania ma nie być uwzględniana w nawigacji, można wyczyścić pole wyboru tej opcji.
14. Opcjonalnie: wybrać opcję dotyczącą optymalizacji pod kątem wyszukiwarek (SEO):
    - **Ukryj stronę przed wyszukiwarkami:** Wyszukiwarki nie będą indeksowały zawartości strony i tym samym nie będą wyświetlały jej w wynikach wyszukiwania.
    - **Ukryj łącza do stron przed wyszukiwarkami:** Powiadomianie wyszukiwarek, że mają nie podążać za linkami ze strony i tym samym nie indeksować miejsc docelowych.
    - **Wyłącz buforowanie stron dla wyszukiwarki:** Powiadomianie wyszukiwarek, że mają nie buforować danej strony.
    - **Ukryj opis strony przed wyszukiwarkami:** Powiadomianie wyszukiwarek, że mają nie dołączać opisu (podany powyżej) po stronie pokazywanej w wynikach wyszukiwania.
  15. Opcjonalnie: wybrać, czy ta strona ma zawierać przycisk Cobrowse umożliwiający korzystanie z sesji wspólnego przeglądania. Sesja wspólnego przeglądania używa usługi Oracle Cobrowse Cloud Service do zarządzania środowiskiem udostępniania ekranów osobom odwiedzającym serwis. Zob. [Włączanie integracji z usługą Cobrowse](#) i [Korzystanie z usługi Cobrowse na stronie](#).
  16. Opcjonalnie: przesłonić snippet śledzenia analiz serwisu, po czym dodać snippet właściwy dla danej strony. Dodanie snippetu — zawierającego kod JavaScript śledzenia — w celu śledzenia analiz internetowych ułatwia integrację z zewnętrznymi dostawcami analiz, takimi jak Google, Adobe lub Oracle Infinity. Zob. [Dodawanie śledzenia analiz](#).
  17. Aby zapisać wszystkie oczekujące zmiany w bieżącej aktualizacji, nacisnąć przycisk **Zapisz**.

#### Dodawanie strony powiązanej

Aby dodać do serwisu stronę powiązaną za pomocą łącza zewnętrznego, należy:

1. Otworzyć serwis do edycji.
2. Przejść do strony na tym poziomie drzewa serwisu, na którym ta strona ma zostać dodana. Strony są dodawane na bieżącym poziomie, ale z łatwością można je przenieść w inne miejsce w drzewie serwisu.
3. Nacisnąć przycisk **Dodaj stronę**. Na dole drzewa serwisu jest dodawana "Nowa strona"; pojawia się wezwanie do nadania jej nazwy i określenia innych ustawień.
4. Jako typ strony wybrać **Łącze zewnętrzne**.
5. Nadać stronie nazwę. Może się ona składać tylko z liter, cyfr, znaków podkreślenia ( \_ ) i łączników ( - ). Nazwa ta będzie używana w drzewie serwisu do identyfikacji strony. Spacje są automatycznie zastępowane łącznikami.
6. Podać pełny adres URL strony. Na przykład: `https://www.example.com/sharedpage.htm`
7. Jeśli strona ma być otwierana w nowym oknie (lub na nowej karcie) przeglądarki, wybrać opcję **Otwórz łącze w nowym oknie**. W przeciwnym razie strona ta zostanie wyświetlona w bieżącym oknie, zastępując stronę serwisu i nawigację.
8. Opcjonalnie: wybrać dowolną z dostępnych opcji:

- **Strona błędu:** Jeśli jakiś błąd uniemożliwi wyświetlenie żądanej strony, zamiast niej zostanie wyświetlona domyślna strona błędu. Domyślnie, po wybraniu tej opcji, zostaje również wybrana opcja **Ukryj stronę w nawigacji**. Jeśli strona błędów ma nie być uwzględniana w nawigacji, można wyczyścić pole wyboru tej opcji.
  - **Ukryj stronę w nawigacji:** Strona nie będzie uwzględniana w nawigacji automatycznie generowanej dla serwisu.
  - **Strona szczegółów:** Strona ta służy do wyświetlania szczegółowych informacji o elemencie zawartości wybranym na innej stronie. Zawartość strukturalna jest dostępna tylko dla użytkowników firmowych. Domyślnie, po wybraniu tej opcji, zostaje również wybrana opcja **Ukryj stronę w nawigacji**. Jeśli strona szczegółów ma nie być uwzględniana w nawigacji, można wyczyścić pole wyboru tej opcji.  
Jeśli elementy zawartości strukturalnej zostały skonfigurowane do używania strony szczegółów, użytkownik może kliknąć na łączy w celu uzyskania szczegółowych informacji o elemencie zawartości.
  - **Strona wyszukiwania:** Strona ta służy do wyświetlania wyników wyszukiwania. Można wybrać tę stronę w ustawieniach "Łącze" składnika "wyszukiwanie zawartości". Zob. [Wyszukiwanie zawartości](#). Domyślnie, po wybraniu tej opcji, zostaje również wybrana opcja **Ukryj stronę w nawigacji**. Jeśli strona wyszukiwania ma nie być uwzględniana w nawigacji, można wyczyścić pole wyboru tej opcji.
9. Aby zapisać wszystkie oczekujące zmiany w bieżącej aktualizacji, nacisnąć przycisk **Zapisz**.

## Przenoszenie stron


Stronę można przeciągnąć w inne miejsce na bieżącym poziomie drzewa serwisu; aby przenieść stronę na inny poziom, należy ją wyciąć, a następnie wkleić.



W strukturze hierarchicznej zawsze istnieje strona najwyższego poziomu — strona startowa — pod którą są wyszczególniane wszystkie pozostałe strony. Strony można przeciągać na różne poziomy, zmieniając w ten sposób kolejność stron. Ponadto zawsze można wycinać strony i wklejać je w innym miejscu w hierarchii.


Strony mogą mieć swoje zagnieżdżone strony. Jeśli strona mająca zagnieżdżone strony zostanie przeniesiona, to i one zostaną przeniesione i pozostaną pod przeniesioną stroną. Jeśli używany motyw zawiera zautomatyzowaną nawigację, zmiana kolejności stron powoduje zmianę kolejności ich wyświetlania w menu nawigacyjnym.


Aby przenieść stronę do innego miejsca w drzewie serwisu, należy:

1. Otworzyć serwis do edycji.
2. Przejść do strony, która ma zostać przeniesiona.
3. Aby przenieść stronę do innego miejsca na bieżącym poziomie drzewa serwisu, kliknąć na nazwie strony, po czym przeciągnąć ją do innej lokalizacji na liście stron.
4. Aby przenieść stronę na inny poziom drzewa serwisu, kliknąć na nazwie strony,

która ma zostać przeniesiona, po czym kliknąć na ikonie . Kliknąć na ikonie

, aby wyciąć stronę z jej bieżącej lokalizacji, lub kliknąć na ikonie , aby pozostawić kopię strony w bieżącej lokalizacji. Przejść na ten poziom, na który ma zostać

przeniesiona strona, po czym kliknąć na ikonie . Jeśli na tym poziomie zostanie wybrana jakaś strona, to wklejona strona zostanie zagnieżdżona pod tą wybraną stroną.

Kliknąć na ikonie , aby wkleić stronę w bieżącej lokalizacji. Kolejność strony można zmienić, przeciągając ją w inne miejsce na bieżącym poziomie drzewa serwisu.

## Usuwanie stron


Strony można usuwać w ramach aktualizacji.

Usunięta strona jest rejestrowana w aktualizacji (gdy zostanie naciśnięty przycisk **Zapisz**). Strona zostanie usunięta z serwisu podstawowego dopiero wtedy, gdy aktualizacja zostanie opublikowana. Stronę może usunąć każdy użytkownik mający przypisaną rolę "Menedżer" lub "Współtwórca".

### Uwaga:


Gdy aktualizacja usuwająca stronę zostanie opublikowana, nastąpi trwałe usunięcie strony i nie będzie można jej przywrócić. Jeśli zostanie usunięta strona z serwisu wielojęzycznego, to — gdy serwis zostanie opublikowany — zostanie także usunięta przetłumaczona wersja strony. Jeśli zarządzanie dodawaniem i usuwaniem stron ma istotne znaczenie, warto rozważyć używanie osobnych aktualizacji.


Aby usunąć stronę z serwisu, należy:

1. Otworzyć serwis do edycji.
2. Przejść do strony, która ma zostać usunięta, po czym kliknąć na ikonie .
3. Nacisnąć przycisk **OK**, aby potwierdzić decyzję o usunięciu. Aby zapisać wszystkie oczekujące zmiany w bieżącej aktualizacji, nacisnąć przycisk **Zapisz**.

## Zmiana ustawień strony

Można zmienić ustawienia strony, takie jak nazwa, tytuł, adres URL, nagłówki i stopki.

1. Otworzyć serwis do edycji.
2. Kliknąć na stronie, dla której mają zostać zmienione ustawienia, po czym kliknąć na ikonie .
3. Zmienić odpowiednio ustawienia:
  - **Typ strony:**
    - **Strona internetowa:** Strona i jej zawartość rezydują w serwisie. Można nadać nazwę stronie i zdefiniować jej zawartość, określić miejsce strony w serwisie oraz ustalić, jak ma się ona zachowywać w kontekście bieżącego serwisu.

- **Łącze zewnętrzne:** Strona, pochodząca z innej lokalizacji, jest określana przez adres URL. Można nadać nazwę stronie, określić jej miejsce w serwisie oraz ustalić, jak ma się ona zachowywać w kontekście bieżącego serwisu. Ponieważ jest używana strona z aktywnego serwisu, nie można zmienić jej zawartości. W drzewie serwisu, przed nazwą strony, która jest powiązana za pomocą łącza z zewnętrznym adresem URL, jest wyświetlana ikona .

Jeśli wybrano stronę internetową, zostaną wyświetlone następujące ustawienia:

- **Nazwa strony:** Można używać liter, cyfr, znaków podkreślenia ( \_ ) i łączników ( - ). Dla szablonów, motywów, składników, serwisów ani stron serwisów nie wolno używać następujących nazw: authsite, content, pages, scsTEMPLATE\_\*, \_comps, \_components, \_compsdelivery, \_idcSERVICE, \_sitescloud, \_sitesclouddelivery, \_themes, \_themesdelivery. Następujących nazw można używać dla stron serwisów, ale nie można używać dla szablonów, motywów, składników ani serwisów: documents, sites.
- **URL strony :** Domyślnie nazwa ta będzie używana w adresie URL strony. Spacje są automatycznie zastępowane łącznikami. Aby użyć adresu URL innego niż domyślny, należy nacisnąć przycisk **Przesłoń**, po czym dodać nazwę pliku używaną w adresie URL. Może się ona składać tylko z liter, cyfr, znaków podkreślenia ( \_ ) i łączników ( - ). Należy pamiętać, aby dołączyć rozszerzenie nazwy pliku. Rozszerzeniem domyślnym jest .html.
- **Układ strony:** Układ strony definiuje ogólną strukturę strony, lecz nie jej zawartość. Układ zawiera nazwane *gniazda* (jedno lub większą ich liczbę), w których można umieszczać zawartość. Liczba i typy układów strony zależą od motywu powiązanego z serwisem.
- **Układ strony dla urządzenia mobilnego:** Układ ten definiuje ogólną strukturę strony, gdy jest ona wyświetlana na urządzeniu mobilnym.
- **Tytuł strony:** Tytuł strony nie pojawia się na samej stronie, lecz na pasku tytułu w przeglądarce, gdy strona zostanie wyświetlona. Aby dodać tytuł pojawiający się na stronie, należy za pomocą edytora dodać składnik "tytuł".
- **Opis strony:** Opis strony nie pojawia się na samej stronie, lecz dostarcza informacji o stronie, z których — oprócz nas — mogą skorzystać inni współtwórcy.
- **Słowa kluczowe:** Opcjonalnie można określić rozdzielone przecinkiem słowa kluczowe pomagające wyszukiwarkom rozpoznawać zawartość strony. Słowa kluczowe są przydatne do identyfikowania terminów i pojęć, które nie występują w tekście strony lub które pojawiają się na obrazach. Nie pojawiają się na stronie, lecz są używane przez wyszukiwarki do odnajdywania i rozpoznawania serwisu. Dobry opis, słowa kluczowe i synonimy mogą zwiększyć liczbę odwiedzin serwisu.
- **Nagłówek strony:** Opcjonalnie można dodać do nagłówka strony skrypty lub tagi na potrzeby analiz lub śledzenia. Każdy kod, użyty w nagłówku lub stopce, trzeba dokładnie sprawdzić, aby się upewnić, że działa poprawnie i nie stwarza zagrożenia dla serwisu.
- **Stopka strony:** Opcjonalnie można dodać do stopki strony skrypty lub tagi na potrzeby analiz lub śledzenia. Domyślnie stopka zawiera tekst pokazywany w wyskakującym oknie, dotyczący użycia plików cookie w serwisie, oraz łącze do oświadczenia o ochronie prywatności. Będąc twórcą oprogramowania, można ten tekst zmienić. Każdy kod, użyty w nagłówku lub stopce, trzeba



dokładnie sprawdzić, aby się upewnić, że działa poprawnie i nie stwarza zagrożenia dla serwisu.

- **Opcje strony:**
  - **Strona błędu:** Jeśli jakiś błąd uniemożliwi wyświetlenie żądanej strony, zamiast niej zostanie wyświetlona domyślna strona błędu. Domyślnie, po wybraniu tej opcji, zostają również wybrane opcje **Ukryj stronę w nawigacji** i wszystkie opcje wyszukiwarki. Jeśli strona błędów ma nie być uwzględniana w nawigacji lub w wynikach wyszukiwarki, można wyczyścić pole wyboru tej opcji.
  - **Ukryj stronę w nawigacji:** Strona nie będzie uwzględniana w nawigacji automatycznie generowanej dla serwisu.
  - **Strona szczegółów:** Strona ta służy do wyświetlania szczegółowych informacji o elemencie zawartości wybranym na innej stronie. Zawartość strukturalna jest dostępna tylko dla użytkowników firmowych. Domyślnie, po wybraniu tej opcji, zostaje również wybrana opcja **Ukryj stronę w nawigacji**. Jeśli strona szczegółów ma nie być uwzględniana w nawigacji, można wyczyścić pole wyboru tej opcji.  
Jeśli elementy zawartości strukturalnej zostały skonfigurowane do używania strony szczegółów, użytkownik może kliknąć na łączy w celu uzyskania szczegółowych informacji o elemencie zawartości.
  - **Strona wyszukiwania:** Strona ta służy do wyświetlania wyników wyszukiwania. Można wybrać tę stronę w ustawieniach "Łącze" składnika "wyszukiwanie zawartości". Zob. [Wyszukiwanie zawartości](#). Domyślnie, po wybraniu tej opcji, zostaje również wybrana opcja **Ukryj stronę w nawigacji**. Jeśli strona wyszukiwania ma nie być uwzględniana w nawigacji, można wyczyścić pole wyboru tej opcji.
- **Opcje SEO:**
  - **Ukryj stronę przed wyszukiwarkami:** Wyszukiwarki nie będą indeksowały zawartości strony i tym samym nie będą wyświetlały jej w wynikach wyszukiwania.
  - **Ukryj łącza do stron przed wyszukiwarkami:** Powiadomianie wyszukiwarek, że mają nie podążać za łączy ze strony i tym samym nie indeksować miejsc docelowych.
  - **Wyłącz buforowanie stron dla wyszukiwarki:** Powiadomianie wyszukiwarek, że mają nie buforować danej strony.
  - **Ukryj opis strony przed wyszukiwarkami:** Powiadomianie wyszukiwarek, że mają nie dołączać opisu (podany powyżej) po stronie pokazywanej w wynikach wyszukiwania.
- **Wspólne przeglądanie:** Jeśli administrator włączył dla systemu usługę wspólnego przeglądania oraz została ona włączona dla tego serwisu, to widoczna będzie opcja wyboru, czy na tej stronie ma się znajdować przycisk wspólnego przeglądania, używany w sesjach przeglądania. Sesja wspólnego przeglądania używa usługi Oracle Cobrowse Cloud Service do zarządzania środowiskiem udostępniania ekranów osobom odwiedzającym serwis. Zob. [Włączanie integracji z usługą Cobrowse](#) i [Korzystanie z usługi Cobrowse na stronie](#).
- **Opcje analiz:** Jeśli serwis ten zawiera snippet śledzenia analiz, można przesłonić snippet dla danej strony. Należy kliknąć na przełączniku przesłaniania, po czym edytować skrypt. Więcej informacji jest dostępnych pod hasłem [Dodawanie śledzenia analiz](#). Jeśli użytkownik edytuje snippet serwisu, pojawi się komunikat informujący, że

skrypt został zmodyfikowany. Aby usunąć swoje dostosowania, należy użyć opcji **Przywróć do najnowszej wersji skryptu serwisu**.

Jeśli wybrano łącze zewnętrzne, zostaną wyświetlone następujące ustawienia:

- **Nazwa strony:** Można używać liter, cyfr, znaków podkreślenia ( \_ ) i łączników (-). Nazwa ta będzie używana w drzewie serwisu do identyfikacji strony. Spacje są automatycznie zastępowane łącznikami.
  - **URL łącza:** Podać pełny adres URL strony. Na przykład: `https://www.example.com/sharedpage.htm`.
  - **Otwórz łącze w nowym oknie:** Jeśli strona ma być otwierana w nowym oknie (lub na nowej karcie) przeglądarki, należy wybrać tę opcję. W przeciwnym razie strona ta zostanie wyświetlona w bieżącym oknie, zastępując stronę serwisu i nawigację.
  - **Opcje strony:**
    - **Strona błędu:** Jeśli jakiś błąd uniemożliwi wyświetlenie żądanej strony, zamiast niej zostanie wyświetlona domyślna strona błędu. Domyślnie, po wybraniu tej opcji, zostaje również wybrana opcja **Ukryj stronę w nawigacji**. Jeśli strona błędów ma nie być uwzględniana w nawigacji, można wyczyścić pole wyboru tej opcji.
    - **Ukryj stronę w nawigacji:** Strona nie będzie uwzględniana w nawigacji automatycznie generowanej dla serwisu.
    - **Strona szczegółów:** Strona ta służy do wyświetlania szczegółowych informacji o elemencie zawartości wybranym na innej stronie. Zawartość strukturalna jest dostępna tylko dla użytkowników firmowych. Domyślnie, po wybraniu tej opcji, zostaje również wybrana opcja **Ukryj stronę w nawigacji**. Jeśli strona szczegółów ma nie być uwzględniana w nawigacji, można wyczyścić pole wyboru tej opcji. Jeśli elementy zawartości strukturalnej zostały skonfigurowane do używania strony szczegółów, użytkownik może kliknąć na łączu w celu uzyskania szczegółowych informacji o elemencie zawartości.
    - **Strona wyszukiwania:** Strona ta służy do wyświetlania wyników wyszukiwania. Można wybrać tę stronę w ustawieniach "Łącze" składnika "wyszukiwanie zawartości". Zob. [Wyszukiwanie zawartości](#). Domyślnie, po wybraniu tej opcji, zostaje również wybrana opcja **Ukryj stronę w nawigacji**. Jeśli strona wyszukiwania ma nie być uwzględniana w nawigacji, można wyczyścić pole wyboru tej opcji.
4. Aby zapisać wszystkie oczekujące zmiany w bieżącej aktualizacji, nacisnąć przycisk **Zapisz**.

## Zmianianie układu strony

Układ wyznacza sposób rozmieszczenia zawartości na stronie. Różne układy mogą zawierać różną liczbę nazwanych *gniazd*, będących obszarami rozciągającymi się na szerokość strony. Gniazdo może zawierać jeden lub większą liczbę typów zawartości.

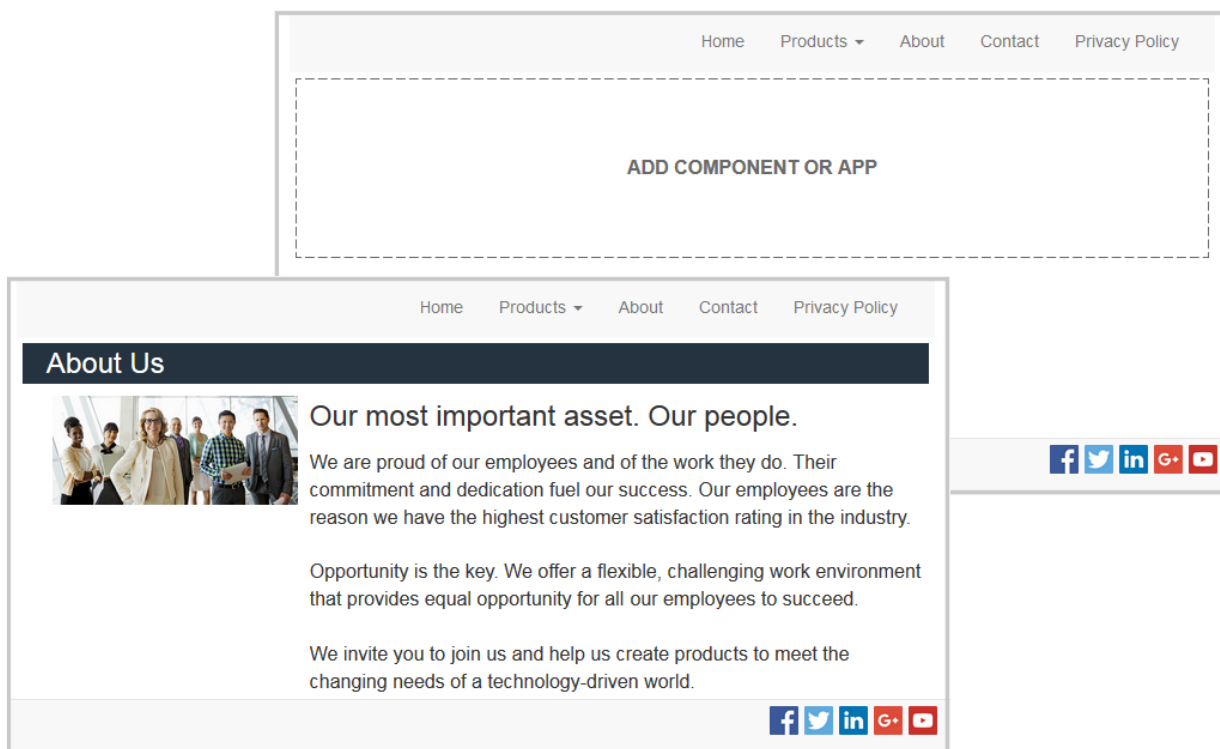
Każdy motyw zawiera kilka układów strony. Dodając stronę do serwisu, wybiera się jej układ. Każdy układ zawiera gniazda, do których można przeciągać zawartość. To, jaka zawartość znajdzie się w tych gniazdach, zależy wyłącznie od nas. Może to być wszystko: od tytułów, tekstów i separatorów do multimediów, galerii i mediów społecznościowych. Można rozmieszczać zawartość w gnieździe, lecz nie można

zmienić liczby ani rozmieszczenia gniazd na stronie. W tym celu trzeba użyć nowego układu strony.


 **Uwaga:**

Można także zastąpić jeden układ innym. Należy tu jednak zachować ostrożność. Jeśli zostanie wybrany układ z mniejszą liczbą gniazd lub z gniazdami mającymi inne nazwy, istniejąca w innych gniazdach zawartość nie zostanie wyświetlona w nowym układzie. Zawartość nie zostanie usunięta; nie będzie jednak można jej wyświetlić, jeśli wybrany układ nie będzie miał gniazda o tej samej nazwie.

Na poniższym rysunku jest pokazany przykładowy układ strony. Widać puste gniazdo w układzie strony oraz gotową stronę z tytułem, obrazem i tekstem dodanym do gniazda.



Aby wybrać inny układ strony, należy:

1. Otworzyć serwis do edycji.
2. Przejść do strony, po czym kliknąć na ikonie .
3. Z listy "Układ strony" wybrać inny układ. Liczba i typy układów strony zależą od motywu powiązanego z serwisem.
4. Aby zapisać wszystkie oczekujące zmiany w bieżącej aktualizacji, nacisnąć przycisk **Zapisz**.

## Zmianianie tła lub motywu

Można określić kolor i obraz tła strony, poszczególnych gniazd na stronie oraz układów sekcji w gnieździe. Można także zmienić cały motyw używany dla serwisu.







Gniazdo jest obszarem rozciągającym się na szerokość strony, mogącym zawierać jeden lub więcej typów zawartości. Określone dla gniazda tło jest stosowane do całego gniazda i wszystkich zawieranych w nim składników.

### Zmianianie tła

Tła są układane warstwowo jedno na drugim. Jeśli zostanie określone tło dla gniazda, znajdzie się ono na wierzchu tła określonego dla strony. W przypadku większości obrazów i kolorów warstwa wierzchnia skutecznie przesłania warstwę spodnią. Jeśli w kolorze lub obrazie tła zostanie wprowadzony pewien stopień przezroczystości, kolory z warstw spodnich mogą prześwitywać przez kolory z warstw wierzchnich lub się z tymi kolorami mieszać.

Niektóre składniki, takie jak "akapity" i "tytuły", mogą udostępniać kolor tła jako element predefiniowanego lub dostosowanego stylu. Jeśli zostanie określony kolor tła składnika, to zostanie ono użyte w postaci warstwy umieszczonej na wierzchu tła układu sekcji, gniazda lub strony. Zob. [Korzystanie ze stylów i formatowania](#).

Opcje tła stron, gniazd i układów sekcji są podobne, lecz różni się sposób uzyskiwania do nich dostępu:

1. Otworzyć serwis do edycji.
2. Przejść do strony, która ma być edytowana, i upewnić się, że przełącznik  jest ustawiony w pozycji **Edytuj**.
3. Aby zmienić tło strony lub gniazda, należy:
  - Kliknąć na ikonie  na pasku bocznym, po czym kliknąć na ikonie .
  - Aby określić tło dla całej strony, wybrać opcję **Cała strona**, po czym kliknąć na ikonie . Aby określić tło dla gniazda na stronie, wybrać gniazdo, po czym kliknąć na ikonie .
4. Aby zmienić tło układu sekcji:
  - Kliknąć na ikonie  menu układu sekcji, po czym wybrać opcję **Ustawienia**.
  - Wybrać opcję **Tło**. Opcje tła są takie same dla stron, gniazd i układów sekcji.
5. Aby użyć obrazu na tło, nacisnąć przycisk **Wybór obrazu**. Można używać obrazów z dowolnej lokalizacji, do której mamy dostęp. Można także używać obrazów, które zostały nam udostępnione lub które wysłaliśmy z lokalizacji lokalnej lub sieciowej.
6. Wybrać obraz, po czym nacisnąć przycisk **OK**. W oknie są wyświetlane wszystkie dostępne pliki. Wystarczy wybrać typ pliku właściwy w danym kontekście. Na przykład, jeśli jest wybierany plik obrazu, należy wybrać plik w formacie graficznym (GIF, JPG, JPEG, PNG lub SVG).
7. Aby dostosować ustawienia obrazu:

- Korzystając z listy **Pozycja**, umieścić obraz na stronie lub w gnieździe.
  - Korzystając z listy **Skala**, dostosować sposób przedstawienia obrazu:
    - **Dopasuj**: Obraz jest skalowany tak, aby bez odkształceń zmieścił się w dostępnym miejscu.
    - **Rozciągnij**: Większy z dwóch wymiarów (szerokość lub wysokość) jest skalowany tak, aby pasował do dostępnego miejsca, a mniejszy wymiar jest rozciągany tak, aby dostępne miejsce zostało wypełnione.
    - **Brak**: Obraz jest używany w jego pełnej rozdzielczości pikselowej; jeśli się nie mieści w dostępnym miejscu jest równomiernie przycinany.
  - Wybrać opcję **Powtarzaj**, aby użyć obrazu (który jest za mały do wypełnienia całego dostępnego miejsca) na kafelki. Opcja ta nie ma zastosowania, jeśli z listy **Skala** została wybrana opcja **Rozciągnij**.
  - Jeśli obraz ma się nie przemieszczać, gdy użytkownik będzie przewijał stronę, wybrać opcję **Nie przewijaj ze stroną**.
8. Aby określić kolor tła, wybrać z menu **Kolor** odpowiedni kolor albo — aby użyć koloru pochodzącego z pełnego zakresu — wybrać opcję **Więcej**.
- Kliknąć na pasku spektrum (z prawej strony), aby wybrać kolor i wyświetlić jego odmiany w obszarze zakresu koloru. W prawym górnym rogu jest wyświetlany wybrany kolor; w lewo są wyświetlane odcienie z rosnącym udziałem bieli, a w dół — z rosnącym udziałem czerni.
  - Aby wybrać wariant koloru, kliknąć w obrębie obszaru zakresu koloru albo przeciągnąć punkt wyboru do miejsca, w którym znajduje się nowy kolor. Obecne pole koloru i 6-cyfrowy kod koloru sygnalizują wybrany kolor.
  - Aby dostosować przezroczystość (kanał alfa), przeciągnąć suwak przezroczystości; przeciągnięcie w lewo zwiększa stopień przezroczystości. Wartość alfa równa 0% czyni kolor przezroczystym, wskutek czego kolor i zawartość warstw spodnich są w pełni widoczne. Wartość alfa równa 100% czyni kolor zupełnie nieprzezroczystym, wskutek czego kolor i zawartość warstw spodnich są niewidoczne. Dla innych wartości obrazy i kolory z warstw spodnich prześwitują, a ich kolory mieszają się w różnym stopniu z kolorami warstw wierzchnich.
  - Aby zastosować bieżący wybór koloru, nacisnąć przycisk **Wybierz**.



### Zmianianie motywu

Zmiana motywu jest rejestrowana w aktualizacji (gdy zostanie naciśnięty przycisk **Zapisz**). Zmiana motywu zostanie zastosowana dopiero wtedy, gdy aktualizacja zostanie opublikowana.

 **Uwaga:**

Nie wszystkie motywy są wzajemnie zgodne. Różne motywy mogą zawierać różne nawigacje, nazwy stylów lub informacje o układzie strony. Na przykład, jeśli wybrano motyw mający inne nazwy układów, istniejące strony mogą używać układów, które nie są dostępne w tym motywie. Wówczas te strony będą wyświetlane jako puste. Zawartość strony nie zostanie usunięta; nie będzie jednak można jej wyświetlić, jeśli wybrany motyw nie będzie miał układu i gniazd o tych samych nazwach. W przypadku wybrania niezgodnego motywu — aby przywrócić zawartość strony — można z powrotem zmienić motyw na oryginalny. Należy się upewnić, że przełączenie nastąpi do zgodnego motywu. Nie mając pewności, należy się skontaktować z twórcą motywu.

Aby zmienić używany dla serwisu motyw, należy:

1. Otworzyć serwis do edycji.
2. Kliknąć na ikonie  na pasku bocznym, po czym kliknąć na ikonie .
3. Wybrać motyw z listy dostępnych motywów, po czym nacisnąć przycisk **OK**. Motyw zostanie zastosowany w bieżącej aktualizacji. Aby zapisać wszystkie oczekujące zmiany w bieżącej aktualizacji, nacisnąć przycisk **Zapisz**.

## Rozmieszczanie zawartości strony

Składniki i elementy zawartości udostępniają funkcje i potrzebną zawartość oraz układy sekcji, umożliwiające automatyczne ich rozmieszczanie.

- [Dodawanie składników i układów sekcji](#)
- [Praca z zasobami i elementami zawartości](#)
- [Używanie wyzwalaczy i czynności](#)
- [Używanie układów sekcji poziomej](#)
- [Używanie układów sekcji pionowej](#)
- [Używanie układów z dwiema i trzema kolumnami](#)
- [Używanie układów sekcji z kartami](#)
- [Używanie układów sekcji suwaka](#)

Informacje, jak korzystać z poszczególnych składników, są zawarte pod hasłem [Używanie składników wbudowanych](#).

Informacje, jak zarządzać składnikami i układami, są dostępne pod hasłem [Zarządzanie niestandardowymi składnikami i układami](#).



Szczegółowe informacje dotyczące tworzenia składników można znaleźć pod hasłem [Tworzenie składników](#).




## Dodawanie składników i układów sekcji

*Składniki* są indywidualnymi komponentami strony internetowej. *Układ sekcji* automatycznie organizuje dodawaną do niej zawartość, ułatwiając współtwórcom dodawanie zawartości bez poświęcania czasu na jej formatowanie.


### Dodawanie składników

Aby dodać do strony składnik, należy:

1. Przejść do strony, która ma być edytowana, i upewnić się, że przełącznik  jest ustawiony w pozycji **Edytuj**.
2. Kliknąć na ikonie , po czym wybrać jeden z następujących typów składników:
  - Aby wyświetlić listę składników, które zostały wybrane do używania w motywie powiązanim z serwisem, wybrać opcję **Używane w motywie**.
  - Aby wyświetlić listę udostępnionych składników niestandardowych, wybrać opcję **Niestandardowe**.
  - Aby wyświetlić listę składników domyślnych (dostępnych wraz z usługą), wybrać opcję **Standardowe**.
  - Aby wyświetlić wszystkie udostępnione składniki, wybrać opcję **Wszystkie**.

3. Kliknąć na ikonie, po czym przeciągnąć tę ikonę w odpowiednie miejsce na stronie. Po przeciągnięciu elementu do strony, pojawią się granice dostępnych gniazd, układów sekcji i wszystkie istniejące elementy. Ikona umieszczania () lub pionowy pasek umieszczania ( — umieszczanie z lewej lub z prawej) sygnalizuje, gdzie znajdzie się nowa zawartość. Ciągłe obramowanie wokół układu sekcji lub elementu zawartości informuje, że można tu przeciągnąć element i zostanie on automatycznie umieszczony. W gnieździe może się znajdować kilka elementów; elementy można przemieszczać na stronie, przeciągając je w nowe miejsca.
4. Gdy element znajdzie się we właściwym miejscu na stronie, "upuścić" go.
5. Aby dostosować właściwości elementu, kliknąć na jego ikonie menu , po czym wybrać opcję **Ustawienia**. W zależności od elementu trzeba będzie dodać tekst, określić łącze do obrazu, określić adres URL innego serwisu lub mapy, zmienić rozmiar elementu lub wykonać inne czynności.

Każdy składnik ma swoje ustawienia (takie jak rozmiar, wyrównanie, odstępy, kolory i krawędzie). Definiują one wygląd i działanie danego składnika. Na przykład ustawienia składnika "akapit" (obejmujące m.in. typ i rozmiar czcionki) decydują o sposobie wyświetlania tekstu w akapicie.

Aby dostosować właściwości składnika, należy go wybrać, a następnie kliknąć na ikonie **Menu** , po czym wybrać opcję **Ustawienia**. Różne składniki mają różne rodzaje ustawień:



Ustawienia	Opis
<b>Ogólne</b>	Ustawienia "Ogólne" obejmują odstępy, wyrównanie i ustawienia wspólne dla składników.
<b>Styl</b>	Styl to nazwany zbiór domyślnych wartości decydujących o wyglądzie. Style są definiowane w motywie. Różne motywy mogą zawierać różne style, a — w obrębie motywu — różne składniki mogą mieć różne style. Można również ręcznie określić ustawienia stylu dla konkretnego wystąpienia składnika.
<b>Łącze</b>	Ustawienia "Łącze" obejmują lokalizację obrazów, dokumentów lub innych zasobów używanych przez składnik. W przypadku przycisków i innych składników, które mogą wykonywać czynności, ustawienia "Łącze" również obejmują wyzwacze i czynności obsługiwane przez składnik.
<b>Składniki</b>	Ustawienia "Składniki", obejmujące inne składniki standardowe (takie jak składnik "artykuł"), udostępniają listę składników i zapewniają dostęp do ustawień każdego z nich.
<b>Niestandardowe</b>	Ustawienia "Niestandardowe" są unikatowe dla składnika i są prezentowane osobno od grup ustawień standardowych. Składniki odległe mogą na przykład przechowywać preferencje pod unikatowym adresem URL i prezentować je jako ustawienia niestandardowe.

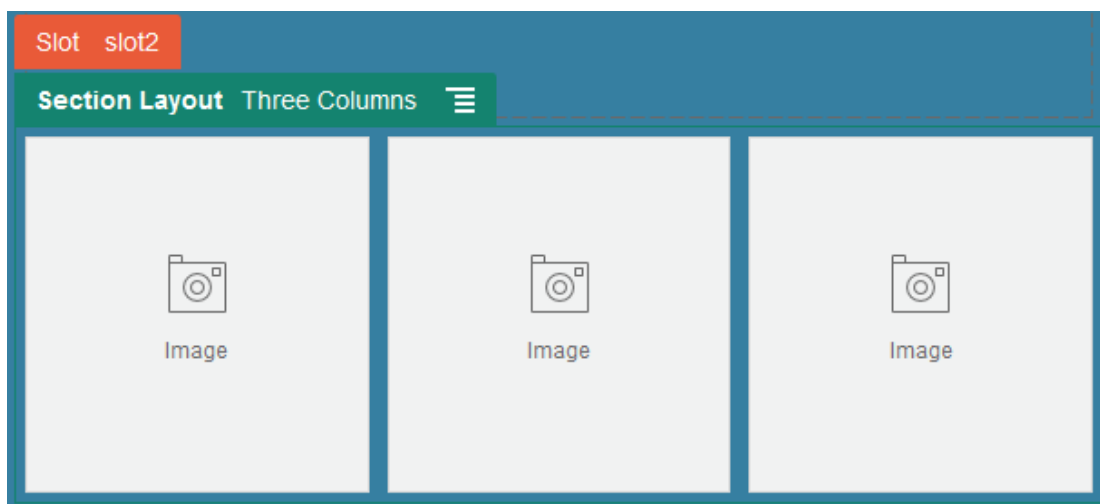
#### Używanie układów sekcji


Układ sekcji może automatycznie organizować zawartość w wielu kolumnach, w formie pionowej listy lub w obszarze z kartami. Współtwórca serwisu może, w celu

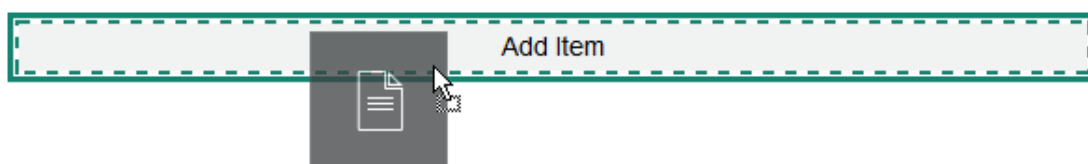


porządkowania zawartości, dodać do gniazda jeden lub większą liczbę układów sekcji. Aby dodać układ do strony, należy:

1. Przejsz do strony, która ma być edytowana, i upewnić się, że przełącznik  jest ustawiony w pozycji **Edytuj**.
2. Aby wstawić układ sekcji na stronie, kliknąć na ikonie . Wybrać typ układu, który ma zostać użyty, po czym postępować zgodnie ze wskazówkami dotyczącymi umieszczania elementów (tymi samymi, które są używane do umieszczania składników).



Ikona umieszczania  sygnalizuje, gdzie znajdzie się układ sekcji (na górze, na dole, z lewej, z prawej). Ciągłe obramowanie wokół układu sekcji lub elementu zawartości informuje, że można tu przeciągnąć element i zostanie on automatycznie umieszczony:






Będąc użytkownikiem firmowym pracującym z elementami zawartości strukturalnej, można przypisać układ sekcji podczas dodawania listy elementów zawartości do strony; układ sekcji będzie automatycznie formatował elementy na stronie.

Programista — aby rozwiązać pewne problemy z układem lub uprościć pracę współtwórców — może utworzyć dodatkowe układy sekcji, jako uzupełnienie już istniejących. Projektant motywu może nawet wbudować układy sekcji w gniazdo w układzie strony. Zob. [Tworzenie układów](#).

## Praca z zasobami i elementami zawartości

Będąc użytkownikiem firmowym, można do serwisu dołączać zasoby cyfrowe i elementy zawartości przechowywane w *kolekcji serwisu* lub powiązonym repozytorium. Kolekcja to podzbiór zasobów przechowywanych w repozytorium, które mogą być używane w serwisie.

Aby dodać zasób cyfrowy lub element zawartości do strony, należy się upewnić, że przełącznik  jest ustawiony w pozycji **Edytuj**, po czym kliknąć na ikonie . Jeśli serwis korzysta z więcej niż jednego repozytorium, należy wybrać repozytorium, które ma być używane. Aby uzyskać więcej informacji o używaniu różnych repozytoriów w serwisie, zob. Zapewnianie serwisowi dostępu do więcej niż jednego repozytorium w podręczniku *Zarządzanie zasobami przy użyciu Oracle Content Management*.

Filtrując zasoby, można znaleźć dokładnie to, co jest potrzebne. Następnie należy kliknąć na ikonie , po czym wybrać preferowany sposób zawężania wyboru. Jeśli nie są widoczne żadne zasoby, może to oznaczać, że zasoby nie są częścią kolekcji serwisu. Aby zobaczyć wszystkie kolekcje w repozytorium, a nie tylko kolekcję serwisu, należy zmienić filtr. Szczegółowe informacje są dostępne na stronie Wyszukiwanie, filtrowanie i sortowanie zasobów.

Następnie należy przeciągnąć zasób cyfrowy lub element zawartości z panelu do gniazda na stronie. Obrazy można osadzać bezpośrednio w składniku "Akapit" w miejscu położenia kursora; można także używać opcji wprowadzenia tekstu alternatywnego, ustawienia wysokości i szerokości obrazu oraz ustawienia wyrównania.

Jeśli do strony jest dodawany element z kolekcji serwisu, to element ten zostaje automatycznie umieszczony w składniku o odpowiednim typie. Na przykład, jeśli jest dodawany zasób cyfrowy "obraz", zostaje on automatycznie umieszczony w składniku "obraz". Jeśli jest dodawany element zawartości, zostaje on automatycznie umieszczony w składniku "element zawartości". Można także najpierw dodać składnik "obraz" lub "element zawartości", a później z panelu "Zawartość" przeciągnąć element zawartości do składnika. Ponadto można wybrać opcję **Ustawienia** składnika, po czym — aby można było wybrać obraz z zasobów lub z listy dokumentów — nacisnąć przycisk **Wybierz**.

## Używanie wyzwalaczy i czynności

Składniki "przycisk", mogą inicjować czynności, takie jak pokazywanie lub ukrywanie składników strony czy wyświetlanie komunikatów. Niektóre składniki, takie jak listy folderów i plików, mogą — bazując na wyborze dokonany przez użytkownika — inicjować czynności w powiązonym składniku.

Na przykład można skonfigurować przycisk, aby użytkownik — gdy go naciśnie (wyzwalacz) — był kierowany na inną stronę lub pod zewnętrzny adres URL i aby był wyświetlany alarm informujący użytkownika o zmianie (czynności). Sposób wyświetlania zmienia się w zależności od rodzaju używanych składników.

### Configure Trigger Actions

Select the actions for the trigger: **Click on Button**

**Page Actions**

- Navigate to Site Page
- Navigate to External URL
- Show Alert
- Show or Hide a Component
- Show or Hide a Slot

**Show Alert** ✕

Message

You are going to the About page.

**And**

**Navigate to Site Page** ✕

Page

About

Target

Open in Same Window

OK
Cancel

Jeśli jest używana więcej niż jedna czynność, należy starannie rozważyć kolejność operacji i umieszczać czynności we właściwej kolejności. W powyższym przykładzie należałoby jako pierwszą umieścić czynność wyświetlania alarmu. Da to użytkownikowi czas na przeczytanie i zamknięcie komunikatu, zanim nastąpi przekierowanie na inną stronę. Jeśli jako pierwsza zostałaby użyta czynność przekierowania, komunikat mógłby zostać zastąpiony nową stroną, co uniemożliwiłoby przeczytanie go.

#### Czynności związane ze stroną

Wszystkie składniki, obsługujące wyzwalacze i czynności, obsługują także czynności związane ze stroną:

- **Przejęcie do strony serwisu:** Wybrać stronę w bieżącym serwisie.
- **Przejęcie pod zewnętrzny URL:** Podać pełny adres URL strony zewnętrznej lub serwisu.
- **Wyświetlenie alarmu:** Wyświetlenie określonego komunikatu w oknie.
- **Pokazanie lub ukrycie składnika:** Wybrać składnik z listy składników bieżącej strony, który ma zostać pokazany, ukryty lub przełączony.
- **Pokazanie lub ukrycie gniazda:** Wybrać gniazdo z listy składników bieżącej strony, które ma zostać pokazane, ukryte lub przełączone.

#### Czynności specyficzne dotyczące składników


Składniki mogą definiować także swoje własne czynności. Czynności te pozwalają składnikowi komunikować się z innymi składnikami oraz inicjować czynności w powiązonym składniku. Na przykład, gdy użytkownik wybierze folder w składniku "lista folderów", składnik ten może wyświetlić pliki zawarte w wybranym folderze. W tym przypadku składnik "lista

folderów" obsługuje wyzwalacz **Wybrano folder**, a składnik "lista plików" obsługuje czynność **Wyświetl pliki**.

Jeśli do strony zostaną dodane składniki obsługujące czynności, to składnik oraz wszelkie obsługiwane czynności zostaną dodane do listy dostępnych czynności. Niektóre składniki zapewniają obsługę tylko czynności (lub tylko określonych czynności). Inne składniki udostępniają wyzwalacze, lecz nie obsługują żadnych czynności.

Informacje, jak tworzyć swoje własne składniki, są dostępne pod hasłem [Tworzenie składników](#).

Aby określić jedną lub więcej czynności np. dla składnika "przycisk", należy:


1. Kliknąć na ikonie menu  przycisku, po czym wybrać opcję **Ustawienia**.
2. W panelu "Ustawienia" kliknąć na pozycji **Łącze**.
3. Z listy **Wybór typu łącza** wybrać opcję **Czynności wyzwalacza**.
4. W obszarze "Dostępne wyzwalacze" wybrać opcję **Naciśnięcie przycisku**.
5. W oknie "Konfiguracja czynności wyzwalacza" przeciągnąć czynność z kolumny (po lewej stronie) do gniazda z etykietą **Wykonywana czynność**.

## Używanie układów sekcji poziomej

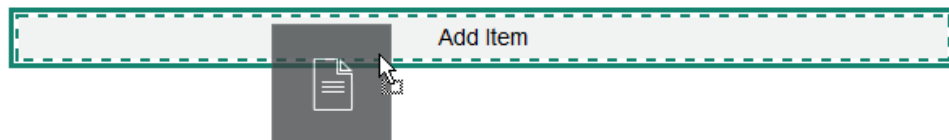
Układu sekcji można używać w celu automatycznego ustalania odstępów i rozmieszczania składników dodawanych do układu.

Układ poziomy służy do rozmieszczania elementów dodanych do tego układu jeden po drugim, w poziomie. Układ zmienia się proporcjonalnie wraz ze zwiększaniem lub zmniejszaniem szerokości strony. Domyślnie elementy są ustawiane w jednym rzędzie i mają przydzielaną w poziomie jednakową ilość miejsca.

Aby dodać układ do strony, należy:

1. Przejść do strony, która ma być edytowana, i upewnić się, że przełącznik  jest ustawiony w pozycji **Edytuj**.
2. Dodać układ do strony.
3. Aby umieścić inną zawartość w układzie sekcji, przeciągnąć odpowiednią zawartość do układu.


Układ zostaje wyróżniony ciągłym obramowaniem i banerem z napisem "Dodaj element".



Można kontynuować dodawanie elementów do układu, a układ odpowiednio je sformatuje. Można nawet dodawać inne układy sekcji, aby uzyskać bardziej złożony układ.

Poniżej został pokazany układ poziomy ze składnikami "tekst", ponumerowanymi w celu pokazania kolejności elementów w układzie:



4. Aby edytować składnik i jego wygląd, kliknąć na ikonie menu  składnika, po czym wybrać opcję **Ustawienia**. Nie chcąc używać domyślnej proporcjonalnej zmiany rozmiaru, można ustawić szerokość poszczególnych obszarów. Można również określić wyrównanie (do lewej, wyśrodkowanie lub do prawej).
5. Aby zmodyfikować ustawienia poszczególnych składników w układzie, wybrać kartę "Ogólne".

Aby zobaczyć ustawienia składnika, należy kliknąć na jego nazwie.


6. Aby zmodyfikować ustawienia tła układu, wybrać kartę "Tło".

Zob. [Zmianie tła lub motywu](#).

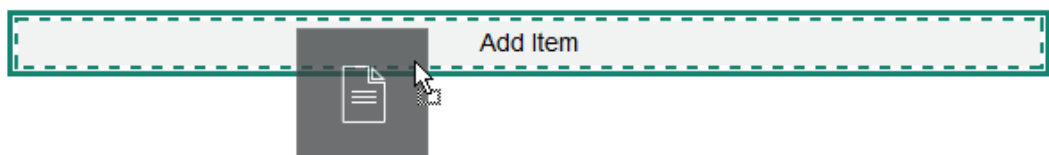
## Używanie układów z dwiema i trzema kolumnami

Układu sekcji z kilkoma kolumnami można używać, aby zapewnić automatyczne ustalanie odstępów i rozmieszczanie składników dodawanych do układu.

Aby dodać układ do strony, należy:

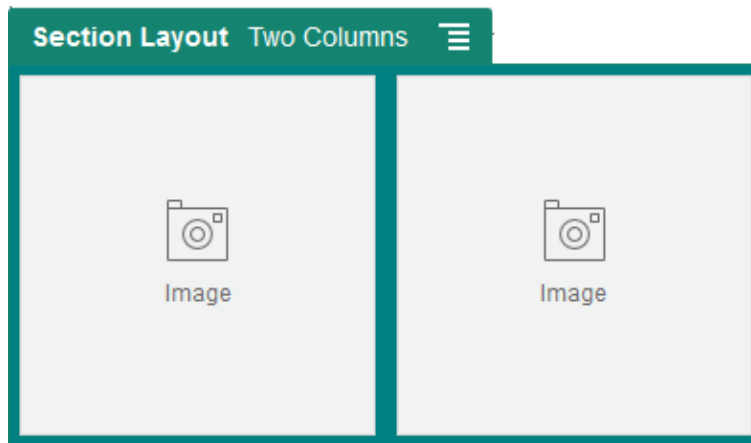
1. Przejść do strony, która ma być edytowana, i upewnić się, że przełącznik  jest ustawiony w pozycji **Edytuj**.
2. Dodać układ do strony.
3. Aby umieścić inną zawartość w układzie sekcji, przeciągnąć odpowiednią zawartość do układu.

Układ zostaje wyróżniony ciągłym obramowaniem i banerem z napisem "Dodaj element".




Można kontynuować dodawanie elementów do układu, a układ odpowiednio je sformatuje. Można także dodawać inne układy sekcji, aby uzyskać układ bardziej złożony, lub dodawać grupy składników.

Poniżej został pokazany układ dwukolumnowy ze składnikami "obraz" użytymi jako elementy zastępcze:



Jeśli zostanie dodana większa liczba elementów, zostaną one wyświetlone w dodatkowych wierszach, każdy z maksymalnie dwoma elementami.

4. Aby edytować składnik i jego wygląd, kliknąć na ikonie menu  składnika, po czym wybrać opcję **Ustawienia**.
5. Aby zmodyfikować ustawienia poszczególnych składników w układzie, wybrać kartę "Ogólne".

Aby zobaczyć ustawienia składnika, należy kliknąć na jego nazwie.
6. Aby zmodyfikować ustawienia tła układu, wybrać kartę "Tło".
7. Wybrać **Ustawienia niestandardowe**, aby określić dodatkowe ustawienia domyślne wyświetlanej zawartości.
  - **Szerokość pierwszej kolumny (%)**: Określić szerokość kolumny jako procent miejsca dostępnego dla układu.
  - **Szerokość drugiej kolumny (%)**: Określić szerokość kolumny jako procent miejsca dostępnego dla układu.
  - **Szerokość trzeciej kolumny (%)**: Określić szerokość kolumny jako procent miejsca dostępnego dla układu.
  - **Responsywny punkt wstrzymania (w pikselach)**: W przypadku projektów stron responsywnych, które automatycznie formatują zawartość, gdy dostępne rozmiary wyświetlania się różnią, określić szerokość w pikselach, przy której układ sekcji ma zostać przełączony ze standardowego układu dwukolumnowego do określonego za pomocą opcji **Działanie responsywne**.
  - **Działanie responsywne**: Wybrać, w jaki sposób układ się zmieni, gdy dostępny rozmiar obszaru wyświetlania jest mniejszy niż wartość podana w polu **Punkt reakcji**.
    - **Brak czynności**: brak dostosowania sposobu działania układu.
    - **Stos z kolumn**: Rozmieszczenie w jednej kolumnie wszystkich elementów, od góry do dołu, z pierwszej kolumny, a następnie wszystkich elementów z drugiej kolumny itd.
    - **Ukryj pierwszą kolumnę**: Ukrycie zawartości z pierwszej kolumny, aby uzyskać więcej miejsca dla pozostałych kolumn.
    - **Ukryj drugą kolumnę**: Ukrycie zawartości z drugiej kolumny, aby uzyskać więcej miejsca dla pozostałych kolumn.


- **Ukryj trzecią kolumnę:** Ukrycie zawartości z trzeciej kolumny, aby uzyskać więcej miejsca dla pozostałych kolumn.
- **Ukryj obie kolumny:** Ukrycie całej zawartości w układzie.
- **Przenieś drugą kolumnę pod pierwszą:** umieszczenie w jednej kolumnie wszystkich elementów z pierwszej kolumny, a pod nimi — wszystkich elementów z drugiej.
- **Przenieś pierwszą kolumnę pod drugą:** umieszczenie w jednej kolumnie wszystkich elementów z drugiej kolumny, a pod nimi — wszystkich elementów z pierwszej.

## Używanie układów sekcji pionowej

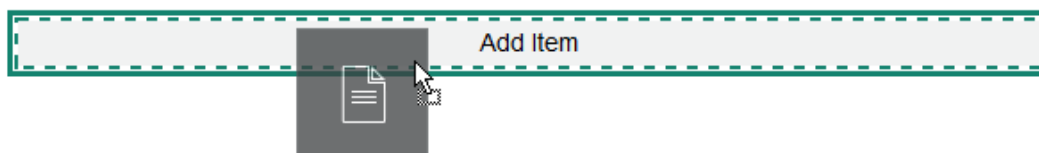
Układu sekcji można używać w celu automatycznego ustalania odstępów i rozmieszczania składników dodawanych do układu.

Układ pionowy służy do rozmieszczania elementów dodanych do tego układu jeden po drugim, w pionie.

Aby dodać układ do strony, należy:

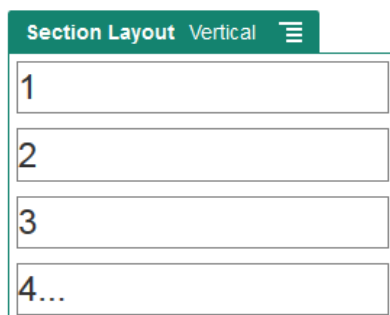
1. Przejść do strony, która ma być edytowana, i upewnić się, że przełącznik  jest ustawiony w pozycji **Edytuj**.
2. Dodać układ do strony.
3. Aby umieścić inną zawartość w układzie sekcji, przeciągnąć odpowiednią zawartość do układu.


Układ zostaje wyróżniony ciągłym obramowaniem i banerem z napisem "Dodaj element".



Można kontynuować dodawanie elementów do układu, a układ odpowiednio je sformatuje. Można nawet dodawać inne układy sekcji, aby uzyskać bardziej złożony układ.

Poniżej został pokazany układ pionowy z ponumerowanymi składnikami "tekst" w celu pokazania kolejności elementów w układzie:




4. Aby edytować składnik i jego wygląd, kliknąć na ikonie menu  składnika, po czym wybrać opcję **Ustawienia**.
5. Aby zmodyfikować ustawienia poszczególnych składników w układzie, wybrać kartę "Ogólne".  
Aby zobaczyć ustawienia składnika, należy kliknąć na jego nazwie.
6. Aby zmodyfikować ustawienia tła układu, wybrać kartę "Tło".


## Używanie układów sekcji z kartami

Układu sekcji z kartami można używać w celu tworzenia odstępów i rozmieszczania składników dodawanych do układu.

Aby dodać układ do strony, należy:

1. Przejść do strony, która ma być edytowana, i upewnić się, że przełącznik  jest ustawiony w pozycji **Edytuj**.
2. Dodać układ do strony. Domyślnie wraz z układem jest dodawana jedna karta. Aby dodać dodatkowe karty, wybrać opcję **Nowa karta**.
3. Aby umieścić inną zawartość na karcie, przeciągnąć odpowiednią zawartość na kartę.


Można kontynuować dodawanie elementów do układu, a układ odpowiednio je sformatuje. Można także dodawać inne układy sekcji, aby uzyskać układ bardziej złożony, lub dodawać grupy składników.

4. Aby edytować kartę i jej wygląd, kliknąć na ikonie  menu składnika, po czym wybrać opcję **Ustawienia**.
5. Aby zmodyfikować ustawienia tła karty, wybrać sekcję "Tło".  
Można użyć obrazu dla karty, zmienić jego pozycję itd. Zob. [Zmianie tła lub motywu](#).
6. Aby zmodyfikować inne ustawienia wyglądu karty, wybrać sekcję "Styl". Wybrać styl (np. linia włosowa czy ramka) lub dostosować obramowanie i narożniki karty.

## Używanie układów sekcji suwaka

Układu sekcji suwaka można używać do tworzenia zawartości, która pozostaje w miejscu, dopóki nie zostanie przesunięta w jedną stronę, robiąc miejsce dla nowej zawartości. Użytkownicy przemieszczają się między slajdami, klikając na punktach pod sekcją slajdu.


Aby dodać układ suwaka do strony, należy:

1. Przejść do strony, która ma być edytowana, i upewnić się, że przełącznik  jest ustawiony w pozycji **Edytuj**.
2. Dodać układ do strony. Domyślnie wraz z układem jest dodawany jeden slajd. Aby dodać dodatkowe slajdy, wybrać opcję **Nowy slajd**.



3. Aby umieścić zawartość w konkretnym slajdzie, wyświetlić slajd w suwaku, po czym przeciągnąć element składnika lub zawartości do tego slajdu.

Jeśli trzeba, można nadal dodawać elementy do slajdu i odpowiednio je rozmieszczać. Można dodawać inne układy sekcji (aby uzyskać układ bardziej złożony) lub dodawać grupy składników umożliwiające umieszczanie zawartości w wybranym miejscu.

4. Aby edytować układ slajdu i jego wygląd, kliknąć na jego ikonie menu , po czym wybrać opcję **Ustawienia**.
5. Aby zmodyfikować ustawienia tła całego obszaru slajdu, wybrać sekcję "Tło".  
Zob. [Zmienianie tła lub motywu](#). Można użyć obrazu, zmienić jego pozycję itd.
6. Aby zmodyfikować inne ustawienia wyglądu, wybrać sekcję "Styl". Wybrać styl (np. linia włosowa czy ramka) lub dostosować obramowanie i narożniki slajdu.
7. Aby zmodyfikować ustawienia poszczególnych slajdów, kliknąć na nazwie slajdu (aby otworzyć jego ustawienia). Dla każdego slajdu zmienić tło, styl itp.

# 10

## Używanie składników wbudowanych

Składniki, układy sekcji, zasoby cyfrowe i zawartość strukturalna zapewniają elastyczność w udostępnianiu zawartości i funkcji potrzebnych użytkownikom.

Zamieszczono tu informacje dotyczące korzystania z poszczególnych składników. Ponadto:

- Informacje, jak używać składników w swoim serwisie, są dostępne pod hasłem [Rozmieszczanie zawartości](#).
- Informacje, jak zarządzać składnikami, są dostępne pod hasłem [Zarządzanie niestandardowymi składnikami i układami](#).
- Informacje, jak tworzyć swoje własne składniki, są dostępne pod hasłem [Tworzenie składników](#).

### Składniki podstawowe

- [Tytuły](#)
- [Akapity](#)
- [Zwykły tekst](#)
- [Przyciski](#)

### Składniki struktury

- [Separatory](#)
- [Rozdzielacze](#)

### Składniki multimedialne

- [Obrazy](#)
- [Galerie](#)
- [Siatka galerii](#)
- [Wideo z serwisu YouTube](#)
- [Wideo](#)

### Składniki "dokumenty"

- [Dokumenty](#)
- [Listy folderów](#)
- [Listy plików](#)
- [Menedżer dokumentów](#)
- [Biblioteka projektu](#)

### Składniki społecznościowe

- [Pasek społecznościowy](#)

- [Facebook — Lubię to/Rekomendacja](#)
- [Twitter - Udostępnij/Obserwuj](#)
- [Składnik "konwersacja"](#)
- [Lista konwersacji](#)

#### Składniki procesu

- [Formularz uruchamiania procesu](#)
- [Lista zadań procesu](#)
- [Formularz szczegółów zadania](#)

#### Elementy zawartości

- [Składnik "element zawartości"](#)
- [Element zastępczy zawartości](#)
- [Lista zawartości](#)
- [Wyszukiwanie zawartości](#)
- [Rekomendacje](#)

#### Inne składniki

- [Mapy](#)
- [Nagłówki](#)
- [Artykuły](#)
- [Obrazy z tekstem](#)
- [Grupy składników](#)
- [Korzystanie z usługi Cobrowse na stronie](#)
- [Oracle Intelligent Advisor](#)
- [Oracle Visual Builder](#)


## Składniki podstawowe


Składniki podstawowe stanowią bloki konstrukcyjne zawartości tekstowej.

- [Tytuły](#)
- [Akapity](#)
- [Zwykły tekst](#)
- [Przyciski](#)

### Tytuły

Aby dodać i sformatować składnik "tytuł", należy:

1. Przejść do strony, która ma być edytowana, i upewnić się, że przełącznik  jest ustawiony w pozycji **Edytuj**.


2. Dodać składnik do strony.
3. Kliknąć w składniku "tytuł", po czym wpisać jego tekst. Tekst przyjmuje formatowanie zgodne ze stylem domyślnym składnika.
4. Aby dodać łącze do tekstu tytułu:
  - a. Wprowadzić tekst, który ma zostać użyty jako tekst łącza, po czym zaznaczyć ten tekst, a następnie kliknąć na ikonie .
  - b. Z listy **Wybór typu łącza** wybrać jedną z następujących opcji:
    - **Strona internetowa:** Należy podać pełny adres URL strony zewnętrznej lub serwisu, a następnie wybrać miejsce otwierania łącza.
    - **Strona serwisu:** Można, używając selektora stron, wybrać stronę w bieżącym serwisie oraz miejsce otwierania łącza. Można określić dodatkowe parametry adresu URL w następującym formacie: `klucz1=wartość1&klucz2=wartość2`. Wartości puste są obsługiwane. Przykład: `klucz1=&klucz2=wartość2`. Można także określić kotwicę URL, ale trzeba dodać specjalny układ sekcji "Anchor" strony serwisu docelowego, na której ma być rozstrzygane łącze, oraz określić tę samą nazwę kotwicy w ustawieniach układu, używanych podczas definiowania czynności uaktywniającej.

 **Uwaga:**

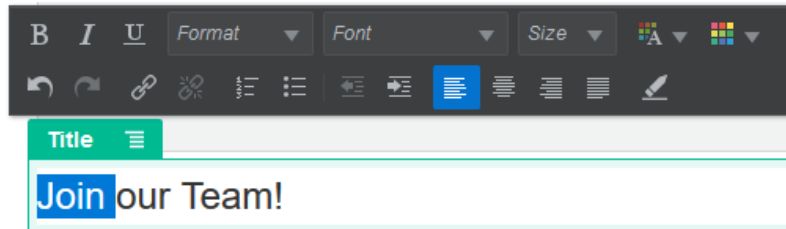
Układ sekcji "Anchor", wymagany do korzystania z kotwic adresów URL, jest dystrybuowany z zestawem OCE Toolkit. Informacje, jak ten zestaw uzyskać, są zawarte pod hasłem Tworzenie za pomocą zestawu OCE Toolkit.

- **Pobieranie pliku:** Pobieranie wybranego pliku z repozytorium. Należy wybrać plik.


Jeśli wybrano zasób cyfrowy, można wybrać określoną realizację. Jeśli realizacja nie zostanie wybrana, będzie stosowany rozmiar oryginalny. Chcąc opublikować najnowszą wersję zasobu (gdy serwis zostanie opublikowany), wybrać opcję **Użyj najnowszej**. Jeśli nie zostanie wybrana opcja **Użyj najnowszej wersji zasobu**, zostanie użyta najnowsza *opublikowana* wersja, zamiast najnowszej wersji roboczej (jeśli taka istnieje).
- **Podgląd pliku:** Podgląd (i opcjonalne pobranie) wybranego pliku na przyciemnionej i nieaktywnej wersji strony.
- **Element zawartości:** Należy wybrać element zawartości z powiązanego repozytorium zasobów, następnie wybrać stronę szczegółów, która ma zostać wyświetlona, po czym przekierować stronę do otwierania jej w tym samym oknie lub nowym.
- **E-mail:** Należy podać poprawny adres e-mail oraz (opcjonalnie) temat. Wynikowa wiadomość zostaje otwarta w domyślnym kliencie poczty elektronicznej i jest przez niego wysyłana.
- **Mapa:** Należy podać poprawny adres e-mail lub poprawne współrzędne, po czym wybrać miejsce otwierania mapy w przeglądarkach stacjonarnych i mobilnych.
- **Połączenie telefoniczne:** Należy podać poprawny numer telefonu.

Aby usunąć łącze, kliknąć w dowolnym miejscu w tekście łącza, a następnie na ikonie .

5. Aby zmienić formatowanie domyślne, zaznaczyć tekst, który ma zostać sformatowany, po czym użyć dowolnych opcji z paska formatowania, np. określających czcionkę, kolor lub wyrównanie. Nie wszystkie opcje mogą być od razu widoczne — zależy to od motywu.








Dokonywane zmiany są od razu stosowane. Te zmiany formatowania są stosowane w odniesieniu do stylu podstawowego. Jeśli zmienimy styl podstawowy, zmiany te pozostaną na stałe.

6. Aby usunąć formatowanie zastosowane przy użyciu tych opcji, zaznaczyć tekst, po czym kliknąć na ikonie .

## Akapity

Aby dodać i sformatować składnik "akapit", należy:

1. Przejść do strony, która ma być edytowana, i upewnić się, że przełącznik  jest ustawiony w pozycji **Edytuj**.
2. Dodać składnik do strony.
3. Kliknąć w składniku "akapit", po czym wpisać jego tekst. Tekst przyjmuje formatowanie zgodne ze stylem domyślnym składnika.
4. Aby dodać wideo () , obraz () lub element zawartości () do akapitu, kliknąć na odpowiedniej ikonie. Wybrać zasób, który ma zostać dodany, po czym odpowiednio dostosować ustawienia. Na przykład wstawić element zawartości, a następnie wybrać różne widoki pozwalające kontrolować złożone układy w akapicie.
5. Aby dodawać dodatkowe akapity w składniku "akapit", wystarczy naciskać klawisz Enter.
6. Aby dodać łącze w obrębie akapitu, należy:
  - a. Wprowadzić tekst, który ma zostać użyty jako tekst łącza, po czym zaznaczyć ten tekst, a następnie kliknąć na ikonie .
  - b. Z listy **Wybór typu łącza** wybrać jedną z następujących opcji:
    - **Strona internetowa:** Należy podać pełny adres URL strony zewnętrznej lub serwisu, a następnie wybrać miejsce otwierania łącza.
    - **Strona serwisu:** Można, używając selektora stron, wybrać stronę w bieżącym serwisie oraz miejsce otwierania łącza. Można określić

dotatkowe parametry adresu URL w następującym formacie:

klucz1=wartość1&klucz2=wartość2. Wartości puste są obsługiwane.

Przykład: klucz1=&klucz2=wartość2. Można także określić kotwicę URL, ale trzeba dodać specjalny układ sekcji "Anchor" strony serwisu docelowego, na której ma być rozstrzygane łącze, oraz określić tę samą nazwę kotwicy w ustawieniach układu, używanych podczas definiowania czynności uaktywniającej.


 **Uwaga:**

Układ sekcji "Anchor", wymagany do korzystania z kotwic adresów URL, jest dystrybuowany z zestawem OCE Toolkit. Informacje, jak ten zestaw uzyskać, są zawarte pod hasłem Tworzenie za pomocą zestawu OCE Toolkit.

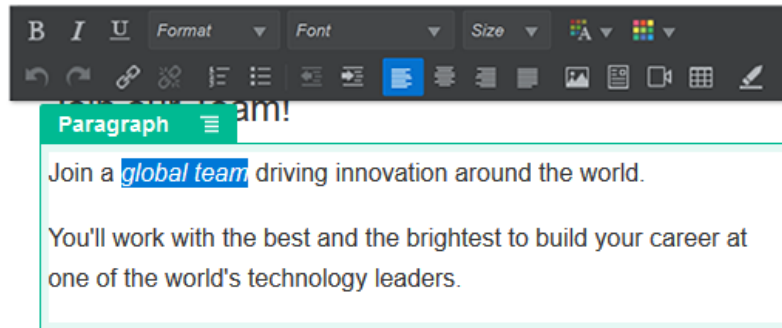
- **Pobieranie pliku:** Pobieranie wybranego pliku z repozytorium. Należy wybrać plik.

Jeśli wybrano zasób cyfrowy, można wybrać określoną realizację. Jeśli realizacja nie zostanie wybrana, będzie stosowany rozmiar oryginalny. Chcąc opublikować najnowszą wersję zasobu (gdy serwis zostanie opublikowany), wybrać opcję **Użyj najnowszej**. Jeśli nie zostanie wybrana opcja **Użyj najnowszej wersji zasobu**, zostanie użyta najnowsza *opublikowana* wersja, zamiast najnowszej wersji roboczej (jeśli taka istnieje).


- **Podgląd pliku:** Podgląd (i opcjonalne pobranie) wybranego pliku na przyciemnionej i nieaktywnej wersji strony.
- **Element zawartości:** Należy wybrać element zawartości z powiązanego repozytorium zasobów, następnie wybrać stronę szczegółów, która ma zostać wyświetlona, po czym przekierować stronę do otwierania jej w tym samym oknie lub nowym.
- **E-mail:** Należy podać poprawny adres e-mail oraz (opcjonalnie) temat. Wynikowa wiadomość zostaje otwarta w domyślnym kliencie poczty elektronicznej i jest przez niego wysyłana.
- **Mapa:** Należy podać poprawny adres e-mail lub poprawne współrzędne, po czym wybrać miejsce otwierania mapy w przeglądarkach stacjonarnych i mobilnych.
- **Połączenie telefoniczne:** Należy podać poprawny numer telefonu.

Aby usunąć łącze, kliknąć w dowolnym miejscu w tekście łącza, a następnie na ikonie .

7. Aby zmienić formatowanie domyślne dowolnego fragmentu, zaznaczyć tekst, który ma zostać sformatowany, po czym użyć dowolnych opcji z paska formatowania, np. określających czcionkę, kolor lub wyrównanie. Nie wszystkie opcje mogą być od razu widoczne — zależy to od motywu.



Dokonywane zmiany są od razu stosowane. Te zmiany formatowania są stosowane w odniesieniu do stylu podstawowego. Jeśli zmienimy styl podstawowy, zmiany te pozostaną na stałe.


8. Aby usunąć formatowanie zastosowane przy użyciu tych opcji, zaznaczyć tekst, po czym kliknąć na ikonie .

## Zwykły tekst

Składnik "tekst" umożliwia dodawanie tekstu do strony i formatowanie go wyłącznie przy użyciu stylów zdefiniowanych w motywie. Pasek narzędzi formatowania nie jest dostępny podczas edytowania zawartości. W związku z tym nie można przesłonić wybranego stylu formatowaniem niestandardowym.

Jest to użyteczne, jeśli dla elementów, takich jak nagłówki, jest stosowane formatowanie standaryzowane. Można zaktualizować style zdefiniowane w motywie i automatycznie zaktualizować w całym serwisie powiązany tekst bez przesłonięć formatu, które by to uniemożliwiły.


Aby dodać składnik "tekst", należy:

1. Przejść do strony, która ma być edytowana, i upewnić się, że przełącznik  jest ustawiony w pozycji **Edytuj**.
2. Dodać składnik do strony.
3. Kliknąć w składniku "akapit", po czym wpisać jego tekst. Tekst przyjmuje formatowanie stylu domyślnego dla składnika (<p> lub element "akapit").

Aby dodawać dodatkowe akapity, wystarczy naciskać klawisz Enter.

### Uwaga:

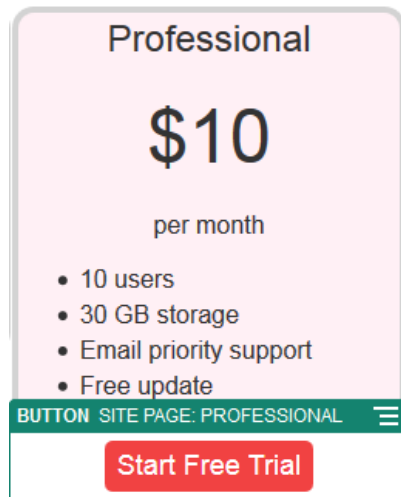
Można kopiować i przeklejać zawartość z innych źródeł do składnika "tekst", ale obrazy i używane znaczniki HTML są wówczas usuwane.

4. Aby zmienić podstawowy styl składnika "tekst":
  - a. Kliknąć na ikonie menu  składnika, po czym wybrać opcję **Ustawienia**.
  - b. Nacisnąć przycisk **Wybór stylu**, a następnie wybrać styl z menu.  
Lista zawiera standardowe znaczniki HTML akapitów i nagłówków. Wybrany styl przypisuje powiązany znacznik do zawartości.



Wybrany styl podstawowy jest odzwierciedlany w składniku.

## Przyciski

Używając przycisku, można uczynić łącze (lub funkcję) bardziej widocznym na stronie.



Aby dodać przycisk do strony, należy:

1. Przejść do strony, która ma być edytowana, i upewnić się, że przełącznik  jest ustawiony w pozycji **Edytuj**.
2. Dodać składnik do strony.
3. Aby dostosować właściwości przycisku, kliknąć na jego ikonie menu , po czym wybrać opcję **Ustawienia**.

Można określić etykietę (tekst), rozmiar, wyrównanie i inne opcje wyświetlania przycisku.

### Uwaga:

Jeśli w polach **Szerokość** i **Wysokość** zostanie ustawiona wartość 0 (zero), rozmiar przycisku będzie automatycznie dopasowywany do tekstu etykiety.

Aby określić kolor tła, czcionkę, krawędzie i inne ustawienia, należy kliknąć na karcie **Styl**. Można wybrać **Dostosuj** w celu określenia własnych wartości.

4. Aby powiązać z przyciskiem łącze lub inne czynności, należy:
  - a. W panelu "Ustawienia" kliknąć na pozycji **Łącze**.
  - b. Z listy **Wybór typu łącza** wybrać jedną z następujących opcji:
    - **Brak łącza**: Gdy użytkownik naciśnie przycisk, nie zostanie wykonana żadna czynność.
    - **Strona internetowa**: Należy podać pełny adres URL strony zewnętrznej lub serwisu, a następnie wybrać miejsce otwierania łącza.



- **Strona serwisu:** Można, używając selektora stron, wybrać stronę w bieżącym serwisie oraz miejsce otwierania łącza. Można określić dodatkowe parametry adresu URL w następującym formacie: `klucz1=wartość1&klucz2=wartość2`. Wartości puste są obsługiwane. Przykład: `klucz1=&klucz2=wartość2`. Można także określić kotwicę URL, ale trzeba dodać specjalny układ sekcji "Anchor" strony serwisu docelowego, na której ma być rozstrzygane łącze, oraz określić tę samą nazwę kotwicy w ustawieniach układu, używanych podczas definiowania czynności uaktywniającej.

 **Uwaga:**

Układ sekcji "Anchor", wymagany do korzystania z kotwic adresów URL, jest dystrybuowany z zestawem OCE Toolkit. Informacje, jak ten zestaw uzyskać, są zawarte pod hasłem Tworzenie za pomocą zestawu OCE Toolkit.

- **Pobieranie pliku:** Pobieranie wybranego pliku z repozytorium. Należy wybrać plik.  
Jeśli wybrano zasób cyfrowy, można wybrać określoną realizację. Jeśli realizacja nie zostanie wybrana, będzie stosowany rozmiar oryginalny. Chcąc opublikować najnowszą wersję zasobu (gdy serwis zostanie opublikowany), wybrać opcję **Użyj najnowszej wersji zasobu**. Jeśli nie zostanie wybrana opcja **Użyj najnowszej wersji zasobu**, zostanie użyta najnowsza *opublikowana* wersja, zamiast najnowszej wersji roboczej (jeśli taka istnieje).
- **Podgląd pliku:** Podgląd (i opcjonalne pobranie) wybranego pliku na przyciemnionej i nieaktywnej wersji strony.
- **Element zawartości:** Należy wybrać element zawartości z powiązanego repozytorium zasobów, następnie wybrać stronę szczegółów, która ma zostać wyświetlona, po czym przekierować stronę do otwierania jej w tym samym oknie lub nowym.
- **E-mail:** Należy podać poprawny adres e-mail oraz (opcjonalnie) temat. Wynikowa wiadomość zostaje otwarta w domyślnym kliencie poczty elektronicznej i jest przez niego wysyłana.
- **Czynności wyzwalacza:** Wybrać czynności (jedną lub więcej) związane ze stroną, które mają zostać wykonane, gdy przycisk zostanie naciśnięty. Czynności związane ze stroną obejmują:
  - **Przejdź do strony serwisu:** Wybrać stronę w bieżącym serwisie.
  - **Przejdź pod zewnętrzny URL:** Podać pełny adres URL strony zewnętrznej lub serwisu.
  - **Wyświetlenie alarmu:** Wyświetlenie określonego komunikatu w oknie.
  - **Pokazanie lub ukrycie składnika:** Wybrać składnik z bieżącej strony, który ma zostać pokazany, ukryty lub przełączony.
  - **Pokazanie lub ukrycie gniazda:** Wybrać gniazdo z bieżącej strony, które ma zostać pokazane, ukryte lub przełączone.

- **Mapa:** Należy podać poprawny adres e-mail lub poprawne współrzędne, po czym wybrać miejsce otwierania mapy w przeglądarkach stacjonarnych i mobilnych.
- **Połączenie telefoniczne:** Należy podać poprawny numer telefonu.

Zob. [Używanie wyzwalaczy i czynności](#).

## Składniki struktury


Składniki struktury pomagają rozdzielać zawartość strony.

- [Separatory](#)
- [Rozdzielacze](#)


## Separatory

Separator (linia pozioma) służy do wprowadzenia wizualnego podziału w kolumnie lub na stronie.

Aby dodać separator do strony, należy:

1. Przejść do strony, która ma być edytowana, i upewnić się, że przełącznik  jest ustawiony w pozycji **Edytuj**.
2. Dodać składnik do strony.

Before	After
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi in leo turpis. Ut ex neque, cursus vulputate facilisis sed, tempor quis ligula. Pellentesque sodales sagittis fringilla. Praesent id enim ut orci pretium faucibus a et massa.</p> <p>Get in touch</p>	<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi in leo turpis. Ut ex neque, cursus vulputate facilisis sed, tempor quis ligula. Pellentesque sodales sagittis fringilla. Praesent id enim ut orci pretium faucibus a et massa.</p> <hr/> <p>Get in touch</p>

3. Aby dostosować właściwości separatora, kliknąć na jego ikonie menu , po czym wybrać opcję **Ustawienia**.


Dostosowując odstępy z lewej i prawej strony, można określić położenie poziome i długość separatora.

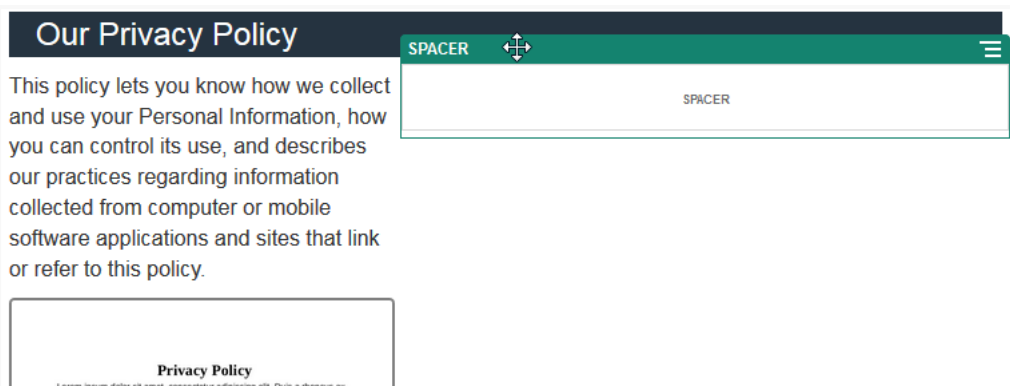
Kolor i grubość linii oraz inne ustawienia są zdefiniowane przez styl składnika. Można wybrać odpowiedni predefiniowany styl z bieżącego motywu albo wybrać opcję **Dostosuj** i podać własne wartości.


## Rozdzielacze

Wprowadzając na stronie puste miejsca, można czynić ją bardziej estetyczną i bardziej czytelną. Składnik "rozdzielacz" służy do dodania pustych miejsc bez konieczności przesłaniania odstępów zdefiniowanych w stylach lub w innych składnikach.

Aby dodać puste pionowe miejsca między składnikami na stronie, należy:

1. Przejdź do strony, która ma być edytowana, i upewnij się, że przełącznik  jest ustawiony w pozycji **Edytuj**.
2. Dodać składnik do strony.



3. Aby dostosować wysokość rozdzielacza, kliknąć na jego ikonie menu , po czym wybrać opcję **Ustawienia**.

## Składniki multimedialne


Składniki multimedialne umożliwiają wstawianie obrazów i filmów wideo na stronie.

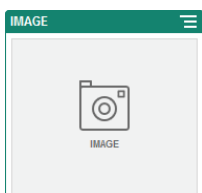
- [Obrazy](#)
- [Galerie](#)
- [Siatka galerii](#)
- [Wideo z serwisu YouTube](#)
- [Wideo](#)


## Obrazy

Obrazy mogą przyciągać uwagę użytkowników i zapraszać do eksplorowania innych obszarów zawartości.

Aby dodać obraz do strony, należy:

1. Przejdź do strony, która ma być edytowana, i upewnij się, że przełącznik  jest ustawiony w pozycji **Edytuj**.
2. Dodać składnik do strony. Składnik "obraz" pokazuje obraz zastępczy, dopóki nie zostanie wybrany obraz, który będzie używany.

















3. Aby wybrać obraz, kliknąć na jego ikonie , wybrać pozycję **Ustawienia**, po czym kliknąć na opcji **Wybór** obok pola "Obraz".
4. Wybrać obraz z repozytorium serwisów lub z folderu "Dokumenty", który został udostępniony bieżącemu użytkownikowi, bądź wysłać obraz do folderu "Dokumenty".

 **Uwaga:**

W oknie są wyświetlane wszystkie dostępne pliki. Trzeba wybrać typ pliku właściwy w danym kontekście. Na przykład, jeśli jest wybierany plik obrazu, trzeba wybrać plik w formacie graficznym (GIF, JPG, JPEG, PNG lub SVG).

- a. Odszukać i wybrać obraz, który ma zostać użyty.  

Jeśli zasoby cyfrowe nie są widoczne, kliknąć na ikonie , po czym zmienić filtrowanie kolekcji na **Wszystkie**.
  - b. Jeśli wybrano obraz z folderu "Dokumenty", plik obrazu można powiązać (tworząc łącze), zamiast kopiować go do serwisu. Aby utworzyć łącze do pliku, wybrać opcję **Zamiast kopio**. Jeśli ta opcja nie zostanie wybrana, w serwisie będzie składowana kopia pliku, do której będą się odnosiły odwołania z serwisu. Tworząc łącze do oryginalnego pliku, unika się duplikowania zawartości. Łącze pozwala odwiedzającym serwis oglądać zawartość, nawet jeśli ulegną zmianie uprawnienia do pliku lub jego wyświetlanie zostanie ograniczone w inny sposób.
  - c. Nacisnąć przycisk **OK**.
  - d. Jeśli wybrano zasób cyfrowy, można wybrać określoną realizację. Jeśli realizacja nie zostanie wybrana, będzie stosowany rozmiar oryginalny. Chcąc opublikować najnowszą wersję zasobu (gdy serwis zostanie opublikowany), wybrać opcję **Użyj najnowszej wersji zasobu**.
5. Aby edytować obraz, kliknąć na ikonie , po czym edytować obraz w dowolny z następujących sposobów:
    - Aby przyciąć obraz, kliknąć na ikonie **Przytnij** (  ). Z paska narzędzi przycinania wybrać jedną z predefiniowanych proporcji obrazu albo przeciągnąć na obraz uchwyty do przycinania. Gdy wynik będzie satysfakcjonujący, nacisnąć na pasku narzędzi przycinania przycisk **Przytnij**.
    - Aby obrócić lub przerzucić obraz, kliknąć na ikonie **Obróć** (  ). Na pasku narzędzi obracania podać niestandardową wartość stopni obrotu bądź użyć przycisków do obracania obrazu w lewo lub w prawo albo wybrać, czy obraz ma zostać przerzucony w poziomie czy w pionie.
    - Aby dodać znak wodny do obrazu, kliknąć na ikonie **Znak wodny** (  ). Dodać tekst do obrazu; korzystając z narzędzi znaku wodnego, można odpowiednio zmienić rozmiar tekstu, styl, kolor i przezroczystość.
    - Aby zmienić format obrazu, kliknąć na ikonie **Opcje** (  ), po czym wybrać z rozwijanej listy **Format** nowy format.

- Aby zmienić kolor tła, kliknąć na ikonie **Opcje** () , po czym wybrać z rozwijanego menu **Kolor tła** odpowiednią opcję.
  - Jeśli jest edytowany obraz w formacie .jpg lub .webp (dostępne w przeglądarce Google Chrome), można zmienić jakość obrazu w celu utworzenia pliku o mniejszym rozmiarze. Kliknąć na ikonie **Opcje** () , po czym w obszarze **Jakość** podać nową wartość procentową.
  - Aby cofnąć lub ponowić zmianę, kliknąć na ikonie  lub  . Aby usunąć wszelkie dokonane zmiany, nacisnąć przycisk **Resetuj**.
  - Aby zmienić powiększenie obrazu, użyć elementów sterujących powiększaniem (   ).
6. Aby dodać nazwę opisową lub zmodyfikować odstępy, wyrównanie, styl i inne opcje, kliknąć na ikonie  , znajdującej się w panelu **Ogólne** menu **Ustawienia**.
7. Z obrazem można powiązać łącze lub inne czynności:
- a. W panelu "Ustawienia" kliknąć na pozycji **Łącze**.
  - b. Wybrać jedną z następujących opcji:
    - **Brak łącza**: Gdy użytkownik kliknie na obrazie, nie zostanie wykonana żadna czynność.
    - **Strona internetowa**: Należy podać pełny adres URL strony zewnętrznej lub serwisu, a następnie wybrać miejsce otwierania łącza.
    - **Strona serwisu**: Można, używając selektora stron, wybrać stronę w bieżącym serwisie oraz miejsce otwierania łącza. Można określić dodatkowe parametry adresu URL w następującym formacie: `klucz1=wartość1&klucz2=wartość2`. Wartości puste są obsługiwane. Przykład: `klucz1=&klucz2=wartość2`. Można także określić kotwicę URL, ale trzeba dodać specjalny układ sekcji "Anchor" strony serwisu docelowego, na której ma być rozstrzygane łącze, oraz określić tę samą nazwę kotwicy w ustawieniach układu, używanych podczas definiowania czynności uaktywniającej.

 **Uwaga:**

Układ sekcji "Anchor", wymagany do korzystania z kotwic adresów URL, jest dystrybuowany z zestawem OCE Toolkit. Informacje, jak ten zestaw uzyskać, są zawarte pod hasłem Tworzenie za pomocą zestawu OCE Toolkit.



- **Pobieranie pliku**: Pobieranie wybranego pliku z repozytorium. Należy wybrać plik.  
Jeśli wybrano zasób cyfrowy, można wybrać określoną realizację. Jeśli realizacja nie zostanie wybrana, będzie stosowany rozmiar oryginalny. Chcąc opublikować najnowszą wersję zasobu (gdy serwis zostanie opublikowany), wybrać opcję **Użyj najnowszej wersji zasobu**. Jeśli nie zostanie wybrana opcja **Użyj najnowszej wersji zasobu**, zostanie użyta najnowsza *opublikowana* wersja, zamiast najnowszej wersji roboczej (jeśli taka istnieje).

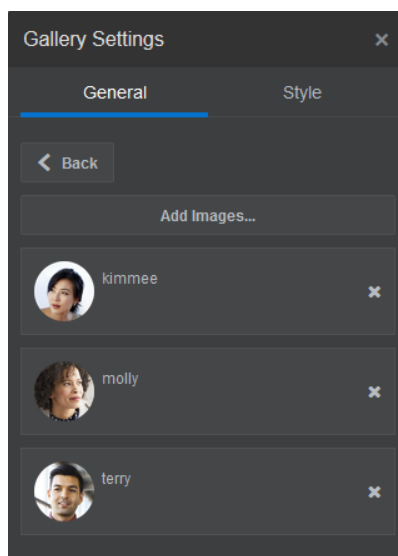
- **Podgląd pliku:** Wybrany plik zostanie wyświetlony jako nakładka na stronę.
- **Element zawartości:** Należy wybrać element zawartości z powiązanego repozytorium zasobów, następnie wybrać stronę szczegółów, która ma zostać wyświetlona, po czym przekierować stronę do otwierania jej w tym samym oknie lub nowym.
- **E-mail:** Należy podać poprawny adres e-mail oraz (opcjonalnie) temat. Wynikowa wiadomość zostaje otwarta w domyślnym kliencie poczty elektronicznej i jest przez niego wysyłana.
- **Podgląd obrazu:** Wybrany obraz zostanie wyświetlony jako nakładka na stronę.
- **Mapa:** Należy podać poprawny adres e-mail lub poprawne współrzędne, po czym wybrać miejsce otwierania mapy w przeglądarkach stacjonarnych i mobilnych.
- **Połączenie telefoniczne:** Należy podać poprawny numer telefonu.

## Galerie

Za pomocą galerii obrazów można przedstawić szereg obrazów. Można wybrać, że obrazy mają być automatycznie przewijane, a można także zezwolić użytkownikom na samodzielne przechodzenie między obrazami.

Aby dodać galerię obrazów do strony, należy:

1. Przejść do strony, która ma być edytowana, i upewnić się, że przełącznik  jest ustawiony w pozycji **Edytuj**.
2. Dodać składnik do strony. Składnik "galeria" pokazuje obraz zastępczy, dopóki nie zostaną wybrane obrazy, które będą używane.
3. Aby dodać jeden lub większą liczbę obrazów do galerii, kliknąć na jej ikonie , wybrać pozycję **Ustawienia**, po czym kliknąć na opcji **Obrazy** na karcie **Ogólne**.
4. Nacisnąć przycisk **Dodaj obrazy**.




5. Wybrać jeden lub większą liczbę obrazów.

 **Uwaga:**

W oknie są wyświetlane wszystkie dostępne pliki. Trzeba wybrać typ pliku właściwy w danym kontekście. Na przykład, jeśli jest wybierany plik obrazu, trzeba wybrać plik w formacie graficznym (GIF, JPG, JPEG, PNG lub SVG).

- a. Odszukać i wybrać obrazy, które mają zostać użyte.

Jeśli zasoby cyfrowe nie są widoczne, kliknąć na ikonie , po czym zmienić filtrowanie kolekcji na **Wszystkie**.

- b. Jeśli wybrano obraz z folderu "Dokumenty", plik obrazu można powiązać (tworząc łącze), zamiast kopiować go do serwisu. Aby utworzyć łącze do pliku, wybrać opcję **Zamiast kopiować plik do serwisu, użyj odwołania do pliku oryginalnego**. Jeśli ta opcja nie zostanie wybrana, w serwisie będzie składowana kopia pliku, do której będą się odnosiły odwołania z serwisu. Tworząc łącze do oryginalnego pliku, unika się duplikowania zawartości. Łącze pozwala odwiedzającym serwis oglądać zawartość, nawet jeśli ulegną zmianie uprawnienia do pliku lub jego wyświetlanie zostanie ograniczone w inny sposób.
- c. Nacisnąć przycisk **OK**.
- d. Jeśli wybrano zasób cyfrowy, można wybrać określoną realizację. Jeśli realizacja nie zostanie wybrana, będzie stosowany rozmiar oryginalny. Chcąc opublikować najnowszą wersję zasobu (gdy serwis zostanie opublikowany), wybrać opcję **Użyj najnowszej wersji zasobu**.

Wybrane obrazy zostają dodane do listy obrazów. Można zmienić ich rozmieszczenie na liście, przeciągając je. Tytułem domyślnym każdego obrazu jest nazwa pliku bez rozszerzenia.

6. Aby zmienić tytuł, opis lub inne opcje określonego obrazu, kliknąć na nim na liście, po czym dokonać zmian.

Z obrazem w galerii można także powiązać łącze lub inne czynności:


- a. W panelu "Ustawienia" dla danego obrazu kliknąć w polu "Łącze".
- b. Wybrać jedną z następujących opcji:
- **Brak łącza:** Gdy użytkownik kliknie na obrazie, nie zostanie wykonana żadna czynność.
  - **Strona internetowa:** Należy podać pełny adres URL strony zewnętrznej lub serwisu, a następnie wybrać miejsce otwierania łącza.
  - **Strona serwisu:** Można, używając selektora stron, wybrać stronę w bieżącym serwisie oraz miejsce otwierania łącza. Można określić dodatkowe parametry adresu URL w następującym formacie: `klucz1=wartość1&klucz2=wartość2`. Wartości puste są obsługiwane. Przykład: `klucz1=&klucz2=wartość2`. Można także określić kotwicę URL, ale trzeba dodać specjalny układ sekcji "Anchor" strony serwisu docelowego, na której ma być rozstrzygane łącze, oraz określić tę samą nazwę kotwicy w ustawieniach układu, używanych podczas definiowania czynności uaktywniającej.

 **Uwaga:**

Układ sekcji "Anchor", wymagany do korzystania z kotwic adresów URL, jest dystrybuowany z zestawem OCE Toolkit. Informacje, jak ten zestaw uzyskać, są zawarte pod hasłem Tworzenie za pomocą zestawu OCE Toolkit.

- **Pobieranie pliku:** Pobieranie wybranego pliku z repozytorium. Należy wybrać plik.  
Jeśli wybrano zasób cyfrowy, można wybrać określoną realizację. Jeśli realizacja nie zostanie wybrana, będzie stosowany rozmiar oryginalny. Chcąc opublikować najnowszą wersję zasobu (gdy serwis zostanie opublikowany), wybrać opcję **Użyj najnowszej**. Jeśli nie zostanie wybrana opcja **Użyj najnowszej wersji zasobu**, zostanie użyta najnowsza *opublikowana* wersja, zamiast najnowszej wersji roboczej (jeśli taka istnieje).
  - **Element zawartości:** Należy wybrać element zawartości z powiązanego repozytorium zasobów, następnie wybrać stronę szczegółów, która ma zostać wyświetlona, po czym przekierować stronę do otwierania jej w tym samym oknie lub nowym.
  - **E-mail:** Należy podać poprawny adres e-mail oraz (opcjonalnie) temat. Wynikowa wiadomość zostaje otwarta w domyślnym kliencie poczty elektronicznej i jest przez niego wysyłana.
  - **Mapa:** Należy podać poprawny adres e-mail lub poprawne współrzędne, po czym wybrać miejsce otwierania mapy w przeglądarkach stacjonarnych i mobilnych.
  - **Połączenie telefoniczne:** Należy podać poprawny numer telefonu.
- c. Nacisnąć przycisk **Wstecz**, aby wrócić do panelu ustawień obrazu. Ponownie nacisnąć przycisk **Wstecz**, aby wrócić do listy obrazów, z której można wybrać inny obraz do aktualizacji.
- d. Po ukończeniu aktualizacji obrazów nacisnąć przycisk **Wstecz**, aby określić opcje galerii.
7. Aby dopasować rozmiar i skalę obrazów w galerii:
- a. Korzystając z opcji **Szerokość**, można określić (w pikselach) szerokość galerii w obrębie gniazda. Aby określić szerokość, należy wybrać opcję wyrównania inną niż **Wypełnij**. Po ustawieniu szerokości można za pomocą opcji **Wypełnij** powiększyć obraz do podanej szerokości.
  - b. Korzystając z opcji **Skalowanie**, dostosować sposób przedstawienia obrazów w galerii:
    - **Przytnij:** Mniejszy z dwóch wymiarów (szerokość lub wysokość) jest skalowany tak, aby pasował do dostępnego miejsca, a większy wymiar jest przycinany tak, aby zapobiec rozciągnięciu obrazu.
    - **Dopasuj:** Każdy obraz jest skalowany tak, aby bez odkształceń zmieścił się w dostępnym miejscu.
    - **Rozciągnij:** Większy z dwóch wymiarów (szerokość lub wysokość) jest skalowany tak, aby pasował do dostępnego miejsca, a mniejszy wymiar jest rozciągany tak, aby dostępne miejsce zostało wypełnione.





- **Brak:** Obraz jest używany w jego pełnej rozdzielczości pikselowej; jeśli się nie mieści w dostępnym miejscu jest równomiernie przycinany.
8. Aby pomóc użytkownikowi poruszać się po galerii:
    - a. Wybrać metodę **nawigacji**:
      - **Miniatury:** Pod galerią będzie pokazywana lista obrazów galerii, zgodnie z ich kolejnością. Użytkownik, gdy kliknie na miniaturze, przejdzie do powiązanego obrazu w galerii.
      - **Indekser:** Pod galerią będzie pokazywany szereg przycisków  reprezentujących poszczególne obrazy galerii. Użytkownik, gdy naciśnie przycisk, przejdzie do powiązanego obrazu w galerii.
      - **Brak:** Nawigacja wizualna nie będzie udostępniana. Użytkownik, aby wyświetlić sąsiadujący obraz, może przeciągać go w prawo lub w lewo. Ta metoda ręczna jest zawsze dostępna.
    - b. Wybrać opcję **Pokaż "Dalej/Wstecz"**, aby na każdym obrazie były wyświetlane ikony ze strzałkami, służące do przechodzenia do poprzedniego lub następnego obrazu w galerii.
  9. Można pokazać lub ukryć nazwę opisową obrazu, zaznaczając lub czyszcząc pole wyboru **Pokaż nazwę opisową**.
  10. Aby obrazy były automatycznie wyświetlane, wybrać opcję **Automatyczne odtwarzanie** oraz określić **Czas przejścia** i **Czas wyświetlania**.

Użytkownicy nadal będą mogli korzystać ze wszystkich skonfigurowanych funkcji nawigacji ręcznej.

## Siatka galerii

Korzystając z siatki galerii, można prezentować zestaw obrazów rozmieszczonych w rzędach i kolumnach.


Aby dodać galerię obrazów do strony, należy:

1. Przejść do strony, która ma być edytowana, i upewnić się, że przełącznik  jest ustawiony w pozycji **Edytuj**.
2. Dodać składnik do strony. Składnik "siatka galerii" pokazuje obraz zastępczy, dopóki nie zostaną wybrane obrazy, które będą używane.
3. Aby dodać jeden lub większą liczbę obrazów do galerii, kliknąć na jej ikonie , wybrać pozycję **Ustawienia**, po czym kliknąć na opcji **Obrazy** na karcie **Ogólne**.
4. Nacisnąć przycisk **Dodaj obrazy**.
5. Wybrać jeden lub większą liczbę obrazów.

### Uwaga:

W oknie są wyświetlane wszystkie dostępne pliki. Trzeba wybrać typ pliku właściwy w danym kontekście. Na przykład, jeśli jest wybierany plik obrazu, trzeba wybrać plik w formacie graficznym (GIF, JPG, JPEG, PNG lub SVG).

- a. Odszukać i wybrać obrazy, które mają zostać użyte.

Jeśli zasoby cyfrowe nie są widoczne, kliknąć na ikonie , po czym zmienić filtrowanie kolekcji na **Wszystkie**.

- b. Jeśli wybrano obraz z folderu "Dokumenty", plik obrazu można powiązać (tworząc łącze), zamiast kopiować go do serwisu. Aby utworzyć łącze do pliku, wybrać opcję **Zamiast kopio**. Jeśli ta opcja nie zostanie wybrana, w serwisie będzie składowana kopia pliku, do której będą się odnosiły odwołania z serwisu. Tworząc łącze do oryginalnego pliku, unika się duplikowania zawartości. Łącze pozwala odwiedzającym serwis oglądać zawartość, nawet jeśli ulegną zmianie uprawnienia do pliku lub jego wyświetlanie zostanie ograniczone w inny sposób.
- c. Nacisnąć przycisk **OK**.
- d. Jeśli wybrano zasób cyfrowy, można wybrać określoną realizację. Jeśli realizacja nie zostanie wybrana, będzie stosowany rozmiar oryginalny. Chcąc opublikować najnowszą wersję zasobu (gdy serwis zostanie opublikowany), wybrać opcję **Użyj najnowszej wersji zasobu**.

Wybrane obrazy zostają dodane do listy obrazów. Można zmienić ich rozmieszczenie na liście (i na siatce), przeciągając je. Tytułem domyślnym każdego obrazu jest nazwa pliku bez rozszerzenia.

6. Aby zmienić tytuł, opis lub inne opcje określonego obrazu, kliknąć na nim na liście, po czym dokonać zmian.

Z obrazem w galerii można także powiązać łącze lub inne czynności:

- a. W panelu "Ustawienia" dla danego obrazu kliknąć w polu "Łącze".

- b. Wybrać jedną z następujących opcji:

- **Brak łącza:** Gdy użytkownik kliknie na obrazie, nie zostanie wykonana żadna czynność.
- **Strona internetowa:** Należy podać pełny adres URL strony zewnętrznej lub serwisu, a następnie wybrać miejsce otwierania łącza.
- **Strona serwisu:** Można, używając selektora stron, wybrać stronę w bieżącym serwisie oraz miejsce otwierania łącza. Można określić dodatkowe parametry adresu URL w następującym formacie: `klucz1=wartość1&klucz2=wartość2`. Wartości puste są obsługiwane. Przykład: `klucz1=&klucz2=wartość2`. Można także określić kotwicę URL, ale trzeba dodać specjalny układ sekcji "Anchor" strony serwisu docelowego, na której ma być rozstrzygane łącze, oraz określić tę samą nazwę kotwicy w ustawieniach układu, używanych podczas definiowania czynności uaktywniającej.

 **Uwaga:**

Układ sekcji "Anchor", wymagany do korzystania z kotwic adresów URL, jest dystrybuowany z zestawem OCE Toolkit. Informacje, jak ten zestaw uzyskać, są zawarte pod hasłem Tworzenie za pomocą zestawu OCE Toolkit.

- **Pobieranie pliku:** Pobieranie wybranego pliku z repozytorium. Należy wybrać plik.

Jeśli wybrano zasób cyfrowy, można wybrać określoną realizację. Jeśli realizacja nie zostanie wybrana, będzie stosowany rozmiar oryginalny. Chcąc opublikować najnowszą wersję zasobu (gdy serwis zostanie opublikowany), wybrać opcję **Użyj najnowszej wersji zasobu**. Jeśli nie zostanie wybrana opcja **Użyj najnowszej wersji zasobu**, zostanie użyta najnowsza *opublikowana* wersja, zamiast najnowszej wersji roboczej (jeśli taka istnieje).

- **Element zawartości:** Należy wybrać element zawartości z powiązanego repozytorium zasobów, następnie wybrać stronę szczegółów, która ma zostać wyświetlona, po czym przekierować stronę do otwierania jej w tym samym oknie lub nowym.
  - **E-mail:** Należy podać poprawny adres e-mail oraz (opcjonalnie) temat. Wynikowa wiadomość zostaje otwarta w domyślnym kliencie poczty elektronicznej i jest przez niego wysyłana.
  - **Podgląd obrazu:** Wybrany obraz zostanie wyświetlony jako nakładka na stronę.
  - **Mapa:** Należy podać poprawny adres e-mail lub poprawne współrzędne, po czym wybrać miejsce otwierania mapy w przeglądarkach stacjonarnych i mobilnych.
  - **Połączenie telefoniczne:** Należy podać poprawny numer telefonu.
- c. Nacisnąć przycisk **Wstecz**, aby wrócić do panelu ustawień obrazu. Ponownie nacisnąć przycisk **Wstecz**, aby wrócić do listy obrazów, z której można wybrać inny obraz do aktualizacji.
- d. Po ukończeniu aktualizacji obrazów nacisnąć przycisk **Wstecz**, aby określić opcje galerii.
7. Za pomocą opcji **Układ** rozmieścić obrazy w siatce.
- Ścianka
  - Kolumny
  - Niestandardowe

Każda z tych opcji jest opisana w kolejnych punktach tej procedury.

8. Wybrać układ **Ścianka**, aby automatycznie rozmieścić obrazy w rzędach, w obrębie dostępnego miejsca.

Utworzone rzędy mają jednakową wysokość, lecz nie mają zdefiniowanych kolumn.



- a. Określić **Wysokość**, aby proporcjonalnie przeskalować wszystkie obrazy do podanej wysokości (w pikselach).
  - b. Określić **Odstęp między obrazami**, aby zwiększyć lub zmniejszyć odstęp między obrazami w rzędzie.
9. Wybrać układ **Kolumny**, aby rozmieścić obrazy w rzędach i kolumnach.
- a. Korzystając z opcji **Skalowanie**, dostosować sposób przedstawienia obrazów w siatce:
    - **Przytnij:** Mniejszy z dwóch wymiarów (szerokość lub wysokość) jest skalowany tak, aby pasował do dostępnego miejsca, a większy wymiar jest przycinany tak, aby zapobiec rozciągnięciu obrazu.
    - **Dopasuj:** Każdy obraz jest skalowany tak, aby bez odkształceń zmieścił się w dostępnym miejscu.

Na przykład w poniższej siatce są używane cztery kolumny, a sześć obrazów jest skalowanych tak, aby się w całości mieściły:



Poniżej jest przedstawiona ta sama siatka z przyciętymi obrazami:



- b. Określić **Proporcje**, wyznaczając kształt komórek siatki.
  - **Kwadrat:** Proporcje 1:1.
  - **Panorama:** Proporcje 16:9.
  - **Portret:** Proporcje 9:16.
  - **Niestandardowe:** Można podać własne wartości liczbowe wyznaczające proporcje.



- c. Określić liczbę kolumn (pole **Kolumny**).  
Siatka zostanie automatycznie dostosowana tak, aby zawierała kolumny jednakowej szerokości.
  - d. Określić **Odstęp między obrazami**, aby zwiększyć lub zmniejszyć odstęp między obrazami w rzędach i kolumnach.
10. Wybrać układ **Niestandardowe**, aby розміścić obrazy w rzędach i kolumnach, na podstawie podanej szerokości i wysokości obrazów.
- a. Korzystając z opcji **Skalowanie**, dostosować sposób przedstawienia obrazów w siatce:
    - **Przytnij**: Mniejszy z dwóch wymiarów (szerokość lub wysokość) jest skalowany tak, aby pasował do dostępnego miejsca, a większy wymiar jest przycinany tak, aby zapobiec rozciągnięciu obrazu.
    - **Dopasuj**: Każdy obraz jest skalowany tak, aby bez odkształceń zmieścił się w dostępnym miejscu.
  - b. Określić **Wysokość obrazu** i **Szerokość obrazu**, wyznaczając kształt komórek siatki.  
Siatka zostanie automatycznie dostosowana tak, aby zawierała komórki o podanych wymiarach.
  - c. Określić **Odstęp między obrazami**, aby zwiększyć lub zmniejszyć odstęp między obrazami w rzędach i kolumnach.
11. Określić opcje **Wyrównanie**, **Szerokość** i **Odstęp** decydujące o położeniu siatki w gnieździe.  
Korzystając z opcji **Szerokość**, można określić (w pikselach) szerokość galerii w obrębie gniazda. Aby określić szerokość, należy wybrać opcję wyrównania inną niż **Wypełnij**.  
. Po ustawieniu szerokości można za pomocą opcji **Wypełnij** powiększyć obraz do podanej szerokości.

## Wideo z serwisu YouTube

Stronę można wzbogacać o filmy wideo z serwisu YouTube.

Informacje, jak używać filmów wideo innych niż z serwisu YouTube, są dostępne pod hasłem [Wideo](#).

Aby dodać do strony film wideo z serwisu YouTube, należy:

1. Przejść do strony, która ma być edytowana, i upewnić się, że przełącznik  jest ustawiony w pozycji **Edytuj**.
2. Dodać składnik do strony. Dopóki nie zostanie wybrany film wideo, składnik prezentuje obraz zastępczy.
3. Aby określić film wideo z serwisu YouTube i dostosować właściwości wyświetlania, kliknąć na ikonie menu  składnika, po czym wybrać opcję **Ustawienia**.
4. Odszukać film wideo, a następnie skopiować i wkleić jego adres URL do pola **URL w serwisie YouTube**.
5. Określić opcje wyświetlania:

- **Wyświetlanie formantów:** Włączenie opcji ekranowych i sprzętowych umożliwiających użytkownikowi samodzielne sterowanie odtwarzaniem wideo.
  - **Pokaż informacje:** Chwilowe wyświetlanie opisu wideo (w jego górnym rogu).
  - **Automatyczne odtwarzanie:** Automatyczne rozpoczęcie odtwarzania wideo.
  - **Pętla:** Automatyczne powtarzanie wideo w pętli.
6. Użyć ustawień **Proporcje** zdefiniowanych dla wideo (opcja **Automatycznie**) albo wybrać inne proporcje ekranu wideo.
  7. Określić opcje wyrównania lub odstępów w celu odpowiedniego umiejscowienia wideo.


## Wideo

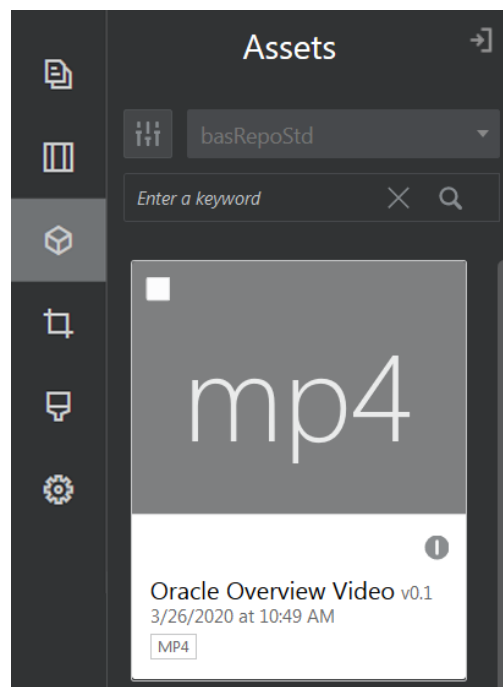
W celu uatrakcyjnienia strony można dodawać filmy wideo z Oracle Content Management.


Oracle Content Management oferuje różne opcje dodawania filmów wideo do strony. Aby uzyskać funkcje automatycznej konwersji kodowania i zoptymalizowanego strumieniowania, oferowane przez usługę "Wideo plus", można osadzić [Wideo z serwisu YouTube](#), które jest tam hostowane, lub dodać wideo przechowywane, zarządzane i dostarczane przez Oracle Content Management. Jest to przydatne, gdy serwisy są obsługiwane przez różne urządzenia wyposażone w różne funkcje. Jeśli administrator systemu nie włączył usługi "Wideo plus", pozostaje opcja wideo standardowego.

Informacje, jak używać filmów wideo z serwisu YouTube, są dostępne pod hasłem [Wideo z serwisu YouTube](#).

Aby można było używać filmy wideo przechowywanego w repozytorium dokumentów lub zasobów Oracle Content Management, należy:

1. Przejść do strony, która ma być edytowana, i upewnić się, że przełącznik edycji () jest ustawiony w pozycji **Edytuj**.
  - a. Chcąc użyć wideo przechowywanego jako zasób cyfrowy, otworzyć panel **Zasoby**, wyszukać odpowiednie wideo, po czym wybrać je i przeciągnąć na stronę.



- b. Alternatywnie, z listy składników dodać składnik "wideo" do strony. Dopóki nie zostanie wybrany film wideo, składnik prezentuje obraz zastępczy.
2. Jeśli zasób został przeciągnięty z panelu "Zasoby" lub składnik "wideo" z listy składników, kliknąć na ikonie menu () , aby dostosować właściwości wyświetlania wideo bądź przejść do wideo znajdującego się w składniku "wideo". Kliknąć na menu, po czym wybrać opcję **Ustawienia**.
3. Aby wybrać wideo dla składnika "wideo", nacisnąć przycisk **Wybór** w celu otwarcia kart **Dokumenty** i **Zasoby cyfrowe**, po czym przejść do wideo, które ma zostać dodane z usługi Oracle Content Management. Wybrać wideo, po czym nacisnąć przycisk **OK**.
  - a. Jeśli jest włączona usługa "Wideo plus", można wybrać różne repozytoria, kanały wyszukiwania, kolekcje i słowa kluczowe; podczas przechodzenia do wideo można także filtrować zasoby w celu zawężenia wyszukiwania.
  - b. Jeśli jest używane wideo standardowe, wybrać plik wideo typu MP4. Format wideo MP4 jest typowy dla wszystkich obsługiwanych przeglądark. Także niektóre pliki wideo standardowego mają predefiniowane obrazy podglądu. Dla tych, które nie mają, jako obraz podglądu jest wyświetlany pusty ekran. Aby zamiast pustego ekranu użyć grafiki, w oknie dialogowym "Ustawienia wideo" nacisnąć przycisk **Wybór**, po czym z usługi Oracle Content Management wybrać odpowiedni obraz podglądu.
4. Po wybraniu wideo (przez przeciągnięcie z panelu zasobów albo wybranie z okna dialogowego "Ustawienia") określić za pomocą okna dialogowego "Ustawienia" opcje wyświetlania:
  - **Wyświetlanie formantów**: Włączenie opcji ekranowych i sprzętowych umożliwiających użytkownikowi samodzielne sterowanie odtwarzaniem wideo.
  - **Automatyczne odtwarzanie**: Automatyczne rozpoczynanie odtwarzania wideo.
  - **Pętla**: Automatyczne powtarzanie wideo w pętli.

- **Wyciszone:** Automatycznie wycisza wideo podczas ładowania strony.
5. Określić opcje szerokości, wyrównania lub odstępów w celu odpowiedniego umiejscowienia wideo.
  6. Aby wybrać lub zdefiniować niestandardowy styl obramowania filmu wideo, nacisnąć przycisk **Styl**.

## Składniki "dokumenty"

Przedstawiono tu niektóre składniki, pozwalające uzyskiwać dostęp do plików i folderów oraz je wyświetlać.


- [Dokumenty](#)
- [Listy folderów](#)
- [Listy plików](#)
- [Menedżer dokumentów](#)
- [Biblioteka projektu](#)

## Dokumenty


Dokumenty wielostronicowe i prezentacje slajdów można oglądać bezpośrednio ze strony.



Aby dodać dokument do strony, należy:

1. Przejść do strony, która ma być edytowana, i upewnić się, że przełącznik  jest ustawiony w pozycji **Edytuj**.
2. Dodać składnik do strony. Składnik "dokument" pokazuje obraz zastępczy, dopóki nie zostanie określony dokument, które będzie używany.




3. Aby wybrać dokument, kliknąć na jego ikonie , wybrać pozycję **Ustawienia**, po czym kliknąć na opcji **Wybór** obok pola "Dokument".

Dokument musi być przechowywany w repozytorium serwisu albo w innym repozytorium, do którego mamy dostęp. Można także używać dokumentów, które zostały nam udostępnione lub które wysłaliśmy z lokalizacji lokalnej lub sieciowej.

4. Wybrać dokument, po czym nacisnąć przycisk **OK**.

 **Uwaga:**

W oknie są wyświetlane wszystkie dostępne pliki. Trzeba wybrać typ pliku właściwy w danym kontekście. Na przykład, jeśli jest wybierany plik dokumentu, trzeba wybrać plik w formacie dokumentu, na przykład TXT lub DOC.


5. Za pomocą panelu "Ustawienia" dodać nazwę opisową bądź zmodyfikować odstępy, wyrównanie i inne opcje prezentacji.
6. Aby pomóc użytkownikowi poruszać się po dokumencie:
  - a. Wybrać opcję **Pokaż numery stron**, a pod każdą stroną będzie wyświetlany jej numer.
  - b. Wybrać metodę **nawigacji**:
    - **Miniatury**: Pod dokumentem będzie pokazywana lista stron dokumentu, zgodnie z ich kolejnością. Użytkownik, gdy kliknie na miniaturze, przejdzie do powiązanej z nią strony.
    - **Indekser**: Pod dokumentem będzie pokazywany szereg przycisków  reprezentujących poszczególne strony dokumentu. Użytkownik, gdy kliknie na przycisku, przejdzie do powiązanej z nim strony.
    - **Brak**: Nawigacja wizualna nie będzie udostępniana. Użytkownik, aby wyświetlić sąsiadującą stronę, może przeciągać ją w prawo lub w lewo. Ta metoda ręczna jest zawsze dostępna.
  - c. Wybrać opcję **Pokaż "Dalej/Wstecz"**, aby na każdej stronie były wyświetlane ikony ze strzałkami, służące do przechodzenia do poprzedniej lub następnej strony w dokumencie.


## Listy folderów

Za pomocą listy folderów można uzyskać wykaz folderów z określonego folderu ze swojego konta Oracle Content Management.

Jeśli ten składnik zostanie użyty w połączeniu z jednym lub większą liczbą składników "lista plików" lub "menedżer dokumentów" (zawartymi na tej stronie), to składniki te będą mogły automatycznie wyświetlić zawartość folderu wybranego na liście folderów.

Aby dodać składnik "lista folderów" do strony, należy:

1. Przejść do strony, która ma być edytowana, i upewnić się, że przełącznik  jest ustawiony w pozycji **Edytuj**.

2. Dodać składnik do strony.
3. Aby edytować składnik i jego wygląd, kliknąć na ikonie menu składnika , po czym wybrać opcję **Ustawienia**.
4. Wybrać **Ustawienia niestandardowe**, aby określić ustawienia domyślne wyświetlanej zawartości.
5. Nacisnąć przycisk **Wybierz** (obok pola **Wybór folderu**), aby zmienić folder używany do wyświetlania. Trzeba wybrać folder inny niż strona startowa repozytorium dokumentów. Po ukończeniu nacisnąć przycisk **Wstecz**.

 **Uwaga:**

Lista folderów nadaje wszystkim użytkownikom uprawnienia "Pobierający". Użytkownicy, niezależnie od przypisanej im roli, mogą wyświetlać i pobierać pliki. Jeśli osoba odwiedzająca serwis ma uprawnienia wyższe niż określone dla składnika, to ustawienia w składniku zostaną przesłonięte przez indywidualne uprawnienia tej osoby.


6. Wybrać odpowiednie opcje, aby określić dodatkowe ustawienia domyślne wyświetlanej zawartości.
  - **Wybór domyślny:** Jeśli została wybrana opcja **Folder Oracle Documents**, wybrać folder (jeśli trzeba), który będzie pokazywany na liście jako wybrany.
  - **Pokaż nagłówek "Nazwa folderu":** Wybrać, aby w nagłówku osadzonego składnika była wyświetlana nazwa folderu.
  - **Sortowanie folderów:** Wybrać początkowy sposób wyświetlania elementów — alfabetycznie według nazwy albo według daty ostatniej aktualizacji.
  - **Pokaż podfoldery:** Używając suwaka, można ograniczyć liczbę wyświetlanych podfolderów.
7. Korzystając z karty "Ogólne", zmodyfikować ustawienia, odstępy i inne opcje prezentacji.
8. Korzystając z karty "Styl", sformatować ramkę zawierającą składnik; można użyć predefiniowanych stylów lub swoich własnych wyborów.


## Listy plików

Za pomocą listy plików można uzyskać widok plików z określonego folderu ze swojego konta Oracle Content

Jeśli ten składnik zostanie użyty w połączeniu z jednym lub większą liczbą składników "lista folderów" (zawartymi na tej stronie), to lista plików będzie mogła automatycznie wyświetlić zawartość folderu wybranego na liście folderów. Można również skonfigurować składnik tak, aby były wykonywane określone czynności, gdy użytkownik kliknie na pliku na liście. Na przykład można wyświetlić podgląd wybranego pliku w osobnym oknie przeglądarki, w nakładce usługi "Lightbox" lub w składniku "menedżer dokumentów" na stronie.

Aby dodać składnik "lista plików" do strony, należy:

1. Przejść do strony, która ma być edytowana, i upewnić się, że przełącznik  jest ustawiony w pozycji **Edytuj**.
2. Dodać grupę składników do strony.

3. Aby edytować składnik i jego wygląd, kliknąć na ikonie menu składnika , po czym wybrać opcję **Ustawienia**.
4. Wybrać **Ustawienia niestandardowe**, aby określić ustawienia domyślne wyświetlanej zawartości.
5. Nacisnąć przycisk **Wybierz** (obok pola **Wybór folderu**), aby zmienić folder używany do wyświetlania. Trzeba wybrać folder inny niż strona startowa dokumentu. Po ukończeniu nacisnąć przycisk **Wstecz**.

 **Uwaga:**

Lista plików nadaje wszystkim użytkownikom uprawnienia "Pobierający". Użytkownicy, niezależnie od przypisanej im roli, mogą wyświetlać i pobierać pliki. Jeśli osoba odwiedzająca serwis ma uprawnienia wyższe niż określone dla składnika, to ustawienia w składniku zostaną przesłonięte przez indywidualne uprawnienia tej osoby.

6. Wybrać odpowiednie opcje, aby określić dodatkowe ustawienia domyślne wyświetlanej zawartości.
  - **Opcje wyświetlania:** Wybrać szczegóły, które będą pokazywane wraz z plikami wyświetlanymi na liście.
    - **Nagłówek "Nazwa folderu":** pokazanie (w nagłówku) nazwy folderu.
    - **Opis pliku:** pokazanie opisu pliku (o ile opis jest dostępny).
    - **Separatory plików:** rozdzielenie poszczególnych plików linią.
    - **Ikona "Pobierz":** dołączenie ikony "Pobierz", tak aby użytkownicy mogli pobrać plik.
    - **Ostatnia aktualizacja:** pokazanie daty ostatniej aktualizacji pliku.
    - **Rozmiar pliku:** pokazanie rozmiaru pliku.
    - **Obraz:** pokazanie miniaturowego obrazu zawartości pliku.
  - **Wyzwalacze i czynności:** Wybrać, aby lista plików była automatycznie odświeżana o zawartość folderu wybranego w składniku "lista folderów". Można także wybrać uaktywnianie — gdy plik zostanie wybrany — wyzwalacza "Wybranie pliku". Korzystając z karty "Łącze", można powiązać czynności z wyzwalaczem **Wybranie pliku**.
  - **Sortowanie plików:** Wybrać początkowy sposób wyświetlania elementów — alfabetycznie według nazwy albo według daty ostatniej aktualizacji.
  - **Pokaż pliki:** Wybrać, czy mają być wyświetlane wszystkie pliki czy ograniczona ich liczba (z użyciem suwaka).
7. Korzystając z karty "Ogólne", zmodyfikować ustawienia, odstępy i inne opcje prezentacji.
8. Korzystając z karty "Styl", sformatować ramkę zawierającą składnik; można użyć predefiniowanych stylów lub swoich własnych wyborów.
9. Korzystając z karty "Łącze", można powiązać czynności z wyzwalaczem **Wybranie pliku**. Na przykład, aby skonfigurować składnik "lista plików" do uzyskiwania podglądu wybranego pliku w nakładce usługi "Lightbox":

- a. Upewnić się, że w oknie "Ustawienia listy plików" została wybrana opcja **Uaktywnij wyzwalacz, gdy plik zostanie wybrany**.
- b. Na karcie "Łącze" ustawień listy plików, kliknąć na wyzwalaczu **Wybranie pliku**.
- c. Przeciągnąć na listę czynności czynność **Podgląd u usłudze Lightbox** dot. strony.
- d. W polu "ID pliku" lub "Adres URL", wybrać **Łącze pliku**.



**Łącze pliku** korzysta z łącza referencyjnego, zawierającego uprawnienia zezwalające na pobieranie, dzięki czemu wszyscy odwiedzający serwis mogą wyświetlić podgląd i opcjonalnie pobrać plik. Jeśli zostanie użyty **ID pliku**, to zostanie użyte łącze dla współużytkowników. Łącze dla współużytkowników umożliwia — tylko zarejestrowanym użytkownikom — uzyskać podgląd wybranego pliku i opcjonalnie go pobrać.

Gdy użytkownik kliknie na pliku na liście, zostanie otwarty podgląd pliku na przyciemnionej i nieaktywnej wersji strony.

## Menedżer dokumentów

Za pomocą menedżera dokumentów można uzyskać widok strony startowej lub plików w Oracle Content Management.

Aby dodać składnik "menedżer dokumentów" do strony, należy:

1. Przejść do strony, która ma być edytowana, i upewnić się, że przełącznik  jest ustawiony w pozycji **Edytuj**.
2. Dodać składnik do strony.
3. Aby edytować składnik i jego wygląd, kliknąć na ikonie menu składnika , po czym wybrać opcję **Ustawienia**.
4. Wybrać **Ustawienia niestandardowe**, aby określić ustawienia domyślne wyświetlanej zawartości.
5. Nacisnąć przycisk **Wybierz** (obok pola **Wybór folderu**), aby zmienić folder używany do wyświetlania.

Aby wybrać folder, należy zaznaczyć znajdujące się obok niego pole wyboru. Aby otworzyć folder, wystarczy kliknąć na jego nazwie. Aby wrócić do folderu, należy kliknąć na jego nazwie występującej w ścieżce lub — aby wrócić do folderu podstawowego — nacisnąć przycisk **Przejdź do folderu podstawowego**. Po ukończeniu nacisnąć przycisk **Wstecz**.

6. Nacisnąć przycisk **Wybór dostępu do folderu**, po czym wybrać rolę dostępu przypisywaną odwiedzającym.

Odwiedzający będą mogli wyświetlać zawartość folderu i nad nią pracować, zależnie od ich roli i **opcji przeglądania**, które zostaną włączone w następnym punkcie.

- **Dostęp współużytkowników:** Odwiedzający będą mogli korzystać z dowolnych funkcji dostępnych dla współużytkowników folderu, takich jak wyświetlanie konwersacji, adnotacji czy niestandardowych właściwości elementów.
- **Oglądający:** Oglądający może przyglądać się plikom i folderom, lecz nie może niczego zmieniać.
- **Pobierający:** Pobierający może dodatkowo pobierać pliki i zapisywać je w swoim komputerze.


- **Współtwórca:** Współtwórca może dodatkowo modyfikować, aktualizować i usuwać pliki, a także wysyłać nowe pliki.

Podczas konfigurowania dostępu do folderu należy pamiętać, że:

- Autor serwisu nie może do folderu przypisać praw dostępu wyższych niż te, które sam posiada. Na przykład, jeśli autor ma w odniesieniu do folderu uprawnienia "Pobierający", to nie może odwiedzającym serwis nadać uprawnień "Współtwórca".
  - Uprawnienia odwiedzającego mogą być modyfikowane przez uprawnienia ustawione w składniku w odniesieniu do folderu. Na przykład, jeśli odwiedzający ma uprawnienia "Oglądający" (lub nie ma żadnych uprawnień) do folderu, składnik może przypisać — w zależności od wybranej roli — wyższe uprawnienia. Te rozszerzone uprawnienia obowiązują tylko w samym składniku.
  - Jeśli osoba odwiedzająca serwis ma uprawnienia wyższe niż określone dla składnika, to ustawienia w składniku zostaną przesłonięte przez indywidualne uprawnienia tej osoby.
  - Uprawnienia, nadane w odniesieniu do folderu, są stosowane do folderów i plików zagnieżdżonych w tym folderze.
7. Wybrać odpowiednie opcje, aby określić dodatkowe ustawienia domyślne wyświetlanej zawartości.
- **Układ:** Wybrać dla folderów i plików początkowy układ: tabeli, listy lub listy kompaktowej. Użytkownicy, wyświetlając gotowy osadzony składnik, mogą zmieniać układ.
  - **Schemat kolorów:** Wybrać dla osadzanego wykazu zawartości folderu jeden z dostępnych schematów kolorów.
  - **Porządek sortowania:** Wybrać początkowy sposób wyświetlania elementów — alfabetycznie według nazwy albo według daty ostatniej aktualizacji.
  - **Opcje przeglądania:** Wybrać opcje, które będą dostępne dla użytkowników podczas wybierania elementu. Na przykład można zezwolić na wyświetlanie plików, pobieranie plików, udostępnianie lub kopiowanie plików i folderów oraz usuwanie plików. Aby ograniczyć możliwość wykonywania niektórych czynności odnoszących się do plików lub folderów, należy wyczyścić pole wyboru odpowiedniej opcji na liście. Jeśli w poprzednim punkcie została wybrana opcja "Dostęp współużytkowników", można wyłączyć pokazywanie okienka bocznego, w którym są wyświetlane wszystkie konwersacje, adnotacje i właściwości niestandardowe.
  - **Opcje przeglądarki:** Wybrać sposób wyświetlania plików przez użytkowników. Można zezwolić na wyświetlanie plików w osadzonej ramce lub na innej karcie lub — w zależności od ustawień przeglądarki — w nowym oknie. Można także ukryć lub pokazać miniatury oraz dostosować sposób wyświetlania wideo.
  - **Pokaż elementy sterujące powiększaniem:** Można wybrać, czy w osadzonym widoku folderu, mają być pokazywane suwak i przyciski powiększania.
  - **Tryb dopasowania do przeglądarki:** Można wybrać, czy pliki mają wypełniać całą stronę, być dopasowywane do jej szerokości czy zachowywać rozmiar oryginalny.
  - **Wyzwalacze i czynności:** Wybrać, aby odświeżać listę plików, jeśli jest ona także używana w składniku obsługującym listę folderów.

8. Korzystając z karty "Ogólne", zmodyfikować ustawienia, odstępy i inne opcje prezentacji.
9. Korzystając z karty "Styl", sformatować ramkę zawierającą składnik; można użyć predefiniowanych stylów lub swoich własnych wyborów.

Gdy menedżer dokumentów zostanie dodany, oglądający zobaczą w serwisie widok wybranego folderu, prezentowany w ramce. Za pomocą udostępnionych opcji wyświetlania użytkownicy mogą zmieniać sposób wyszczególniania folderów i plików. Jeśli użytkownik wybierze plik lub folder, będzie mógł wybrać z paska menu lub z menu podręcznego (kliknięcie prawym przyciskiem myszy) dowolną dozwoloną czynność, wynikającą z roli i określonych **opcji przeglądania**.

Jeśli z elementem jest powiązana konwersacja, pojawi się obok niego ikona . Wystarczy na niej kliknąć, aby otworzyć okienko konwersacji, w którym można zobaczyć adnotacje i komentarze. Konwersacje niezależne (niepowiązane z folderem) muszą zostać dodane przy użyciu składnika "konwersacja".

Jeśli element ma właściwości niestandardowe, można je w tym okienku wyświetlić. Należy nacisnąć przycisk **Więcej**, po czym — aby otworzyć okienko właściwości — wybrać opcję **Właściwości niestandardowe**.



## Biblioteka projektu

Za pomocą składnika "biblioteka projektu" można w repozytorium zestawiać foldery z różnych lokalizacji, bez konieczności zmiany oryginalnego folderu lub jego lokalizacji. Używając składnika "biblioteka projektu", można przypisywać osobne uprawnienia do poszczególnych folderów w celu zapewnienia obsługi różnych ról zespołu projektowego.

Na przykład, jeśli do folderu znajdującego się w składniku "biblioteka projektu" zostanie przypisana jedna z ról "odwiedzający" (Oglądający, Pobierający lub Współtwórca), to odwiedzający mogą zobaczyć zawartość folderu i wchodzić z nią w interakcję. Jeśli zostanie ustawiony dostęp dla współużytkowników, to tylko współużytkownicy zobaczą ten folder wyszczególniony w składniku "biblioteka projektu". Współużytkownicy mogą wchodzić w interakcję z zawartością folderu, mając określone uprawnienia w odniesieniu do oryginalnego folderu.

Jeśli ten składnik zostanie użyty w połączeniu z jednym lub większą liczbą składników "lista plików" lub "menedżer dokumentów" (zawartymi na tej stronie), to składniki te będą mogły automatycznie wyświetlić zawartość folderu wybranego w składniku "biblioteka projektu".

Aby dodać do strony składnik "biblioteka projektu", należy:

1. Przejść do strony, która ma być edytowana, i upewnić się, że przełącznik  jest ustawiony w pozycji **Edytuj**.
2. Dodać składnik do strony.
3. Aby edytować składnik i jego wygląd, kliknąć na ikonie menu składnika , po czym wybrać opcję **Ustawienia**.
4. Aby wybrać konwersacje i opcje wyświetlania, wybrać opcję **Ustawienia niestandardowe**.
5. Podać tytuł. Domyślny tytuł to "Biblioteka projektu". Można wybrać wyświetlanie lub ukrywanie tytułu, wybierając poniżej opcję wyświetlania **Tytuł** lub anulując jej wybór.
6. Wybrać opcję **Schemat kolorów**.

 **Uwaga:**

Jeśli do powiązania biblioteki projektu ze składnikiem "menedżer dokumentów" są używane wyzwalacze i czynności, wybrany dla biblioteki projektów schemat kolorów zostanie również zastosowany do folderu wybranego do wyświetlania w składniku "menedżer dokumentów". Nie dotyczy to przypadku, w którym biblioteka projektu została powiązana ze składnikiem "lista folderów" lub "lista plików"; te składniki nie obsługują schematów kolorów.

7. Nacisnąć przycisk **Dodaj** (obok listy folderów), aby dodać istniejący folder lub utworzyć nowy.
  - a. Przejsć do przynajmniej jednego folderu i wybrać go lub — aby utworzyć folder — nacisnąć przycisk **Utwórz**. Każdy wybrany w tym oknie folder jest dodawany do składnika "biblioteka projektu".

Można użyć dowolnej opcji z paska narzędzi, w tym opcji wyświetlania i sortowania.
  - b. Po ukończeniu nacisnąć przycisk **Wstecz**.
8. Aby uporządkować listę, wybrać folder, po czym — w celu zmiany jej położenia — nacisnąć przycisk **Przenieś w górę** lub **Przenieś w dół** albo — aby usunąć folder z listy — przycisk **Usuń**.
9. Aby określić dostęp do folderu na podstawie roli użytkownika, wybrać folder z listy folderów, po czym wybrać opcję **Wybór uprawnień do folderu**, a następnie rolę dostępu.
  - **Dostęp współużytkowników:** Tylko zarejestrowani użytkownicy mający uprawnienia do folderu mogą zobaczyć ten folder w składniku "biblioteka projektu". Współużytkownicy mogą wchodzić w interakcję z zawartością folderu, mając określone uprawnienia w odniesieniu do oryginalnego folderu.
  - **Oglądający:** Oglądający może przyglądać się plikom i folderom, lecz nie może niczego zmieniać.
  - **Pobierający:** Pobierający może dodatkowo pobierać pliki i zapisywać je w swoim komputerze.
  - **Współtwórca:** Współtwórca może dodatkowo modyfikować, aktualizować i usuwać pliki, a także wysyłać nowe pliki.

Podczas konfigurowania dostępu do folderu należy pamiętać, że:

- Autor serwisu nie może do folderu przypisać praw dostępu wyższych niż te, które sam posiada. Na przykład, jeśli autor ma w odniesieniu do folderu uprawnienia "Pobierający", to nie może odwiedzającym serwis nadać uprawnień "Współtwórca".
- Uprawnienia odwiedzającego mogą być modyfikowane przez uprawnienia ustawione w składniku w odniesieniu do folderu. Na przykład, jeśli odwiedzający ma uprawnienia "Oglądający" (lub nie ma żadnych uprawnień) do folderu, składnik może przypisać — w zależności od wybranej roli — wyższe uprawnienia. Te rozszerzone uprawnienia obowiązują tylko w samym składniku.
- Jeśli osoba odwiedzająca serwis ma uprawnienia wyższe niż określone dla składnika, to ustawienia w składniku zostaną przesłonięte przez indywidualne uprawnienia tej osoby.

- Upewnienia, nadane w odniesieniu do folderu, są stosowane do folderów i plików zagnieżdżonych w tym folderze.
10. W przypadku folderów z uprawnieniem **Dostęp współużytkowników**, można wyświetlić konwersacje powiązane z folderami lub zawartością, wybierając opcję **Pokaż okienko konwersacji w menedżerze dokumentów**.

Jeśli zostanie wybrana ta opcja oraz na stronie zostanie skonfigurowany składnik "menedżer dokumentów" do wyświetlania wybranego folderu, to użytkownik może kliknąć na ikonie konwersacji w celu wyświetlenia dowolnej konwersacji powiązanej z folderem. Jeśli ta opcja nie zostanie wybrana, to ikona konwersacji nie pojawi się.
  11. Po ukończeniu zamknąć okno.
  12. Korzystając z karty "Ogólne", zmodyfikować ustawienia, odstępy i inne opcje prezentacji.
  13. Korzystając z karty "Styl", sformatować ramkę zawierającą składnik; można użyć predefiniowanych stylów lub swoich własnych wyborów.
  14. Korzystając z karty "Łącze", powiązać czynności z wyzwalaczem **Wybrany folder**. Na przykład, jeśli do strony zostanie dodany składnik "menedżer dokumentów", to składnik ten będzie mógł zostać użyty do wyświetlania zawartości folderu wybranego w folderze projektu:
    - a. Na karcie "Łącze" ustawień składnika "biblioteka projektu" kliknąć na wyzwalaczu **Wybrany folder**.
    - b. W oknie "Konfiguracja czynności wyzwalacza" wybrać opcję "Menedżer dokumentów".
    - c. Przeciągnąć na listę czynności czynność **Wyświetl dokumenty**.
    - d. W polu "ID folderu" lub "Adres URL" wybrać **Wybrany folder**.

Gdy użytkownik kliknie na folderze znajdującym się w składniku "biblioteka projektu", zawartość folderu zostanie wyświetlona w składniku "menedżer dokumentów" na stronie.

## Składniki społecznościowe

Składniki społecznościowe pomagają użytkownikom pozostawać w kontakcie (komunikować się).

- [Pasek społecznościowy](#)
- [Facebook — Lubię to/Rekomendacja](#)
- [Twitter - Udostępnij/Obserwuj](#)
- [Składnik "konwersacja"](#)
- [Lista konwersacji](#)



## Pasek społecznościowy

Z łatwością można dodawać ikony i łącza prowadzące do popularnych mediów społecznościowych, takich jak Facebook czy Twitter. Pasek społecznościowy domyślnie zawiera niektóre cele, lecz można samodzielnie dodawać i usuwać jego elementy.







Aby dodać i zmodyfikować pasek społecznościowy, należy:

1. Przejść do strony, która ma być edytowana, i upewnić się, że przełącznik  jest ustawiony w pozycji **Edytuj**.
2. Dodać składnik do strony. Na pasku społecznościowym są pokazywane domyślnie dodane ikony.
3. Aby dodać lub zmienić ikony na pasku społecznościowym, kliknąć na jego ikonie , wybrać pozycję **Ustawienia**, po czym kliknąć na opcji **Ikony** na górze karty **Ogólne**.

Aby usunąć ikonę społecznościową, kliknąć na znaku x obok jej nazwy. Aby można było dodać ikonę, musi być ona przechowywana w repozytorium serwisu albo w innym repozytorium, do którego mamy dostęp. Można także używać obrazów, które zostały nam udostępnione lub które wysłaliśmy z lokalizacji lokalnej lub sieciowej.



4. Aby wysłać ikony z lokalizacji lokalnej lub sieciowej:
  - a. Nacisnąć przycisk **Dodaj ikony** znajdujący się na górze panelu.
  - b. Przejść w repozytorium do miejsca, w którym obrazy będą przechowywane albo kliknąć na ikonie , aby dodać nowy folder w bieżącej lokalizacji.
  - c. Kliknąć na ikonie .
  - d. Odszukać i wybrać pliki obrazów, po czym nacisnąć przycisk **Otwórz**.  
Pliki zostaną wysłane do bieżącej lokalizacji w repozytorium.
5. Wybrać jeden lub większą liczbę obrazów w repozytorium, po czym nacisnąć przycisk **OK**.
  - a. Odszukać obraz (lub obrazy), który ma zostać użyty, po czym na nim kliknąć.
  - b. Aby utworzyć łącze do pliku w repozytorium, wybrać opcję **Zamiast kopiować plik do serwisu, użyj odwołania do pliku oryginalnego**. Jeśli ta opcja nie zostanie wybrana, w serwisie będzie składowana kopia pliku, do której będą się odnosiły odwołania z serwisu. Tworząc łącze do oryginalnego pliku, unika się duplikowania zawartości. Łącze pozwala odwiedzającym serwis oglądać zawartość, nawet jeśli ulegną zmianie uprawnienia do pliku lub jego wyświetlanie zostanie ograniczone w inny sposób.
  - c. Nacisnąć przycisk **OK**.

Wybrane obrazy zostają dodane do listy obrazów. Można zmienić ich rozmieszczenie na liście, przeciągając je. Tytułem domyślnym każdego obrazu jest nazwa pliku bez rozszerzenia.

6. Aby zmienić domyślny URL, tytuł, opis lub inne opcje określonego obrazu, kliknąć na nim na liście, po czym dokonać zmian.  
Po ukończeniu aktualizacji ikon nacisnąć przycisk **Wstecz**, aby określić opcje ogólne.
7. Określić rozmiar, odstępy, orientację i wyrównanie wszystkich ikon na pasku społecznościowym.



## Facebook — Lubię to/Rekomendacja

Można dodać do strony przycisk "Facebook - Lubię to", aby umożliwić użytkownikom zasygnalizowanie (w serwisie Facebook) polubienia serwisu.

1. Przejść do strony, która ma być edytowana, i upewnić się, że przełącznik  jest ustawiony w pozycji **Edytuj**.
2. Dodać składnik do strony.
3. Aby edytować aplikację i jej wygląd, kliknąć na jej ikonie menu , po czym wybrać opcję **Ustawienia**.
4. Wybrać **Ustawienia: Facebook - Lubię to** lub **Ustawienia: Facebook - Rekomendacja**, aby określić URL oraz opcjonalnie dodać przycisk **Udostępnij**, tak aby można było łatwo przesłać (do serwisu Facebook) łącze prowadzące do serwisu.
5. Korzystając z karty "Ogólne", zmodyfikować ustawienia, odstępy i inne opcje prezentacji.
6. Korzystając z karty "Styl", sformatować ramkę zawierającą aplikację; można użyć predefiniowanych stylów lub swoich własnych wyborów.

## Twitter - Udostępnij/Obserwuj

Można dodać do strony przycisk "Twitter - Udostępnij", aby umożliwić użytkownikom szybkie udostępnianie zawartości na koncie serwisu Twitter.

1. Przejść do strony, która ma być edytowana, i upewnić się, że przełącznik  jest ustawiony w pozycji **Edytuj**.
2. Dodać składnik do strony.
3. Aby edytować aplikację i jej wygląd, kliknąć na jej ikonie menu , po czym wybrać opcję **Ustawienia**.
4. Wybrać opcję **Ustawienia: Twitter - Obserwuj**, aby ustawić nazwę użytkownika serwisu Twitter, wybrać, czy nazwa ma być pokazywana, oraz wybrać rozmiar przycisku **Obserwuj**. Wybrać **Ustawienia: Twitter - Udostępnij**, aby określić poniższe ustawienia domyślne. Użytkownicy, gdy korzystają z tego przycisku, mogą te wartości zmienić.
  - **Udostępnij URL**: URL serwisu, który może zostać udostępniony przez użytkownika.
  - **Tekst tweeta**: Tekst wpisu odnoszącego się do strony.
  - **Via @**: Konto użytkownika serwisu Twitter używane dla tego tweeta.
  - **Rekomendacje @**: Konto użytkownika serwisu Twitter używane do rekomendacji na Twitterze.
  - **Hasztag #**: Hasztag, który ma być używany dla tego wpisu.
  - **Liczba**: Liczba udostępnień (wyświetlane pionowo lub poziomo).
  - **Duży przycisk**: Wybrać duży lub mały przycisk dla aplikacji.
5. Korzystając z karty "Ogólne", zmodyfikować ustawienia, odstępy i inne opcje prezentacji.
6. Korzystając z karty "Styl", sformatować ramkę zawierającą aplikację; można użyć predefiniowanych stylów lub swoich własnych wyborów.



## Składnik "konwersacja"

Za pomocą konwersacji można promować dyskusję na dany temat bezpośrednio ze swojej strony.

### Uwaga:

Aby składnik "konwersacja" mógł działać w serwisie, musi to być serwis zabezpieczony, ograniczony do określonych użytkowników albo do użytkowników z rolą "Użytkownik Oracle Content Management". Zob. [Zmianie zabezpieczeń serwisu](#).

Aby dodać do strony składnik "konwersacja", należy:

1. Przejść do strony, która ma być edytowana, i upewnić się, że przełącznik  jest ustawiony w pozycji **Edytuj**.
2. Dodać składnik do strony.
3. Aby edytować składnik i jego wygląd, kliknąć na ikonie menu składnika , po czym wybrać opcję **Ustawienia**.
4. Aby wybrać konwersację i schemat kolorów prezentacji, wybrać opcję **Ustawienia niestandardowe**.
5. Aby wybrać istniejącą konwersację lub utworzyć nową, nacisnąć przycisk **Wybierz** znajdujący się obok opcji **Wybór konwersacji**.
  - a. Wybrać konwersację z listy dostępnych konwersacji albo nacisnąć przycisk **Utwórz** w celu utworzenia nowej konwersacji i nazwania jej.

Lista zawiera wszystkie konwersacje, do których bieżący użytkownik ma dostęp.

Można przefiltrować listę:

- **Wszystkie**: Pokazuje wszystkie konwersacje z wyjątkiem tych, które zostały oznaczone jako wyciszone.
- **Ulubione**: Pokazuje konwersacje, które zostały oznaczone jako ulubione.
- **Wyciszone**: Pokazuje konwersacje, które zostały wyciszone i wykluczone z listy konwersacji bieżącego użytkownika.
- **Zamknięte**: Pokazuje wszystkie konwersacje, które zostały oznaczone jako zamknięte.

Listę tę można również posortować:

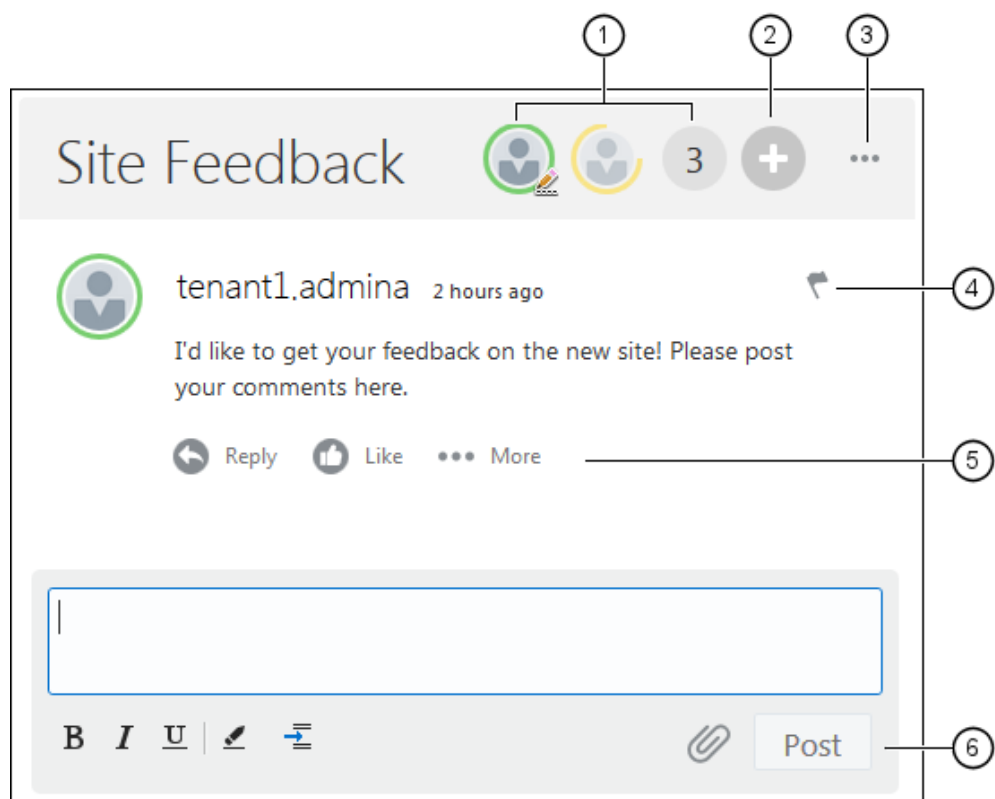
- **Ostatnia aktualizacja**: Wyświetla konwersacje w kolejności od ostatnio do najdawniej aktualizowanych.
- **Nazwa**: Wyświetla konwersacje w porządku alfanumerycznym, od wartości najniższej do najwyższej.
- **Nieprzeczytane**: Wyświetla nieprzeczytane konwersacje w kolejności od konwersacji z największą liczbą nieprzeczytanych komentarzy do konwersacji z najmniejszą liczbą nieprzeczytanych komentarzy.

- b. Po ukończeniu nacisnąć przycisk **Wstecz**.
  - c. Jeśli została wybrana konwersacja niezależna (nieoparta na pliku czy folderze), można ustawić uprawnienia dostępu dla osób wyświetlających konwersację.
  - d. Wybrać opcję **Schemat kolorów**.
  - e. Aby składnik "konwersacja" był używany na stronie w połączeniu ze składnikiem "lista konwersacji", wybrać opcję **Automatyczne odświeżanie konwersacji na podstawie wyboru w składniku "Lista konwersacji"**.  
Gdy użytkownik wybierze z listy konwersację, zostanie ona wyświetlona w składniku "konwersacja".
  - f. Po ukończeniu zamknąć okno.
6. Korzystając z karty "Ogólne", zmodyfikować ustawienia, odstępy i inne opcje prezentacji.
  7. Korzystając z karty "Styl", sformatować ramkę zawierającą składnik; można użyć predefiniowanych stylów lub swoich własnych wyborów.

W opublikowanym serwisie odwiedzający będą widzieć konwersację w ramce. Użytkownicy — aby przeczytać komentarze i na nie odpowiedzieć — mogą nawigować w obrębie konwersacji.

 **Uwaga:**

Jeśli odwiedzający nie został jednoznacznie dodany do konwersacji jako jej uczestnik, będzie mógł czytać i odpowiadać na komentarze, ale nie będzie widzieć elementów oznaczonych od 1 do 4 na poniższym rysunku.



- **Pasek menu (1)** zawiera informacje o poszczególnych uczestnikach konwersacji. Klikając na ikonie użytkownika, można uzyskać informacje o jego statusie oraz wyświetlić opcje pracy z użytkownikiem. Ikona "Liczba" pokazuje łączną liczbę uczestników konwersacji. Aby wyświetlić pełną listę, należy na tej ikonie kliknąć.
- Aby dodać użytkowników do konwersacji, należy kliknąć na ikonie **Dodaj użytkowników (2)**.
- Klikając na ikonie **Więcej opcji (3)**, można wyświetlić listę opcji do pracy z konwersacją. Lista opcji zmienia się w zależności od roli użytkownika. Dla konwersacji powiązanych z folderami i plikami lista opcji zależy również od uprawnień ustawionych w odniesieniu do danego folderu lub pliku.
- Aby zaalarmować konkretnego użytkownika, poprzez przypisanie mu flagi powiadomienia, należy kliknąć na ikonie **Flagi (4)**. Użytkownik zostanie powiadomiony e-mailowo (jak określono w jego preferencjach).
- Korzystając z **opcji dot. komentarza (5)**, można w odniesieniu do konkretnego komentarza w konwersacji wykonać takie czynności, jak: odpowiedzenie na komentarz bądź polubienie, edytowanie lub usunięcie komentarza.
- Dodając lub edytując komentarz, można za pomocą **opcji edycji (6)** dodać lub usunąć podstawowe formatowanie (takie jak pogrubienie lub podkreślenie), dodać załącznik oraz przesłać komentarz do konwersacji. Osoba odwiedzająca serwis, niebędąca uczestnikiem konwersacji, nie może dodawać załączników.

## Lista konwersacji



Za pomocą konwersacji można promować dyskusję na dany temat bezpośrednio ze swojej strony.

Jeśli ten składnik zostanie użyty w połączeniu z jednym lub większą liczbą składników "lista konwersacji" (zawartymi na tej stronie), to lista konwersacji będzie mogła automatycznie wyświetlić zawartość konwersacji wybranej na liście konwersacji.

### Uwaga:

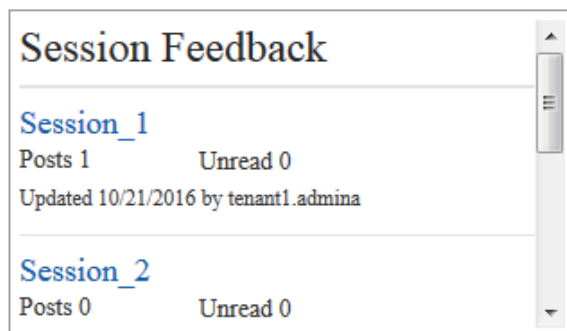
Aby składnik "konwersacja" lub składnik "lista konwersacji" mógł działać w serwisie, musi to być serwis zabezpieczony, ograniczony do określonych użytkowników albo do użytkowników z rolą "Użytkownik Oracle Content Management". Zob. [Zmianie zabezpieczeń serwisu](#).

Aby dodać składnik "lista konwersacji" do strony, należy:

1. Przejść do strony, która ma być edytowana, i upewnić się, że przełącznik  jest ustawiony w pozycji **Edytuj**.
2. Dodać składnik do strony.
3. Aby edytować składnik i jego wygląd, kliknąć na ikonie menu składnika , po czym wybrać opcję **Ustawienia**.
4. Aby wybrać konwersacje i opcje wyświetlania, wybrać opcję **Ustawienia niestandardowe**.

5. Podać tytuł. Domyślny tytuł to "Lista konwersacji". Można wybrać wyświetlanie lub ukrywanie tytułu, wybierając poniżej opcję wyświetlania **Tytuł** lub anulując jej wybór.
6. Nacisnąć przycisk **Dodaj** (obok listy konwersacji), aby dodać istniejącą konwersację lub utworzyć nową.
  - a. Wybrać konwersację/konwersacje z listy dostępnych konwersacji albo nacisnąć przycisk **Utwórz** w celu utworzenia nowej konwersacji i nazwania jej.
  - b. Po ukończeniu nacisnąć przycisk **Wstecz**.
  - c. Wybrać uprawnienia dozwolone dla tych osób, które wyświetlają składnik "lista konwersacji".
7. Aby uporządkować listę konwersacji, wybrać konwersację, po czym — w celu zmiany jej położenia — nacisnąć przycisk **Przenieś w górę** lub **Przenieś w dół** albo — aby usunąć konwersację z listy — przycisk **Usuń**.
8. Wybrać **Schemat kolorów** oraz dodatkowe **Opcje wyświetlania**. Wybrane opcje wyświetlania są pokazywane pod nazwą konwersacji na liście.
9. Po ukończeniu zamknąć okno.
10. Korzystając z karty "Ogólne", zmodyfikować ustawienia, odstępy i inne opcje prezentacji.
11. Korzystając z karty "Styl", sformatować ramkę zawierającą składnik; można użyć predefiniowanych stylów lub swoich własnych wyborów.

Na poniższym rysunku jest pokazywana lista konwersacji, zatytułowana *Session Feedback*, z wybranymi wszystkimi opcjami wyświetlania.



## Składniki procesu

Składniki procesu umożliwiają użytkownikowi serwisu inicjowanie zadań dla predefiniowanych procesów i zarządzanie tymi zadaniami.

### Uwaga:

Aby można było korzystać ze składników procesu, trzeba mieć dostępną usługę Oracle Process Cloud Service wydanie 17.1.3 lub nowsze.

- [Formularz uruchamiania procesu](#)
- [Lista zadań procesu](#)

- [Formularz szczegółów zadania](#)



## Formularz uruchamiania procesu

W celu zainicjowania procesu, zdefiniowanego w Oracle Process Cloud Service, można skorzystać z formularza uruchamiania procesu.

Aby formularz uruchamiania procesu działał w serwisie, muszą zostać spełnione następujące warunki:

- Aby powiązane procesy oraz formularze uruchamiania procesów można było wyświetlić za pomocą tego składnika, trzeba je zdefiniować w Oracle Process Cloud Service. Zob. *Developing Structured Processes* w podręczniku *Using Processes in Oracle Integration*.
- Aby można było użyć formularza uruchamiania procesu, użytkownik musi mieć rolę powiązaną z torem procesu zawierającym formularz uruchamiający.
- Administrator musi skonfigurować integrację Oracle Process Cloud Service i Oracle Content Management. Zob. *Integrowanie z Oracle Process Cloud Service* w podręczniku *Administering Oracle Content Management*. Integracja tych usług wymaga użycia funkcji jednokrotnego logowania (SSO), dlatego też obie usługi muszą się znajdować w tej samej domenie tożsamości.

Aby dodać składnik "formularz uruchamiania procesu" do strony, należy:

1. Przejść do strony, która ma być edytowana, i upewnić się, że przełącznik  jest ustawiony w pozycji **Edytuj**.
2. Dodać składnik do strony.
3. Aby edytować składnik i jego wygląd, kliknąć na ikonie menu  składnika, po czym wybrać opcję **Ustawienia**.
4. Aby wybrać formularz uruchamiania procesu i ustawić właściwości domyślne formularza, wybrać opcję **Ustawienia niestandardowe**.
  - a. Jeśli serwis ma być serwisem publicznym, wybrać usługę proxy.
  - b. Wybrać partycję dla formularza uruchamiającego. Można użyć partycji testowej do sprawdzenia, czy proces działa zgodnie z planem, lub partycji produkcyjnej do wdrożenia procesu w celu ogólnego zastosowania.
  - c. Jeśli ma być zawsze używana wersja procesu ustawiona jako domyślna, wybrać opcję **Użyj domyślnej wersji procesu**. Jeśli ta opcja nie zostanie wybrana, trzeba wybrać konkretną wersję i — jeśli proces zostanie zaktualizowany — trzeba będzie w tych ustawieniach zaktualizować dokonany wybór.
  - d. Wybrać proces.
  - e. Wybrać formularz uruchamiający. Formularze mają następującą składnię: `typ_procesu:wersja:nazwa_procesu:start`. Na przykład `Basic Approval:1.0:Process:Start Basic Approval`.

Jeśli na górze okna jest wyświetlany komunikat "Brak połączenia z Process Cloud Service", to sygnalizuje on, że prawdopodobnie nie została skonfigurowana integracja między Oracle Process Cloud Service i Oracle Content Management. Proszę się skontaktować z administratorem.

Aby można było na liście zobaczyć proces, jego autor musi dodać użytkownika jako inicjatora procesu. Autor procesu musi dodać wszystkich odwiedzających jako inicjatorów procesu; w przeciwnym razie odwiedzający będą mogli wypełnić formularz, ale nie będą mogli zainicjować procesu.

- f. Wybrać odpowiednie opcje, aby określić dodatkowe ustawienia domyślne formularza.
  - **Tytuł formularza:** Opcjonalnie zastąpić domyślny tytuł formularza własnym tytułem.
  - **Nazwa przycisku "Wyślij":** Opcjonalnie zmienić nazwę przycisku **Wyślij** na inną.
  - **Pokazywanie przycisku "Prześlij":** Opcjonalnie wybrać pokazywanie lub ukrywanie przycisku **Prześlij** składnika (domyślnie jest pokazywany). Można ukryć przycisk **Prześlij** składnika, jeśli podobny przycisk jest udostępniany w samym formularzu.
  - **Potwierdzenie przesłania:** Opcjonalnie zastąpić domyślny komunikat potwierdzający swoim własnym komunikatem.
  - **Pokazywanie potwierdzenia przesłania:** Opcjonalnie wybrać pokazywanie lub ukrywanie komunikatu potwierdzającego, gdy zostanie naciśnięty przycisk **Prześlij**. Przycisk ten jest domyślnie pokazywany i wyświetlany w składniku "formularz uruchamiania procesu".
  - **Pokazywanie przycisku "Zapisz":** Opcjonalnie wybrać pokazywanie lub ukrywanie przycisku **Zapisz** składnika (domyślnie jest ukryty). Pokazywanie przycisku **Zapisz** składnika można wybrać, jeśli powiązany proces zezwala użytkownikowi na zapisanie wykonanej pracy i późniejszy do niej powrót.
  - **Pokazywanie przycisku "Odrzuć":** Opcjonalnie wybrać pokazywanie lub ukrywanie przycisku **Odrzuć** składnika (domyślnie jest ukryty). Pokazywanie przycisku **Odrzuć** można wybrać, aby zezwolić użytkownikowi na odrzucenie zawartości formularza i rozpoczęcie pracy od nowa.
  - **Pokazywanie załączników:** Opcjonalnie wybrać pokazywanie lub ukrywanie obszaru **Załączniki** formularza, umożliwiającego wysyłanie pliku lub plików (domyślnie jest ukryty). Wysłane pliki są w ramach procesu składowane w Oracle Process Cloud Service.
  - **Dostosowywanie wartości domyślnych:** Można ustawić pary nazw i wartości w celu wstępnego wypełnienia formularza uruchamiającego. Należy nacisnąć przycisk **Dodaj pole**, po czym podać nazwę pola i wartość. Nazwa jest jednym z pól formularza, a nie etykietą używaną w formularzu, natomiast wartość określa, co jest dozwolone dla tego pola. Aby wstawić dane bieżącego zalogowanego użytkownika, można użyć wartości specjalnych — %%username%% i %%userid%%.
5. Korzystając z karty "Ogólne", zmodyfikować ustawienia, odstępy i inne opcje prezentacji.
6. Korzystając z karty "Styl", sformatować ramkę zawierającą składnik; można użyć predefiniowanych stylów lub swoich własnych wyborów.
7. Za pomocą karty "Łącza" można przypisać czynności do wyzwalaczy udostępnionych przez składnik:
  - **Przesłano formularz uruchamiający:** Ten wyzwalacz pojawia się, gdy użytkownik naciśnie przycisk **Prześlij**.
  - **Zapisano formularz uruchamiający:** Ten wyzwalacz pojawia się, gdy użytkownik naciśnie przycisk **Zapisz**.



- **Odrzucono formularz uruchamiający:** Ten wyzwalacz pojawia się, gdy użytkownik naciśnie przycisk **Odrzuć**.

Aby przypisać czynność, należy kliknąć na wyzwalaczu. Więcej informacji dotyczących przypisywania wyzwalaczy i czynności jest dostępnych pod hasłem [Używanie wyzwalaczy](#)



## Lista zadań procesu

W celu wyszczególnienia procesów, zdefiniowanych w Oracle Process Cloud Service, można skorzystać ze składnika "lista zadań procesu". Można wyświetlić szczegółowe informacje dotyczące zadań w składniku "lista zadań" albo użyć składnika "lista zadań procesu" w połączeniu ze składnikiem "szczegóły zadania" w celu wyszczególnienia i wyświetlenia szczegółowych informacji dotyczących tylko wybranego zadania.

Aby składnik "lista zadań procesu" mógł działać w serwisie, muszą zostać spełnione następujące warunki:

- Aby powiązane procesy można było wyświetlić za pomocą tego składnika, trzeba je zdefiniować w Oracle Process Cloud Service. Zob. *Developing Structured Processes* w podręczniku *Using Processes in Oracle Integration*.
- Serwis musi być serwisem zabezpieczonym, ograniczonym do określonych użytkowników albo do użytkowników z rolą "Użytkownik Oracle Content Management". Zob. [Zmienianie zabezpieczeń serwisu](#).
- Administrator musi skonfigurować integrację Oracle Process Cloud Service i Oracle Content Management. Zob. *Integrowanie z Oracle Process Cloud Service* w podręczniku *Administering Oracle Content Management*. Integracja między dwiema usługami wymaga systemu jednokrotnego logowania (SSO), dlatego też obie usługi muszą być w tej samej domenie tożsamości.

Aby dodać składnik "lista zadań procesu" do strony, należy:

1. Przejść do strony, która ma być edytowana, i upewnić się, że przełącznik  jest ustawiony w pozycji **Edytuj**.
2. Dodać składnik do strony.
3. Aby edytować składnik i jego wygląd, kliknąć na ikonie menu  składnika, po czym wybrać opcję **Ustawienia**.
4. Aby ustawić opcje wyświetlania i przefiltrować dostępne zadania, wybrać opcję **Ustawienia niestandardowe**.
5. Aby ustawić opcje wyświetlania listy zadań, wybrać jedną z następujących opcji.

### Uwaga:

Dla składnika można ustawić opcje wyświetlania, lecz prezentacja zadań jest zależna od projektu w usłudze Oracle Process Cloud Service.

- **Pokaż szczegóły:** Należy wybrać, aby w samej liście zadań były zawarte szczegółowe informacje dotyczące zadania. Można użyć składnika "lista zadań procesu" w połączeniu ze składnikiem "szczegóły zadania" w celu

wyszczególnienia i wyświetlenia szczegółowych informacji dotyczących tylko wybranego zadania.

- **Pokaż wyszukiwanie:** Należy wybrać, aby umieścić pasek wyszukiwania na górze listy zadań. Jeśli zostanie wybrana opcja **Pokaż wyszukiwanie**, to pojawi się dodatkowa opcja wyboru **Pokaż filtr** dodająca do paska wyszukiwania opcję filtrowania. Należy użyć poniższych opcji filtrowania w celu ustawienia domyślnych wartości filtrowania.
  - **Pokaż opcję "Wybierz wszystkie":** Należy wybrać, aby umożliwić użytkownikowi wybór wszystkich wyświetlanych zadań do przetwarzania.
  - **Rozmiar strony:** Należy dostosować do maksymalnej liczby wyświetlanych zadań. Jeśli istniejących zadań jest mniej niż podana maksymalna liczba, to składnik automatycznie dostosuje się do mniejszej liczby. Jeśli istniejących zadań jest więcej niż podana maksymalna liczba, to w składniku zostanie wyświetlona maksymalna liczba zadań oraz pojawią się łącza do dodatkowych stron.
6. Aby można było filtrować zadania dostępne na liście, wybrać jedną z następujących opcji.
- Stosowane filtry określają początkową listę zadań. Jeśli zostaną wybrane powyższe opcje **Pokaż wyszukiwanie** i **Pokaż filtr**, to użytkownik będzie mógł dostosować ustawienia w celu zmodyfikowania filtra dla wszystkich ustawień, z wyjątkiem **Od użytkownika**.
- **Słowa kluczowe wyszukiwania:** Opcjonalnie określić przynajmniej jeden szukany termin. W wynikach wyszukiwania są zwracane tylko te zadania, w których tytułach są zawarte podane słowa kluczowe. Jeśli została wybrana opcja **Pokaż wyszukiwanie**, to te słowa kluczowe zostaną wyświetlone na pasku wyszukiwania, a użytkownik będzie mógł je usunąć lub zmodyfikować.
  - **Status**  
: Należy wybrać jeden z dostępnych statusów. Domyślny status to **Przypisane**.
  - **Osoba odpowiedzialna:** Należy wybrać jedną z dostępnych osób odpowiedzialnych. Domyślnym ustawieniem jest **Wszystkie moje i mojej grupy**, obejmujące wszystkie zadania dostępne dla użytkownika i jego grupy — w tym te, które zostały przyjęte, ale nie rozpoczęto nad nimi żadnych prac.
  - **Od użytkownika:** Opcjonalnie wybrać przynajmniej jednego użytkownika, od którego pochodzi zadanie. Aby rozpocząć wyszukiwanie dostępnych użytkowników, powiązanych z Oracle Process Cloud Service, należy zacząć wpisywać nazwę użytkownika. Podane wartości nie są wyświetlane użytkownikowi końcowemu i nie można ich usunąć. Użytkownicy mogą dodawać dodatkowe nazwy użytkowników w celu rozszerzenia listy użytkowników.
  - **Termin:** Opcjonalnie wybrać termin "W dniu", "Przed", "Po" lub "Pomiędzy" powiązany z datą wybraną z kalendarza.
  - **Aplikacja:** Opcjonalnie wybrać nazwę konkretnej aplikacji. Z listy aplikacji dostępnych dla użytkownika można wybrać jedną lub więcej aplikacji.
7. Korzystając z karty "Ogólne", zmodyfikować ustawienia, odstępy i inne opcje prezentacji.
8. Korzystając z karty "Styl", sformatować ramkę zawierającą składnik; można użyć predefiniowanych stylów lub swoich własnych wyborów.

 **Uwaga:**

Jeśli na stronie znajduje się lista zadań procesu i formularz szczegółów zadania, to składnik "szczegóły zadania" automatycznie będzie wyświetlać szczegóły zadania wybranego z listy. Więcej informacji dotyczących wyzwalaczy i czynności jest dostępnych w rozdziale [Używanie wyzwalaczy i czynności](#).



## Formularz szczegółów zadania

W celu wyświetlenia szczegółów wybranego zadania można skorzystać z formularza szczegółów zadania w połączeniu z listą zadań procesu. Jeśli na stronie znajdują się lista zadań procesu i formularz szczegółów procesu, to składnik "szczegóły zadania procesu" automatycznie będzie wyświetlać szczegóły zadania wybranego z listy.

Aby formularz szczegółów zadania mógł działać w serwisie, muszą zostać spełnione następujące warunki:

- Aby powiązane procesy można było wyświetlić za pomocą tego składnika, trzeba je zdefiniować w Oracle Process Cloud Service. Zob. *Developing Structured Processes* w podręczniku *Using Processes in Oracle Integration*.
- Serwis musi być serwisem zabezpieczonym, ograniczonym do określonych użytkowników albo do użytkowników z rolą "Użytkownik Oracle Content Management". Zob. [Zmianianie zabezpieczeń serwisu](#).
- Administrator musi skonfigurować integrację Oracle Process Cloud Service i Oracle Content Management. Zob. *Integrowanie z Oracle Process Cloud Service* w podręczniku *Administering Oracle Content Management*. Integracja między dwiema usługami wymaga systemu jednokrotnego logowania (SSO), dlatego też obie usługi muszą być w tej samej domenie tożsamości.

Aby dodać składnik "szczegóły zadania" do strony, należy:

1. Przejść do strony, która ma być edytowana, i upewnić się, że przełącznik  jest ustawiony w pozycji **Edytuj**.
2. Dodać składnik do strony.
3. Aby edytować składnik i jego wygląd, kliknąć na ikonie menu  składnika, po czym wybrać opcję **Ustawienia**.
4. Aby ustawić opcje wyświetlania, wybrać opcję **Ustawienia niestandardowe**.
5. Aby ustawić opcje wyświetlania formularza szczegółów zadania, wybrać jedną z następujących opcji.

 **Uwaga:**

Dla składnika można ustawić opcje wyświetlania, ale prezentacja szczegółów zadania jest zależna od projektu w usłudze Oracle Process Cloud Service.

- **Pokaż czynności:** Należy wybrać, aby wyświetlić dostępne dla użytkownika czynności, takie jak "Zatwierdź" czy "Odrzuć".
  - **Pokaż przycisk "Zapisz":** Należy wybrać, aby był wyświetlany przycisk **Zapisz**.
  - **Pokaż przycisk "Zamknij":** Należy wybrać, aby był wyświetlany przycisk **Zamknij**.
  - **Pokaż sekcję "Załącznik":** Należy wybrać, aby w formularzu szczegółów była wyświetlana sekcja **Załącznik**.
  - **Pokaż sekcję "Komentarze":** Należy wybrać, aby w formularzu szczegółów była wyświetlana sekcja **Komentarze**.
  - **Pokaż sekcję "Historia":** Należy wybrać, aby w formularzu szczegółów była wyświetlana sekcja **Historia**.
  - **Pokaż sekcję "Więcej informacji":** Należy wybrać, aby w formularzu szczegółów była wyświetlana sekcja **Więcej informacji**.
  - **Pokaż sekcję "Łączy":** Należy wybrać, aby w formularzu szczegółów była wyświetlana sekcja **Łączy**.
6. Korzystając z karty "Ogólne", zmodyfikować ustawienia, odstępy i inne opcje prezentacji. Domyślnie składnik "szczeóły zadania" rozszerzy się i zostaną wyświetlone wszystkie podane szczeóły. Można wybrać opcję **Ustaw wysokość** i dostosować wysokość do określonej wartości.
7. Korzystając z karty "Styl", sformatować ramkę zawierającą składnik; można użyć predefiniowanych stylów lub swoich własnych wyborów.
8. Za pomocą karty "Łączy" można przypisać czynności do wyzwalaczy udostępnionych przez składnik:
- **Przesłano szczeóły zadania:** Wyzwalacz ten pojawia się, gdy użytkownik naciśnie przycisk **Prześlij**.
  - **Zatwierdzono zadanie:** Wyzwalacz ten pojawia się, gdy użytkownik naciśnie przycisk **Zatwierdź**.
  - **Odrzucono zadanie:** Wyzwalacz ten pojawia się, gdy użytkownik naciśnie przycisk **Odrzuć**.
  - **Zamknięto zadanie:** Wyzwalacz ten pojawia się, gdy użytkownik naciśnie przycisk **Zamknij**.
  - **Zapisano zadanie:** Wyzwalacz ten pojawia się, gdy użytkownik naciśnie przycisk **Zapisz**.
  - **Dodano komentarz do zadania:** Wyzwalacz ten pojawia się, gdy użytkownik doda tekst komentarza, a następnie naciśnie przycisk **Prześlij komentarz**.

Aby przypisać czynność, należy kliknąć na wyzwalaczu.

 **Uwaga:**

Składnik "lista zadań procesu" nie obsługuje ręcznych wyzwalaczy ani czynności; jeśli jednak do strony zostanie dodany składnik "szczeóły zadania", to będzie on automatycznie wyświetlać szczeóły zadania wybranego z listy. Więcej informacji dotyczących wyzwalaczy i czynności jest dostępnych w rozdziale [Używanie wyzwalaczy i czynności](#).

Informacje, jak używać Oracle Process Cloud Service, są dostępne pod hasłem Getting Started with Process.

## Elementy zawartości

Dla użytkowników firmowych każdy serwis ma kolekcję, która zawiera zasoby cyfrowe i elementy zawartości powiązane z serwisem. Składniki "element zawartości" ułatwiają dodawanie elementów do serwisu.

Można przeciągać zasoby cyfrowe i elementy zawartości bezpośrednio z panelu "Zawartość" edytora — zostanie wówczas automatycznie użyty składnik odpowiedni dla obrazu zasobu cyfrowego lub dla elementu zawartości strukturalnej.

Składniki dodatkowe, wybrane z panelu "Zawartość", umożliwiają dynamiczne wyświetlanie elementów zawartości na podstawie jej typu.




- [Składnik "element zawartości"](#)
- [Element zastępczy zawartości](#)
- [Lista zawartości](#)
- [Wyszukiwanie zawartości](#)
- [Rekomendacje](#)

## Składnik "element zawartości"

Będąc użytkownikiem firmowym, można używać składnika "element zawartości", który pomaga — przed dodaniem elementów zawartości — opracować układ strony i skonfigurować interakcje strony.



Jeśli element zawartości zostanie przeciągnięty z panelu "Zawartość" na stronę, to zostanie automatycznie wstawiony składnik "element zawartości" przeznaczony na ten element, chyba że element zawartości jest niestandardowym typem zasobu cyfrowego. Przeciągnięcie niestandardowego zasobu cyfrowego na stronę spowoduje wstawienie zasobu jako składnika "obraz" lub "wideo". Trzeba go usunąć, a następnie najpierw dodać składnik "element zawartości", przeciągnąć niestandardowy zasób cyfrowy lub standardowy element zawartości z panelu "Zawartość" do składnika.

Aby dodać do strony składnik "element zawartości", należy:

1. Przejść do strony, która ma być edytowana, i upewnić się, że przełącznik  jest ustawiony w pozycji **Edytuj**.
2. Dodać składnik do strony.
3. Aby edytować zawartość składnika elementu zawartości, kliknąć na jego ikonie menu , a następnie wybrać opcję **Edytuj**, lub wybrać opcję **Ustawienia** w celu otwarcia panelu "Ustawienia", po czym kliknąć na ikonie "Edytuj" (.

 **Uwaga:**

Jeśli typem zawartości, użytym w składniku "element zawartości", jest niestandardowy zasób cyfrowy, a nie standardowy typ zawartości, nie będzie można edytować typu niestandardowego zasobu cyfrowego. W menu składnika nie będzie dostępna opcja **Edytuj**, a w panelu ustawień nie będzie żadnej ikony umożliwiającej edycję.

4. Wprowadzić zmiany w elemencie zawartości, po czym nacisnąć przycisk **Zapisz**. Element zawartości zostanie zapisany w repozytorium zawartości jako nowa wersja.
5. Aby edytować wygląd składnika, kliknąć na ikonie menu składnika , po czym wybrać opcję **Ustawienia**.
6. Korzystając z karty "Ogólne", zmodyfikować ustawienia, odstępy i inne opcje prezentacji. Jeśli element zawartości jest powiązany ze składnikiem, zostanie wyświetlony widok miniatur elementów zawartości. Jeśli żaden element zawartości nie został jeszcze powiązany, zostanie wyświetlony obraz zastępczy i wiadomość.
7. Wybrać odpowiednie opcje, aby określić dodatkowe ustawienia domyślne wyświetlanej zawartości.
  - **Używana wersja:** Jeśli zostanie wybrana opcja **Użyj najnowszej wersji zasobu** i istnieje nowsza, nieopublikowana wersja elementu zawartości, to zostanie ona automatycznie opublikowana po opublikowaniu bieżącej aktualizacji serwisu. Jeśli nie zostało to jednoznacznie określone, wszystkie elementy będą wersjami roboczymi lub najnowszymi.
  - **Widok elementu:** Wybrać układ używany do wyświetlania elementu zawartości. Udostępniany układ **Domyślny** pokazuje wszystkie pola elementu zawartości. Jeśli dla elementu zawartości istnieją inne układy niestandardowe, można wybrać dowolny dostępny układ. Jeśli zostanie wybrany układ niestandardowy, dla którego jest włączona opcja **Dodaj obsługę ustawień niestandardowych, gdy są one używane w serwisach**, można dodać dane niestandardowe do układu, klikając na ikonie , następnie wprowadzając właściwe dane do pola **Dane niestandardowe**, po czym wracając do karty "Ogólne" za pomocą przycisku **Wstecz**.
  - **Strona do wyświetlania indywidualnego elementu:** Jeśli na strony szczegółów zostało wyznaczonych kilka stron (lub jedna), to są one tutaj wyświetlane. Wybrać stronę, na której — gdy użytkownik kliknie na łączy dla elementu zawartości, służącym do wyświetlania szczegółowych informacji — zostaną wyświetlone szczegółowe informacje.

 **Uwaga:**

Jeśli strona szczegółów nie zostanie utworzona, to (w układzie domyślnym) łączy do wyświetlania szczegółów elementu zawartości nie będzie pokazywane.



8. Korzystając z karty "Styl", sformatować ramkę zawierającą składnik; można użyć predefiniowanych stylów lub swoich własnych wyborów.

## Element zastępczy zawartości

Będąc użytkownikiem firmowym, można używać składnika "element zastępczy zawartości", który dynamicznie wyświetla elementy zawartości jednego lub większej liczby typów.

Na przykład, składnika "element zastępczy zawartości" można użyć na stronie szczegółów, a gdy użytkownik kliknie na łączu, aby uzyskać bardziej szczegółowe informacje o danym elemencie zawartości, zostanie automatycznie załadowany widok szczegółów powiązanego elementu zawartości. Na przykład, jeśli strona zawiera kilka artykułów, każdy z nagłówkiem i obrazem, to gdy użytkownik kliknie na konkretnym artykule, na stronie szczegółów zostanie wyświetlony pełny artykuł.

Aby dodać do strony składnik "element zastępczy zawartości", należy:

1. Przejść do strony, która ma być edytowana, i upewnić się, że przełącznik  jest ustawiony w pozycji **Edytuj**.
2. Dodać składnik do strony.
3. Aby edytować składnik i jego wygląd, kliknąć na ikonie menu składnika , po czym wybrać opcję **Ustawienia**.
4. Korzystając z karty "Ogólne", zmodyfikować ustawienia, odstępy i inne opcje prezentacji.
5. Wybrać odpowiednie opcje, aby określić dodatkowe ustawienia domyślne wyświetlanej zawartości.
  - **Typ zawartości:** Wybrać jeden lub większą liczbę dostępnych typów zawartości. Typami zawartości są typy elementów zawartości dostępnych w kolekcji serwisu. Obejmują niestandardowe typy zasobów cyfrowych.
  - **Widok elementu:** Wybrać układ używany do wyświetlania elementu zawartości. Udostępniany układ **Domyślny** pokazuje wszystkie pola elementu zawartości. Jeśli dla elementu zawartości istnieją inne układy niestandardowe, można wybrać dowolny dostępny układ.
  - **Strona do wyświetlania indywidualnego elementu:** Jeśli na stronie szczegółów zostało wyznaczonych kilka stron (lub jedna), to są one tutaj wyświetlane. Wybrać stronę, na której — gdy użytkownik kliknie na łączu dla elementu zawartości, służącym do wyświetlania szczegółowych informacji — zostaną wyświetlone szczegółowe informacje.
6. Korzystając z karty "Styl", sformatować ramkę zawierającą składnik; można użyć predefiniowanych stylów lub swoich własnych wyborów.

### Uwaga:



Jeśli strona szczegółów nie zostanie utworzona, to (w układzie domyślnym) łącze do wyświetlania szczegółów elementu zawartości nie będzie pokazywane.

## Lista zawartości

Będąc użytkownikiem firmowym, można używać składnika "lista zawartości", który dynamicznie wyświetla elementy zawartości konkretnego typu.

Na przykład, składnika "lista elementów zawartości" można użyć na stronie szczegółów, a gdy użytkownik kliknie na łączu, aby uzyskać bardziej szczegółowe informacje o danym elemencie zawartości, zostanie automatycznie załadowany widok szczegółów powiązanego elementu zawartości. Na przykład, jeśli strona zawiera kilka artykułów, każdy z nagłówkiem i obrazem, to gdy użytkownik kliknie na konkretnym artykule, na stronie szczegółów zostanie wyświetlony pełny artykuł.

Aby dodać składnik "lista zawartości" do strony, należy:

1. Przejść do strony, która ma być edytowana, i upewnić się, że przełącznik  jest ustawiony w pozycji **Edytuj**.
2. Dodać składnik do strony.
3. Aby edytować składnik i jego wygląd, kliknąć na ikonie menu składnika , po czym wybrać opcję **Ustawienia**.
4. Korzystając z karty "Ogólne", zmodyfikować ustawienia, odstępy i inne opcje prezentacji.
5. Wybrać odpowiednie opcje, aby określić dodatkowe ustawienia domyślne wyświetlanej zawartości.
  - **Typ zawartości:** Wybrać jeden z dostępnych typów zawartości. Typami zawartości są typy elementów zawartości dostępnych w kolekcji serwisu. Obejmują niestandardowe typy zasobów cyfrowych.
  - **Maksymalna liczba elementów i Zaczynj od elementu:** Określić maksymalną liczbę wyświetlanych elementów oraz określić początek wyświetlania. Dodatkowe elementy nie będą wyświetlane.
  - **Podział na strony:** Określić, czy do listy mają zostać dołączone opcje podziału strony. Jeśli opcja ta zostanie wybrana, można będzie dostosować sposób wyświetlania podziału stron z użyciem przycisków lub numerów stron oraz różnych etykiet.
  - **Data:** Użyć dostępnych opcji w celu wybrania elementów zawartości do wyświetlenia (na podstawie daty utworzenia elementu, przypadającej przed, po, między lub w wybranym przedziale dat).
  - **Kategorie:** Wybrać kategorie do przefiltrowania listy elementów zawartości, tak aby były wyświetlane tylko elementy z konkretnej kategorii lub poszczególnych kategorii. Aby otworzyć wysuwany panel, za pomocą którego będzie można wybrać kategorie z repozytorium serwisów, nacisnąć przycisk **Wybierz kategorie**. Jeśli zostanie wybrana więcej niż jedna kategoria w tej samej taksonomii, to tylko jedna z wybranych kategorii musi być przypisana do wyświetlanych elementów. Jeśli zostanie wybrana więcej niż jedna kategoria z różnych taksonomii, to wszystkie wybrane kategorie muszą być przypisane do wyświetlanych elementów. Na przykład, jeśli zostaną wybrane kategorie **Miasta** i **Parki** z taksonomii **Miejsce docelowe**, zostaną wyświetlone elementy, które są miastami lub parkami. Jeśli zostanie wybrana kategoria **Miasta** z taksonomii **Miejsce docelowe** i kategoria **Europa** z taksonomii **Regiony**, zostaną wyświetlone elementy, które są miastami w Europie.




Domyślnie wszystkie podrzędne wybranej kategorii będą dostępne. Aby ograniczyć listę do elementów w określonym węźle kategorii, należy wybrać węzeł, a następnie wyczyścić pole wyboru **Uwzględniaj kategorie podrzędne**.

- **Język:** Wybrać język spośród powiązanych z repozytorium.
- **Dodatkowy napis-zapytanie** (opcjonalne): Określić dodatkowe parametry zapytania w celu uszczegółowienia listy wyświetlanych elementów, używając następującej składni: `field.dept eq "Finance"`. W poniższej tabeli przedstawiono listę dostępnych operatorów.
- **Uporządkuj wg:** Posortować elementy według nazwy lub daty w porządku rosnącym lub malejącym. Jeśli typ zawartości zawiera inne pola daty, liczbowe lub dziesiętne, można także sortować według tych pól.

Można również wybrać opcję **Niestandardowe**, po czym podać wyrażenie niestandardowe. Na przykład, aby posortować wg numeru działu w porządku rosnącym, należy wpisać: `fields.deptno:asc`. Do sortowania można też zdefiniować parametr adresu URL. Na przykład można wpisać `{{URLParams.sortBy}}`, po czym dodać na końcu adresu URL serwisu następujący parametr: `?sortBy=fields.deptno:asc`.

- **Widok elementu:** Wybrać układ używany do wyświetlania elementu zawartości. Udostępniany układ **Domyślny** pokazuje wszystkie pola elementu zawartości. Jeśli dla elementu zawartości istnieją inne układy niestandardowe, można wybrać dowolny dostępny układ. Jeśli zostanie wybrany układ niestandardowy, dla którego jest włączona opcja **Dodaj obsługę ustawień niestandardowych, gdy są one używane w serwisach**, można dodać dane

niestandardowe do układu, klikając na ikonie , następnie wprowadzając właściwe dane do pola **Dane niestandardowe**, po czym wracając do karty "Ogólne" za pomocą przycisku **Wstecz**.

- **Strona do wyświetlania indywidualnego elementu:** Jeśli na stronie szczegółów zostało wyznaczonych kilka stron (lub jedna), to są one tutaj wyświetlane. Wybrać stronę, na której — gdy użytkownik kliknie na łączu dla elementu zawartości, służącym do wyświetlania szczegółowych informacji — zostaną wyświetlone szczegółowe informacje.

#### Uwaga:

Jeśli strona szczegółów nie zostanie utworzona, to (w układzie domyślnym) łącze do wyświetlania szczegółów elementu zawartości nie będzie pokazywane.

- **Widok listy:** Wybrać układ sekcji do rozmieszczenia elementów. Ta lista może zawierać układy niestandardowe. Omówiono tu następujące układy:
  - **Poziomy:** Rozmieszcza elementy jeden po drugim, w poziomie.
  - **Pionowy:** Rozmieszcza elementy jeden po drugim, w pionie. Jeśli nie został wybrany żaden układ, jest to ustawienie domyślne.
  - **Dwie kolumny:** Rozmieszcza po dwa elementy w wielu rzędach.
  - **Trzy kolumny:** Rozmieszcza po trzy elementy w wielu rzędach.

Aby zmodyfikować układu, kliknąć na strzałce skierowanej w prawo obok wybranego elementu.

- **Widok pustej listy:** Wybrać, w jaki sposób ma być wyświetlana pusta lista.
  - **Opcje:** Określić, czy ma być używane automatyczne odświeżanie zapytania wyszukiwania w składniku "wyszukiwanie zawartości".
  - **Odstęp:** Określić, w jaki sposób będą wyświetlane elementy.
6. Korzystając z karty "Styl", sformatować ramkę zawierającą składnik; można użyć predefiniowanych stylów lub swoich własnych wyborów.

**Tabela 10-1 Operatory zapytań**

Operator	Przykład	Obsługiwane typy danych	Opis
eq	?q=name eq "John" ?q=type eq "DigitalAsset" ?q=type eq "Employee" and fields.DOB eq "1994/09/26T16:23:45.208" ?q=type eq "Employee" and fields.DOB eq "1994/09/26T16:23:45.208"	tekstowe, odwołanie, liczbowe, liczba dziesiętna, wartość logiczna, data-godzina	Operator "równa się" (eq) uzgadnia dokładną wartość podaną w zapytaniu. Operator ten nie ma zastosowania dla wielowartościowych typów danych. W wartości podanej przy użyciu tego operatora nie jest uwzględniana wielkość liter, z wyjątkiem pól standardowych. Operator ten uwzględnia w wartości nawet znaki specjalne.

Tabela 10-1 (Kont.) Operatory zapytań

Operat or	Przykład	Obsługiwane typy danych	Opis
co	?q=(type eq "Employee" AND name co "john alex") ?q=(type eq "Car" AND fields.features co "manual")	tekstowe, odwołanie, liczbowe, liczba dziesiętna, data-godzina, duży tekst	Operator "zawiera" (co) uzgadnia każde słowo podane w kryterium. Słowa te są tworzone poprzez podzielenie wartości według znaków specjalnych. Daje to wyniki, które zawierają przynajmniej jeden z wyrazów (w tym przykładzie będzie to "john" lub "alex" lub oba te wyrazy). Podczas wyszukiwania operator ten nie uwzględnia w wartości znaków specjalnych. Nie przeprowadza wyszukiwania dla wyrazów niedozwolonych. Więcej informacji o wyrazach niedozwolonych można znaleźć w dokumentacji "Apache Lucene". Operator ten jest stosowany w przypadku atrybutów jednowartościowych <i>tekstowych i w formacie d</i> ; dla atrybutów wielowartościowych ma zastosowanie dla typu danych: <i>tekstowe, odwołanie, liczbowe, liczba dziesiętna, data-godzina, duży tekst</i> . Objasnienie możliwych formatów daty-godziny zawarto w poniższej tabeli <i>Obsługiwane formaty daty/daty-godziny (24 godz.)</i> . W wartości podanej przy użyciu tego operatora nie jest uwzględniana wielkość liter.
sw	?q=type eq "Employee" AND name sw "Joh" ?q=type eq "Employee" AND fields.city sw "Los"	tekstowe	Operator "zaczyna się od" (sw) uzgadnia tylko początkowe wartości znaku podane w warunku pola. Operator ten nie ma zastosowanie dla wielowartościowych typów danych. W wartości podanej przy użyciu tego operatora nie jest uwzględniana wielkość liter.

Tabela 10-1 (Kont.) Operatory zapytań

Operator	Przykład	Obsługiwane typy danych	Opis
ge	?q=(type eq "Employee" AND fields.age ge "40") ?q=type eq "DigitalAsset" AND updatedDate ge "20171026"	liczbowe, liczba dziesiętna, data-godzina	Operator "większe niż lub równe" (ge) uzgadnia tylko wartości liczbowe oraz datę-godzinę. Objasnienie możliwych formatów daty-godziny zawarto w poniższej tabeli <i>Obsługiwane formaty daty/daty-godziny (24 godz.)</i> . Operator ten nie ma zastosowanie dla wielowartościowych typów danych.
le	?q=(type eq "Employee" AND fields.weight le "60.6")	liczbowe, liczba dziesiętna, data-godzina	Operator "mniejsze niż lub równe" (le) uzgadnia tylko wartości liczbowe oraz datę-godzinę. Objasnienie możliwych formatów daty-godziny zawarto w poniższej tabeli <i>Obsługiwane formaty daty/daty-godziny (24 godz.)</i> . Operator ten nie ma zastosowanie dla wielowartościowych typów danych.
gt	?q=(type eq "Employee" AND fields.age gt "20")	liczbowe, liczba dziesiętna, data-godzina	Operator "większe niż" (gt) uzgadnia tylko wartości liczbowe oraz datę-godzinę. Objasnienie możliwych formatów daty-godziny zawarto w poniższej tabeli <i>Obsługiwane formaty daty/daty-godziny (24 godz.)</i> . Objasnienie możliwych formatów daty-godziny zawarto w poniższej tabeli <i>Obsługiwane formaty daty/daty-godziny (24 godz.)</i> . Operator ten nie ma zastosowanie dla wielowartościowych typów danych.
lt	?q=(type eq "Employee" AND fields.age lt "20") ?q=type eq "Employee" AND createdDate lt "1994/09/26T16:23:45.208"	liczbowe, liczba dziesiętna, data-godzina	Operator "mniejsze niż" (lt) uzgadnia tylko wartości liczbowe oraz datę-godzinę. Objasnienie możliwych formatów daty-godziny zawarto w sekcji <b>Obsługiwane formaty daty/daty-godziny (24 godz.)</b> . Operator ten nie ma zastosowanie dla wielowartościowych typów danych.

Tabela 10-1 (Kont.) Operatory zapytań

Operator	Przykład	Obsługiwane typy danych	Opis
mt	<pre>?q=(type eq "Car" AND fields.review mt "petrol 20KMPL") ?q=(type eq "Employee" AND name mt "Jo?n") ?q=(type eq "Employee" AND name mt "Jo*") ?q=(type eq "Employee" AND fields.role mt "senior*")</pre>	tekstowe, duży tekst	<p>Operator zapytania frazowego lub wyszukiwania wg sąsiedztwa (mt) umożliwia wyszukiwanie wyrazów, które dzieli pewna odległość odległości. Wyniki są sortowane według najlepszego uzgodnienia. Jest przydatny do wyszukiwania elementów zawartości, gdy wartości, podane w kryterium "petrol 20kmpl", mają zostać użyte do wyszukiwania faktycznej zawartości typu "petrol fuel mileage runs 20KMPL in the speed way".</p> <p>W operatorze zgodności można używać wieloznaczników — w obrębie jednej wartości jest obsługiwany wieloznacznik jednoznakowy (reprezentujący zarówno jeden znak), jak i wieloznakowy (reprezentujący więcej niż jeden znak). Wieloznacznikiem jednoznakowym jest "?" (znak zapytania), a wieloznakowym — "*" (gwiazdka). Na przykład imiona "John" i "Joan" można wyszukiwać jako "Jo?n", używając wieloznacznika jednoznakowego, lub jako "Jo*", używając wieloznacznika wieloznakowego.</p> <p>Operator ten ma zastosowanie dla jedno- i wielowartościowych typów danych. Nie przeprowadza wyszukiwania dla wyrazów niedozwolonych. Więcej informacji o wyrazach niedozwolonych można znaleźć w dokumentacji "Apache Lucene". W wartości podanej przy użyciu tego operatora nie jest uwzględniana wielkość liter.</p>

Tabela 10-1 (Kont.) Operatory zapytań



Operator	Przykład	Obsługiwane typy danych	Opis
sm	<code>?q=(type eq "Employee" And fields.city sm "Rome")</code>	tekstowe, duży tekst	Operator podobieństwa. Umożliwia wyszukiwanie wartości podobnych do podanego kryterium. Wyszukiwanie to jest nazywane również wyszukiwaniem rozmytym (ang. fuzzy search), które — w celu uzgodnienia wyniku — używa domyślnie maksymalnie dwóch edycji. Angielskie słowo "Rome" jest zbliżone do słowa "Dome". Operator ten ma zastosowanie dla jedno- i wielowartościowych typów danych. W wartości podanej przy użyciu tego operatora nie jest uwzględniana wielkość liter.
AND	<code>?q=(type eq "Employee" AND name eq "John" AND fields.age ge "40")</code>	ND	Operatorem AND można używać do wstawiania warunku AND między różnymi warunkami zapytania. Operator ten ma pierwszeństwo przed operatorem OR.
OR	<code>type eq "Employee" AND name eq "John" OR fields.age ge "40"</code>	ND	Operatorem OR można używać do wstawiania warunku OR między różnymi warunkami zapytania.
()	<code>?q=type eq "Employee" AND (name eq "John" AND fields.age ge "40") ?q=type eq "Employee" AND ( (name eq "John" AND fields.age ge "40") OR fields.weight ge 60)</code>	ND	Nawiasy — obejmując operatory — służą do grupowania warunków w kryteria. Kryterium ujęte w nawias ma najwyższy priorytet; następnym w kolejności wyznacza operator AND, a po nim operator OR.

## Wyszukiwanie zawartości

Będąc użytkownikiem firmowym, można używać składnika "wyszukiwanie zawartości" oraz określać czynności zwracane przez wyszukiwanie.

Można wstawiać dostosowany pasek wyszukiwania, aby zmienić lub odświeżyć zawartość wyświetlaną na stronie albo wybrać inną czynność, taką jak otwieranie strony wyników wyszukiwania czy wyświetlanie alarmów.

Aby dodać do strony składnik "wyszukiwanie zawartości", należy:

1. Przejść do strony, która ma być edytowana, i upewnić się, że przełącznik  jest ustawiony w pozycji **Edytuj**.
2. Dodać do strony składnik "wyszukiwanie zawartości".
3. Aby edytować składnik i jego wygląd, kliknąć na ikonie menu składnika , po czym wybrać opcję **Ustawienia**.
4. Za pomocą karty "Ogólne" zmodyfikować tekst elementu zastępczego, odstępy, wyrównanie i inne opcje prezentacji.
5. Za pomocą karty "Styl" sformatować ramkę, zawierającą składnik z predefiniowanymi stylami lub z własnymi niestandardowymi ustawieniami czcionki, obramowania, koloru tła itd.
6. Jeśli składnik "wyszukiwanie" ma być używany do odświeżania danych na bieżącej stronie, dodać na stronie składnik "lista zawartości". Przejść do **ustawień** i wybrać typ zawartości oraz inne informacje dotyczące zapytania, takie jak liczba wyświetlanych elementów, czy dzielić wyniki na strony lub czy uruchamiać skrypt Lazy Load przy przewijaniu (ładowanie zawartości w momencie przewijania strony). Można teraz przejść do trybu "Widok" i wypróbować funkcję wyszukiwania.
7. Jeśli składnik "wyszukiwanie" ma być używany do odświeżania danych na bieżącej stronie, a na stronie jest więcej niż jeden składnik "lista zawartości", należy wyłączyć funkcję automatycznego kierowania zapytań we wszystkich listach zawartości, z wyjątkiem tej listy, na której mają być wyświetlane wyniki. Przejść do ustawień "Lista zawartości", po czym wyczyścić pole wyboru **Automatyczne odświeżanie z** .
8. Jeśli na stronie ma się znajdować więcej niż jeden składnik "wyszukiwanie zawartości" (z każdym składnikiem "wyszukiwanie" kierującym określoną listę zawartości dla wyników), trzeba wyczyścić pole wyboru wszystkich opcji automatycznego odświeżania w składnikach "lista zawartości", po czym — w celu powiązania składnika "wyszukiwanie zawartości" z odpowiednią listą zawartości — użyć funkcji "Wyzwalacze/Czynności".
9. W przypadku składnika "wyszukiwanie", użyć karty "Łącze" do powiązania czynności ze składnikiem. Wybrać opcję z rozwijanej listy "Wybór typu łącza". Aby wyświetlić dostępne wyzwalacze lub utworzyć nowy, wybrać kolejno opcje **Czynności wyzwalacza** i **Przy zapytaniu wyszukiwania**. Znaleźć listę zawartości, która ma być przeszukiwana, rozwinąć ją, po czym przeciągnąć na listę czynności składnik **szukaj w zawartości**. W obszarze **Wyszukiwanie** wybrać opcję **Szukany napis** . Wybrana lista zawartości jest zaznaczona na liście **Wykonaj tę czynność w** .
10. Składnika "wyszukiwanie zawartości" można także użyć do przekazywania zapytania na stronę "Wyniki wyszukiwania". Na karcie "Łącze" składnika "wyszukiwanie", wybrać — aby wyświetlić stronę "Wyniki wyszukiwania" — opcję **Strona wyszukiwania** . Można użyć strony domyślnej lub łącza prowadzącego do utworzonej strony wyników wyszukiwania. Z rozwijanej listy "Strona" wybrać stronę, a następnie czynności wyświetlania strony. Można dostosować wyniki do określonego typu zawartości, utworzyć wyniki w nowym oknie itd.

Przy projektowaniu efektywnej strony "Wyniki wyszukiwania" mogą być pomocne następujące wskazówki:

- Utworzyć stronę i wyznaczyć ją jako stronę wyszukiwania. Domyślnie jest ona oznaczona jako ukryta, lecz ustawienie to można zmienić w jej właściwościach.



- Edytować stronę oraz dodać do niej listę zawartości. Ustawienia listy zawartości można edytować, określając typ zawartości. Typ zawartości można także określić w ustawieniach łącza składnika "wyszukiwanie" (powyżej). Jeśli zostały wybrane ustawienia łącza, to można użyć strony "Wyniki wyszukiwania", która — w zależności od składnika używanego do zainicjowania wyszukiwania — może wyświetlać wyniki z różnych typów zawartości.
- Aby edytować sposób wyświetlania, zmienić ustawienia, takie jak wybór paginacji dla skryptu Lazy Load (ponieważ ta strona prawdopodobnie będzie używana wyłącznie dla wyników wyszukiwania).
- Na stronie "Wyniki wyszukiwania" można również umieścić składnik "wyszukiwanie zawartości". Spowoduje to powtórzenie szukanego napisu, użytego do uruchomienia strony, umożliwiając użytkownikowi (jeśli jest to konieczne) uszczegółowienie wyszukiwania.

## Rekomendacje

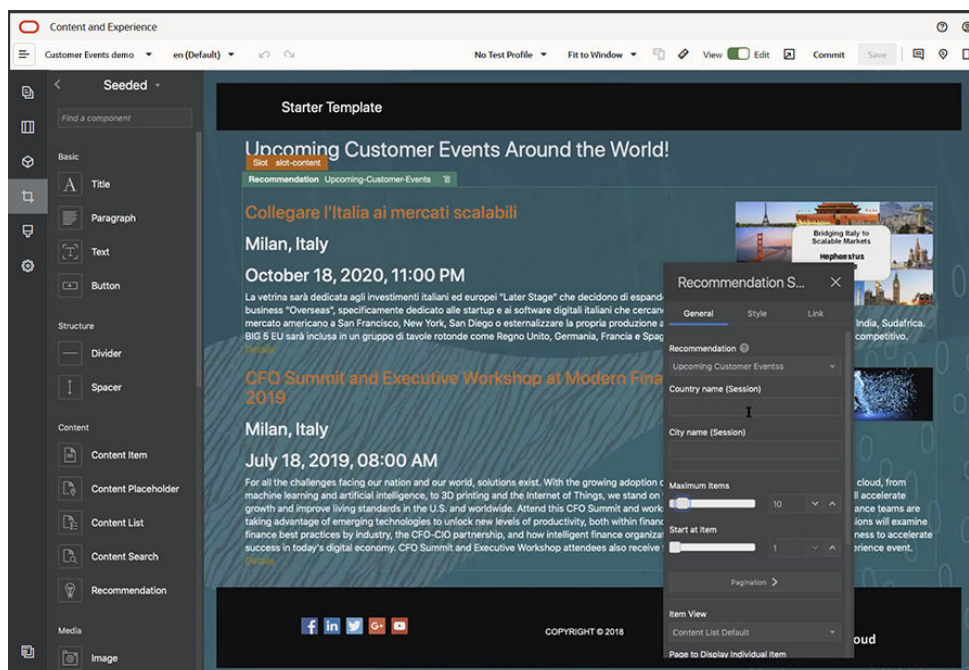
Będąc użytkownikiem firmowym, można korzystać ze składnika **rekomendacja** w celu zapewnienia osobom, które odwiedzają serwis internetowy, spersonalizowaną obsługę, wyświetlając zasoby na podstawie lokalizacji lub obszarów zainteresowania. Współtwórca repozytorium — po utworzeniu rekomendacji — definiuje zbiór reguł, za pomocą którego są wyszukiwane zasoby zgodne z atrybutami odbiorców, takimi jak informacje o geolokalizacji odwiedzającego serwis. Na przykład odwiedzający z europejskimi adresami IP mogą najpierw na stronie startowej serwisu widzieć ogłoszenia o wydarzeniach w Europie, zaś odwiedzający z północnoamerykańskimi adresami IP — o wydarzeniach w Stanach Zjednoczonych i Kanadzie.

Rekomendacje, podobnie jak zasoby cyfrowe, są powiązane z repozytorium; mogą być edytowane (i przekazywane w procesie Workflow do sprawdzenia i opublikowania) przez dowolną osobę, która jest współtwórcą zawartości repozytorium. Nawet nie mając uprawnień współtwórcy, można wyświetlać i testować rekomendacje, aby — przed użyciem ich w serwisie lub w architekturze "head-less" — zobaczyć, jak będą działać.

Aby dodać rekomendację do strony, należy:

1. Z menu bocznego wybrać opcję **Serwisy**, następnie wybrać serwis, do którego ma zostać dodana rekomendacja, po czym wybrać z menu podręcznego opcję **Otwórz** albo kliknąć na ikonie  (na pasku czynności).
2. Przełączyć serwis do trybu **Edytuj**, po czym wybrać aktualizację, która ma zostać użyta, lub utworzyć nową.
3. Z bocznego menu wybrać opcję **Składniki**.
4. Otworzyć menu **Standardowe**.
5. W sekcji "Zawartość" kliknąć na składniku **rekomendacja**, po czym przeciągnąć go na stronę.
6. Z menu rekomendacji () wybrać opcję **Ustawienia**.





7. Na karcie **Ogólne**, w ustawieniach rekomendacji, wybrać rekomendację, która ma zostać użyta. Zostanie odświeżona strona oraz zostaną wyświetlone zalecane zasoby.
8. Opcjonalnie podać wartości domyślne atrybutów (atrybutów odbiorców), używanych przez tę rekomendację, klikając na strzałkę obok wybranej rekomendacji. Po ukończeniu dodawania wartości nacisnąć przycisk **Wstecz**.
9. Wprowadzić zmiany we wszelkich dodatkowych właściwościach (na przykład: która wersja ma zostać użyta, maksymalna liczba wyświetlanych elementów, domyślny układ zawartości). Po ukończeniu zamknąć ustawienia rekomendacji, po czym nacisnąć przycisk **Zapisz**.

#### Uwaga:

Jeśli zostały utworzone profile testowe z predefiniowanymi wartościami atrybutów odbiorców, należy wybrać profil z menu profili (na pasku menu) w celu uzyskania podglądu, w jaki sposób rekomendacja odpowiada na te dane wejściowe. Wartości profilu testowego zastępują wartości domyślne, które zostały ustawione w ustawieniach składnika.

10. Jeśli rekomendacja używa w swoich regułach bieżącej daty (daty systemowej), a ma zostać przetestowana z użyciem innej daty, wybrać z menu profilu testowego opcję **Data systemowa**, po czym wybrać odpowiednią datę.

## Inne składniki

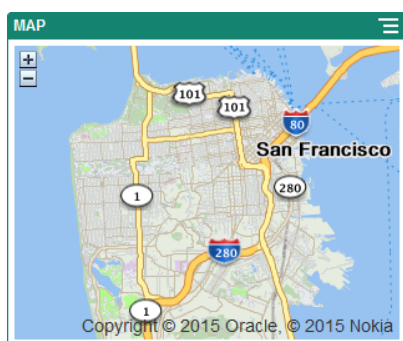
Przedstawiono tu sposoby używania niektórych składników, łączących różne typy zawartości.

- [Mapy](#)
- [Nagłówki](#)



- [Artykuły](#)
- [Obrazy z tekstem](#)
- [Grupy składników](#)
- [Korzystanie z usługi Cobrowse na stronie](#)
- [Oracle Intelligent Advisor](#)
- [Oracle Visual Builder](#)

## Mapy

Do swojego serwisu można dodać mapę pozwalającą użytkownikom interaktywnie eksplorować obszar wokół określonego miejsca.



Aby dodać mapę do strony, należy:

1. Przejść do strony, która ma być edytowana, i upewnić się, że przełącznik  jest ustawiony w pozycji **Edytuj**.
2. Dodać składnik do strony. Składnik "mapa Oracle" pokazuje lokalizację domyślną.
3. Aby zmienić lokalizację przedstawianą przez mapę i dostosować właściwości wyświetlania, kliknąć na jej ikonie menu , po czym wybrać opcję **Ustawienia**.
4. Wpisać w polu **Lokalizacja** początkową lokalizację dla mapy. Można użyć adresu, kodu pocztowego lub — rozdzielonej przecinkiem — szerokości i długości geograficznej (na przykład 40.5,-57.6).





Jeśli jest używany adres, upewnić się, że podane informacje odpowiadają jednej lokalizacji. Jeśli adres będzie pasował do więcej niż jednej lokalizacji, mapa pozostanie pusta.


5. W polu **Powiększenie** określić początkowy poziom powiększenia mapy. Opcjonalnie można zezwolić użytkownikom na zmienianie powiększenia za pomocą myszy, gładzika lub ekranowych elementów sterujących.
6. Określić opcje stylu, wyrównania i wyświetlania:
  - **Powiększenie:** Włączanie/wyłączanie opcji ekranowych i sprzętowych pozwalających użytkownikom dostosowywać powiększenie mapy.
  - **Przesuwanie:** Włączanie/wyłączanie opcji ekranowych i sprzętowych pozwalających użytkownikom zmieniać obszar prezentowany na mapie.
  - **Wyświetlany znacznik:** Oznaczenie początkowej lokalizacji ikoną przypięcia.










- **Skala:** Dołączenie wskaźnika pokazującego skalę bieżącego poziomu powiększenia.
- **Przeгляд:** Dołączanie wstawkowej mapy pokazującej bieżący widok w szerszym kontekście.

## Nagłówki

Składnik "nagłówek" dodaje się, aby zwrócić uwagę na określone miejsce w serwisie z użyciem tytułu oraz wspomagającego obrazu i akapitu.


1. Przejść do strony, która ma być edytowana, i upewnić się, że przełącznik  jest ustawiony w pozycji **Edytuj**.
2. Dodać składnik do strony.
3. Kliknąć w różnych obszarach składnika, aby dodać odpowiednie teksty. Dodaje się je do głównego nagłówka oraz do znajdującego się pod nim akapitu. Tekst przyjmuje formatowanie zgodne ze stylem domyślnym składnika. Aby wprowadzać dodatkowe wiersze tekstu, należy naciskać klawisz Enter.
4. Aby zmienić formatowanie domyślne dowolnego fragmentu, zaznaczyć tekst, który ma zostać sformatowany, po czym użyć dowolnych opcji z paska formatowania, np. określających czcionkę, kolor lub wyrównanie.
5. Aby usunąć formatowanie zastosowane przy użyciu tych opcji, zaznaczyć tekst, po czym kliknąć na ikonie .
6. Aby edytować składnik i jego wygląd, kliknąć na ikonie menu składnika , po czym wybrać opcję **Ustawienia**.
7. Wybrać **Składniki**, aby określić szczegóły wyświetlanej zawartości.
  - **Obraz:**
    - Nacisnąć przycisk **Wybierz**, aby użyć tego obrazu w nagłówku. Wybrać obraz z repozytorium serwisów lub z folderu "Dokumenty", który został udostępniony bieżącemu użytkownikowi, bądź wysłać obraz do folderu "Dokumenty". Odszukać i wybrać obraz, który ma zostać użyty. Jeśli zasoby cyfrowe nie są widoczne, kliknąć na ikonie , po czym zmienić filtrowanie kolekcji na **Wszystkie**. Jeśli wybrano obraz z folderu "Dokumenty", plik obrazu można powiązać (tworząc łącze), zamiast kopiować go do serwisu. Aby utworzyć łącze do pliku, wybrać opcję **Zamiast kopiować plik do** . Jeśli ta opcja nie zostanie wybrana, w serwisie będzie składowana kopia pliku, do której będą się odnosiły odwołania z serwisu. Tworząc łącze do oryginalnego pliku, unika się duplikowania zawartości. Łącze pozwala odwiedzającym serwis oglądać zawartość, nawet jeśli ulegną zmianie uprawnienia do pliku lub jego wyświetlanie zostanie ograniczone w inny sposób. Nacisnąć przycisk **OK**. Jeśli wybrano zasób cyfrowy, można wybrać określoną realizację. Jeśli realizacja nie zostanie wybrana, będzie stosowany rozmiar oryginalny. Chcąc opublikować najnowszą wersję zasobu (gdy serwis zostanie opublikowany), wybrać opcję **Użyj najnowszej wersji zasobu**.



Aby edytować obraz, kliknąć na ikonie , po czym edytować obraz w dowolny z następujących sposobów:

- \* Aby przyciąć obraz, kliknąć na ikonie **Przytnij** (). Z paska narzędzi przycinania wybrać jedną z predefiniowanych proporcji obrazu albo przeciągnąć na obraz uchwyty do przycinania. Gdy wynik będzie satysfakcjonujący, nacisnąć na pasku narzędzi przycinania przycisk **Przytnij**.
  - \* Aby obrócić lub przerzucić obraz, kliknąć na ikonie **Obróć** (). Na pasku narzędzi obracania podać niestandardową wartość stopni obrotu bądź użyć przycisków do obracania obrazu w lewo lub w prawo albo wybrać, czy obraz ma zostać przerzucony w poziomie czy w pionie.
  - \* Aby dodać znak wodny do obrazu, kliknąć na ikonie **Znak wodny** (). Dodać tekst do obrazu; korzystając z narzędzi znaku wodnego, można odpowiednio zmienić rozmiar tekstu, styl, kolor i przezroczystość.
  - \* Aby zmienić format obrazu, kliknąć na ikonie **Opcje** (), po czym wybrać z rozwijanej listy **Format** nowy format.
  - \* Aby zmienić kolor tła, kliknąć na ikonie **Opcje** (), po czym wybrać z rozwijanego menu **Kolor tła** odpowiednią opcję.
  - \* Jeśli jest edytowany obraz w formacie .jpg lub .webp (dostępne w przeglądarce Google Chrome), można zmienić jakość obrazu w celu utworzenia pliku o mniejszym rozmiarze. Kliknąć na ikonie **Opcje** (), po czym w obszarze **Jakość** podać nową wartość procentową.
  - \* Aby cofnąć lub ponowić zmianę, kliknąć na ikonie  lub . Aby usunąć wszelkie dokonane zmiany, nacisnąć przycisk **Resetuj**.
  - \* Aby zmienić powiększenie obrazu, użyć elementów sterujących powiększaniem ().
- **Tytuł:** Wpisać tekst, który ma być wyświetlany w etykietce.
  - **Tekst alternatywny:** Wpisać tekst alternatywny, który będzie wyświetlany w celu ułatwienia dostępu.
  - **Podpis:** Wpisać tekst podpisu, który będzie wyświetlany pod obrazem.
  - **Wyrównanie, Szerokość, Odstęp:** Jeśli trzeba, zmienić układ obrazu.
- **Tytuł:** Zmienić odstępy dla tytułu nagłówka.
  - **Akapit:** Zmienić odstępy w tekście, który będzie wyświetlany pod nagłówkiem.
8. Korzystając z karty "Ogólne", zmodyfikować ustawienia, odstępy i inne opcje prezentacji.
  9. Korzystając z karty "Styl", dodać formatowanie tekstu oraz dostosować kolor tła, czcionki i krawędzie.

## Artykuły


Do serwisu można dodać składnik "artykuł" łączący w sobie składniki "nagłówek", "akapit" i "obraz".


1. Przejść do strony, która ma być edytowana, i upewnić się, że przełącznik  jest ustawiony w pozycji **Edytuj**.



2. Dodać składnik do strony.
3. Klikać w różnych obszarach składnika, aby dodać odpowiednie teksty. Można dodać zawartość nagłówka artykułu oraz znajdującego się poniżej tytułu niższego poziomu. Następnie można dodać pod nagłówkiem niższego poziomu tekst artykułu. Cały tekst przyjmie formatowanie domyślnego stylu składnika. Aby wprowadzać dodatkowe wiersze tekstu, należy naciskać klawisz Enter.
4. Aby zmienić formatowanie domyślne dowolnego fragmentu, zaznaczyć tekst, który ma zostać sformatowany, po czym użyć dowolnych opcji z paska formatowania, np. określających czcionkę, kolor lub wyrównanie.
5. Aby usunąć formatowanie zastosowane przy użyciu tych opcji, zaznaczyć tekst, po czym kliknąć na ikonie .
6. Aby edytować składnik i jego wygląd, kliknąć na ikonie menu składnika , po czym wybrać opcję **Ustawienia**.
7. Wybrać **Składniki**, aby określić szczegóły wyświetlanej zawartości.








- **Obraz:**

- Nacisnąć przycisk **Wybierz**, aby użyć tego obrazu w artykule. Wybrać obraz z repozytorium serwisów lub z folderu "Dokumenty", który został udostępniony bieżącemu użytkownikowi, bądź wysłać obraz do folderu "Dokumenty". Odszukać i wybrać obraz, który ma zostać użyty. Jeśli

zasoby cyfrowe nie są widoczne, kliknąć na ikonie , po czym zmienić filtrowanie kolekcji na **Wszystkie**. Jeśli wybrano obraz z folderu "Dokumenty", plik obrazu można powiązać (tworząc łącze), zamiast kopiować go do serwisu. Aby utworzyć łącze do pliku, wybrać opcję **Zamiast kopiować plik do** . Jeśli ta opcja nie zostanie wybrana, w serwisie będzie składowana kopia pliku, do której będą się odnosiły odwołania z serwisu. Tworząc łącze do oryginalnego pliku, unika się duplikowania zawartości. Łącze pozwala odwiedzającym serwis oglądać zawartość, nawet jeśli ulegną zmianie uprawnienia do pliku lub jego wyświetlanie zostanie ograniczone w inny sposób. Nacisnąć przycisk **OK**. Jeśli wybrano zasób cyfrowy, można wybrać określoną realizację. Jeśli realizacja nie zostanie wybrana, będzie stosowany rozmiar oryginalny. Chcąc opublikować najnowszą wersję zasobu (gdy serwis zostanie opublikowany), wybrać opcję **Użyj najnowszej wersji zasobu**.


Aby edytować obraz, kliknąć na ikonie , po czym edytować obraz w dowolny z następujących sposobów:



- \* Aby przyciąć obraz, kliknąć na ikonie **Przytnij** (  ). Z paska narzędzi przycinania wybrać jedną z predefiniowanych proporcji obrazu albo przeciągnąć na obraz uchwyty do przycinania. Gdy wynik będzie satysfakcjonujący, nacisnąć na pasku narzędzi przycinania przycisk **Przytnij** .
- \* Aby obrócić lub przerzucić obraz, kliknąć na ikonie **Obróć** (  ). Na pasku narzędzi obracania podać niestandardową wartość stopni obrotu bądź użyć przycisków do obracania obrazu w lewo lub w prawo albo wybrać, czy obraz ma zostać przerzucony w poziomie czy w pionie.

- \* Aby dodać znak wodny do obrazu, kliknąć na ikonie **Znak wodny** ()  
Dodać tekst do obrazu; korzystając z narzędzi znaku wodnego, można odpowiednio zmienić rozmiar tekstu, styl, kolor i przezroczystość.
  - \* Aby zmienić format obrazu, kliknąć na ikonie **Opcje** ()  
z rozwijanej listy **Format** nowy format.
  - \* Aby zmienić kolor tła, kliknąć na ikonie **Opcje** ()  
z rozwijanego menu **Kolor tła** odpowiednią opcję.
  - \* Jeśli jest edytowany obraz w formacie .jpg lub .webp (dostępne w przeglądarce Google Chrome), można zmienić jakość obrazu w celu utworzenia pliku o mniejszym rozmiarze. Kliknąć na ikonie **Opcje** ()  
po czym w obszarze **Jakość** podać nową wartość procentową.
  - \* Aby cofnąć lub ponowić zmianę, kliknąć na ikonie  lub . Aby usunąć wszelkie dokonane zmiany, nacisnąć przycisk **Resetuj**.
  - \* Aby zmienić powiększenie obrazu, użyć elementów sterujących powiększaniem ()  
    - **Tytuł**: Wpisać tekst, który ma być wyświetlany w etykietce.
    - **Tekst alternatywny**: Wpisać tekst alternatywny, który będzie wyświetlany w celu ułatwienia dostępu.
    - **Podpis**: Wpisać tekst podpisu, który będzie wyświetlany pod obrazem.
    - **Wyrównanie, Szerokość, Odstępy**: Jeśli trzeba, zmienić układ obrazu.
  - **Tytuł**: Zmienić odstępy dla tytułu artykułu.
  - **Akapit** (nagłówek niższego poziomu) i **Akapit**: Zmienić odstępy w tekście znajdującym się pod tytułem artykułu.
8. Korzystając z karty "Ogólne", zmodyfikować ustawienia, odstępy i inne opcje składnika.
  9. Korzystając z karty "Styl", dodać formatowanie tekstu oraz dostosować kolor tła, czcionki i krawędzie.

## Obrazy z tekstem


Można dodać składnik "obraz i tekst" łączący w sobie składniki "akapit" i "obraz".


1. Przejść do strony, która ma być edytowana, i upewnić się, że przełącznik  jest ustawiony w pozycji **Edytuj**.
2. Dodać składnik do strony.
3. Kliknąć w obszarze akapitu składnika, aby dodać odpowiedni tekst. Cały tekst przyjmie formatowanie domyślnego stylu składnika. Aby wprowadzać dodatkowe wiersze tekstu, należy naciskać klawisz Enter.
4. Aby zmienić formatowanie domyślne dowolnego fragmentu, zaznaczyć tekst, który ma zostać sformatowany, po czym użyć dowolnych opcji z paska formatowania, np. określających czcionkę, kolor lub wyrównanie.





5. Aby usunąć formatowanie zastosowane przy użyciu tych opcji, zaznaczyć tekst, po czym kliknąć na ikonie .
6. Aby edytować składnik i jego wygląd, kliknąć na ikonie menu składnika , po czym wybrać opcję **Ustawienia**.
7. Wybrać **Składniki**, aby określić szczegóły wyświetlanej zawartości.






- **Obraz:**

- Nacisnąć przycisk **Wybierz**, aby użyć tego obrazu. Wybrać obraz z repozytorium serwisów lub z folderu "Dokumenty", który został udostępniony bieżącemu użytkownikowi, bądź wysłać obraz do folderu "Dokumenty". Odszukać i wybrać obraz, który ma zostać użyty. Jeśli

zasoby cyfrowe nie są widoczne, kliknąć na ikonie , po czym zmienić filtrowanie kolekcji na **Wszystkie**. Jeśli wybrano obraz z folderu "Dokumenty", plik obrazu można powiązać (tworząc łącze), zamiast kopiować go do serwisu. Aby utworzyć łącze do pliku, wybrać opcję **Zamiast kopiować plik do** . Jeśli ta opcja nie zostanie wybrana, w serwisie będzie składowana kopia pliku, do której będą się odnosiły odwołania z serwisu. Tworząc łącze do oryginalnego pliku, unika się duplikowania zawartości. Łącze pozwala odwiedzającym serwis oglądać zawartość, nawet jeśli ulegną zmianie uprawnienia do pliku lub jego wyświetlanie zostanie ograniczone w inny sposób. Nacisnąć przycisk **OK**. Jeśli wybrano zasób cyfrowy, można wybrać określoną realizację. Jeśli realizacja nie zostanie wybrana, będzie stosowany rozmiar oryginalny. Chcąc opublikować najnowszą wersję zasobu (gdy serwis zostanie opublikowany), wybrać opcję **Użyj najnowszej wersji zasobu**.

Aby edytować obraz, kliknąć na ikonie , po czym edytować obraz w dowolny z następujących sposobów:




- \* Aby przyciąć obraz, kliknąć na ikonie **Przytnij** (  ). Z paska narzędzi przycinania wybrać jedną z predefiniowanych proporcji obrazu albo przeciągnąć na obraz uchwyty do przycinania. Gdy wynik będzie satysfakcjonujący, nacisnąć na pasku narzędzi przycinania przycisk **Przytnij** .
- \* Aby obrócić lub przerzucić obraz, kliknąć na ikonie **Obróć** (  ). Na pasku narzędzi obracania podać niestandardową wartość stopni obrotu bądź użyć przycisków do obracania obrazu w lewo lub w prawo albo wybrać, czy obraz ma zostać przerzucony w poziomie czy w pionie.
- \* Aby dodać znak wodny do obrazu, kliknąć na ikonie **Znak wodny** (  ). Dodać tekst do obrazu; korzystając z narzędzi znaku wodnego, można odpowiednio zmienić rozmiar tekstu, styl, kolor i przezroczystość.
- \* Aby zmienić format obrazu, kliknąć na ikonie **Opcje** (  ), po czym wybrać z rozwijanej listy **Format** nowy format.

- \* Aby zmienić kolor tła, kliknąć na ikonie **Opcje** () , po czym wybrać z rozwijanego menu **Kolor tła** odpowiednią opcję.
  - \* Jeśli jest edytowany obraz w formacie .jpg lub .webp (dostępne w przeglądarce Google Chrome), można zmienić jakość obrazu w celu utworzenia pliku o mniejszym rozmiarze. Kliknąć na ikonie **Opcje** () , po czym w obszarze **Jakość** podać nową wartość procentową.
  - \* Aby cofnąć lub ponowić zmianę, kliknąć na ikonie  lub  . Aby usunąć wszelkie dokonane zmiany, nacisnąć przycisk **Resetuj**.
  - \* Aby zmienić powiększenie obrazu, użyć elementów sterujących powiększaniem ().
- **Tytuł:** Wpisać tekst, który ma być wyświetlany w etykietce.
  - **Tekst alternatywny:** Wpisać tekst alternatywny, który będzie wyświetlany w celu ułatwienia dostępu.
  - **Podpis:** Wpisać tekst podpisu, który będzie wyświetlany pod obrazem.
  - **Wyrównanie, Szerokość, Odstęp:** Jeśli trzeba, zmienić układ obrazu.
- **Akapit:** Zmienić odstępy w tekście.
8. Korzystając z karty "Ogólne", zmodyfikować ustawienia, odstępy i inne opcje składnika.
  9. Korzystając z karty "Styl", dodać formatowanie tekstu oraz dostosować kolor tła, czcionki i krawędzie.

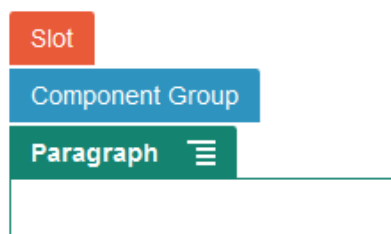
## Grupy składników

Można połączyć jeden lub więcej składników, tworząc grupę składników, którą można nazwać i której można wielokrotnie używać.


Zapisując grupę składników, zapisuje się ją jako składnik niestandardowy o podanej nazwie, który pojawia się (w edytorze) na liście składników niestandardowych.

1. Przejść do strony, która ma być edytowana, i upewnić się, że przełącznik  jest ustawiony w pozycji **Edytuj**.
2. Dodać grupę składników do strony. Grupa składników jest identyfikowana za pomocą ikony .
3. Przeciągnąć jeden lub więcej składników do ramki składnika niestandardowego.
4. Określić pozycję i rozmiar składników w grupie składników, postępując tak samo, jak w przypadku składników w gnieździe.
5. Aby edytować składnik i jego wygląd, kliknąć na ikonie  menu składnika, po czym wybrać opcję **Ustawienia**. Po kliknięciu na nazwie składnika (zamiast na ikonie menu), będzie można zobaczyć i wybrać ikonę "Menu" dla grupy (lub gniazda) składników:





Dla grupy składników, karta "Ustawienia" umożliwia określenie pozycji grupy składników, obraz tła oraz inne ustawienia stosowane dla całej grupy składników.

6. Gdy zmiany wprowadzone w grupie składników będą gotowe do zapisania, kliknąć na grupie składników, następnie kliknąć na jej ikonie , po czym wybrać opcję **Zapisz**.

- a. W oknie dialogowym wpisać nazwę grupy składników. Może się ona składać tylko z liter, cyfr, znaków podkreślenia ( \_ ) i łączników (-). Użyta spacja zostanie automatycznie zastąpiona łącznikiem.

Jeśli jest to nowa grupa składników, nie może mieć ona takiej samej nazwy jak już istniejący składnik niestandardowy.

Jeśli do strony została dodana istniejąca grupa składników, po czym grupa ta została zmodyfikowana, a następnie została podjęta próba zapisania zmian, to jest udostępniana opcja podania nazwy (w celu utworzenia nowej grupy składników) oraz opcja **Zastąp istniejącą grupę składników** (w celu zaktualizowania istniejącej grupy składników z użyciem dokonanych zmian).

- b. Nacisnąć przycisk **Zapisz**.



Grupa składników o podanej nazwie zostanie zapisana jako składnik niestandardowy. Grupa ta pojawi się w swoim folderze w menedżerze składników oraz (w edytorze) na liście składników niestandardowych, jeśli bieżący użytkownik jest właścicielem tego składnika lub ktoś ten składnik mu udostępnił. Można udostępnić grupę składników, postępując jak w przypadku każdego innego składnika niestandardowego.

## Korzystanie z usługi Cobrowse na stronie

Usługa Oracle Cobrowse Cloud Service jest to narzędzie do pracy zespołowej, umożliwiające udostępnianie ekranów lub inicjowanie sesji wspólnego przeglądania z inną osobą. Na przykład można dodać to narzędzie do formularza zamówienia, dzięki czemu przedstawiciel firmy będzie mógł zobaczyć ekran klienta podczas składania przez niego zamówienia.



Aby można było używać tej funkcji, trzeba ją najpierw włączyć dla serwisu. Istnieją dwa rodzaje skryptów procesu uruchamiającego: jeden wykorzystujący dostosowywany przycisk (Launch Point 2) i jeden wykorzystujący domyślny przycisk Cobrowse (Launch Point 1). Użytkownik sam decyduje, który rodzaj procesu uruchamiającego będzie używany, gdy zostanie włączona funkcja dla serwisu i zostanie dodany wymagany skrypt. Szczegółowe informacje są dostępne pod hasłem [Włączanie integracji](#).

Po włączeniu usługi Cobrowse dla serwisu, można skonfigurować dowolną jego stronę, tak aby było możliwe wspólne przeglądanie.

1. Przejdź do strony, która ma być edytowana, i upewnij się, że przełącznik  jest ustawiony w pozycji **Edytuj**.
2. Kliknąć na ikonie . Zaznaczyć pole wyboru w sekcji usługi Cobrowse.
3. Nacisnąć przycisk **Zamknij**.

Jeśli został włączony skrypt "Launch Point 1", to podczas następnego wyświetlenia strony lub po naciśnięciu klawisza dostępu (jeśli został skonfigurowany) pojawi się domyślny przycisk Cobrowse.

Jeśli został włączony skrypt "Launch Point 2", trzeba dodać niestandardowy przycisk na stronie.

1. Upewnij się, że przełącznik  jest ustawiony w pozycji **Edytuj**.
2. Dodać składnik do strony. Składnik "proces uruchamiający Cobrowse" jest wyszczególniony w sekcji składników "Integracja".
3. Aby edytować składnik i jego wygląd, kliknąć na ikonie menu składnika , po czym wybrać opcję **Ustawienia**. Można dostosowywać etykietę, wygląd, rozmiar i wyrównanie przycisku. Korzystając z karty "Styl", dodać formatowanie tekstu oraz dostosować kolor tła, czcionki i krawędzie. Aby zastosować więcej zaawansowanych stylów, należy edytować lub dodać klasy stylów w plikach `design.json` i `design.css`, zawartych w folderze projektów motywu dla bieżącego szablonu serwisu. Prefiks klasy stylu to `scs-cobrowse`.

Gdy serwis zostanie opublikowany, odwiedzający będą mogli korzystać z przycisku Cobrowse w celu zainicjowania sesji z przedstawicielem organizacji. Odwiedzający, po naciśnięciu przycisku Cobrowse, dostaje bezpieczny ID sesji. Korzystając z telefonu, przekazuje ten ID do przedstawiciela organizacji, który ma dostęp do konsoli agenta Cobrowse. Agent używa konsoli do zainicjowania sesji, która trwa do momentu jej zakończenia przez odwiedzającego lub agenta. Gdy jest używane ustawienie "Launch Point 2", składnik "proces uruchamiający" zawiera ustalony ID "cec-start-cobrowse". Należy użyć tego ID w konsoli Cobrowse.

#### Uwagi dotyczące użycia

Jeśli usługa Cobrowse działa w trybie natychmiastowym (ICB), pliki wideo lub osadzone ramki iFrame nie będą widoczne na stronie, dopóki zawartość ramki iFrame nie zostanie aktywowana tym samym ID serwisu Cobrowse. Z tego powodu niektóre składniki usługi Oracle Content Management nie są renderowane w konsoli agenta Cobrowse w trybie ICB. Należy użyć trybu zaawansowanego usługi Cobrowse, aby można było renderować następujące składniki:

- Wideo
- YouTube
- Menedżer dokumentów
- Facebook - Lubię to
- Twitter - Obserwuj
- Twitter - Udostępnij
- Facebook - Rekomendacja
- Konwersacja

Składnik niestandardowy, który używa ramki iFrame do uzyskiwania zawartości, również nie jest renderowany w trybie natychmiastowym.

Więcej informacji dotyczących usługi Oracle Cobrowse Cloud Service można znaleźć w rozdziale [Cobrowse Overview](#) w podręczniku *Cobrowse Deployment and Use Guide*. Szczegółowe informacje dotyczące korzystania z usługi Cobrowse w serwisach zabezpieczonych lub będących na etapie rozwoju są dostępne pod hasłem [Włączanie integracji z usługą Cobrowse](#).



## Oracle Intelligent Advisor

Usługa Oracle Intelligent Advisor (dawniej Oracle Policy Automation) jest używana do implementowania scenariuszy "wywiad" online, takich jak informacje zwrotne związane z rozwiązywaniem problemów czy przypisywaniem uprawnień w odniesieniu do usług. Zapewnia porady z użyciem różnych kanałów, rejestrując reguły w naturalnym języku dokumentów Microsoft Word i Microsoft Excel, a następnie tworząc wokół tych reguł interaktywne środowiska obsługi klientów, nazywane wywiadami.

Aby można było używać funkcji usługi Intelligent Advisor, trzeba ją najpierw włączyć i skonfigurować. Administrator usługi może włączyć funkcje usługi, takie jak dodawanie nazwy hosta, adresu URL, nazwy użytkownika i hasła dla używanego środowiska Intelligent Advisor. Integracja między dwiema usługami wymaga systemu jednokrotnego logowania (SSO), dlatego też obie usługi muszą być w tej samej domenie tożsamości. Zob. rozdział *Integrowanie z Intelligent Advisor* w podręczniku *Integrowanie i rozszerzanie*.

Po stronie usługi Intelligent Advisor wywiady muszą być tworzone i przechowywane w serwisie hosta. Dodatkowo usługa Oracle Content Management musi być autoryzowana do używania przez hosta usługi Intelligent Advisor.

Po skonfigurowaniu i włączeniu usługi Intelligent Advisor można dodać składnik Intelligent Advisor do strony serwisu.

1. Przejść do strony, która ma być edytowana, i upewnić się, że przełącznik  jest ustawiony w pozycji **Edytuj**.
2. Dodać składnik do strony. Składnik pojawi się w sekcji "Integracja" listy składników.
3. Aby edytować składnik i jego wygląd, kliknąć na ikonie menu  składnika, po czym wybrać opcję **Ustawienia**. Można dostosowywać etykietę, wygląd, rozmiar i wyrównanie składnika. Korzystając z karty "Styl", można użyć domyślnego stylu powiązanego z wywiadem z hosta Intelligent Advisor. Aby zastosować więcej zaawansowanych stylów, należy edytować lub dodać klasy stylów w plikach design.css, zawartych w folderze projektów motywu dla bieżącego szablonu serwisu. Prefiks klasy stylu to scs-opainterview-.

Gdy serwis zostanie opublikowany, odwiedzający zobaczą interaktywny wywiad, który został wybrany w składniku Intelligent Advisor. Więcej informacji o usłudze Intelligent Advisor można znaleźć w dokumentacji [Intelligent Advisor Documentation Library](#).

## Oracle Visual Builder

Oracle Visual Builder jest hostowanym środowiskiem dla infrastruktury tworzenia aplikacji. Udostępnia oparte na standardach rozwiązanie typu "open-source" do tworzenia aplikacji, współpracy nad nimi i ich wdrażania w Oracle Cloud.

### Etapy początkowe

Aby można było używać usługi Oracle Visual Builder, trzeba ją najpierw włączyć i skonfigurować. Administrator usługi może włączyć funkcje usługi, takie jak dodawanie nazwy hosta, w którym aplikacje są tworzone i przechowywane. Zob. Integrowanie z Oracle Visual Builder w podręczniku *Integrating and Extending Oracle Content Management*. Integracja tych usług wymaga użycia funkcji jednokrotnego logowania SSO, dlatego też obie usługi muszą się znajdować w tej samej domenie tożsamości.

Zanim będzie można używać tej funkcji z Oracle Content Management, trzeba wykonać po stronie usługi Oracle Visual Builder następujące czynności:

- Włączyć w serwisie usługi Oracle Visual Cloud Service udostępnianie zasobów (CORS).
- Utworzyć aplikacje, udostępnić je do osadzania i skonfigurować do użycia z Oracle Content Management.
- Utworzyć aplikacje internetowe i udostępnić je do osadzania w ramce iFrame. Zaimportować narzędzie Sites SDK; w aplikacjach internetowych musi występować odwołanie do tego narzędzia. Dodać parametr "ID" adresu URL strony do aplikacji internetowych.

### Tworzenie składników Oracle Visual Builder

Po włączeniu integracji, utworzeniu aplikacji i aplikacji internetowych oraz przygotowaniu ich do użytkowania trzeba utworzyć nowy składnik dla każdej aplikacji, która będzie dodawana do stron serwisu.

1. W usłudze Oracle Visual Builder uzyskać URL opublikowanej aplikacji internetowej. Kliknąć na aktywnym projekcie, zawierającym aplikację internetową, po czym kliknąć na tej aplikacji. Skopiować adres URL z paska adresu.
2. W usłudze Oracle Content Management wybrać kolejno opcje **Programista i Wyświetl wszystkie składniki**. Zostaną wyświetlone zarejestrowane składniki i układy odległe.
3. Nacisnąć przycisk **Utwórz**, po czym wybrać opcję **Utwórz składnik usługi Visual Builder**.
4. Wpisać nazwę składnika. Nie może to być nazwa już używana dla innego składnika lub układu.

Może się ona składać tylko z liter, cyfr, znaków podkreślenia ( \_ ) i łączników ( - ). Użyta spacja zostanie automatycznie zastąpiona znakiem podkreślenia.



Dla szablonów, motywów, składników, serwisów ani stron serwisów nie wolno używać następujących nazw: authsite, content, pages, scstemplate\_\*, \_comps, \_components, \_compsdelivery, \_idcservice, \_sitescloud, \_sitesclouddelivery, \_themes, \_themesdelivery. Następujących nazw można używać dla stron serwisów, ale nie można używać dla szablonów, motywów, składników ani serwisów: documents, sites.

5. Opcjonalnie: podać opis składnika.
6. Wkleić URL do aktywnej aplikacji internetowej lub aktywnej aplikacji klasycznej.

7. Nacisnąć przycisk **Utwórz**. Po utworzeniu składnika jego nazwa pojawi się na liście składników. Foldery i pliki, zawarte w składniku, można eksplorować po kliknięciu na nazwie składnika (lub układu) wyświetlanej na liście.
8. Aby wybrać ikonę inną niż ta, która jest domyślnie przypisywana do składnika, należy:
  - a. Wybrać z listy składnik.
  - b. Wybrać opcję **Właściwości**
  - c. Kliknąć na karcie **Logo składnika**.
  - d. Kliknąć na obrazie loga, wyświetlanym w galerii, po czym nacisnąć przycisk **Gotowe**

#### Dodawanie składnika do strony serwisu

Teraz można dodać składnik aplikacji do jednej ze stron serwisu. Składnik ten można dodać do strony publicznej albo strony zabezpieczonej.

1. Przejść do strony, która ma być edytowana, i upewnić się, że przełącznik  jest ustawiony w pozycji **Edytuj**.
2. Dodać składnik do strony. Składnik pojawi się w sekcji "Niestandardowe".
3. Aby edytować składnik i jego wygląd, kliknąć na ikonie menu składnika , po czym wybrać opcję **Ustawienia**. Można dostosowywać etykietę, wygląd, rozmiar i wyrównanie składnika. Korzystając z karty "Styl", można użyć domyślnego stylu powiązanego ze składnikiem z hosta Oracle Visual Builder. Aby zastosować więcej zaawansowanych stylów, należy edytować lub dodać klasy stylów w plikach design.css, zawartych w folderze projektów motywu dla bieżącego szablonu serwisu. Prefiks klasy stylu to scs-component.

Podczas edytowania serwisu można wyświetlić składnik w trybie podglądu. Gdy serwis zostanie opublikowany, odwiedzający będą widzieli wybraną do używania z tym składnikiem aplikację Oracle Visual Builder, działającą w ramce iFrame na stronie.

# Część III

## Publikowanie serwisów i zarządzanie nimi

W tej części opisano, w jaki sposób pracować z serwisami, zabezpieczać je, udoskonalać i publikować. Część ta zawiera następujące rozdziały:

- [Zarządzanie serwisami](#)
- [Publikowanie serwisów](#)
- [Serwisy zabezpieczone](#)
- [Praca z serwisami wielojęzycznymi](#)
- [Korzystanie z przekierowań lub odwzorowań adresów URL](#)
- [Zwiększanie wydajności serwisu](#)

# 11

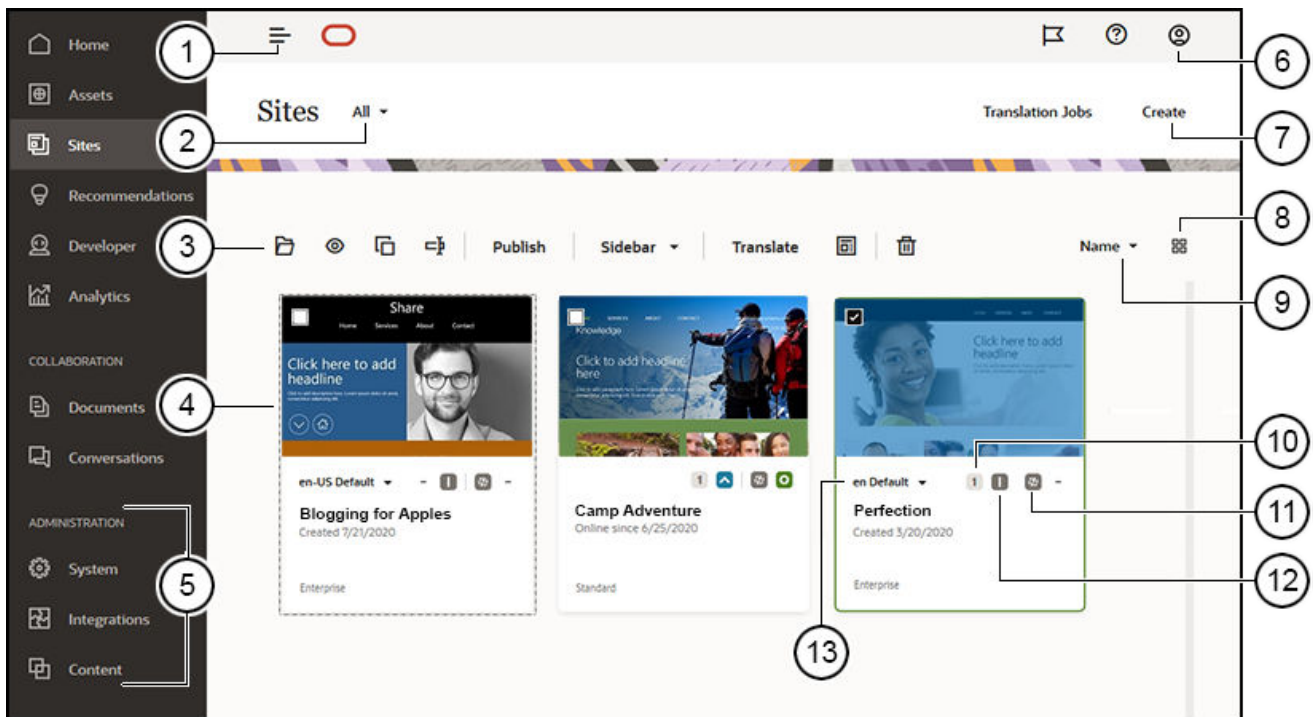
## Zarządzanie serwisami

Aby zmienić układ lub zawartość serwisu, należy w edytorze utworzyć i otworzyć aktualizację. W celu utworzenia oraz zarządzania samym serwisem i jego właściwościami należy skorzystać z opcji dostępnych w menedżerze serwisów.

- [Omówienie strony "Serwisy"](#)
- [Zarządzanie serwisem i ustawienia serwisów](#)
- [Przełączanie serwisu do trybu online lub offline](#)
- [Zmianie opisu, loga lub właściwości osadzania serwisu](#)
- [Ustawienie właściwości dotyczących wyszukiwarek](#)
- [Dostosowywanie ustawień serwisów](#)
- [Włączanie integracji z usługą Cobrowse](#)
- [Dodawanie śledzenia analiz](#)

### Omówienie strony "Serwisy"

Strona "Serwisy" stanowi bramę do obszarów umożliwiającą pracę nad serwisami internetowymi.



Numer	Opis
1	<b>Przełącznik paneli</b> pozwala ukryć lub pokazać menu nawigacyjne.
2	<b>Menu filtrowania</b> umożliwia filtrowanie listy serwisów w celu wyświetlenia wszystkich serwisów, do których bieżący użytkownik ma dostęp, lub wyświetlenia ich podzbioru. Jeśli funkcja zarządzania serwisem jest włączona, można wyświetlić oczekujące wystąpienia o serwis.
3	Gdy z listy zostanie wybrany jakiś serwis, stanie się dostępny <b>pasek czynności</b> . Za pomocą opcji z paska czynności można otwierać i wyświetlać swoje serwisy, zmieniać ich status i nazwę oraz wykonywać inne zadania. Pokazywane opcje zależą od roli przypisanej użytkownikowi w odniesieniu do wybranego serwisu. Na przykład użytkownik, będący twórcą serwisu, ma w odniesieniu do tego serwisu rolę "Menedżer" i może wykonywać wszystkie wymienione zadania. Jeśli serwis został udostępniony użytkownikowi, któremu przypisano rolę "Oglądający", to będzie mógł on wyświetlać właściwości serwisu, lecz nie będzie mógł niczego zmienić. Opcje dostępne na pasku czynności zależą także od statusu serwisu. Opcje <b>Zmień nazwę i Usuń</b> są dostępne tylko dla serwisu przełączonego do trybu offline. <b>Wskazówka:</b> Dostępne są także skróty. Wystarczy kliknąć prawym przyciskiem myszy na serwisie, a zostanie otwarte menu podręczne, z którego można wybrać odpowiednią opcję.
4	<b>Lista serwisów</b> pokazuje wszystkie serwisy, których właścicielem jest bieżący użytkownik lub które zostały mu udostępnione. Podawana jest nazwa serwisu, liczba jego aktualizacji, jego status (online, offline lub oczekujący na aktualizację) oraz czy serwis jest zabezpieczony.
5	<b>Menu "Administrowanie"</b> jest pokazywane, gdy użytkownik zalogował się jako administrator zawartości. Za pomocą dostępnych opcji można zmienić ustawienia usługi, skonfigurować integracje oraz skonfigurować repozytoria i inne funkcje powiązane z zasobami.
6	<b>Menu użytkownika</b> zawiera opcje umożliwiające ustawianie preferencji, przekazywanie informacji zwrotnych, uzyskiwanie dostępu do Pomocy oraz wylogowywanie się. Klikając na zdjęciu użytkownika, wyświetla się opcje z menu.
7	Aby rozpocząć proces tworzenia serwisu internetowego, należy kliknąć na opcji <b>Utwórz</b> . Aby można było przystąpić do tworzenia serwisu internetowego, administrator musi umożliwić tworzenie serwisów oraz udostępnić przynajmniej jeden szablon. Jeśli opcja <b>Utwórz</b> na stronie serwisu lub szablonu na stronie szablonów nie są widoczne, należy się skontaktować z administratorem usługi.
8	Klikając na <b>ikonie widoku</b> można wybrać, czy serwisy mają być wyświetlane w widoku tabeli, widoku listy czy widoku tabeli.
9	Korzystając z <b>opcji sortowania</b> , można zmieniać kolejność wyświetlania serwisów.
10	Aby edytować istniejącą aktualizację albo aby utworzyć nową aktualizację dla serwisu, należy kliknąć na ikonie <b>Numer aktualizacji</b> . Jeśli dla serwisu nie ma żadnych aktualizacji, można utworzyć nową aktualizację, otwierając serwis i przełączając konstruktor serwisów do trybu edycji.
11	Aby zmienić status serwisu lub opublikować zmiany, które zostały zatwierdzone, lecz jeszcze nie zostały opublikowane w aktywnym serwisie, należy kliknąć na ikonie <b>Offline</b> , <b>Online</b> lub <b>Publikuj ponownie</b> .
12	Ikony <b>zabezpieczeń</b> pokazują, czy serwis wymaga czy nie wymaga logowania.











Numer	Opis
13	Za pomocą <b>listy języków</b> można zobaczyć języki określone w założeniu systemowym dot. lokalizacji, powiązanych z serwisem. Aby uzyskać podgląd lokalizowanej (tłumaczonej) wersji, należy wybrać język, po czym otworzyć serwis.



## Zarządzanie serwisem i ustawienia serwisów

Zarządzanie serwisem obejmuje tworzenie i edytowanie serwisu oraz zarządzanie jego zawartością. Do tworzenia, kopiowania i usuwania stron serwisu oraz zawartości stron służy *konstruktor ser*. Do tworzenia, kopiowania, udostępniania i usuwania całych serwisów służy *menedżer folderów i plików*.

Do wykonania przedstawionych poniżej zadań służą opcje z paska menu, z menu podręcznego ze strony "Serwisy" lub z menu podręcznego w konstruktorze serwisów.


Zadanie	Opis
Filtrowanie widoku serwisów	<p>Aby przefiltrować listę serwisów, wybrać z menu filtrowania jedną z następujących opcji:</p> <ul style="list-style-type: none"> <li>• <b>Wszystko</b> — Pokazuje wszystkie serwisy, do których bieżący użytkownik ma dostęp. Administrator serwisów (przy włączonej funkcji zarządzania serwisem), ma dostęp do wszystkich serwisów w środowisku.</li> <li>• <b>Moje</b> — Pokazuje wszystkie serwisy, których właścicielem jest bieżący użytkownik.</li> <li>• <b>Udostępniane mnie</b> — Pokazuje wszystkie serwisy, które zostały udostępnione bieżącemu użytkownikowi.</li> <li>• <b>Online</b> — Pokazuje serwisy, które są w stanie online (są aktywne).</li> <li>• <b>Offline</b> — Pokazuje serwisy, które są w stanie offline.</li> <li>• <b>Żądania</b> (dostępne tylko, gdy funkcja zarządzania serwisem jest włączona) — Pokazuje oczekujące wystąpienia o serwis. Będąc administratorem serwisów, wszystkie wystąpienia o serwis można zobaczyć w swoim systemie; w przeciwnym razie widoczne są tylko wystąpienia o serwis przesłane przez bieżącego użytkownika.</li> <li>• <b>Kosz</b> — Pokazuje usunięte serwisy, które należą do bieżącego użytkownika. Jeśli bieżący użytkownik jest administratorem serwisów (przy włączonej funkcji zarządzania), opcja ta pokazuje również wszystkie usunięte serwisy.</li> </ul>
Tworzenie serwisu	<p>Aby utworzyć serwis, nacisnąć przycisk <b>Utwórz</b>. Tworząc serwis, zaczynamy od szablonu. Szablon zawiera wszystko, co jest potrzebne do rozpoczęcia pracy nad serwisem, w tym strukturę kodu serwisu, serwis domyślny z przykładowymi stronami i zawartością, motyw ze stylami oraz zasoby (takie jak obrazy), czy nawet składniki niestandardowe. Zob. <a href="#">Tworzenie serwisów</a>.</p> <p>W menu filtrowania, na stronie <b>Serwisy</b>, można sprawdzić, czy funkcja zarządzania serwisem jest włączona. Jeśli jest widoczna opcja <b>Żądania</b>, funkcja zarządzania serwisem jest włączona. Zob. <a href="#">Omówienie funkcji zarządzania serwisem</a>.</p>

Zadanie	Opis
Wyświetlanie wystąpień o serwis i zarządzanie nimi	Aby wyświetlić oczekujące wystąpienia o serwis (opcja dostępna tylko wtedy, gdy funkcja zarządzania serwisem jest włączona), wybrać z menu filtrowania opcję <b>Żądania</b> . Będąc administratorem serwisów, wszystkie wystąpienia o serwis można zobaczyć w swoim systemie; w przeciwnym razie widoczne są tylko wystąpienia o serwis przesłane przez bieżącego użytkownika. Zob. <a href="#">Zarządzanie wystąpieniami o serwis</a> .
Edytowanie serwisu	Aby edytować serwis, należy wybrać z menu podręcznego opcję <b>Otwórz</b> albo kliknąć na ikonie  (na pasku czynności). Zob. <a href="#">Edytowanie serwisów</a> .
Omawianie serwisu lub dołączanie do niego adnotacji	Możliwość omawiania serwisów oraz dołączania do nich adnotacji za pomocą konstruktora serwisów. Należy wybrać serwis, po czym wybrać z menu podręcznego opcję <b>Otwórz</b> albo kliknąć na ikonie  (na pasku czynności). Chcąc omówić serwis, kliknąć na ikonie  — zostanie otwarty panel konwersacji. Aby dołączyć adnotację do serwisu, kliknąć na ikonie  . Zob. <a href="#">Omówienie strony "Konstruktor serwisów"</a> .
Wyświetlanie serwisu	Aby zobaczyć, jak serwis będzie wyglądał w trybie online, należy wybrać serwis, po czym wybrać z menu podręcznego opcję <b>Wyświetl</b> albo kliknąć na ikonie  (na pasku czynności).
Publikowanie serwisu	Aby opublikować serwis, należy go wybrać, po czym wybrać z menu podręcznego lub z paska czynności opcję <b>Publikuj</b> . Zob. <a href="#">Publikowanie zmian serwisu</a> .
Ponowne publikowanie	Jeśli serwis został wcześniej opublikowany, lecz zmieniono powiązany element lub założenie systemowe (na przykład powiązane założenie systemowe dot. lokalizacji zostało zaktualizowane, tak aby obejmowało więcej języków), można <b>ponownie opublikować</b> serwis w celu odświeżenia go z uwzględnieniem tych zmian. Należy wybrać serwis, po czym wybrać z menu podręcznego lub z paska czynności opcję <b>Publikuj ponownie</b> .
Anulowanie publikacji	Chcąc usunąć pliki serwisu (wcześniej opublikowanego) z lokalizacji w Oracle Cloud, należy wybrać serwis, po czym wybrać z menu podręcznego lub z paska czynności opcję <b>Anuluj publikację</b> . Aby anulować publikację serwisu, trzeba przełączyć go do trybu offline.
Przełączanie serwisu do trybu online lub offline	Serwis, zanim będzie można go przełączyć do trybu online, trzeba najpierw opublikować. Aby przełączyć serwis do trybu online lub offline, należy go wybrać, po czym wybrać z menu podręcznego opcję <b>Przełącz do trybu online</b> lub <b>Przełącz do trybu offline</b> . Można także kliknąć na kafelku serwisu na ikonie  (aby przełączyć serwis do trybu online) lub  (aby przełączyć serwis do trybu offline). Serwis działający w trybie <i>online</i> można zobaczyć pod adresem URL serwisu, używając przeglądarki internetowej. Gdy serwis zostanie przełączony do trybu <i>offline</i> , przestanie być dostępny do publicznego odbioru. Serwis będzie można zobaczyć tylko w Oracle Content Management. Zob. <a href="#">Przełączanie serwisu do trybu online lub offline</a> .
Zmianie nazwy serwisu	Aby zmienić nazwę serwisu, należy kliknąć na nim prawym przyciskiem myszy, po czym wybrać z menu podręcznego opcję <b>Zmień nazwę</b> albo kliknąć na ikonie  (na pasku czynności). Wpisać nazwę serwisu, różniącą się od każdej nazwy serwisu istniejącego na tym samym serwerze. Nie można zmienić nazwy serwisu będącego w stanie online. Status serwisu (przełączyć go do trybu offline) może zmienić tylko właściciel serwisu lub osoba mająca przypisaną rolę menedżera.

Zadanie	Opis
Kopiowanie serwisu	<p>Aby zacząć szybko konstruować serwis, można skopiować serwis już istniejący. Do nowego serwisu, o podanej nowej nazwie, zostanie skopiowane wszystko z oryginalnego serwisu, w tym: motyw, wszystkie oczekujące aktualizacje, strony, zawartość stron oraz wszystkie inne zasoby, takie jak obrazy. Nowy serwis znajduje się w stanie offline i jest gotów do edycji.</p> <p><b>Uwaga:</b> Jeśli funkcja zarządzania serwisem jest włączona, serwis — zanim kopia zostanie utworzona — może wymagać zatwierdzenia.</p> <p>Jeśli jest kopiowany serwis, który używa zawartości z więcej niż jednego repozytorium, serwis ten trzeba skopiować przy użyciu narzędzia OCE Toolkit. Zob. <a href="#">Tworzenie za pomocą zestawu OCE Toolkit</a> i <a href="#">Korzystanie z narzędzia cec opartego na wierszu polecenia</a>.</p> <p>Należy wybrać serwis, który ma zostać skopiowany, po czym wybrać z menu podręcznego opcję <b>Kopiuj</b> albo kliknąć na ikonie  (na pasku czynności). Zob. <a href="#">Kopiowanie serwisów</a>.</p>
Tworzenie szablonu z serwisu	<p>Mając serwis, który może zostać użyty jako punkt startowy dla innych serwisów, można z tego serwisu utworzyć szablon. Należy wybrać serwis, po czym wybrać z menu podręcznego opcję <b>Utwórz szablon</b> albo kliknąć na ikonie  (na pasku czynności). Zob. <a href="#">Tworzenie szablonu z serwisu</a>.</p>

 **Uwaga:**

Jeśli szablon jest tworzony z serwisu, który używa zawartości z więcej niż jednego repozytorium, szablon ten trzeba utworzyć przy użyciu narzędzia OCE Toolkit. Zob. [Tworzenie za pomocą zestawu OCE Toolkit](#) i [Korzystanie z narzędzia cec opartego na wierszu polecenia](#).

Zadanie	Opis
Usuwanie lub przywracanie serwisu	<p>Mając odpowiednie uprawnienia, można usunąć serwis i jego zawartość. Wskutek usunięcia serwisu zostanie umieszczone w Koszu wszystko, co znajduje się w folderze serwisu, w tym: wszystkie oczekujące aktualizacje, strony serwisu, zawartość stron oraz wszystkie inne zasoby, takie jak obrazy, które zostały dodane do stron.</p> <p>Serwis może zostać usunięty lub przywrócony przez jego twórcę (właściciela serwisu) lub osobę, której ten serwis został udostępniony i której została przypisana rola "Współtwórca" lub "Menedżer".</p> <p>Nie można usunąć serwisu działającego online. Status serwisu (przełączyć go do trybu offline) może zmienić tylko właściciel serwisu lub osoba mająca przypisaną rolę menedżera.</p> <p>Nie można również usunąć serwisu, który został opublikowany. Trzeba najpierw anulować publikację zawartości. Jeśli serwis zawiera zawartość z więcej niż jednego repozytorium, trzeba w każdym repozytorium anulować publikację powiązanej zawartości.</p> <p>Aby usunąć serwis, należy go wybrać, po czym wybrać z menu podręcznego opcję <b>Usuń</b> albo kliknąć na ikonie  (na pasku czynności). Zostanie wyświetlone pytanie, czy serwis i wszystkie jego aktualizacje mają zostać przeniesione do Kosza. Usunięty serwis pozostaje w Koszu, aż:</p> <ul style="list-style-type: none"><li>• Serwis zostanie przywrócony.</li><li>• Serwis zostanie trwale usunięty.</li><li>• Zostanie osiągnięty przydział miejsca na Kosz.</li><li>• Kosz zostanie automatycznie opróżniony zgodnie z interwałem ustawionym przez administratora serwisu. Wartością domyślną jest 90 dni.</li></ul> <p>Aby przywrócić serwis, wybrać z menu "Serwisy" opcję <b>Kosz</b>. Wybrać serwis z listy, po czym opcję <b>Przywróć</b>.</p>

Zadanie	Opis
Dodawanie współużytkowników do serwisu	<p>Jeśli administrator włączył udostępnianie, można udostępnić swój serwis innym użytkownikom Oracle Content Management i zezwolić im — zgodnie z nadanymi im uprawnieniami — na wyświetlanie lub modyfikowanie serwisu w Oracle Content Management bądź na zarządzanie nim.</p> <p>Każdy, kto w taki sposób może uzyskać dostęp do naszego serwisu, jest nazywany współużytkownikiem. Udostępniając serwis, przypisuje się współużytkownikom role definiujące, co dany współużytkownik może zrobić z naszym serwisem. Współużytkownicy mogą używać serwisu tylko zgodnie z przypisaną im rolą.</p> <p>Należy wybrać serwis, który ma zostać udostępniony, po czym wybrać z menu podręcznego opcję <b>Współużytkownicy</b> albo nacisnąć na pasku czynności przycisk <b>Pasek boczny</b>, a następnie — aby otworzyć pasek boczny "Współużytkownicy" — wybrać opcję <b>Współużytkownicy</b>. Po otwarciu paska bocznego "Współużytkownicy" należy nacisnąć przycisk <b>Dodaj współużytkownika</b>.</p> <p>Podać adres e-mail lub adresy e-mail oraz przypisać jedną z następujących ról:</p> <ul style="list-style-type: none"> <li>• <b>Oglądający:</b> Osoby te mogą wyświetlać serwis w edytorze, lecz nie mogą niczego zmieniać.</li> <li>• <b>Pobierający:</b> Rola "Pobierający" zapewnia w odniesieniu do serwisu te same uprawnienia co rola "Oglądający" oraz dodatkowo pozwala utworzyć nowy szablon z serwisu.</li> <li>• <b>Współtwórca:</b> Ma te same uprawnienia co oglądający, a ponadto może edytować serwis i usuwać strony serwisu oraz może usunąć serwis przełączony do trybu offline.</li> <li>• <b>Menedżer:</b> Ma te same uprawnienia co współtwórca, a ponadto może dodawać użytkowników i przypisywać im role, publikować zmiany do serwisu online oraz przełączać serwis online/offline. Twórcy (właścicielowi) serwisu jest automatycznie przypisywana rola "Menedżer".</li> </ul>
Zmianianie właściwości serwisu (opis, logo itp.)	<p>Aby zmienić opis serwisu, dodać logo, umożliwić osadzanie serwisu, dodać zindywidualizowany adres URL oraz wyświetlić inne informacje o serwisie (takie jak URL serwisu, właściciel serwisu i inne szczegóły), należy wybrać serwis, po czym wybrać z menu podręcznego opcję <b>Właściwości</b> albo nacisnąć na pasku czynności przycisk <b>Pasek boczny</b>, a następnie wybrać opcję <b>Właściwości</b>. Zob. <a href="#">Zmianianie opisu, loga lub właściwości osadzania serwisu</a>.</p>
Tłumaczenie serwisu	<p>Serwis firmowy — jeśli powiązane repozytorium na to pozwala — można przetłumaczyć na różne języki, eksportując pliki serwisu, tłumacząc je, a następnie importując już przetłumaczone pliki. Zob. <a href="#">Tłumaczenie serwisu</a>.</p>
Ustawienie właściwości dotyczących wyszukiwarek	<p>W konstruktorze serwisów można określić słowa kluczowe i teksty pomagające wyszukiwarkom rozpoznawać zawartość serwisu. Zob. <a href="#">Ustawienie właściwości dotyczących wyszukiwarek</a></p>

 **Uwaga:**

Jeśli jest udostępniany serwis, który ma dostęp do więcej niż jednego repozytorium, to udostępniane jest tylko repozytorium domyślne. Pozostałe repozytoria trzeba udostępnić osobno.


Zadanie	Opis
Dostosowywanie ustawień serwisu (ulubione ikony, pliki kontrolera itp.)	W konstruktorze serwisów można określić ikony serwisu, używane w różnych przeglądarkach i na różnych platformach, lub dodać plik kontrolera określający działanie łączy. Zob. <a href="#">Dostosowywanie ustawień serwisów</a>
Włączanie funkcji wspólnego przeglądania	W konstruktorze serwisów można włączyć usługę Oracle Cobrowse Cloud Service — narzędzie do pracy zespołowej, umożliwiające udostępnianie ekranów lub inicjowanie sesji wspólnego przeglądania z inną osobą. Na przykład można dodać to narzędzie do formularza zamówienia, dzięki czemu przedstawiciel firmy będzie mógł zobaczyć ekran klienta podczas składania przez niego zamówienia. Zob. <a href="#">Włączanie integracji z usługą Cobrowse</a>
Dodawanie obiektów śledzenia analiz do serwisów	W konstruktorze serwisów można do serwisu dodać snippet — zawierający kod JavaScript śledzenia — w celu śledzenia analiz internetowych i ułatwienia integracji z zewnętrznymi dostawcami analiz, takimi jak Google, Adobe lub Oracle Infinity. Zob. <a href="#">Dodawanie śledzenia analiz</a> .
Zabezpieczanie serwisu aktywnego	Serwis, opublikowany i udostępniony online, staje się publicznie dostępny dla wszystkich osób. Będąc właścicielem serwisu lub mając przypisaną rolę menedżera, można jednak ograniczyć dostęp do serwisu aktywnego, czyniąc go dostępnym dla zarejestrowanych użytkowników lub dla podzbioru użytkowników. Status serwisu może zmienić tylko właściciel serwisu lub osoba mająca przypisaną rolę menedżera.  Należy wybrać serwis, który ma zostać zabezpieczony, po czym wybrać z menu podręcznego opcję <b>Właściwości</b> albo nacisnąć na pasku czynności przycisk <b>Pasek boczny</b> , wybrać opcję <b>Właściwości</b> , po czym kliknąć na karcie <b>Zabezpieczenia serwisu</b> . Zob. <a href="#">Zmianianie zabezpieczeń serwisu</a> .
Dodawanie przekierowania do serwisu	Jeśli URL serwisu ulegnie zmianie, proces przekierowania przekaże jeden URL (źródłowy) do drugiego URL (docelowo). Pomaga to zachować zakładki użytkownika i klasyfikacje wyszukiwarki. Zob. <a href="#">Dodawanie przekierowań do serwisu</a> .

## Zmianianie opisu, loga lub właściwości osadzania serwisu

Większość właściwości serwisu jest ustawianych podczas tworzenia lub aktualizowania serwisu. Wyświetlając właściwości, można uzyskać wartościowe informacje o serwisie, w tym: adres URL, kto jest właścicielem oraz inne szczegóły.

Aby zmienić opis serwisu lub jego logo, dodać zindywidualizowany serwis bądź umożliwić osadzanie serwisu, należy:

- Wybrać serwis, po czym wybrać z menu podręcznego opcję **Właściwości** albo nacisnąć na pasku czynności przycisk **Pasek boczny**, a następnie wybrać opcję **Właściwości**.
- Aby zmienić opcjonalny opis serwisu, kliknąć na karcie **Właściwości serwisu**, po czym dodać lub zmienić opis.
- Adresu URL nie można zmienić bezpośrednio, ale można go kopiować i wklejać do dokumentów, prezentacji oraz e-maili, ułatwiając w ten sposób dostęp do serwisu.

Jeśli serwis działa w trybie online, można kliknąć na ikonie , aby przejść bezpośrednio do niego online.

4. Aby serwis mógł być używany jako serwis osadzony, ustawić dla opcji **Serwis osadzalny** wartość **Tak**.
5. Aby zmienić obraz używany dla serwisu, kliknąć na karcie **Logo serwisu**, a następnie nacisnąć przycisk **Zmień**. Odszukać i wybrać obraz, który ma zostać użyty. Użyty obraz musi być w formacie .png, .jpeg, lub .jpg i powinien mieć proporcje 4:3 (kształt prostokątny). Najlepszy rozmiar to 300x225 pikseli, ponieważ mniejsze obrazy mogą być wyświetlane w postaci zniekształconej, a większe mogą mieć wpływ na wydajność.
6. Po zakończeniu nacisnąć przycisk **Zamknij**.

## Ustawienie właściwości dotyczących wyszukiwarek

Można określić słowa kluczowe i teksty pomagające wyszukiwarkom rozpoznawać zawartość serwisu.

Ustawienia optymalizacji pod kątem wyszukiwarek (SEO — Search Engine Optimization) można definiować na poziomie serwisu i na poziomie strony. Ustawienia poziomu serwisu dodają się do podobnych ustawień dla poszczególnych stron albo te ustawienia przesłaniają, jak opisano w poniższej tabeli.



Opcja	Poziom serwisu	Poziom strony
"Opis" lub "Opis strony"	Dostarcza ogólnych informacji o serwisie, które nie są w nim samym zawarte. Opis serwisu jest dołączany do każdej jego strony. Ten opis jest także używany jako opis strony startowej (opis poziomu strony), jeśli w obszarze <b>Ustawienia strony</b> nie ustawiono żadnej wartości dla strony startowej.	Dostarcza ogólnych informacji o stronie, które nie są w niej samej zawarte. Opis strony stanowi dodatek do opisu serwisu, który (opis serwisu) jest dołączany do każdej jego strony.
"Słowa kluczowe"	Określa terminy lub pojęcia, które odnoszą się do wszystkich stron serwisu. Wartości te są dodawane (dołączane) do słów kluczowych, określonych dla poszczególnych stron.	Określa terminy lub pojęcia, które odnoszą się do danej strony. Słowa kluczowe strony mogą być przydatne do identyfikowania terminów i pojęć, które nie występują w tekście strony lub które pojawiają się na obrazach.
"Nagłówek" lub "Nagłówek strony"	Dodawanie skryptów lub tagów na potrzeby analiz lub śledzenia serwisu. Zawartość nagłówka serwisu jest dołączana do każdej jego strony.	Dodawanie skryptów lub tagów na potrzeby analiz lub śledzenia serwisu. Zawartość nagłówka strony stanowi dodatek do zawartości nagłówka serwisu, która jest dołączana do każdej jego strony.
"Stopka" lub "Stopka strony"	Dodawanie skryptów lub tagów na potrzeby analiz lub śledzenia serwisu. Zawartość stopki serwisu jest dołączana do każdej jego strony.	Dodawanie skryptów lub tagów na potrzeby analiz lub śledzenia serwisu. Zawartość stopki strony stanowi dodatek do zawartości stopki serwisu, która jest dołączana do każdej jego strony.

Jeśli poniższe opcje wykluczania wyszukiwania zostaną wybrane na poziomie serwisu, ustawienie to będzie się odnosiło do wszystkich stron i przesłoni ustawienia poszczególnych

stron. Jeśli opcja nie zostanie wybrana na poziomie serwisu, to będzie się ona odnosiła tylko do tych stron, dla których będzie indywidualnie użyta.

Opcja	Poziom serwisu	Poziom strony
"Ukryj przed wyszukiwarkami"	Jeśli ta opcja zostanie wybrana, do każdej strony zostanie dodany metaznacznik NOINDEX, aby przeglądarki nie indeksowały zawartości żadnej strony tego serwisu. Serwis ani żadna z jego stron nie pojawią się wówczas w wynikach wyszukiwania internetowego.	Jeśli ta opcja zostanie wybrana, do bieżącej strony zostanie dodany metaznacznik NOINDEX, aby przeglądarki nie indeksowały zawartości tej strony. Strona ta nie pojawi się wówczas w wynikach wyszukiwania internetowego.
Ukryj łącza do stron przed wyszukiwarkami	Jeśli ta opcja zostanie wybrana, do każdej strony zostanie dodany metaznacznik NOFOLLOW, aby przeglądarki nie podążały za łączami (i nie indeksowały miejsc docelowych) występującymi na dowolnej stronie tego serwisu.	Jeśli ta opcja zostanie wybrana, do bieżącej strony zostanie dodany metaznacznik NOFOLLOW, aby przeglądarki nie podążały za łączami (i nie indeksowały miejsc docelowych) występującymi na tej stronie.
Ukryj opisy stron przed wyszukiwarkami	Jeśli ta opcja zostanie wybrana, do każdej strony zostanie dodany metaznacznik NOSNIPPET, aby przeglądarki nie dołączały opisu (podany powyżej) po stronie pokazywanej w wynikach wyszukiwania.	Jeśli ta opcja zostanie wybrana, do bieżącej strony zostanie dodany metaznacznik NOSNIPPET, aby przeglądarki nie dołączały opisu (podany powyżej) po tej stronie pokazywanej w wynikach wyszukiwania.

Aby zmienić ustawienia optymalizacji pod kątem wyszukiwarek (SEO), należy:

- Otworzyć serwis do edycji.
- Kliknąć na ikonie  na pasku bocznym, po czym kliknąć na ikonie **SEO** .
- Podać opcjonalny opis serwisu. Opis serwisu jest dołączany do każdej jego strony. Ten opis jest także używany jako opis strony startowej (opis poziomu strony), jeśli w obszarze **Ustawienia strony** nie ustawiono żadnej wartości dla strony startowej.
- Opcjonalnie: określić rozdzielone przecinkiem słowa kluczowe pomagające wyszukiwarkom rozpoznawać zawartość serwisu.  
Za pomocą słów kluczowych są określane terminy lub pojęcia, które odnoszą się do wszystkich stron serwisu. Wartości te są dodawane (dołączane) do słów kluczowych, określonych dla poszczególnych stron za pomocą opcji **Ustawienia strony**.
- Opcjonalnie: dodać do nagłówka skrypty lub tagi na potrzeby analiz lub śledzenia serwisu. Zawartość nagłówka serwisu jest dołączana do każdej jego strony. Każdy kod, użyty w nagłówku, trzeba dokładnie sprawdzić, aby się upewnić, że działa poprawnie i nie stwarza zagrożenia dla serwisu.
- Opcjonalnie: dodać do stopki skrypty lub tagi na potrzeby analiz lub śledzenia serwisu. Zawartość stopki serwisu jest dołączana do każdej jego strony. Każdy kod, użyty w stopce, trzeba dokładnie sprawdzić, aby się upewnić, że działa poprawnie i nie stwarza zagrożenia dla serwisu.



7. Opcjonalnie: wybrać jedną lub kilka opcji w celu wykluczenia informacji z pojawiających się wyników wyszukiwania (zgodnie z opisem podanym w poprzedniej tabeli).
8. Aby zapisać wszystkie oczekujące zmiany w bieżącej aktualizacji, nacisnąć przycisk **Zapisz**.

## Włączanie usługi Prerender Service na potrzeby optymalizacji pod kątem wyszukiwarek (SEO)

W serwisach tworzonych za pomocą Oracle Content Management można włączyć funkcję wstępnego renderowania statycznych stron HTML, tak aby reagowały na żądania robotów przeszukujących.

Jeśli usługa Prerender Service jest włączona i żądanie pochodzi z robota przeszukującego, strona jest wyszukiwana w pamięci podręcznej.

- Jeśli strona zostanie znaleziona w pamięci podręcznej i będzie nie starsza niż 15 dni, to żądanie zostanie obsłużone przez stronę znajdującą się w pamięci podręcznej.
- Jeśli strona w pamięci podręcznej jest starsza niż 15 dni lub jeśli strona nie zostanie znaleziona w pamięci podręcznej, to jest pobierana z usługi Oracle Content Management nowa wersja, pamięć podręczna jest aktualizowana oraz jest udostępniana nowa wstępnie renderowana strona.
- Jeśli strona nie zostanie znaleziona w pamięci podręcznej ani w usłudze Oracle Content Management, to jest zwracana strona błędu.

Wygasłe strony w serwisach publicznych lub nowych serwisach, które jeszcze nie są przechowywane w pamięci podręcznej, są renderowane raz dziennie. Jeśli strona jest już przechowywana w pamięci podręcznej, to nie aktualizuje pamięci podręcznej. Po włączeniu usługi Prerender Service wstępnie renderowana pamięć podręczna jest aktualizowana co godzinę dla serwisów, które zostały zaktualizowane i ponownie opublikowane. Jeśli użytkownik dokonuje zmian w stronach serwisu zmiany i chce przetestować, jak te zmiany wpłyną na optymalizację, może w ustawieniach serwisu ręcznie zaktualizować wstępnie renderowaną pamięć podręczną.

1. Z nawigacji bocznej wybrać opcję **Serwisy**.
2. Wybrać serwis, po czym wybrać z menu podręcznego opcję **Właściwości** albo nacisnąć na pasku czynności przycisk **Pasek boczny**, a następnie wybrać opcję **Właściwości**.
3. Na pasku bocznym "Właściwości" wybrać opcję **SEO**.
4. Zostanie wyświetlona data i godzina ostatniego odświeżenia pamięci podręcznej. Aby zaktualizować pamięć podręczną, nacisnąć przycisk **Odśwież teraz**. Na pasku postępu będzie pokazywany status buforowania.

### Uwaga:

Jeśli (zgodnie z ustawieniami SEO dla serwisu) strona jest ukryta przed wyszukiwarkami, to nie jest wstępnie renderowana ani nie jest przechowywana w pamięci podręcznej.

Aby włączyć usługę Prerender Service w Oracle Content Management, należy:

1. W obszarze "Administrowanie" nawigacji bocznej wybrać opcję **System**.
2. Z menu "Ustawienia systemu" wybrać opcję **Ustawienia SEO dla serwisów**.
3. Nacisnąć przycisk **Włączone**.
4. Zdefiniować wymaganych dodatkowych użytkowników-agentów.

**Tabela 11-1 Trwale zakodowani użytkownicy-agenci w Oracle Content Management**

Użytkownik-agent	Użytkownik-agent	Użytkownik-agent	Użytkownik-agent
baiduspider	facebookexternalhit	twitterbot	rogerbot
linkedinbot	embedly	quora link preview	showyoubot
outbrain	pinterest/0.	developers.google.co m/+web/snippet	slackbot
vkShare	W3C_Validator	redditbot	Applebot
WhatsApp	flipboard	tumblr	bitlybot
SkypeUriPreview	nuzzel	Discordbot	Google Page Speed
Qwantify	pinterestbot		

**Tabela 11-2 Dodatkowi, wstępnie skonfigurowani użytkownicy-agenci we właściwości "PrerenderUserAgents" pliku config.cfg**

Użytkownik-agent	Użytkownik-agent	Użytkownik-agent	Użytkownik-agent
AddSearchBot	AdIdxBot	AdsBot-Google	AdsBot-Google-Mobile-Apps
AppEngine-Google	Baidu-YunGuanCe	Bingbot	BingPreview
DuckDuckBot	DuckDuckGo-Favicons-Bot	endeca webcrawler	Exabot
Facebot	Feedfetcher-Google	FeedValidator	Fetch
FlipboardProxy	Google Favicon	Google Web Preview	Google-Adwords-Instant
Googlebot	Googlebot-Image	Googlebot-Mobile	Googlebot-News
Googlebot-Video	Google-PhysicalWeb	Google-Structured-Data-Testing-Tool	HubSpot,ia_archiver
Mediapartners-Google	MSNBot	NetcraftSurveyAgent	nutch
Oracle Secure Enterprise Search	pinterest.com	PIs-Google	SEOkicks
seoscanners	Siteimprove.com	Slurp	Sogou web spider
VSE/1.0	W3C_CSS_Validator	W3C_I18n-Checker	W3C_Unicorn
W3C-checklink	W3C-mobileOK	Y!J	Yahoo Link Preview
Yahoo! Slurp	Yandex	YandexBot	YunGuanCe

## Ustawianie opcji dostarczania serwisu statycznego

Jeśli firma używa skompilowanych serwisów, można kontrolować, jak długo serwisy statyczne są przechowywane w pamięci podręcznej oraz jakiego rodzaju funkcje mobilni użytkownicy-agenci wywołują dla adaptacyjnych układów mobilnych obsługiwanych przez kompilację serwisu.

- Włączanie automatycznej kompilacji w trakcie publikowania
- Przesłanie domyślnych nagłówków Cache-Control skompilowanych serwisów
- Określanie mobilnych użytkowników-agentów do obsługi skompilowanych adaptacyjnych układów

## Włączanie automatycznej kompilacji w trakcie publikowania

Oracle Content Management zawiera wbudowaną usługę kompilacji, umożliwiającą łatwe skompilowanie serwisu bez żadnych dodatkowych konfiguracji. Administrator usługi może także skonfigurować ręcznie usługę kompilacji do celów testowych (np. weryfikacji kompilacji serwisu) lub do używania bibliotek niestandardowych. Bez względu na używaną usługę kompilacji można wybrać kompilowanie serwisu, gdy jest on publikowany lub ponownie publikowany.

1. Po zalogowaniu się do usługi Oracle Content Management wybrać opcję **Serwisy** (boczne menu nawigacyjne).
2. Wybrać serwis, który ma być modyfikowany, po czym nacisnąć przycisk **Właściwości**.
3. W oknie dialogowym "Właściwości" wybrać opcję **Statyczne dostarczanie**.
4. W sekcji **Włączanie automatycznej kompilacji** włączyć opcję **Kompiluj serwis po opublikowaniu**.
5. Po zakończeniu nacisnąć przycisk **Gotowe**.

Po opublikowaniu lub ponownym opublikowaniu serwisu, gdy jest włączona funkcja automatycznej kompilacji, status publikacji jest śledzony i wyświetlany w kafelku serwisu na stronie **Serwisy**

. Po ukończeniu publikacji proces kompilacji jest śledzony w sekcji **Dostarczanie statyczne** okna dialogowego "Właściwości serwisu". Gdy kompilacja zostanie ukończona, w sekcji "Statyczne dostarczanie" właściwości serwisu zostanie wyświetlona data i godzina ostatniej kompilacji oraz zostanie udostępnione łącze do pobrania dziennika kompilacji.

## Przesłanie domyślnych nagłówków Cache-Control skompilowanych serwisów

Domyślnie skompilowane serwisy są przechowywane w podręcznej pamięci przeglądarki użytkownika przez 300 sekund (5 minut). Administrator usługi może zmienić tę wartość domyślną, lecz będąc programistą serwisu, można przesłonić — w ustawieniach serwisu — wartość domyślną dla określonego serwisu.

1. Po zalogowaniu się do usługi Oracle Content Management wybrać opcję **Serwisy** (boczne menu nawigacyjne).
2. Wybrać serwis, który ma być modyfikowany, po czym nacisnąć przycisk **Właściwości**.
3. W oknie dialogowym "Właściwości" wybrać opcję **Statyczne dostarczanie**.
4. W sekcji **Nagłówki buforowania odpowiedzi** wpisać `Cache-control: max-age=`, po czym podać liczbę sekund przechowywania strony w pamięci podręcznej przeglądarki użytkownika. Na przykład ustawienie `Cache-control: max-age=600` spowoduje, że strona będzie przechowywana w pamięci podręcznej przez 10 minut.
5. Po zakończeniu nacisnąć przycisk **Gotowe**.

Jeśli instancja używa Akamai, pozostawić — aby zachować istniejące ustawienia Akamai — pustą sekcję **Nagłówki buforowania odpowiedzi**. Aby przesłonić istniejące ustawienia

Akamai, podać `Edge-Control: !no-store,max-age=1800,downstream-ttl=1800`, gdzie elementy wyróżnione czcionką pogrubioną są ustawieniami domyślnymi podanymi w sekundach.

- `!no-store` sygnalizuje, że to ustawienie powinno dla tej właściwości przesłonić odpowiednią konfigurację buforowania Akamai.
- `max-age` określa, jak długo Akamai przechowuje tę stronę w pamięci podręcznej. Wartością domyślną jest 1800 sekund (30 minut). W tym czasie Akamai spełni żądania dotyczące strony bez konieczności występowania o stronę z usługi Oracle Content Management.
- `downstream-ttl` informuje Akamai, że trzeba wysłać do przeglądarek klientów nagłówek "Cache-Control: max-age" wraz z jego odpowiedzią, zlecając tym przeglądarkom przechowywanie strony w pamięci podręcznej przez określony czas. Wartością domyślną jest 1800 sekund (30 minut).

## Określanie mobilnych użytkowników-agentów do obsługi skompilowanych adaptacyjnych układów

Podczas kompilowania serwisu można tworzyć strony mobilne do obsługi adaptacyjnych układów. Można określić użytkowników-agentów, którzy spowodują, że serwer będzie dostarczał — zamiast standardowych skompilowanych stron serwisu — strony mobilne. Wprowadzone tu wartości są traktowane jako fragmenty napisowe, w których — gdy są uzgadniane z wysyłanymi przez przeglądarkami nagłówkami użytkowników-agentów — nie jest uwzględniana wielkość liter. Należy zwrócić uwagę, że słowo kluczowe `Mobile` jest powszechnie używane w napisach użytkowników-agentów dla przeglądarek z urządzeń mobilnych.

1. Po zalogowaniu się do usługi Oracle Content Management wybrać opcję **Serwisy** (boczne menu nawigacyjne).
2. Wybrać serwis, który ma być modyfikowany, po czym nacisnąć przycisk **Właściwości**.
3. W oknie dialogowym "Właściwości" wybrać opcję **Statyczne dostarczanie**.
4. W sekcji **Mobilni użytkownicy-agenti** podać listę fragmentów napisów użytkowników-agentów (rozdzielając wpisy przecinkiem), którym mają być udostępniane strony mobilne. Jeśli którakolwiek część fragmentu napisu jest zgodna z napisem użytkownika-agenta przeglądarki, to wówczas strony mobilne będą udostępniane.

## Określanie i konfigurowanie zindywidualizowanych adresów URL

Konfigurując zindywidualizowany adres URL, umożliwia się użytkownikom uzyskiwanie dostępu do serwisu Oracle Content Management w określonej domenie, a nie w domenie instancji Oracle Content Management. Na przykład adresem URL instancji może być `https://myinstance.cec.ocp.oraclecloud.com/site/MyCustomerSite/`, lecz bardziej przyjazny adres URL, taki jak `https://www.example.com`, jest łatwiejszy do zapamiętania, potencjalnie lepszy do promocji marki oraz generalnie łatwiejszy do użycia. W zależności od wymagań serwis Oracle Content Management może być hostowany z użyciem niestandardowej ścieżki, takiej jak `https://www.example.com/store/`

Aby móc korzystać ze zindywidualizowanych adresów, trzeba wykonać kilka czynności.

- Użyć sieci CDN (Content Delivery Network).
- Zarządzać zindywidualizowaną domeną za pomocą usługi DNS (Domain Name System), tak aby rekord nazwy kanonicznej (CNAME) był odwzorowywany na sieć CDN.
- Wdrożyć w sieci CDN ważny certyfikat, mający na celu ochronę zindywidualizowanej domeny.
- Skonfigurować serwis Oracle Content Management tak, aby akceptował żądania z użyciem zindywidualizowanej domeny.
- Skonfigurować sieć CDN, tak aby poprawnie kierowała żądania do usługi Oracle Content Management.

### Używanie sieci CDN (Content Delivery Network)

Sieć CDN (Content Delivery Network) to platforma obsługi globalnie rozproszonych serwerów, której zadaniem jest poprawa wydajności i bezpieczeństwa serwisów internetowych. CDN minimalizuje odległość między użytkownikami i serwerami, optymalizując jednocześnie wydajność żądań dotyczących zawartości. Mimo że głównym celem sieci CDN jest poprawa obsługi użytkowników, to jednak sieć CDN może być także używana do modyfikowania przesyłanych żądań, tak aby użytkownik widział oczyszczone zasoby, nawet jeśli działający na zapleczu proces ich takimi nie widzi.

Chcąc zapewnić obsługę hostowania serwisu Oracle Content Management w zindywidualizowanej domenie, trzeba skonfigurować sieć CDN, tak aby obsługiwała wszystkie żądania przychodzące ze skonfigurowanej zindywidualizowanej domeny, poprawnie je kierowała z powrotem do usługi Oracle Content Management oraz tak je modyfikowała, aby były poprawnie i bezpiecznie obsługiwane przez usługę Oracle Content Management.

### Zarządzanie domeną za pomocą usługi DNS (Domain Name System)

Jako zindywidualizowana domena dla serwisu Oracle Content Management może zostać użyta dowolna domena. Przed przystąpieniem do konfigurowania zindywidualizowanej domeny dla serwisu Oracle Content Management trzeba sobie zapewnić jej kontrolę.

Ze względu na ograniczenia systemu DNS, użycie domeny głównej (takiej jak *example.com*) bez prefiksu *www* lub innej poddomeny (takiej jak *store.example.com*) jest prawdopodobnie niemożliwe. Aby ustalić, czy używanie domeny głównej jest możliwe, należy się skontaktować z dostawcami usług DNS i sieci CDN.

Ponieważ DNS działa nie na poziomie ścieżki, lecz na poziomie domeny, to — aby usługa Oracle Content Management hostowała niektóre ścieżki domeny, a inna usługa hostowała inne ścieżki — rozsyłanie musi być obsługiwane przez sieć CDN. Usługi DNS można używać tylko do segregowania ruchu na poziomie domeny i poddomeny.

### Wdrażanie certyfikatów

Certyfikat chroniący zindywidualizowaną domenę musi zostać utworzony i być hostowany przez sieć CDN. Certyfikat może chronić jedną domenę, kilka domen lub poddomen, a także domeny określone przy użyciu wieloznaczników, takie jak *\*.example.com*. Dla zindywidualizowanej domeny jest akceptowana dowolna kombinacja. Wszystkie chronione domeny będą widoczne w szczegółach certyfikatu, a zatem jeśli nie jest zamierzone publiczne udostępnianie tych informacji, należy używać osobnych certyfikatów.

 **Uwaga:**

Proces tworzenia i hostowania certyfikatów jest często specyficzny dla danej sieci CDN; jej dostawca powinien określić, jak najlepiej to zrobić.

### Konfigurowanie serwisu Oracle Content Management do użycia ze zindywidualizowaną domeną

Aby serwis Oracle Content Management został poprawnie załadowany, gdy jest używana zindywidualizowana domena, trzeba najpierw powiadomić usługę Oracle Content Management, jakiego adresu URL serwisu używa ta zindywidualizowana domena. Wykonuje się to w usłudze Oracle Content Management, na karcie właściwości serwisu.

1. W usłudze Oracle Content Management wybrać opcję **Serwisy** (boczna nawigacja).
2. Wybrać serwis, z którym będzie używana zindywidualizowana domena, wybrać z menu podręcznego lub z paska bocznego (na pasku czynności) opcję **Właściwości**.
3. Wprowadzić w polu "Zindywidualizowana domena" zindywidualizowaną domenę, po czym nacisnąć przycisk **Zapisz**.

 **Uwaga:**

Może upłynąć godzina lub więcej, zanim usługa Oracle Content Management będzie gotowa do przyjmowania żądań z użyciem tej domeny. Postęp można w dowolnej chwili sprawdzić w panelu właściwości serwisu.

### Konfigurowanie sieci CDN (Content Delivery Network) do rozsyłania żądań

Gdy usługa Oracle Content Management będzie już poprawnie skonfigurowana i gotowa do rozsyłania żądań, żądania wychodzące ze zindywidualizowanej domeny będą kierowane poprzez usługę DNS do sieci CDN, która musi poprawnie je przekazywać do usługi Oracle Content Management. Na przykład, jeśli serwis Oracle Content Management, którego adresem URL jest `https://myinstance.cec.ocp.oraclecloud.com/site/MyCustomerSite/`, zostanie skonfigurowany z użyciem zindywidualizowanej domeny `https://www.example.com/store`, to trzeba skonfigurować sieć CDN tak, aby:

- rozpoznawała zindywidualizowaną domenę `https://www.example.com/store`
- rozpoznawała źródłową instancję Oracle Content Management za pomocą zindywidualizowanej domeny `https://myinstance.cec.ocp.oraclecloud.com/`
- dołączała do określonego serwisu ścieżkę serwisu `site/MyCustomerSite/`
- i wysyłała do źródłowej instancji Oracle Content Management pełny adres URL `https://myinstance.c`
- Oracle Content Management odbiera żądanie i odpowiada sieci CDN, która spełnia żądanie w przeglądarce użytkownika, pokazując mu jedynie zindywidualizowaną domenę `https://www.example.com/store`

Etapy konfiguracji sieci CDN są często specyficzne dla danej sieci CDN i dlatego należy współpracować z dostawcą sieci CDN przy konfigurowaniu definicji źródła, która informuje sieć CDN, w jaki sposób ma uzyskać dostęp do źródłowej instancji CDN Oracle Content Management, oraz określa dołączaną ścieżkę serwisu.

 **Uwaga:**

Żądania nie mogą zawierać żadnej z następujących ścieżek:

- /documents\*
- /content/published\*
- /osn\*
- /pxysvc\*
- /site\*

Żądanie kierowane z użyciem tych ścieżek nie będą zawierały ścieżki serwisu.

Będą — aby zapewnić ich poprawną obsługę — przekładane na źródłową instancję Oracle Content



## Włączanie integracji z usługą Cobrowse

Funkcja Cobrowse (wspólne przeglądanie) stanowi narzędzie do pracy zespołowej, używane z usługą Oracle Cobrowse Cloud Service.

Integracja z usługą Cobrowse Cloud Service musi najpierw zostać dodana jako integracja zaakceptowana przez administratora usługi. Zob. rozdział *Integrowanie z Oracle Cobrowse Cloud Service* w podręczniku *Integrowanie i rozszerzanie usługi Oracle Content Management*.

Po włączeniu integracji z usługą Oracle Cobrowse Cloud Service można skonfigurować funkcję dla serwisu, a następnie dodać ją do określonych stron serwisu.

Aby włączyć w serwisie funkcję wspólnego przeglądania, należy:

1. Otworzyć serwis do edycji.
2. Kliknąć na ikonie  na pasku bocznym, po czym kliknąć na ikonie .
3. W sekcji usługi Cobrowse, wybrać opcję **Włącz używanie usługi Oracle Cobrowse w tym serwisie**.
4. Wprowadzić skrypt procesu uruchamiającego dla serwisu. Administrator usługi Oracle Cobrowse Cloud Service ma dostęp do konsoli administrowania (usługą Cobrowse), za pomocą której może uzyskać odpowiedni wycinek skryptu JavaScript procesu uruchamiającego. Dostępne są dwa różne typy procesów uruchamiających.
  - Launch Point 1: Przycisk Cobrowse jest automatycznie dodawany na stronie.
  - Launch Point 2: Umożliwia dostosowanie przycisku i interfejsu, który jest dodawany na stronie.
5. Nacisnąć przycisk **Zamknij**.

Po włączeniu usługi Cobrowse dla serwisu, można ją dodać do strony lub dostosować sposób jej używania na stronie. Aby uzyskać szczegółowe informacje, zob. [Korzystanie z usługi Cobrowse na stronie](#).

#### Korzystanie z usługi Cobrowse w serwisach zabezpieczonych lub podczas testowania przy użyciu konstruktora serwisów

Usługa Cobrowse działa w dwóch trybach: natychmiastowym (ICB) i zaawansowanym (ACB). Aby można było używać usługi Cobrowse w trybie ICB — w przypadku serwisów zabezpieczonych lub w celu uzyskania podglądu serwisu będącego na etapie tworzenia — jest wymagana dodatkowa konfiguracja. Tę konfigurację można przeprowadzić w konsoli administrowania usługą Cobrowse. Zob. *Logowanie do konsoli agenta* w podręczniku [Używanie autonomicznej usługi Cobrowse](#).

W polu "Funkcje niestandardowe" należy dodać funkcję umożliwiającą agentom zobaczenie zasobów chronionych hasłem w opublikowanym serwisie zabezpieczonym:

```
function () {
return {
passwordProtectedPatterns: [
"<PROTOCOL>://<DOMAIN>/authsite/*?*\"",
"<PROTOCOL>://<DOMAIN>/documents/*?*\"",
"<PROTOCOL>://<DOMAIN>/content/*?*#*\""
]
}
}
```

Aby można było skorzystać z tej funkcji w przypadku serwisów zabezpieczonych i w celu wyświetlenia/uzyskania podglądu serwisu będącego na etapie tworzenia, należy dodać dodatkowy kod:

```
function () {
return {
passwordProtectedPatterns: [
"<PROTOCOL>://<DOMAIN>/authsite/*?*\"",
"<PROTOCOL>://<DOMAIN>/documents/*?*\"",
"<PROTOCOL>://<DOMAIN>/content/*?*#*\"",
"<PROTOCOL>://<DOMAIN>/sites/*?*#*\"",
"<PROTOCOL>://<DOMAIN>/_themes/*?*\"",
"<PROTOCOL>://<DOMAIN>/_sitescloud/*?*\"",
"<PROTOCOL>://<DOMAIN>/_compdelivery/*?*\""
]
}
}
```



Konfiguracja zasobów chronionych jest nową funkcją usługi Cobrowse. Funkcja ta używa tych samych wieloznacznych wzorców adresu URL co maskowanie stron Cobrowse. Więcej szczegółów można znaleźć w rozdziale [Configure page masking](#) w podręczniku *Cobrowse Deployment and Use Guide*.



## Dodawanie śledzenia analiz

W konstruktorze serwisów można do serwisu lub strony dodać snippet — zawierający kod JavaScript śledzenia — w celu śledzenia analiz internetowych i ułatwienia integracji z zewnętrznymi dostawcami analiz, takimi jak Google, Adobe lub Oracle Infinity.

Aby dodać obiekty śledzenia analiz do serwisów, należy:

1. Otworzyć serwis do edycji.
2. Kliknąć na ikonie  na pasku bocznym, po czym kliknąć na ikonie **Analizy** ().
3. Aby włączyć śledzenie analiz, kliknąć na przełączniku.
4. W polu **Snippet śledzenia (JavaScript)** dodać nowy snippet lub edytować już istniejący skrypt.  
Administrator może dostarczyć dla środowiska snippet kodu. W takim przypadku kod ten pojawi się w polu. Można dostosować skrypt lub dodać własny. Jeśli użytkownik edytuje snippet dostarczony przez administratora, pojawi się komunikat informujący, że skrypt został zmodyfikowany. Aby usunąć swoje dostosowania, należy użyć opcji **Przywróć do najnowszej wersji skryptu dzierżawy**.

Oto przykład snippetu śledzenia Google Analytics:

```
<!-- Global site tag (gtag.js) - Google Analytics -->
<script async src="https://www.googletagmanager.com/gtag/js?
id=UA-85172963-3"></script>
<script>
  window.dataLayer = window.dataLayer || [];
  function gtag(){dataLayer.push(arguments);}
  gtag('js', new Date());

  gtag('config', 'UA-85172963-3');
</script>
```

Trzeba zapisać i opublikować tę zmianę oraz, zanim zaczną być gromadzone analizy dotyczące serwisu, przełączyć serwis do trybu online (jeśli trzeba).

### Wyświetlanie danych analitycznych

Po opublikowaniu serwisu i przełączeniu go do trybu online można przeglądać śledzone dane analityczne dla serwisu dostawcy, takiego jak Google Analytics. Jeśli został użyty snippet do śledzenia analiz Oracle Infinity, należy przejść na stronę startową Oracle Infinity, po czym — aby wyświetlić dane i wybrać lub utworzyć raporty — wybrać opcję **Analytics**.

## Publikowanie serwisów

Informacje o przełączaniu serwisów do trybu online lub offline oraz o publikowaniu zmian serwisu.



- [Przełączanie serwisu do trybu online lub offline](#)
- [Publikowanie zmian serwisu](#)

### Przełączanie serwisu do trybu online lub offline

Gdy serwis działa w trybie *online*, użytkownicy z właściwym dostępem mogą wyświetlać serwis przy użyciu standardowej przeglądarki internetowej, korzystając z jego adresu URL. Gdy serwis zostanie przełączony do trybu *offline*, przestanie być dostępny do publicznego odbioru. Serwis będzie można zobaczyć tylko w Oracle Content Management.



Serwis, zanim będzie można go przełączyć do trybu online, trzeba najpierw opublikować. Aby opublikować serwis, zob. [Publikowanie zmian serwisu](#).

Ikona statusu (z prawej strony) pokazuje, czy serwis jest serwisem online czy offline:

- Jeśli serwis nigdy nie był wcześniej opublikowany, będzie wyświetlany znak łącznika (-).
- Jeśli serwis jest serwisem online, będzie wyświetlana ikona .
- Jeśli serwis jest serwisem offline, będzie wyświetlana ikona .

Informacje, kto ma dostęp do serwisu będącego w stanie online lub offline, są dostępne pod hasłem [Omówienie zabezpieczeń serwisu](#).

Aby zmienić status serwisu, bieżący użytkownik musi być właścicielem serwisu lub mieć przypisaną rolę "Menedżer" w odniesieniu do tego serwisu. Jeśli funkcja zarządzania serwisem jest włączona, administratorzy serwisów — bez względu na to, czy serwis został im udostępniony czy nie — mogą także zmienić status dowolnego serwisu.

1. Na stronie "Serwisy" wybrać serwis z listy.
2. Aby przełączyć serwis do trybu online lub offline, wybrać z menu podręcznego opcję **Przełącz do trybu online** lub **Przełącz do trybu offline**. Można także kliknąć na kafelku serwisu na ikonie  (aby przełączyć serwis do trybu online) lub  (aby przełączyć serwis do trybu offline).

Zostanie wyświetlone wezwanie do potwierdzenia dokonanego wyboru.

Gdy serwis zostaje przełączony do trybu online, jest tworzona w pełni renderowana wersja HTML serwisu, która następnie jest kopiowana do lokalizacji w Oracle Cloud. Pod nazwą serwisu online jest pokazywany jego adres URL. Domyślny adres URL ma postać:


`https://service_name.identity_domain.sites.oraclecloud.com/site_name`

Serwis — gdy zostanie przełączony do stanu offline — jest, wraz ze swoimi folderami i plikami, zabierany z lokalizacji w Oracle Cloud.

## Publikowanie zmian serwisu

Zmiany, dokonane w serwisie, może opublikować tylko jego właściciel lub osoba mająca przypisaną rolę "Menedżer".

Podczas publikowania zmian w serwisie warto pamiętać o następujących zasadach:

- Wszelkie zatwierdzone zmiany — po ich opublikowaniu w serwisie, który działa w trybie online — stają się od razu widoczne dla każdego, kto ma dostęp do danego serwisu.
  - Publikując zmiany w serwisie, można opublikować wszystkie zasoby kierowane do kanału serwisu lub tylko te zasoby, które są używane na stronach serwisu, w tym rekomendacje. Na przykład, jeśli są publikowane wszystkie zasoby kierowane do kanału serwisu, a serwis ten obejmuje listę odnoszącą się do elementów zawartości, niebędących bezpośrednio częścią serwisu, to zostaną także opublikowane te zasoby, do których występują odniesienia.
  - Podczas publikowania zmian w serwisie wielojęzycznym zostaną opublikowane tłumaczenia tylko dla języków zdefiniowanych w założeniu systemowym dot. lokalizacji, powiązanych z tym serwisem.
  - Jeśli wcześniej zostały opublikowane tłumaczenia dla języków, które nie są już zdefiniowane w powiązanej z serwisem lokalizacji systemowej dot. lokalizacji, to przetłumaczona zawartość zostanie usunięta z opublikowanego serwisu.
  - Jeśli zostanie usunięta strona z serwisu wielojęzycznego, to — gdy serwis zostanie opublikowany — przetłumaczone strony także zostaną usunięte.
1. Aby opublikować zmiany w serwisie, wybrać go, po czym wybrać z menu podręcznego lub z paska czynności opcję **Publikuj**.
  2. Wybrać, czy opublikować serwis i wszystkie zasoby kierowane do kanału publikowania dla serwisu, czy publikować serwis i wszystkie jego zasoby dodane do stron serwisu.
  3. Oracle Content Management sprawdza, czy wszystkie napisy i zasoby serwisu zawierają wymagane zatwierdzenia i tłumaczenia. Nie można opublikować serwisu, jeśli brakuje tłumaczeń wymaganych przez powiązane z serwisem założenie systemowe dot. lokalizacji bądź zasoby nie zostały oznaczone jako przetłumaczone lub zasób wymaga zatwierdzenia, lecz nie został oznaczony jako zatwierdzony. Aby zobaczyć więcej szczegółów, rozwinąć elementy. Jeśli występują problemy, rozwiązać je, po czym ponowić próbę opublikowania. Jeśli wszystkie elementy są poprawne, nacisnąć przycisk **Publikuj**. Zostanie wyświetlone wezwanie do potwierdzenia dokonanego wyboru.
  4. Jeśli serwis nie jest jeszcze w trybie online () , przełączyć go do trybu online; stanie się serwisem dostępnym dla użytkowników.

## Serwisy zabezpieczone

Po opublikowaniu serwisu i udostępnieniu go online można kontrolować, kto ma dostęp do serwisu.

Gdy serwis jest zabezpieczony, można określić, które grupy użytkowników mogą uzyskiwać — na podstawie przypisanej im roli — dostęp do opublikowanego serwisu online. Są to role poziomu usługi, przypisywane przez administratora jej instancji.

- [Omówienie zabezpieczeń serwisu](#)
- [Zmianie zabezpieczeń serwisu](#)

Będąc administratorem, można wykonywać dodatkowe czynności związane z zabezpieczeniem serwisu, takie jak włączenie logowania niestandardowego, zezwolenie na udostępnianie serwisów i motywów, zezwolenie określonym osobom na tworzenie serwisów, szablonów i składników oraz włączenie funkcji zarządzania. Zob. *Konfigurowanie ustawień serwisów i zasobów w podręczniku [Administrowanie Oracle Content Management](#)*.

## Omówienie zabezpieczeń serwisu

Można zastosować zabezpieczenia pozwalające decydować, kto może zobaczyć opublikowany serwis (online), kto może zobaczyć zabezpieczoną zawartość serwisu i wchodzić z nią w interakcję oraz kto może zobaczyć i edytować nieopublikowany serwis (offline).

### Zabezpieczenia serwisu

Po opublikowaniu serwisu i udostępnieniu go online można kontrolować, kto ma dostęp do serwisu. W zależności od tego, w jaki sposób administratorzy serwisów i systemu skonfigurowali środowisko, serwis można uczynić publicznie dostępnym dla każdego, można ograniczyć dostęp do serwisu do zarejestrowanych użytkowników albo można ograniczyć dostęp do serwisu do określonych użytkowników.

Aby zmienić zabezpieczenia serwisu lub inne ustawienia, trzeba być właścicielem serwisu lub mieć przypisaną rolę "Menedżer" w odniesieniu do tego serwisu. Wymóg logowania można zmienić tylko wtedy, gdy serwis działa w trybie offline. Zmiana określonych użytkowników lub ich ról jest jednak możliwa także w trybie online. Serwis — gdy zostanie przełączony do stanu offline — jest, wraz ze swoimi folderami i plikami, zabierany z lokalizacji w Oracle Cloud.

Dostępne opcje zabezpieczeń mogą być ograniczone przez założenie systemowe dot. szablonu (jeśli funkcja zarządzania serwisem jest włączona) lub przez założenie systemowe dot. dzierżawy (jeśli funkcja zarządzania serwisem jest wyłączona). Zob. [Omówienie funkcji zarządzania serwisem](#).

Gdy serwis jest zabezpieczony, można określić, które grupy użytkowników mogą uzyskiwać — na podstawie przypisanej im roli — dostęp do opublikowanego serwisu online. Są to role poziomu usługi, przypisywane przez administratora jej instancji.

- **Użytkownicy chmury:** Zidentyfikowani użytkownicy logują się do instancji serwisu, używając nazwy użytkownika i hasła. Obejmuje to wszystkich zidentyfikowanych użytkowników z rolą lub bez roli **Odwiedzający** lub **Użytkownik** poziomu aplikacji .
- **Odwiedzający:** Dostęp do serwisu będą mieli tylko użytkownicy z tą rolą. Na przykład rola ta może zostać przypisana użytkownikom, którzy mogą zobaczyć opublikowane serwisy, ale nie mają dostępu do folderów i plików w tej instancji Oracle Content Management .

 **Uwaga:**

Nie są przy tym uwzględniani użytkownicy z rolą **Użytkownik**, chyba że użytkownik jest właścicielem serwisu albo serwis został udostępniony tym użytkownikom w sposób jawny.

- **Użytkownicy usługi:** Dostęp do serwisu będą mieli tylko użytkownicy z tą rolą. Na przykład ta rola może zostać nadana użytkownikom, którzy mogą zobaczyć opublikowane serwisy oraz mają prawa dostępu do folderów i plików w danej instancji usługi Oracle Content Management.
- **Określeni użytkownicy:** Opublikowany serwis mogą zobaczyć tylko użytkownicy dodani jako współużytkownicy serwisu.

#### Udostępnianie serwisu

Za pomocą ustawień *udostępniania serwisu* można określić indywidualnych użytkowników, którzy mogą uzyskiwać dostęp do nieopublikowanego serwisu offline, i zezwolić im — zgodnie z przypisanymi im uprawnieniami — na wyświetlanie lub modyfikowanie serwisu bądź zarządzanie nim. Użytkownik może udostępnić serwis, którego jest właścicielem lub który został udostępniony użytkownikowi z przypisaniem roli "Menedżer".

 **Uwaga:**

Każda rola związana z udostępnianiem i przypisana do użytkownika rozszerza jego rolę związaną z zabezpieczeniami. Na przykład, jeśli użytkownik ma przypisaną rolę **Odwiedzający**, lecz zostanie mu udostępniony serwis i przypisana rola "Współtwórca", to może modyfikować ten serwis w trybie offline, podczas gdy inni użytkownicy mający rolę **Odwiedzający** mogą jedynie zobaczyć ten serwis online.

- **Oglądający:** Osoby te mogą wyświetlać serwis w edytorze, lecz nie mogą niczego zmieniać.
- **Pobierający:** W odniesieniu do serwisu zapewnia te same uprawnienia co rola "Oglądający".
- **Współtwórca:** Ma te same uprawnienia co oglądający, a ponadto może edytować serwis i usuwać strony serwisu oraz może usunąć serwis przełączony do trybu offline.
- **Menedżer:** Ma te same uprawnienia co współtwórca, a ponadto może dodawać użytkowników i przypisywać im role, publikować zmiany do serwisu online oraz

przełączać serwis online/offline. Twórcy (właścicielowi) serwisu jest automatycznie przypisywana rola "Menedżer".

Gdy jest tworzony serwis, jest tworzony również kanał o nazwie identycznej z nazwą serwisu. Aby udostępnić kanał innej osobie, trzeba udostępnić serwis i nadać jej (jako minimum) rolę "Współtwórca", dzięki czemu osoba ta będzie mogła używać kanału do publikowania zasobów. Użytkownik, aby móc opublikować serwis, musi mieć przypisaną rolę "Menedżer".

### Udostępnianie składowika

Niektóre składowiki mogą umożliwiać dostęp do współużytkowanych zasobów, takich jak foldery, pliki lub konwersacje. *Udostępnianie składowika* uwzględnia zabezpieczenia serwisu (kto może wyświetlać opublikowany serwis) oraz udostępnianie zasobów (kto może wyświetlać foldery, pliki i konwersacje oraz pracować z nimi).

Na przykład, jeśli użytkownik doda do swojego serwisu składowik "menedżer dokumentu", wszyscy odwiedzający serwis będą mogli zobaczyć zawartość folderu oraz (w zależności od przypisanej im roli oraz nadanych innych uprawnień) będą mogli dodawać, modyfikować lub usuwać zawartość folderu.

Uwagi ogólne:

- Autor serwisu nie może do folderu przypisać praw dostępu wyższych niż te, które sam posiada. Na przykład, jeśli autor ma w odniesieniu do folderu uprawnienia "Pobierający", to nie może odwiedzającym serwis nadać uprawnień "Współtwórca".
- Uprawnienia odwiedzającego mogą być modyfikowane przez uprawnienia ustawione w składowiku. Na przykład, jeśli odwiedzający ma uprawnienia "Oglądający" (lub nie ma żadnych uprawnień) do folderu, składowik "menedżer dokumentów" może przypisać — w zależności od roli wybranej w składowiku — wyższe uprawnienia. Te rozszerzone uprawnienia obowiązują tylko w samym składowiku.
- Jeśli osoba odwiedzająca serwis ma uprawnienia wyższe niż określone dla składowika, to ustawienia w składowiku zostaną przesłonięte przez indywidualne uprawnienia tej osoby.
- Uprawnienia, nadane w odniesieniu do folderu, są stosowane do folderów i plików zagnieżdżonych w tym folderze.

W przypadku serwisów publicznych:

- Składowiki konwersacji są obsługiwane tylko w zabezpieczonych serwisach.
- Składowiki "menedżer dokumentów" wyposażają domyślnie wszystkich odwiedzających w uprawnienia "Pobierający" odnoszące się do powiązanego folderu. Za pomocą ustawień samego składowika można zmienić rolę, bazując na zamieszczonych powyżej wskazówkach, oraz można ograniczyć opcje prezentowane użytkownikowi.
- Składowiki "lista folderów" i "lista plików" nadają wszystkim użytkownikom uprawnienia "Pobierający". Użytkownicy, niezależnie od przypisanej im roli, mogą wyświetlać i pobierać pliki.

### URL zabezpieczonych serwisów

Gdy serwis zostaje przełączony do trybu online, jest tworzona w pełni renderowana wersja HTML serwisu, która następnie jest kopiowana do lokalizacji w Oracle Cloud. Pod nazwą serwisu online jest pokazywany jego adres URL.

Domyślny adres URL niezabezpieczonego serwisu ma postać:

`https://service_name.identity_domain.sites.oraclecloud.com/site_name`

Domyślny adres URL zabezpieczonego serwisu ma postać:


```
https://service_name.identity_domain.sites.oraclecloud.com/authsite/  
site_name
```

Proszę zwrócić uwagę na element `authsite` adresu URL.

## Zmianianie zabezpieczeń serwisu

Po opublikowaniu serwisu i udostępnieniu go online można kontrolować, kto ma dostęp do serwisu. W zależności od tego, w jaki sposób administratorzy serwisów i systemu skonfigurowali środowisko, serwis można uczynić publicznie dostępnym dla każdego, można ograniczyć dostęp do serwisu do zarejestrowanych użytkowników albo można ograniczyć dostęp do serwisu do określonych użytkowników.

Aby zmienić zabezpieczenia serwisu lub inne ustawienia, trzeba być właścicielem serwisu lub mieć przypisaną rolę "Menedżer" w odniesieniu do tego serwisu. Wymóg logowania można zmienić tylko wtedy, gdy serwis działa w trybie offline. Zmiana określonych użytkowników lub ich ról jest jednak możliwa także w trybie online. Serwis — gdy zostanie przełączony do stanu offline — jest, wraz ze swoimi folderami i plikami, zabierany z lokalizacji w Oracle Cloud. W celu przełączenia serwisu do trybu offline należy wybrać serwis ze strony "Serwisy", po czym wybrać z menu


podręcznego opcję **Przełącz do trybu offline** albo kliknąć na ikonie  (na pasku czynności). Zostanie wyświetlone wezwanie do potwierdzenia dokonanego wyboru.

Aby zmienić zabezpieczenia serwisu, należy:

1. Na stronie "Serwisy" wybrać serwis, po czym wybrać z paska czynności lub z menu podręcznego (kliknięcie prawym przyciskiem myszy) opcję **Właściwości**.
2. Kliknąć na karcie **Zabezpieczenia serwisu**.

Opcje dostępne na karcie "Zabezpieczenia serwisu" zależą od tego, w jaki sposób administrator systemu skonfigurował ustawienia zabezpieczeń serwisu oraz (jeśli funkcja zarządzania serwisem jest włączona) w jaki sposób administrator serwisów skonfigurował ustawienia zabezpieczeń w szablonie, na którym ten serwis bazuje. Zob. [Serwisy — pierwsze kroki](#) i [Omówienie funkcji zarządzania serwisem](#).

3. Aby wprowadzić wymóg logowania się zarejestrowanych użytkowników (żeby mogli zobaczyć serwis online), wybrać opcję **Tak** obok ustawienia "Wymagane logowanie". Aby usunąć wymóg logowania się i uczynić serwis online publicznie dostępnym, wybrać opcję **Nie** obok ustawienia "Wymagane logowanie".
4. Wybrać, które grupy zarejestrowanych użytkowników mogą uzyskiwać dostęp do serwisu online. Aby można było wybierać poszczególne grupy, trzeba najpierw wyczyścić pole wyboru **Użytkownicy chmury**.
  - **Użytkownicy chmury**: Dostęp do serwisu będą mieli tylko zidentyfikowani użytkownicy. Zidentyfikowani użytkownicy logują się do domeny, używając nazwy użytkownika i hasła. Obejmuje to użytkowników z rolą **Odwiedzający Oracle Content Management Cloud** albo z rolą **Użytkownik Oracle Content Management Cloud**.
  - **Odwiedzający**: Dostęp do serwisu będą mieli tylko użytkownicy z tą rolą. Nie obejmuje to użytkowników z rolą **Użytkownik Oracle Content Management Cloud**.

- **Użytkownicy usługi:** Dostęp do serwisu mogą uzyskać tylko użytkownicy, którzy mogą się zalogować do tej instancji Oracle Content Management.
  - **Określeni użytkownicy:** Określić osoby, które będą miały dostęp do serwisu. Nacisnąć przycisk **Dodaj współużytkowników**. Wpisać w polu wyszukiwania nazwę użytkownika lub jej fragment. Wybrać użytkownika z wyświetlanej listy, po czym powtórzyć, aby dodać dalszych użytkowników. Po ukończeniu nacisnąć przycisk **Dodaj**. Aby usunąć użytkownika, wybrać z menu (pod nazwą użytkownika) opcję **Usuń**.
5. Nacisnąć przycisk **Zapisz**, aby zapisać zmiany i zamknąć okno.  
Jest pokazywana informacja, że serwis działa w trybie offline i że jest wymagane logowanie.
6. W celu przełączenia serwisu do trybu online wybrać z menu podręcznego opcję **Przełącz do trybu online** albo kliknąć na ikonie  (na pasku czynności). Wybrać opcję **Potwierdzenie kontynuacji**, po czym nacisnąć przycisk **OK**.

Gdy serwis zostaje przełączony do trybu online, jest tworzona w pełni renderowana wersja HTML serwisu, która następnie jest kopiowana do lokalizacji w Oracle Cloud. Pod nazwą serwisu online jest pokazywany jego adres URL.

Domyślny adres URL niezabezpieczonego serwisu ma postać:

```
https://service_name-identity_domain.cec.ocp.oraclecloud.com/site/site_name
```

Domyślny adres URL zabezpieczonego serwisu ma postać:

```
https://service_name-identity_domain.cec.ocp.oraclecloud.com/site/authsite/site_name
```

Proszę zwrócić uwagę na element authsite adresu URL.

Można dodać adres URL wylogowywania i zaimplementować go jako łącze, przycisk albo stronę, która będzie widoczna w menu. Zob. [Akapity](#), [Przyciski](#) i [Dodawanie stron](#).

Adres URL wylogowywania ma postać:

```
https://service_name-identity_domain.cec.ocp.oraclecloud.com/cloudgate/logout.html?postlogouturl=%2Fsite%2Fauthsite%2Fsite_name
```

 **Uwaga:**

Adres postlogouturl musi mieć zakodowany format, jak powyżej.



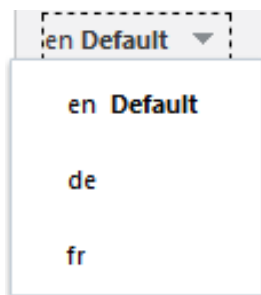
## Praca z serwisami wielojęzycznymi

Informacje o tłumaczeniach serwisów w celu zapewnienia ich wielojęzyczności.

- [Serwisy wielojęzyczne — omówienie](#)
- [Tłumaczenie serwisu](#)
- [Zarządzanie zleceniami tłumaczeń serwisu](#)

### Serwisy wielojęzyczne — omówienie

Serwis można przetłumaczyć na dowolny język, który został określony w założeniu systemowym dot. lokalizacji. W celu wyświetlenia języków, określonych w powiązonym z serwisem założeniu systemowym dot. lokalizacji, wystarczy — na kafelku serwisu — kliknąć na znajdującej się obok języka strzałce skierowanej w dół. Aby uzyskać podgląd lokalizowanej (tłumaczonej) wersji, należy wybrać język, po czym otworzyć serwis.



#### Uwaga:

Korzystając z usługi Oracle Content Management w edycji Starter, nie można tworzyć serwisów wielojęzycznych. Aby uzyskać dostęp do pełnego zestawu funkcji i nieograniczonej liczby serwisów, należy uaktualnić usługę Oracle Content Management do edycji Premium.

Po wybraniu serwisu do tłumaczenia zostanie utworzony plik .zip zawierający pliki serwisu.

#### Uwaga:

Jeśli serwis zawiera zasoby z więcej niż jednego repozytorium, plik .zip będzie zawierał tylko zasoby z repozytorium domyślnego.

Plik .zip zawiera następujące pliki i foldery:

- Folder *zasobów* — ten folder istnieje tylko wtedy, gdy do tłumaczenia zostanie wybrany ukończony serwis lub tylko zasoby docelowe.
  - Folder *główny*
    - \* Plik `<ID_elementu_zawartosci>.json` dla każdego elementu zawartości — zawiera nazwę, opis i możliwe do przetłumaczenia napisy z pól elementu zawartości, a także dodatkowe informacje o elemencie zawartości, których nie można edytować.
  - `job.json` — plik opisujący zlecenie tłumaczenia. Tego pliku nie należy edytować.
- Folder *serwisów* — ten folder istnieje tylko wtedy, gdy do tłumaczenia zostanie wybrany ukończony serwis lub tylko elementy zawartości.
  - Folder *główny*
    - \* Plik `<ID_strony>.json` dla każdej strony serwisu — zawiera nazwę, tytuł, opis, słowa kluczowe, nagłówek, stopkę i możliwe do przetłumaczenia napisy ze składników na stronie, a także dodatkowe informacje o stronie, których nie można edytować. Jeśli występują składniki niestandardowe, prawdopodobnie zostały one skonfigurowane do używania możliwych do przetłumaczenia napisów. Zob. [Tworzenie możliwych do przetłumaczenia składników dla serwisów wielojęzycznych](#).
    - \* `siteinfo.json` — zawiera opis, słowa kluczowe, nagłówek i stopkę.
    - \* `structure.json` — zawiera nawigację i strukturę serwisu.
  - `job.json` — plik opisujący zlecenie tłumaczenia. Tego pliku nie należy edytować.

## Tłumaczenie serwisu

Jeśli na stronie w przetłumaczonym serwisie występują zasoby (bezpośrednio albo na liście zawartości), a bieżący użytkownik je przetłumaczył, to zostaną one wyświetlone w tym samym języku, w którym jest wyświetlany serwis. Każdy element zawartości z serwisu można także przetłumaczyć osobno; zob. [Lokalizowanie elementów zawartości](#).

Jeśli jest tłumaczony serwis, który zawiera zasoby z więcej niż jednego repozytorium, w zleceniu tłumaczenia będą uwzględnione tylko zasoby z repozytorium domyślnego

### Uwaga:

Korzystając z usługi Oracle Content Management w edycji Starter, nie można pracować nad tłumaczeniami. Aby uzyskać dostęp do pełnego zestawu funkcji i nieograniczonej liczby serwisów, należy uaktualnić usługę Oracle Content Management do edycji Premium.

### Tworzenie zlecenia tłumaczenia

Gdy jest eksportowany pakiet tłumaczenia dla tłumaczenia ręcznego lub jest używany łącznik z tłumaczeniem, trzeba utworzyć zlecenie tłumaczenia.

1. Wybrać serwis, który ma zostać przetłumaczony, po czym nacisnąć przycisk **Tłumacz**. Aby wyświetlić opcję **Tłumacz**, trzeba niekiedy nacisnąć przycisk **Więcej**.
2. Wprowadzić informacje dotyczące zlecenia tłumaczenia, po czym nacisnąć przycisk **Utwórz** :
  - Wpisać nazwę zlecenia tłumaczenia.
  - Wybrać języki docelowe, na które serwis ma zostać przetłumaczony.
  - Wybrać, czy wyeksportować pełny serwis z zawartością (strony, strukturę i informacje o serwisie) i zasobami docelowymi, czy tylko zawartość serwisu bądź tylko zasoby docelowe.
  - Wybrać łącznik z tłumaczeniem lub opcję eksportowania pakietu tłumaczenia dla tłumaczenia ręcznego.
  - Po ukończeniu nacisnąć przycisk **Utwórz**.

Po zakończeniu zlecenia tłumaczenia przetłumaczony pakiet zostanie zaimportowany do usługi Oracle Content and Experience Oracle Content Management.

#### Ręczne tłumaczenie plików w języku źródłowym

Jeśli zawartość jest ręcznie tłumaczona, należy — po ukończeniu zlecenia tłumaczenia — pobrać plik .zip zawierający pliki serwisu.

1. Następnie nacisnąć na banerze przycisk **Zlecenia tłumaczeń**, wybrać zlecenie tłumaczenia, po czym nacisnąć przycisk **Pobierz**.
2. Dla każdego języka, wybranego jako język docelowy, utworzyć folder w pliku .zip na tym samym poziomie, na którym znajduje się folder `główny`, na przykład `de`, `es` i `fr`. Można tłumaczyć podzbiór wybranych języków. Na przykład można najpierw przetłumaczyć na język niemiecki (`de`), a później na język hiszpański (`es`) i francuski (`fr`).
3. Skopiować *wszystkie* pliki `.json` z folderu `głównego` do każdego folderu językowego.
4. Przetłumaczyć wartości napisowe we wszystkich plikach `.json` na odpowiednie języki. Nie należy usuwać żadnych wartości napisowych z plików `.json` ani zmieniać nazw plików.
5. Spakować w plik .zip folder `zasobów` (jeśli zlecenie tłumaczenia zawiera zasoby) i `serwisu` (jeśli zlecenie tłumaczenia zawiera zawartość serwisu), a także plik `job.json`, folder `główny` i wszystkie foldery językowe z przetłumaczonymi plikami.

#### Importowanie ręcznie przetłumaczonego pakietu


Niezależnie od tego, czy tłumaczenie zostało przeprowadzone ręcznie czy automatycznie, należy zaimportować zakończony pakiet tłumaczenia.

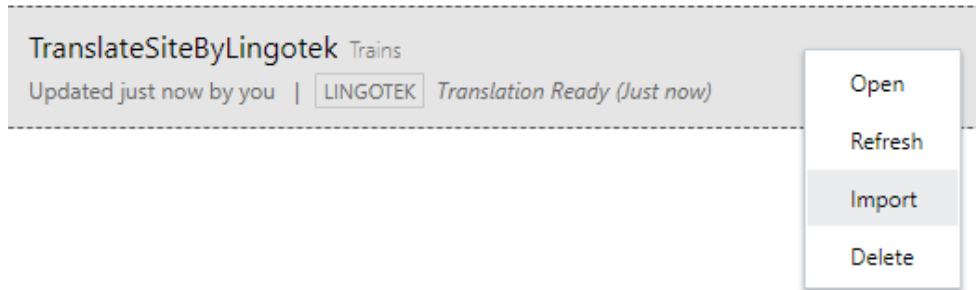
1. Na stronie **Zlecenia tłumaczeń** nacisnąć przycisk **Importuj**.
2. Nacisnąć przycisk **Wyślij**, wybrać plik .zip zawierający przetłumaczone pliki serwisu, po czym nacisnąć przycisk **Otwórz**.
3. Po zakończeniu wysyłania nacisnąć przycisk **OK**.
4. Usługa Oracle Content Management weryfikuje, czy tłumaczenia, które zostały zdefiniowane w zleceniu, są dostępne w pliku .zip. Aby zobaczyć, które strony i zasoby zostały zawarte w zleceniu tłumaczenia, należy kliknąć na łączu dostępnym w oknie dialogowym.
5. Gdy tłumaczenia będą gotowe do importowania, nacisnąć przycisk **Importuj**.

Status importu jest wyświetlany nad banerem. Szczegóły zlecenia można wyświetlić, wybierając opcję **Szczegóły**.

#### Importowanie przetłumaczonego pakietu z łącznika z tłumaczeniem

Niezależnie od tego, czy tłumaczenie zostało przeprowadzone ręcznie czy automatycznie, należy zaimportować zakończony pakiet tłumaczenia.

1. Kliknąć na ikonie , po czym wybrać opcję **Zlecenia tłumaczeń**.
2. Kliknąć prawym przyciskiem myszy na zakończonym zleceniu tłumaczenia, po czym wybrać opcję **Importuj**.



3. Usługa Oracle Content Management weryfikuje tłumaczenia, które zostały zdefiniowane w zleceniu. Aby zobaczyć, które strony i zasoby zostały zawarte w zleceniu tłumaczenia, należy kliknąć na łączu dostępnym w oknie dialogowym.
4. Gdy tłumaczenia będą gotowe do importowania, nacisnąć przycisk **Importuj**. Status importu jest wyświetlany nad banerem. Szczegóły zlecenia można wyświetlić, wybierając opcję **Szczegóły**.

Po zaimportowaniu tłumaczeń należy opublikować serwis oraz — aby tłumaczenia były w nim dostępne — upewnić się, że serwis ten działa w trybie online.

Jeśli po zaimportowaniu pakietu tłumaczenia w usłudze tłumaczeniowej zostaną wprowadzone jakiegokolwiek zmiany w tłumaczeniach, można zaktualizować tłumaczenia poprzez naciśnięcie przycisku **Odśwież**.

Jeśli serwis zostanie zmieniony po przetłumaczeniu go, trzeba przetłumaczyć każdą nową lub edytowaną wartość napisową. Edytując serwis w konstruktorze serwisów, edytowana jest również domyślna wersja językowa serwisu. Wszelkie zmiany dokonane w strukturze, takie jak dodanie składników i zmiana kolejności stron, zostaną powielone w lokalizowanych (tłumaczonych) wersjach serwisu. Następnie można utworzyć nowe zlecenie tłumaczenia zaktualizowanych wartości napisowych.




## Zarządzanie zleceniami tłumaczeń serwisu

Po wybraniu serwisu do tłumaczenia zostanie utworzone *zlecenie tłumaczenia*. Można pobrać pliki do tłumaczenia, przetłumaczyć je, po czym zaimportować.

Informacje, jak utworzyć zlecenie tłumaczenia, są dostępne pod hasłem [Tłumaczenie serwisu](#).

Na stronie "Zlecenia tłumaczeń" są wymienione wszystkie zlecenia tłumaczeń oraz ich statusy:

-  (Gotowe) — Plik .zip zawierający pliki serwisu jest gotowy do pobrania.

-  (W toku) — Plik .zip został pobrany. Statusem będzie "W toku", dopóki wszystkie tłumaczenia dla wszystkich języków docelowych nie zostaną pomyślnie zaimportowane.
-  (Ukończono) — Tłumaczenia dla wszystkich języków docelowych (dla danego zlecenia) zostały pomyślnie zaimportowane.
-  (Niepowodzenie) — Nie udało się wykonać zlecenia tłumaczenia. Nad banerem powinien pojawić się komunikat o niepowodzeniu, informujący o przyczynie niepowodzenia zlecenia. Chcąc ponownie wyświetlić komunikat, można **ponownie przesłać** zlecenie tłumaczenia. Należy rozwiązać problem, po czym **ponownie przesłać** zlecenie.

Można wykonywać następujące czynności:

- Aby wyświetlić szczegóły zlecenia tłumaczenia, należy zlecenie otworzyć. Szczegóły obejmują język źródłowy, wszystkie wybrane języki docelowe oraz statusy tłumaczy.
- Aby pobrać plik .zip zawierający pliki serwisu, należy wybrać zlecenie, po czym nacisnąć przycisk **Pobierz**.
- Aby usunąć zlecenie, należy wybrać zlecenie i nacisnąć przycisk **Usuń**.
- Aby zaimportować tłumaczenia, należy nacisnąć kolejno przyciski **Importuj** i **Wyślij**, następnie wybrać plik .zip zawierający przetłumaczone pliki serwisu, po czym nacisnąć przycisk **OK**.

Usługa Oracle Content Management weryfikuje, czy wszystkie tłumaczenia, które zostały zdefiniowane w zleceniu, są dostępne w pliku .zip. Aby zobaczyć, które strony i zasoby serwisów zostały zawarte w zleceniu tłumaczenia, należy kliknąć na łączu w oknie dialogowym. Gdy tłumaczenia będą gotowe do importowania, nacisnąć przycisk **Importuj**.

## Ustawienia narodowe dla tłumaczeń

Gdy element jest przesyłany do tłumaczenia, język docelowy jest identyfikowany za pomocą kodu, tak że dostawca usług językowych wie, na jaki język trzeba ten element przetłumaczyć. Na przykład **fr** reprezentuje język francuski, a **de** — niemiecki.

Kody te można rozszerzać, określając w ten sposób dialekty regionalne. Na przykład **de-LI** jest kodem języka niemieckiego używanego w Liechtensteinie, **de-LU** — w Luksemburgu. Jeśli jednak dostawca usług językowych nie obsługuje dialektu, to przekazany kod jest obcinany do dwóch znaków określających język podstawowy. Na przykład kody **de-LI** i **de-LU** zostałyby skrócone do postaci **de**.

Jeśli dostawca usług językowych nie obsługuje wszystkich dialektów, lecz tylko jeden, to może dokonać zamiany. Na przykład **ms-BN** jest kodem języka malajskiego używanego w Brunei, lecz dostawca usług językowych — jeśli tego dialektu nie obsługuje — może zmienić kod np. na **ms-MY**, będący kodem języka malajskiego używanego w Malezji. Jeśli dostawca usług językowych nie rozróżnia dialektów — na przykład **en-BZ** dla języka angielskiego używanego w Belize oraz **en-JM** na Jamajce — to przytnie kod do podstawowego, w tym przypadku do **en** dla języka angielskiego.

## Niestandardowe ustawienia narodowe dla tłumaczeń

Niestandardowe ustawienia narodowe mogą być tworzone przez programistę, zgodnie z wymaganiami biznesowymi. Kody niestandardowych ustawień narodowych zawierają język podstawowy, kod dialektu (jeśli ma zastosowanie) oraz literę **x**, określającą ustawienie

narodowe jako niestandardowe, i wszelkie inne informacje wymagane w danej organizacji. Na przykład kodem niestandardowego ustawienia dla języka angielskiego mógłby być **en-JM-x-custom**.

Ponieważ niestandardowe ustawienie narodowe jest unikatowe dla danej organizacji, kody niestandardowych ustawień narodowych są — gdy element jest przesyłany do tłumaczenia — skracane do języka podstawowego i kodu dialektu, o ile jest on obsługiwany przez dostawcę usług językowych. W powyższym przykładzie kod **en-JM-x-custom** zostałby skrócony do **en-JM** przez usunięcie części kodu właściwej dla dostosowania. A jeśli dostawca usług językowych nie obsługuje dialektu Jamajki (JM), to kod ten zostałby skrócony do języka podstawowego **en**.

## Ustawianie aliasów ustawień narodowych dla przekierowań adresu URL


Z łatwością można ustawić alias ustawień narodowych, które są używane w adresie URL serwisu w trybie wykonawczym i w podglądzie trybu wykonawczego. Jest to szczególnie przydatne, jeśli organizacja używa niestandardowych ustawień narodowych, które mogą być długie i zbyt złożone dla adresu URL.

Na przykład można mieć niestandardowe ustawienie narodowe zdefiniowane jako en-GB-x-kornwalijski, które w adresie URL będzie wyglądało następująco:

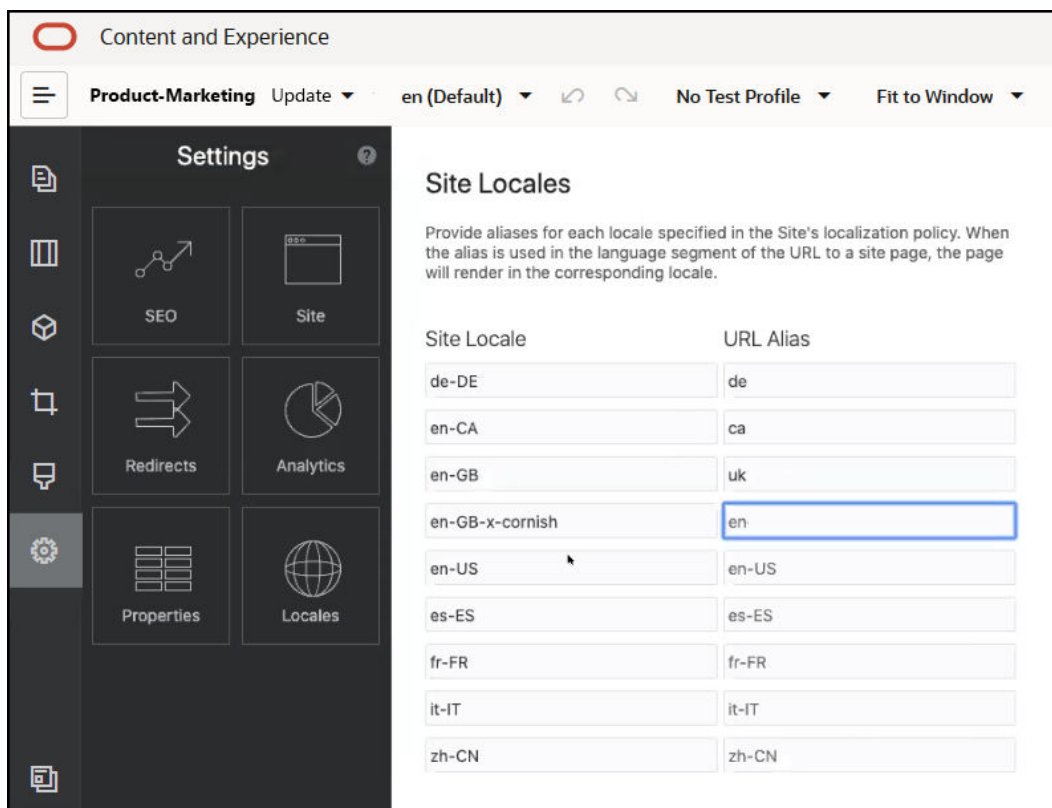
```
https://example.com/site/BlogSite/en-GB-x-cornish/home.html
```


Dodając alias, można przekierować do prostszej formy adresu URL:

```
https://example.com/site/mysite/en/home.html
```

1. Otworzyć serwis do edycji.
2. Kliknąć na ikonie  na pasku bocznym, po czym nacisnąć przycisk **Ustawienia narodowe**.

Lista wszystkich ustawień narodowych używanych w serwisie jest wyświetlana obok odpowiednich pól aliasu adresu URL.



3. Podać alias obok każdego narodowego ustawienia serwisu, które ma mieć alias, po czym nacisnąć kolejno przyciski **Zamknij** i **Zapisz**.
4. Aby uzyskać podgląd aliasu używanego w adresie URL w trybie wykonawczym, kliknąć na ikonie .
5. Gdy aktualizacja zostanie opublikowana, zmiany zostaną opublikowane i udostępnione do użycia.

# Korzystanie z przekierowań lub odwzorowań adresów URL

Po zmianie struktury serwisu lub przeniesieniu go można przekierowywać żądania użytkowników pod nowe adresy URL. Określając kody 30x przekierowywania adresów, można zachować zakładki i opublikowane łącza w różnych projektach serwisów.

Strony, które ze względu na swoją reputację mogły być wysoko klasyfikowane w wyszukiwarkach, mogły zostać przeniesione pod inne adresy, gdy serwisy hostowane w Oracle Content Management zostały przeniesione z innej infrastruktury. Przekierowania pomagają zmienić organizację struktury adresów URL serwisów i utrzymać miejsce w rankingach wyszukiwarek.

- [Planowanie przekierowań](#)
- [Dodawanie przekierowań do serwisu](#)
- [Określanie, w pliku JSON, reguł przekierowywania](#)
- [Wysyłanie — do serwisu — pliku z regułami przekierowania](#)
- [Odwzorowywanie adresu URL serwisu](#)

## Planowanie przekierowań

Można określić przekierowania wysyłające odpowiedzi HTTP 30x pod wyznaczone adresy URL. Jeśli żądanie nie będzie zgodne z jednym z ustalonych przekierowań, nastąpi zwykłe przetwarzanie adresu URL i strona zostanie zwrócona w zwykły sposób.

Można utworzyć plik JSON określający przekierowania i wysłać go do serwera. Serwer będzie używał tego pliku JSON przy przetwarzaniu przychodzących adresów URL żądań.

Dwa rodzaje reguł przekierowywania umożliwiają przekierowywanie przychodzących adresów URL do nowych lokalizacji:

- [Proste uzgadnianie napisów](#)
- [Uproszczone uzgadnianie wieloznaczne](#)

## Proste uzgadnianie napisów

Przy prostym uzgadnianiu napisów i ich zastępowaniu można określić jawne adresy URL, a następnie przekierować każdy z nich — przez odwzorowanie — do docelowego adresu URL.

W poniższej tabeli są pokazane przykłady uzgadniania napisów.

Źródłowy URL	Docelowy URL
/index.html	/home.htm
/products/widget	/items/knickknack



Źródłowy URL	Docelowy URL
/index?page=widgets	/items/widgets

Odwzorowania napisów są łatwe do zrozumienia i testowania. Reguły są szybko oceniane z użyciem prostych uzgodnień napisów i wyszukiwania odwzorowań.

Pewnym ograniczeniem jest mała elastyczność w zakresie parametrów zapytania do URL. Muszą one być w pełni zgodne. Dodatkowe parametry adresu URL lub parametry podane w innej kolejności powodują brak zgodności reguły.

## Uproszczone uzgadnianie wieloznaczne

Uproszczone uzgadnianie wieloznaczne pozwala regule uzgodnić wiele adresów URL, przy jednoczesnym ograniczeniu nawrotów wyrażenia regularnego, wymaganych w celu uzyskania wyniku.

Ponieważ pisanie wyrażeń regularne jest stosunkowo trudne, a źle skonstruowane wyrażenia regularne mogą być obliczane w nieskończoność (ReDoS), drugi typ reguły umożliwia korzystanie z mechanizmu uzgadniania uproszczonego. Są w nim używane wieloznacznik "\*" odpowiadający dowolnej liczbie znaków (w tym zerowej) w przychodzącym adresie URL oraz słowo kluczowe `wildcard` z wartością indeksu określającą te części przychodzącego adresu URL, które mają być kopiowane do adresu URL przekierowania.

W poniższej tabeli są pokazane przykłady uproszczonego uzgadniania wieloznacznego.

Źródłowy URL	Docelowy URL
/old/*	/new/<\$wildcard(1)\$>
/dispatch.asp?page=*&facet=Lang*	/page<\$wildcard(1)\$>/<\$wildcard(2)\$>

Uproszczone uzgadnianie wieloznaczne adresów URL jest bardziej wydajne niż proste uzgadnianie napisów, lecz czyni to bez niepowiązanego przetwarzania wyrażeń regularnych. Składnia jest prosta i — stosując uzgadnianie wzorców adresów URL — można określić za pomocą jednego wzorca dużą liczbę adresów URL.

Reguły wieloznaczne, ponieważ opierają się na wyrażeniach regularnych, są oceniane nieco wolniej niż oparte na prostym uzgadnianiu napisów. Duża liczba reguł może się przyczynić do ogólnego pogorszenia wydajności dostarczania stron.



## Dodawanie przekierowań do serwisu

Jeśli URL serwisu ulegnie zmianie, proces przekierowania przekaże jeden URL (źródłowy) do drugiego URL (docelowy). Pomaga to zachować zakładki użytkownika i klasyfikacje wyszukiwarki.

Mogą być używane dwa typy przekierowań:

- Przekierowanie trwałe, które używa kodu HTTP 301 odpowiedzi usługi
- Przekierowanie tymczasowe, które używa kodu HTTP 302 odpowiedzi usługi

Aby wysłać plik `redirect.json`, należy:

1. Otworzyć serwis do edycji.
2. Kliknąć na ikonie  na pasku bocznym, po czym kliknąć na ikonie **Przekierowania** ().
3. Kliknąć na opcji **Wybór pliku do wysłania**, przejść do pliku, który ma zostać użyty, wybrać go, po czym nacisnąć przycisk **OK**.
4. Gdy aktualizacja zostanie opublikowana, zmiany zostaną opublikowane i zaimplementowane.

## Określanie, w pliku JSON, reguł przekierowywania

W pliku JSON można określić reguły przekierowywania pod adresy URL.

Aby określić reguły przekierowywania pod adresy URL, należy w pliku JSON użyć następującego formatu:

```
{
  "redirectRules":
  [
    {
      "type": "string",
      "comment": "this rule is applied first",
      "expression": "/index.htm",
      "location": "/home.html"
    },
    {
      "type": "wildcard",
      "expression": "/items/*?page=*",
      "location": "/<$page$>?item=<$wildcard(1)$>",
      "code": 302
    }
  ]
}
```

Zewnętrzną strukturą zawierającą jest (w pliku JSON) tablica. Są w niej zawarte poszczególne wystąpienia reguł.

Jako pierwsze będą oceniane kolejno reguły "string", a po nich reguły "wildcard". Jeśli zostanie spełniona jedna z reguł, ocenianie kolejnych reguł jest przerywane i jest generowane przekierowanie odpowiadające tej regule.

Każda z reguł ma następujące właściwości:

- Właściwość "comment" jest opcjonalnym napisem niemającym wpływu na ocenę reguł. Zawiera uwagi lub komentarz.
- Właściwość "expression" jest wymaganą wartością napisową odpowiadającą przychodzącemu adresowi URL, względnemu wobec serwisu. W regule "wildcard" gwiazdka (\*) reprezentuje dowolną liczbę znaków, w tym zero.
- Właściwość "location" jest wymaganą wartością napisową określającą miejsce docelowe przekierowania. Może to być pełny lub względny adres URL.

- Właściwość "code" jest opcjonalną liczbą całkowitą określającą kod odpowiedzi HTTP, który ma zostać użyty podczas uruchamiania przekierowania. Wartością musi być jedna z następujących liczb całkowitych:
  - 301: sygnalizuje, że zasób został przeniesiony trwale. Jest to wartość domyślna, która jest używana, jeśli właściwość "code" została pominięta.
  - 302: sygnalizuje, że zasób został przeniesiony tymczasowo.
- Właściwość "type" jest opcjonalnym napisem sygnalizującym typ reguły przekierowania. Wartością musi być jeden z następujących napisów:
  - "string" określa szybszą regułę, której wyrażenie dokładnie odpowiada całemu wejściowemu adresowi URL.
  - "wildcard" określa regułę wieloznaczną, która może odpowiadać różnym adresom URL. Jest to wartość domyślna, która jest używana, jeśli właściwość została pominięta.

### Tokeny lokalizacji

Korzystając z tokenów lokalizacji, można ułatwić konstruowanie lokalizacji przekierowania. Można używać następujących tokenów lokalizacji:

- `<$urlPath$>`: część uzgadnianego adresu URL, stanowiąca ścieżkę.
- `<$urlQueryString$>`: cały napis-zapytanie zgodnego adresu URL.
- `<$urlQueryStringExcept(nazwa1,nazwa2)$>`: cały napis-zapytanie zgodnego adresu URL z wyjątkiem podanych parametrów.
- `<$wildcard(N)$>`: oparty na jedynej indeks zgodnego wieloznacznika w zgodnym adresie URL. (Analogiczne do `\1..9` w wyrażeniach regularnych.)
- `<$nazwa$>`: wartość nazwanego parametru napisu-zapytania. Na przykład, jeśli wejściowym napisem-zapytaniem jest `msmith: ?page=42`, można w lokalizacji użyć tokenu `<$page$>`, aby umieścić w niej liczbę 42.

### Ograniczenia

Następujące ograniczenia obowiązują zarówno do całego pliku `redirects.json`, jak i zawartych w nim reguł:

- Maksymalny całego pliku, akceptowany przez Oracle Content Management, wynosi 250 KB.
- Maksymalna liczba reguł w pliku `redirects.json` wynosi 1000.
- Maksymalna długość właściwości "expression" reguły wynosi 1000 znaków.
- Maksymalna długość właściwości "location" reguły wynosi 2000 znaków.
- Maksymalna liczba tokenów \* w wyrażeniu reguły wieloznaczej wynosi 10.

### Przykład uzgadniania reguły "string"

Reguła:

```
{
  "type": "string",
  "expression": "/old/page.jsp?id=material&type=glass",
```

```
        "location": "/new/<$id$>.htm"  
    }  
}
```

Z regułą tą jest zgodny następujący adres URL:

```
/old/page.jsp?id=material&type=glass
```

- Lokalizacją docelową jest: `/new/material.htm`
- Zgodny jest cały URL, w tym napis-zapytanie.
- Mimo że we właściwości "location" został użyty token `<$id$>`, to nie jest on tu konieczny, ponieważ zgodny mógł być tylko jeden napis-zapytanie. Lokalizacja mogłaby zostać określona jako `/new/material.htm`.

Z regułą tą nie są zgodne następujące adresy URL:

- `/old/page.jsp`  
(W wyrażeniu reguły jest określony napis-zapytanie, który musi być zgodny.)
- `/old/page.jsp?id=material&type=glass&index=2`  
(Dodatkowy fragment `&index=2` w kandydującym adresie URL powoduje, że adres ten nie jest w pełni zgodny z regułą.)
- `/old/page.jsp?type=glass&id=material`  
(Kolejność parametrów napisu-zapytania musi być identyczna z określoną w regule "string".)

#### Przykład uzgadniania reguły "wildcard"

Reguła:

```
{  
    "type": "wildcard",  
    "expression": "/old/*/pages/*?id=*&item=sheet-*",  
    "location": "/new/<$id$>/<$wildcard(4)$>.html"  
}
```

Z regułą tą są zgodne następujące adresy URL:

- `/old/phones/android/pages/info.asp?id=XT1045&item=sheet-specs`
  - Lokalizacją docelową jest: `/new/XT1045/specs.html`
  - Zgodna jest część adresu URL stanowiąca ścieżkę, a zatem także i napis-zapytanie jest sprawdzany pod kątem spełnienia warunków.
  - W tym przykładzie kolejność parametrów jest zgodna z ich kolejnością w wyrażeniu reguły, lecz nie jest to wymagane.
- `/old/phones/android/pages/info.asp?item=sheet-specs&id=XT1045`
  - Lokalizacją docelową jest: `/new/XT1045/specs.html`
  - Część adresu URL stanowiąca ścieżkę jest zgodna z wyrażeniem reguły przed znakiem zapytania (?), a zatem także i parametry są sprawdzane pod kątem zgodności.

- Mimo że parametry są w wyrażeniu reguły podane w innej kolejności, to jednak są one uzgadniane indywidualnie.
- `/old/phones/android/pages/info.asp?id=XT1045&item=sheet-specs&unrelated=thing`
  - Lokalizacją docelową jest: `/new/XT1045/specs.html`
  - Zgodna jest część adresu URL stanowiąca ścieżkę, a zatem także i napis-zapytanie jest sprawdzany pod kątem spełnienia warunków.
  - Kandydujący URL zawiera dodatkowy parametr `&unrelated=thing`, lecz — ponieważ są zgodne nazwane parametry zapytania występujące w wyrażeniu reguły — reguła jest uważana za zgodną.
  - Parametr `unrelated` mógłby być dostępny we właściwości "loaction" jako token `<$unrelated$>` i mógłby mieć wartość `thing`, mimo że nie ma to wpływu na uzgodnienie reguły.

Następujące adresy URL nie byłyby zgodne:

- `/old/pages/info.jsp`  
(Część adresu URL stanowiąca ścieżkę nie jest zgodna z częścią stanowiącą ścieżkę w wyrażeniu reguły.)
- `/old/phones/android/pages/info.asp`  
(Część adresu URL stanowiąca ścieżkę jest zgodna z częścią stanowiącą ścieżkę w wyrażeniu reguły, lecz nie są zgodne parametry zapytania występujące w wyrażeniu reguły.)
- `/old/phones/android/pages/info.asp?id=cellular`  
(Część adresu URL stanowiąca ścieżkę jest zgodna z częścią stanowiącą ścieżkę w wyrażeniu reguły, lecz nie wszystkie parametry zapytania występujące w wyrażeniu reguły są zgodne.)

### Definiowanie tablicy tokenów

Można także utworzyć w pliku `redirects.json` tablicę definicji tokenów, pomagającą konfigurować przekierowania obsługujące wiele zindywidualizowanych adresów URL. Dzięki temu można dokonywać przekierowań na podstawie charakterystyki przychodzącego adresu URL.

Do zdefiniowania tokenów, przeznaczonych do użycia w adresach URL w regułach przekierowania, należy użyć w pliku `redirects.json` następującego formatu:

```
{
  "tokenDefinitions":
  [
    {
      "token": "sitePrefix",
      "type": "hostmatch",
      "expression": "example.com"
      "value": ""
    },
    {
      "token": "sitePrefix",
      "type": "hostmatch",
      "expression": "*.com"
    }
  ]
}
```

```

        "value": "/site/Starter-Site"
    },
    {
        "token": "gotoRedirect",
        "type": "pathmatch",
        "expression": "*oracle*"
        "value": "https://www.oracle.com"
        "flags": "caseinsensitive"
    },
]
}

```

Definicje tokenDefinitions mają następujące właściwości:

- "token": Nazwa definiowanego tokenu.
- "type": Jeden z następujących typów:
  - "hostmatch" — jest uzgadniana wartość hosta z przychodzącego adresu URL.
  - "pathmatch" — jest uzgadniana ścieżka z przychodzącego adresu URL.
  - "querymatch" — jest uzgadniana wartość zapytania z przychodzącego adresu URL.
- "expression": Wyrażenie używane do uzgadniania. Można stosować wieloznaczniki.
- "value": Wartość używana dla tokenu.
- "flags": Domyślnie w wyrażeniu uzgadniania jest uwzględniana wielkość liter, chyba że wartość flags zostanie ustawiona na caseinsensitive

Podczas obliczania wartości tokenu wpisy z tablicy tokenDefinitions są wyliczane w kolejności ich występowania. Jest używana pierwsza zgodna definicja. Jeśli dla tokenu nie zostanie uzgodniona żadna definicja, będzie użyty napis pusty. Ze względów praktycznych oraz wydajnościowych najczęściej używane tokeny powinny być umieszczane na początku listy tokenDefinitions list.

Lista tokenDefinitions ma następujące ograniczenia:

- Można utworzyć maksymalnie 250 definicji tokenów.
- Nazwa podana we właściwości token musi być krótsza niż 100 znaków.
- Wyrażenie podane we właściwości expression może zawierać maksymalnie 10 wieloznaczników.
- Wyrażenie podane we właściwości expression musi być krótsze niż 1000 znaków.
- Wartość podana we właściwości value musi być krótsza niż 1000 znaków.

### Przykład

Na przykład można mieć następujący plik redirects.json:

```

{
  "redirectRules":
  [
    {
      "type": "string",
      "expression": "/legacy-privacy-policy.html",
      "location": "<$pathPrefix$>/about/new-privacy-policy.html"
    }
  ]
}

```

```
    },  
  ],  
  "tokenDefinitions":  
  [  
    {  
      "token": "pathPrefix",  
      "type": "hostmatch",  
      "expression": "vanity.com"  
      "value": "/fashion"  
    },  
  ],  
]  
}
```

W tym przypadku użyta w regule właściwość `location` zawiera token `<$pathPrefix$>`. Token `pathPrefix` jest zdefiniowany w sekcji `tokenDefinitions`. Jeśli przychodzący adres URL będzie zawierał "vanity.com", to wartość tokenu `pathPrefix` zostanie ustawiona na `/fashion`. Zostanie ona użyta w odpowiedzi `location`, dając w wyniku `/fashion/about/new-privacy-policy.html`.

Założmy, że pierwszym zindywidualizowanym adresem URL domeny jest `http://example.com/legacy-privacy-policy.html`. Zostanie wówczas uzgodniona pierwsza i jedyna reguła przekierowania.

Zadeklarowana lokalizacja `location` dla tej reguły to `<$pathPrefix$>/about/new-privacy-policy.html`. W tej sytuacji token `<$pathPrefix$>` musi zostać ustalony. W tym celu, aby odnaleźć zgodny wpis, jest przeglądana tablica `tokenDefinitions`.

Uwzględniana jest pierwsza definicja tokenu. Jej `token` jest właściwy i dlatego jest on dalej przetwarzany. Wyrażenie `vanity.com` nie jest zgodne z przychodzącym adresem URL `example.com`, a zatem ta definicja nie spełnia wymagań i jest kontynuowane wyliczanie.

Ponieważ nie ma dalszych definicji tokenów, jako wartość tokenu `<$pathPrefix$>` jest używany napis pusty. Ostateczną lokalizacją, zwracaną dla tego przekierowania, jest `/about/new-privacy-policy.html`.

Założmy, że drugim zindywidualizowanym adresem URL domeny jest `http://vanity.com/legacy-privacy-policy.html`. Podobnie jak w przypadku pierwszego adresu URL, zadeklarowana lokalizacja `location` dla tej reguły to `<$pathPrefix$>/about/new-privacy-policy.html`. W tej sytuacji token `<$pathPrefix$>` musi zostać ustalony. W tym celu, aby odnaleźć zgodny wpis, jest przeglądana tablica `tokenDefinitions`.

Uwzględniana jest pierwsza definicja tokenu. Jak poprzednio, jej `token` jest właściwy i dlatego jest on dalej przetwarzany. Wyrażenie `vanity.com` jest zgodne z przychodzącym adresem URL `vanity.com`, a zatem ta definicja spełnia wymagania i jako wartość tokenu jest używana wartość `/fashion`.

Ponieważ został znaleziony zgodny wpis dla tokenu, następuje zatrzymanie wyliczania tablicy definicji tokenów i ostateczna lokalizacja jest obliczana jako `/fashion/about/new-privacy-policy.html`.



### Testowanie przekierowań do serwisu

Przekierowania do serwisu można przetestować, edytując serwis; w tym celu należy otworzyć panel **Ustawienia**, po czym kliknąć na ikonie **Przekierowania**. Wprowadzić testowany URL, po czym nacisnąć przycisk **Test**.

## Wysyłanie — do serwisu — pliku z regułami przekierowania

W Oracle Content Management można do serwisu wysłać reguły przekierowania.

Aby wysłać plik `redirect.json` do serwisu, należy:

1. Otworzyć serwis do edycji.
2. Kliknąć na ikonie  na pasku bocznym, po czym kliknąć na ikonie .
3. Kliknąć na opcji **Wybór pliku do wysłania**, przejść do pliku, który ma zostać użyty, wybrać go, po czym nacisnąć przycisk **OK**.
4. Gdy aktualizacja zostanie opublikowana, zmiany zostaną opublikowane i udostępnione do użycia.



## Odwzorowywanie adresu URL serwisu

Po utworzeniu serwisu i opublikowaniu go przy użyciu Oracle Content Management, można skonfigurować usługę DNS (Domain Name System), tak aby serwis był dostępny za pomocą zarejestrowanej nazwy domeny, takiej jak `www.mysite.com`.

Usługa DNS określa, gdzie można odnaleźć strony serwisu; w tym celu odwzorowuje nazwę domeny na lokalizację serwisu, czyli na nazwę kanoniczną (CNAME).

Do odwzorowania nazwy domeny są potrzebne:

- URL instancji usługi Oracle Content Management. Zazwyczaj ma on następującą postać:  
`service-tenant.documents.datacenter.oraclecloud.com`
- Nazwa domeny w postaci zarejestrowanej przez organizację zajmującą się rejestracją domen.  
Na przykład `www.example.com`. Może to być również poddomena, taka jak `www.example.com/subdomain`.
- Konto u dostawcy sieci CDN (Content Delivery Network). Oracle Content Management zapewnia integrację z Akamai. Należy się skontaktować z Asystą Techniczną Oracle, która pomoże skonfigurować Akamai pod kątem instancji.

Chcąc używać własnej sieci CDN (zamiast Akamai udostępnionej przez Oracle Content Management), należy wykonać opisane poniżej etapy.

Różni dostawcy usług DNS udostępniają różne interfejsy internetowe i wymagają wykonywania innych czynności w celu aktualizacji rekordu CNAME. Poniżej podane są informacje, które będą potrzebne, oraz ogólne czynności, które należy wykonać.

Aby odwzorować URL serwisu na nazwę domeny, należy:

1. Wystąpić do dostawcy sieci CDN (Content Delivery Network) o certyfikat SSL dla domeny. Na przykład `https://www.example.com`.
2. Skonfigurować sieć CDN, tak aby:
  - a. Akceptowała wszystkie żądania przychodzące do domeny i przekazywała je przy użyciu protokołu zabezpieczonego (https).
  - b. Punkty źródłowe do domeny z Oracle Content Management:  
`service-tenant.documents.datacenter.oraclecloud.com`
3. Zmienić plik strefy serwera DNS, odwzorowując nazwę domeny na serwer brzegowy udostępniany przez dostawcę sieci CDN:  
`domain CNAME CDN Server`
4. Począkać na rozprzestrzenienie aktualizacji. W zależności od usługi DNS, może to zająć od 2 do 48 godzin.

Gdy to nastąpi, można uzyskać dostęp do serwisu, używając nazwy domeny. Na przykład:

`https://www.mysite.com/site_name`

Domyślnie punkt końcowy dla Oracle Cloud REST API for Content Management jest dostępny, jeśli jest używany standardowy adres URL podany dla serwisu. Na przykład składniki "lista folderów" i "lista plików" używają REST API do wykonywania operacji

na folderach i plikach. Jeśli jest używany niestandardowy adres URL, należy sprawdzić, czy mamy dostęp do punktu końcowego z użyciem nazwy domeny. Na przykład:

`https://www.mysite.com/documents`

## Zwiększanie wydajności serwisu

Korzystając z podręcznej pamięci przeglądarki, można poprawić wydajność dostarczania i renderowania zawartości w przeglądarce. Także rendering ATF (Above the fold) usprawnia renderowanie serwisu internetowego.

- [Używanie pamięci podręcznej w celu zwiększenia wydajności](#)
- [Rendering ATF \(Above the Fold\)](#)

### Używanie pamięci podręcznej w celu zwiększenia wydajności

Dostarczanie elementów zawartości, zasobów cyfrowych i serwisów powinno — w celu zwiększenia wydajności tych operacji i renderowania — w pełni opierać się na podręcznej pamięci przeglądarki osoby odwiedzającej.

Serwisy, motywy, elementy zawartości i zasoby cyfrowe są przechowywane przez określony czas w podręcznej pamięci przeglądarki osoby odwiedzającej. Gdy serwis, motyw, element zawartości lub zasób cyfrowy zostanie zaktualizowany, zostanie zmieniony klucz "cache-buster" w adresie URL, tak że przeglądarka musi pobrać nowy adres URL i uzyskać nowy element.

Klucz "cache" pomaga zarządzać użytkowaniem podręcznej pamięci przeglądarki, odwołując się tylko do aktualnych zasobów. Mimo że klucz "cache" jest zawarty w adresie URL, to jednak jest elementem logicznym, a nie — jak często ma to miejsce — lokalizacją fizyczną (folderem). Zmiana w kluczu "cache" nie wskazuje innej lokalizacji fizycznej zasobu; informuje jedynie serwer, że należy pobrać aktualną wersję zasobu.

Zasoby mogą być statyczne (np. pliki CSS, pliki JS i pliki obrazów) lub dynamiczne (np. dane, dane serwisu czy dane elementu zawartości). Istnieje pięć kategorii zasobów używanych do konstruowania serwisu:

- **Zasoby produktu** — Zasoby stanowiące część produktu, które ulegają aktualizacji, gdy pojawi się nowa wersja produktu lub do starej wersji zostaną zastosowane poprawki.
- **Zasoby serwisu** — Zasoby stanowiące część serwisu, takie jak `structure.json`, dane stron oraz obrazy. Zasoby te są aktualizowane, gdy serwis jest publikowany. Kontroler jest opisany dalej.
- **Zasoby motywów** — Zasoby stanowiące część motywów, takie jak układy, pliki CSS i obrazy. Zasoby te są aktualizowane, gdy motyw jest publikowany.
- **Zasoby składników** — Zasoby stanowiące część składników niestandardowych. Do zasobów tych należą pliki HTML, JS i CSS oraz pliki obrazów używane w składniku. Zasoby te są aktualizowane, gdy składnik jest publikowany. Jeśli jeden składnik zostanie zmieniony i ponownie opublikowany, to klucz "cache" zmienia się dla wszystkich składników, ponieważ jest używany tylko jeden klucz dla wszystkich składników.
- **Zasoby CaaS** — Zasoby obsługujące elementy zawartości i elementy cyfrowe. Są aktualizowane, gdy elementy zawartości są publikowane lub publikowane bądź gdy zmienia się miejsce docelowe gromadzenia.

W poniższych tematach jest opisane wykorzystywanie pamięci podręcznej (buforowanie) w trybie wykonawczym Oracle Content Management oraz dla konstruktora serwisów:

- [Buforowanie w trybie wykonawczym](#)
- [Buforowanie dla konstruktora serwisów](#)

## Buforowanie w trybie wykonawczym

W trybie wykonawczym nagłówek Cache-Control dla Oracle Content Management jest ustawiany na 15 dni. Do adresu URL każdego z zasobów jest dodawany klucz "cache".

Dopóki URL nie ulegnie zmianie, przeglądarka będzie obsługiwać dany zasób z lokalnej pamięci podręcznej, o ile jest ona dostępna. Gdy zasób zostanie zaktualizowany, nastąpi aktualizacja klucza "cache" w adresie URL, co zmusza przeglądarkę do wysłania nowego żądania do serwera i zaktualizowania lokalnej pamięci podręcznej.

Kontroler zawierający klucze "cache" jest także przechowywany w pamięci podręcznej, przez 1 minutę. Wskutek tego zaktualizowane klucze "cache" nie będą widoczne przez czas sięgający 1 minuty.

W trybie wykonawczym serwer zwraca `controller.html` z najnowszymi kluczami "cache" dla produktu, serwisu, motywu, składników i zasobów CaaS. Do pliku `controller.html` jest dodawany skrypt z kluczami; na przykład:

```
<script type="text/javascript">
  var SCSCacheKeys = {
    product: '123',
    site: '456',
    theme: '789',
    component: '012',
    caas: '345'

  };
</script>
```

Klucze te są używane przez `controller.js` do konstruowania adresów URL podobnych do pokazanych w poniższej tabeli.

Typ zasobów	Przykłady
Zasoby produktu	<code>/sitePrefix/productCacheKey/_sitesclouddelivery/...</code>  <code>/mySite/_cache_947d/_sitesclouddelivery/</code>
Zasoby motywu	<code>/sitePrefix/themeCacheKey/_themesdelivery/themeName/...</code>

Typ zasobów	Przykłady
Zasoby składnika	<code>/sitePrefix/compCacheKey/_compdelivery/compName/...</code>
Zasoby serwisu	<code>/sitePrefix/siteCacheKey/content/...</code> <code>/sitePrefix/siteCacheKey/structure.json</code> <code>/sitePrefix/siteCacheKey/pages/100.json</code>
Zasoby CaaS	<code>RegularCaaSUrl?cacheKey=caasCacheKey</code>

Oracle Content Management, wstawiając w ten sposób klucze "cache" do adresów URL, może zmusić przeglądarkę do ładowania zaktualizowanych zasobów; jeśli zmieni URL, przeglądarka uzna, że ma do czynienia z nowym zasobem.



**Uwaga:**

W przypadku serwisów zabezpieczonych w pamięci podręcznej są przechowywane tylko zasoby produktu, motywów i składników. Nie jest natomiast buforowana zawartość serwisu ani zawartość CaaS.

## Buforowanie dla konstruktora serwisów

W konstruktorze serwisów zasoby statyczne są przechowywane w pamięci podręcznej przez 15 dni.

Gdy jest używany konstruktor serwisów, buforowanie odbywa się dla zasobów produktów, motywów i składników. (Nie obejmuje zasobów serwisów i CaaS.) Klucze "cache" motywów i składników są ponownie generowane podczas uruchamiania lub odświeżania konstruktora serwisów.

Jeśli do motywu lub składnika wprowadzono zmiany, które mają zostać uwzględnione w konstruktorze serwisów, należy odświeżyć konstruktor serwisów, naciskając klawisz F5.

## Rendering ATF (Above the Fold)

Rendering ATF powoduje, że strona internetowa wydaje się ładować szybciej niż faktycznie to następuje. Celem jest wyrenderowanie najpierw wszystkich części strony, które są od razu widoczne, a następnie — zanim użytkownik przewinie stronę w dół — wyrenderowanie pozostałych części strony, które początkowo nie są widoczne.

Gniazdo może mieć desygnację "above the fold", przy której jest na karcie wyświetlana odpowiednia ikona.

Gniazdo, aby było renderowane w ten sposób, musi być oznaczone jako `scs-atf`, jak poniżej:

```
<div class="scs-slot scs-atf" id="headline"></div>
```

Składnik powinien poinformować moduł renderujący o ukończeniu renderowania. Gotowe składniki, dostarczane z produktem, wykonują to domyślnie. Składnik niestandardowy może wykonać dodatkowe wywołania i powinien:

1. Powiadomić moduł renderujący, że ma on poczekać, aż renderowanie zostanie ukończone.
2. Powiadomić moduł renderujący o ukończeniu renderowania.

Aby zrealizować punkt 1, należy dodać w pliku `appinfo.json` dla niestandardowego składnika następującą właściwość:

```
"initialData": {  
  . . .  
  "customRenderComplete": true,  
  . . .  
}
```

Aby zrealizować punkt 2, należy w pliku `render.js` składnika wprowadzić następujące wywołanie, informujące moduł renderujący o ukończeniu renderowania:

```
SitesSDK.setProperty('renderComplete', true);
```

Jeśli nie wszystkie składniki znajdujące się w gnieździe ATF zgłoszą terminowo, że ich renderowanie zostało ukończone, to moduł renderujący odczeka 2 sekundy, po czym zajmie się resztą strony. Jeśli ten czas jest za krótki, można go wydłużyć, deklarując w szablonie strony następującą zmienną globalną:

```
var SCSAtfPassTimeout = 3000;
```



#### Uwaga:

Czas jest określany w milisekundach, a zatem w tym przykładzie wynosi 3 sekundy.

Dane diagnostyczne dla procesu ATF są dostarczane przez API. Z konsoli debugera można wywołać następującą metodę (jeśli trzeba, można do niej uzyskać dostęp ze strony):

```
SCSRenderAPI.getRenderMetrics();
```

Na przykład:

```
{currentTime: 16243.400000000001, renderStartTime: 264.36,  
atfPassEndTime: 306.535, mainPassStartTime: 316.475, mainPassEndTime:  
331.38500000000005, ...}
```

```
1. atfComponentCount:13
2. atfPassEndTime:306.535
3. completionCount:23
4. completionRecords:Array(23)
  1. 0:{atf: true, componentId: "a7afdd33-3fbb-4329-bc1b-6be60056a995",
time: 280.065}
  2. 1:{atf: true, componentId: "edfcfcb4-b0d3-422f-aa59-5c925bbbbebee",
time: 283.54}
  3. 2:{atf: true, componentId: "c1c3aec8-e52f-406c-8c29-ab69c05877ed",
time: 283.56000000000006}
  4. 3:{atf: true, componentId: "b3a31dc6-62a1-44d9-9c80-bdb2c5bedaaa",
time: 284.13000000000005}
  5. 4:{atf: true, componentId: "c05aala2-c11c-4ef5-9051-4799c5bee24a",
time: 284.15500000000003}
  6. 5:{atf: true, componentId: "bafd4047-06ec-4739-9b23-9db74f573f30",
time: 294.665}
  7. 6:{atf: true, componentId: "e7d49528-0357-4b45-801e-b3a2716a086c",
time: 297.995}
  8. 7:{atf: true, componentId: "a5f33674-4022-4138-8cc5-fef00c02a557",
time: 299.78000000000003}
  9. 8:{atf: true, componentId: "ccfedc98-1dbd-440e-b867-5e683cea2ec5",
time: 301.19500000000005}
  10. 9:{atf: true, componentId: "d691bc44-fed9-474a-9806-2191f46a5e2e", time:
302.46}
  11. 10:{atf: true, componentId: "cf613054-05d8-40dd-83a0-718760d7bc73",
time: 303.79}
  12. 11:{atf: true, componentId: "b4a6ef98-ffc8-48c7-987c-63346ee97bcc",
time: 305.115}
  13. 12:{atf: true, componentId: "de1fa2ce-66ba-419b-b517-2cb4a7601c3b",
time: 306.535}
  14. 13:{atf: false, componentId: "ba3f8ed4-31d4-4347-b6f0-f1019783a57c",
time: 318.665}
  15. 14:{atf: false, componentId: "ae8af486-76b3-47cd-9989-db4212eefebb",
time: 320.45500000000004}
  16. 15:{atf: false, componentId: "a48b5abb-49b2-4456-90bd-a3de998150c8",
time: 320.48}
```

```
17. 16:{atf: false, componentId: "a9650e6d-7e7e-42a2-  
b758-58f2aeab18a2", time: 322.61500000000007}  
  
18. 17:{atf: false, componentId: "aca9836a-f955-4aa7-8db2-  
fd3cf1189dea", time: 324.235000000000007}  
  
19. 18:{atf: false, componentId: "e3d7941c-fbc7-4da9-963b-  
e3810b6467d4", time: 325.85}  
  
20. 19:{atf: false, componentId: "eecde809-  
da54-4066-9326-73f9d9c35fe4", time: 327.315}  
  
21. 20:{atf: false, componentId: "e8f4fb16-4e15-4570-  
b7de-304e99e449a7", time: 328.74}  
  
22. 21:{atf: false, componentId: "a7baa06e-7f30-42c7-94f4-  
e171ab2edcd6", time: 330.090000000000003}  
  
23. 22:{atf: false, componentId: "fd603b96-2beb-4e87-  
a54f-12d0e264cd0a", time: 331.385000000000005}  
  
24. length:23  
  
25. __proto__:Array(0)  
  
5. componentCount:23  
  
6. currentTime:16243.400000000001  
  
7. mainPassEndTime:331.385000000000005  
  
8. mainPassStartTime:316.475  
  
9. renderStartTime:264.36  
  
10. __proto__:Object
```



# Część IV

## Tworzenie składników serwisów

W tej części wyjaśniono szczegółowo, jak zacząć tworzyć szablony, motywy, style, składniki i inne przydatne elementy oraz jak je rozwijać. Część ta zawiera następujące rozdziały:

- [Dostosowywanie projektów i stylów](#)
- [Omówienie użycia tła](#)
- [Ustawianie wyzwalaczy i czynności](#)
- [Tworzenie szablonów](#)
- [Tworzenie motywów](#)
- [Tworzenie układów](#)
- [Tworzenie składników](#)

## Dostosowywanie projektów i stylów

W Oracle Content Management każdy motyw musi zawierać projekt, określający wygląd i styl stron oraz ustawienia stylu składników używanych w domyślnym serwisie motywu.

- [Projekty — informacje podstawowe](#)
- [Pliki projektu](#)
- [Dostosowywanie stylów listy konwersacji](#)
- [Dostosowywanie stylów listy folderów i listy plików](#)
- [Dostosowywanie ikon paska społecznościowego](#)
- [Konfigurowanie rozszerzeń stosujących stylizację do wywiadów dla Oracle Intelligent Advisor](#)

### Projekty — informacje podstawowe

W Oracle Content Management każdy motyw musi zawierać projekt, określający wygląd i styl stron oraz ustawienia stylu składników używanych w motywie.

Użytkownicy, tworząc serwis, muszą wybrać szablon zawierający domyślnie motyw. Można utworzyć lub wybrać inny motyw do użycia dla serwisu.

Tworząc motyw, poza zaprojektowaniem układów strony, trzeba określić dostępne style, które będą wyświetlane w panelu "Ustawienia" dla każdego typu składnika dostępnego dla użytkowników (na przykład akapit, tytuł i obraz). Aby określić domyślne ustawienia różnych składników, można w ramach projektu opracować więcej niż jeden styl. Bieżący użytkownik dokonuje wyborów związanych na przykład z czcionkami tekstów, rozmiarami czcionek i obramowaniem obrazów, a następnie zapisuje je jako nazwany projekt. Zawarte w projekcie elementy, gdy zostaną wyrenderowane, wpływają na wygląd układów strony.

Zob. rozdział Praca ze stronami serwisu w podręczniku *Tworzenie serwisów za pomocą Oracle Content Management*.

### Pliki projektu

Dwa pliki są uznawane za domyślne pliki projektu dla motywu. Są to: `design.json` i `design.css`.

- `design.json` określa style składników
- `design.css` udostępnia definicje wartości "class" (na przykład kolor i czcionka)

Pliki te znajdują się w strukturze motywu, w katalogu `/designs/default/`.

**Plik design.json**

Plik `design.json` ma następującą strukturę:

```
{
  "componentStyles": {
    "scs-image": {
      "styles": []
    },
    "scs-map": {
      "styles": []
    },
    "scs-title": {
      "styles": []
    },
    "scs-paragraph": {
      "styles": []
    },
    "scs-divider": {
      "styles": []
    },
    "scs-button": {
      "styles": []
    },
    "scs-app": {
      "styles": []
    },
    "scs-spacer": {
    },
    "scs-gallery": {
      "styles": []
    },
    "scs-youtube": {
      "styles": []
    },
    "scs-socialbar": {
      "styles": []
    },
    "scs-document": {
      "styles": []
    }
  }
}
```

Każdy z wpisów `"styles": []` może zawierać listę stylów dla konkretnego składnika. Na przykład składnik `"tytuł"` udostępnia następujące style domyślne:

```
"styles": [{
  "name": "COMP_STYLE_FLAT",
  "class": "scs-title-default-style"
},
{
  "name": "COMP_STYLE_HIGHLIGHT",
```

```

        "class": "scs-title-style-2"
      },
      {
        "name": "COMP_STYLE_DIVIDER",
        "class": "scs-title-style-3"
      }
    ]
  }
}

```

#### Uwaga:

Jeśli zostaną utworzone style niestandardowe dla składnika i nastąpi jego odwzorowanie na style w pliku `design.json`, nie trzeba używać pełnej nazwy, takiej jak udostępniana przez system. Wystarczy określić napis, jaki ma zostać użyty. Na przykład zamiast nazwy `"COMP_STYLE_BOX"` wystarczy użyć `"Box"`. Znaczy to, że na liście na karcie **Style** (w panelu ustawień) pojawi się dla tego składnika nazwa `"Box"`, a nie `"COMP_STYLE_BOX"`.

Wartości `"name"` są odwzorowywane na wyrazy wyświetlane w interfejsie użytkownika, jak poniżej:

```

"COMP_STYLE_FLAT": "Flat",
"COMP_STYLE_HIGHLIGHT": "Highlight",
"COMP_STYLE_DIVIDER": "Divider",

```

#### Plik `design.css`

Plik `design.css` udostępnia definicje wartości `"class"`. Poniżej przedstawiono kilka przykładów.

```

.scs-title-default-style {
  color: #333333;
  display: block;
  font-family: "Helvetica Neue", "Helvetica", "Arial", sans-serif;
  font-size: 24px;
  font-weight: normal; }

.scs-title-style-2 {
  background-color: #DEF300;
  color: #333333;
  font-family: adobe-clean, sans-serif;
  padding-top: 2em;
  padding-bottom: 2em; }

.scs-button-default-style .scs-button-button:hover {
  background: #f7f8f9;
  border: 1px solid #c4ced7;
  color: #0572ce;

```

```
box-shadow: inset 0 1px 0 #f7f8f9;
text-shadow: 0 1px 0 #f7f8f9; }
```

```
.scs-button-default-style .scs-button-button:active {
  background: #0572ce;
  border: 1px solid #0572ce;
  color: #fff;
  box-shadow: inset 0 1px 0 #0572ce;
  text-shadow: 0 1px 0 #0572ce; }
```

## Projekt tabeli responsywnej

Oracle Content Management udostępnia przykładowy arkusz CSS responsywnej tabeli w obszarze składnika "akapit"; tabela ta umożliwia przedstawianie danych (danych z poszczególnych wierszy) w postaci stosu, gdy jest wyświetlana w urządzeniach mobilnych.

Tabela responsywna dostosowuje się do skutecznego wyświetlania zawartości w zależności od rozmiaru ekranu. Na przykład 5-kolumnowa tabela może się dobrze wyświetlać poziomo na stronie internetowej, lecz — gdy jest wyświetlana w telefonie — dane mogą się lepiej prezentować w postaci stosu. Tabele responsywne do poprawnego działania wymagają wiersza nagłówka.

The image illustrates a responsive table design. On the left, a desktop browser view shows a table with 5 columns: ACCOUNT, DUE DATE, AMOUNT, MINIMUM, and PERIOD. The data is presented horizontally. On the right, a mobile phone view shows the same data stacked vertically, demonstrating how the table adapts to a smaller screen.

ACCOUNT	DUE DATE	AMOUNT	MINIMUM	PERIOD
Visa	04/16/2020	\$3,090	\$25	03/09/2020 - 04/08/2020
Amex	04/22/2020	\$1,451	\$50	03/16/2020 - 04/15/2020

W poniższym wygenerowanym kodzie HTML do każdej komórki tabeli został dodany atrybut "data-label" z wartością odpowiadającą polu z wiersza nagłówka.

```
<thead>
  <tr>
    <th scope="col">ACCOUNT</th>
    <th scope="col">DUE DATE</th>
    <th scope="col">AMOUNT</th>
    <th scope="col">MINIMUM</th>
    <th scope="col">PERIOD</th>
  </tr>
</thead>

<tbody>
  <tr>
    <td data-label="ACCOUNT">Visa</td>
    <td data-label="DUE DATE">04/16/2020</td>
    <td data-label="AMOUNT">$3,090</td>
    <td data-label="PERIOD">03/09/2020 - 04/08/2020</td>
  </tr>
```

Jeśli atrybut ten będzie przypisany do poszczególnych komórek, to wartości TD będą układane w stos, gdy zostaną zastosowane poniższe reguły CSS.

```
.scs-paragraph:not(.scs-paragraph-edit) table td {
  border-bottom: 1px solid #ddd;
  display: block;
  text-align: right;
}


.scs-paragraph:not(.scs-paragraph-edit) table td::before {
  content: attr(data-label);
  float: left;
  font-weight: bold;
}
```

Poniższa reguła "media" kieruje zapytanie o rozmiar ekranu i zaczyna obowiązywać wtedy, gdy rozmiar ten jest mniejszy niż 767 pikseli:

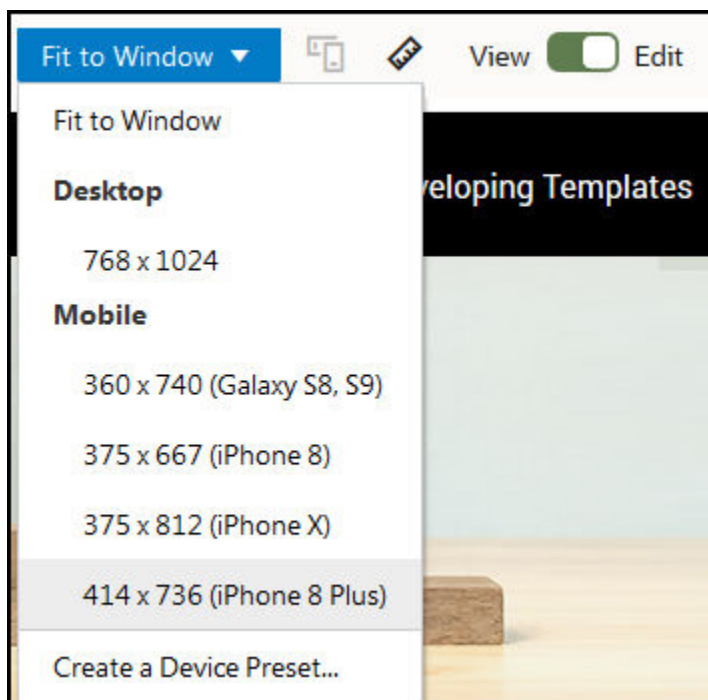
```
@media screen and (max-width: 767px) {
```

Przykładowy kod z użyciem css, umożliwiający wstawianie tabel jako responsywnych, znajduje się w domyślnym pliku design.css udostępnianego motywu początkowego StarterTheme. Jeśli serwis jest konstruowany przy użyciu motywu StarterTheme, tabele wstawiane do gniazda "akapit" będą domyślnie tabelami reponsywnymi. Aby wstawić tabelę, należy:

1. W Oracle Content Management otworzyć serwis, a następnie przełączyć do trybu **edycji**.
2. Aby zmodyfikować, utworzyć nową aktualizację lub wybrać już istniejącą.
3. Przeciągnąć nowy składnik "akapit" na stronę, po czym kliknąć w miejscu (albo w istniejącym akapicie), w którym ma zostać wstawiona tabela.

4. Na pasku narzędzi tekstu z formatowaniem nacisnąć przycisk , po czym ustawić właściwości tabeli. Wybrać pierwszy wiersz jako nagłówek, po czym dostosować szerokość tabeli do przewidywanego najmniejszego ekranu. Na przykład, jeśli serwis ma być wyświetlany w telefonie, to domyślna szerokość 767 pikseli prawdopodobnie będzie zbyt duża, nawet przy wyświetlaniu w postaci stosu. Można wówczas ustawić mniejszą szerokość tabeli (na przykład 300 pikseli) lub na 100%.

Po zakończeniu wrócić do trybu **wyświetlania**, po czym — aby zobaczyć wyniki — wybrać opcję widoku z szerokością ekranu mniejszą niż 767 pikseli. Trzeba się przełączyć do trybu wyświetlania, ponieważ w trybie edycji tabela nie zachowuje się responsywnie.



Chcąc używać tabel responsywnych i konstruować serwis z innego motywu, trzeba skopiować plik design.css z motywu StarterTheme do pliku design.css tego innego motywu.

1. Aby skopiować kod z pliku design.css z motywu StarterTheme, wybrać z bocznej nawigacji (w Oracle Content Management) opcję **Programista**.
2. Nacisnąć przycisk **Wyświetl wszystkie motywy**.
3. Wybrać motyw **StarterTheme**, po czym nacisnąć przycisk **Otwórz**.
4. Wybrać "Projekty", aby otworzyć folder, a następnie wybrać opcję **Domyślne**.
5. Wybrać z motywu StarterTheme plik design.css, po czym nacisnąć przycisk **Pobierz**.
6. Otworzyć plik w edytorze tekstów, po czym odszukać sekcję pliku zaczynającą się komentarzem *An example CSS of how to render a table responsively*.
7. Zaznaczyć kod aż do następnego komentarza i skopiować zaznaczony fragment.

/\*\*

\* An example CSS of how to render a table responsively.

```
* It enables stacking of row data on mobile devices.
* Only do this for view mode (not for edit mode).
*
* On each cell rendered, it adds a user-defined attribute
* 'data-label' with value matching the column header text.
*/
@media screen and (max-width: 767px) {
  .scs-paragraph:not(.scs-paragraph-edit) table {
    border: 0;
  }

  .scs-paragraph:not(.scs-paragraph-edit) table caption {
    font-size: 1.3em;
  }

  .scs-paragraph:not(.scs-paragraph-edit) table thead {
    border: none;
    clip: rect(0 0 0 0);
    height: 1px;
    margin: -1px;
    overflow: hidden;
    padding: 0;
    position: absolute;
    width: 1px;
  }

  .scs-paragraph:not(.scs-paragraph-edit) table tr {
    border-bottom: 3px solid #ddd;
    display: block;
    margin-bottom: .625em;
  }

  .scs-paragraph:not(.scs-paragraph-edit) table td {
    border-bottom: 1px solid #ddd;
    display: block;
    text-align: right;
  }

  .scs-paragraph:not(.scs-paragraph-edit) table td::before {
    content: attr(data-label);
    float: left;
    font-weight: bold;
  }

  .scs-paragraph:not(.scs-paragraph-edit) table td:last-child {
    border-bottom: 0;
  }
}
```

8. Powtórzyć czynności w celu pobrania pliku design.css modyfikowanego szablonu, otworzyć szablon, po czym wkleić skopiowany kod.
9. Zapisać zmiany, po czym wysłać zmodyfikowany plik design.css (jako nową wersję) do modyfikowanego motywu.



## Dostosowywanie stylów listy konwersacji

Dodając selektory do pliku `design.css`, można dostosować styl składnika "lista konwersacji".

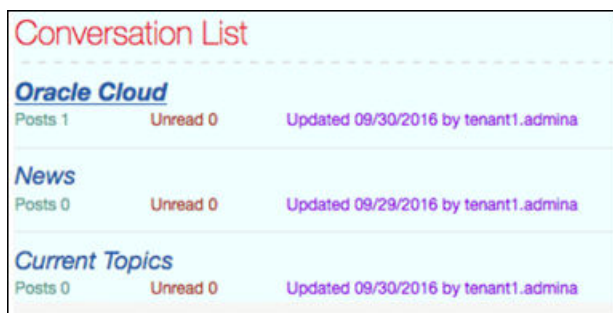
Za pomocą poniższych selektorów CSS można dostosować styl składnika "lista konwersacji".

Nazwa selektora	Opis
<code>.scs-convo-list-cust .scs-convo-list-container</code>	Najbardziej zewnętrzny element DIV składnika
<code>.scs-convo-list-cust .scs-convo-list-title</code>	Tytuł konwersacji na liście, gdy konwersacja jest wybrana
<code>.scs-convo-list-cust .scs-convo-list-line-separator</code>	Separator między tytułem listy i listą
<code>.scs-convo-list-cust .scs-convo-list-convo-title</code>	Tytuł konwersacji na liście
<code>.scs-convo-list-cust .scs-convo-list-convo-line-separator</code>	Separator między każdą konwersacją
<code>.scs-convo-list-cust .scs-convo-list-active</code>	Tytuł konwersacji na liście, gdy konwersacja jest wybrana
<code>.scs-convo-list-cust .scs-convo-list-convo-posts</code>	Liczba wpisów w konwersacji
<code>.scs-convo-list-cust .scs-convo-list-convo-unread</code>	Liczba nieprzeczytanych wiadomości w konwersacji
<code>.scs-convo-list-cust .scs-convo-list-convo-updated</code>	Data ostatniej aktualizacji konwersacji
<code>.scs-convo-list-cust .scs-convo-list-no-convo-msg</code>	Komunikat, gdy lista jest pusta
<code>.scs-convo-list-cust .scs-convo-list-no-auth-msg</code>	Komunikat, gdy lista konwersacji jest renderowana w serwisie publicznym bez identyfikacji użytkownika

Zob. rozdział Korzystanie ze stylów i formatowania w podręczniku *Tworzenie serwisów za pomocą Oracle Content Management*.

### Przykład

Poniższy przykład ilustruje użycie dostosowanego arkusza CSS, zmieniającego typ, styl i kolor czcionki listy konwersacji.



Następujący kod przedstawia dostosowany arkusz CSS, który został użyty do utworzenia przykładu:

```
.scs-convo-list-cust .scs-convo-list-container {
    background-color: azure;
}

.scs-convo-list-cust .scs-convo-list-title {
    color: crimson;
}

.scs-convo-list-cust .scs-convo-list-line-separator {
    border-bottom: 2px dashed #dfe4e7;
}

.scs-convo-list-cust .scs-convo-list-convo-title {
    font-style: italic;
}

.scs-convo-list-cust .scs-convo-list-active {
    text-decoration: underline;
}

.scs-convo-list-cust .scs-convo-list-convo-posts {
    color: cadetblue;
    font-size: 12px;
}

.scs-convo-list-cust .scs-convo-list-convo-unread {
    color: brown;
    font-size: 12px;
    float: left;
}

.scs-convo-list-cust .scs-convo-list-convo-updated {
    color: blueviolet;
    font-size: 12px;
    clear: none;
}

.scs-convo-list-cust .scs-convo-list-no-convo-msg {
    font-size: 18px;
    color: darkorange;
}
```

```
.scs-convo-list-cust .scs-convo-list-no-auth-msg {
  font-size: 18px;
  color: red;
}
```

## Dostosowywanie stylów listy folderów i listy plików

Dodając selektory do pliku `design.css`, można dostosować style składników "lista folderów" i "lista plików".

Za pomocą składnika "lista folderów" można uzyskać wykaz folderów z określonego folderu ze swojego konta Oracle Content Management. Lista folderów automatycznie komunikuje się ze składnikiem "lista plików" i menedżerem dokumentów, umożliwiając wyświetlenie plików zawartych w folderze wybranym na liście folderów.

Za pomocą składnika "lista plików" można uzyskać widok plików z określonego folderu ze swojego konta Oracle Content Management. Lista plików automatycznie komunikuje się ze składnikiem obsługującym listę folderów, umożliwiając wyświetlenie plików zawartych w folderze wybranym na liście folderów.

### Lista folderów — selektory CSS

Za pomocą poniższych selektorów CSS można dostosować styl składnika "lista folderów".

Nazwa selektora	Opis
<code>.scs-folder-list-cust .scs-folder-list-container</code>	Najbardziej zewnętrzny element DIV składnika
<code>.scs-folder-list-cust .scs-folder-list-folder-title</code>	Nazwa folderu
<code>.scs-folder-list-cust .scs-folder-list-line-separator</code>	Separator między nazwą folderu i listą podfolderów
<code>.scs-folder-list-cust .scs-folder-list-subfolder-title</code>	Nazwa podfolderu
<code>.scs-folder-list-cust .scs-folder-list-subfolder-title-active</code>	Nazwa podfolderu, gdy zostanie wybrany
<code>.scs-folder-list-cust .scs-folder-list-no-folder-msg</code>	Komunikat, gdy nie ma żadnego podfolderu do wyświetlenia

### Lista plików — selektory CSS

Za pomocą poniższych selektorów CSS można dostosować styl składnika "lista plików".

Nazwa selektora	Opis
<code>.scs-file-list-cust .scs-file-list-container</code>	Najbardziej zewnętrzny element DIV składnika
<code>.scs-file-list-cust .scs-file-list-folder-title</code>	Nazwa folderu

Nazwa selektora	Opis
<code>.scs-file-list-cust .scs-file-list-line-separator</code>	Separator między nazwą folderu i listą plików
<code>.scs-file-list-cust .scs-file-list-row</code>	Wiersz zawierający informacje o pliku
<code>.scs-file-list-cust .scs-file-list-left-col</code>	Miniatura pliku zlokalizowana w lewej sekcji składnika
<code>.scs-file-list-cust .scs-file-list-mid-col</code>	Środkowa sekcja składnika, zawierająca nazwę, opis, ostatnią modyfikację i rozmiar pliku
<code>.scs-file-list-cust .scs-file-list-file-title</code>	Nazwa pliku zlokalizowana w środkowej sekcji aplikacji
<code>.scs-file-list-cust .scs-file-list-file-desc</code>	Opis pliku zlokalizowana w środkowej sekcji aplikacji
<code>.scs-file-list-cust .scs-file-list-file-lastModified</code>	Ostatnia modyfikacja pliku
<code>.scs-file-list-cust .scs-file-list-file-size</code>	Rozmiar pliku z pionowym separatorem od ostatniej modyfikacji
<code>.scs-file-list-cust .scs-file-list-file-size-no-sep</code>	Rozmiar pliku bez pionowego separatora (nie jest wyświetlana ostatnia modyfikacja)
<code>.scs-file-list-cust .scs-file-list-right-col</code>	Prawa sekcja aplikacji
<code>.scs-file-list-cust .scs-file-list-file-download-icon</code>	Ikona "Pobierz" zlokalizowana w prawej sekcji aplikacji
<code>.scs-file-list-cust .scs-file-list-no-file-msg</code>	Komunikat, gdy nie ma żadnego pliku do wyświetlenia

Zob. Listy plików i Listy folderów w podręczniku *Tworzenie serwisów za pomocą Oracle Content Management*.

## Dostosowywanie ikon paska społecznościowego

Można utworzyć niestandardowe ikony społecznościowe, które będą używane na pasku społecznościowym w domyślnym serwisie motywu.

Ikony społecznościowe, które są widoczne na pasku społecznościowym w serwisie, są określane przez projekt motywu serwisu. Jeśli motyw serwisu zostanie zmieniony, ikony społecznościowe zmieniają się wraz z motywem. Wspólne ikony społecznościowe są zawarte w motywach Oracle Content Management dla następujących serwisów: Facebook, Twitter, LinkedIn, Google+ i YouTube.

Edytując pliki `design.json` i `design.css`, można dodać do składnika "ikony społecznościowe" niestandardowe ikony społecznościowe.

### Plik design.json

Używając nazwy i struktury klasy, można w pliku `design.json` określić nowe ikony, jak pokazano w następującym przykładowym kodzie:

```
"componenticons": {
  "scs-socialbar" {
    "icons": [
      {
        "name": "COMP_ICON_FACEBOOK",
        "class": "scs-facebook-icon"
      },
      {
        "name": "COMP_ICON_LINKEDIN",
        "class": "scs-linkedin-icon"
      },
      {
        "name": "COMP_ICON_TWITTER",
        "class": "scs-twitter-icon"
      },
      {
        "name": "COMP_ICON_GOOGLEPLUS",
        "class": "scs-googleplus-icon"
      },
      {
        "name": "COMP_ICON_YOUTUBE",
        "class": "scs-youtube-icon"
      }
    ]
  }
}
```

### Plik design.css

Używając nazwy i specyfikacji `url`, można w pliku `design.css` dodać nowe ikony, jak pokazano w następującym przykładowym kodzie:

```
.scs-facebook-icon {
  background-image: url("facebook.png"); }
.scs-twitter-icon {
  background-image: url("twitter.png"); }
.scs-linkedin-icon {
  background-image: url("linkedin.png"); }
```

## Konfigurowanie rozszerzeń stosujących stylistykę do wywiadów dla Oracle Intelligent Advisor

Do wywiadów OIA (Oracle Intelligent Advisor, dawniej Oracle Policy Automation) można — w celu ujednoczenia ich wyglądu — stosować stylistykę.

W pliku `design.css` można skonfigurować przedstawione poniżej selektory klas CSS. Wszystkie te selektory klas mają prefiks `scs-opainterview-`.

Każda klasa definiuje stylizację określonego składnika. Klasa sprawuje pełną kontrolę — istniejący styl OIA nie będzie używany.

Selektor	Do czego jest stosowany	Opis
interview	interview	Obszar wywiadu obejmuje całą zawartość wywiadu, w tym nagłówek, stopkę i obszar nawigacyjny.
interviewContent	treść wywiadu	Obszar "treść wywiadu" obejmuje tytuł ekranu oraz formanty, bez nagłówka, stopki i obszaru nawigacyjnego.
screenTitleBlock	blok tytułu ekranu	Blok tytułu ekranu tworzy obszar zawierający tytuł ekranu oraz inne widżety zawarte w tym rzędzie, takie jak lista rozwijana i/lub przyciski "Dalej" i "Wstecz".
screenTitle	tytuł ekranu	Obszar tytułu ekranu jest, jak wskazuje nazwa, obszarem zawierającym tytuł ekranu.
nextButton	przycisk "Dalej"	Przycisk "Dalej".
backButton	Przycisk "Wstecz"	Przycisk "Wstecz".
restartButton	Przycisk "Uruchom ponownie"	Przycisk "Uruchom ponownie".
exitButton	przycisk "Zakończ"	Przycisk "Zakończ".
header	header	Obszar nagłówka.
footer	footer	Obszar stopki.
question	tekst pytania	Stylizacja tekstu pytania.
control	kontener dla formantów	Stylizacja elementu zawierającego formanty.
label	formant "etykieta"	Stylizacja formantów "etykieta".
controlError	wszystkie formanty	Stylizacja kontenera tekstu błędu.
controlErrorText	wszystkie formanty	Stylizacja tekstu błędu.
textInput	pole tekstowe jednowierszowe, hasło i maskowane	Stylizacja formantów "tekst wejściowy".
textAreaInput	pole tekstowe wielowierszowe	Stylizacja formantów "tekst wejściowy wielowierszowy".
calendarInput	calendar	Stylizacja formantów "dane wejściowe z kalendarza". Obsługuje pole <code>iconColor</code> umożliwiające zmianę koloru ikony kalendarza, a także <code>keepIcon</code> sygnalizujące, czy ikona kalendarza ma być wyświetlana.
dropDownInput	lista rozwijana	Stylizacja formantów "lista rozwijana".
filterDropDownInput	filtrowana lista rozwijana	Stylizacja formantów "filtrowana lista rozwijana". Obsługuje pole <code>iconColor</code> umożliwiające zmianę strzałki rozwijającej.
listInput	lista stała	Stylizacja formantów "lista stała".
radioInput	przyciski radiowe	Stylizacja formantów "przyciski radiowe". Można je stylizować za pomocą opcji <code>borderColor</code> i <code>fillColor</code> . Właściwość <code>iconType</code> pozwala zmienić typ <code>tick</code> (zaznaczenie) i <code>fill</code> (wypełnienie).

Selektor	Do czego jest stosowany	Opis
checkboxInput	pole wyboru	Stylistyka formantów "pole wyboru". Można je stylizować za pomocą opcji <code>borderColor</code> i <code>fillColor</code> . Właściwość <code>iconType</code> pozwala zmienić typ używanej ikony. Obecnie są obsługiwane jedynie opcje <code>square</code> (kwadrat) i <code>fill</code> (wypełnienie).
autocompleteInput	wyszukiwanie niestandardowe	Stylizacja pola <code>autocomplete</code> , gdy jest używane rozszerzenie <code>customSearch</code> .
captchaInput	pole wejściowe CAPTCHA	Stylizacja pola wejściowego, w którym użytkownik wprowadza informacje CAPTCHA.
signatureInput	formant "podpis"	Stylistyka formantów "podpis". Obsługuje dodatkowe pole <code>inkColor</code> umożliwiające zmianę koloru podpisu.
explanationHeader	formant "objaśnienie"	Stylistyka rozwijalnego nagłówka najwyższego poziomu dla formantów "objaśnienie"
explanationText	formant "objaśnienie"	Stylistyka tekstu rozwiniętego formantu "objaśnienie".
signatureClearButton	formant "podpis"	Stylistyka przycisku czyszczenia w formancie "podpis".
uploadAddButton	formant "wyślij"	Stylistyka przycisku dodawania w formancie "wyślij".
entityRemoveButton	formant "usuń encję"	Formant "usuń encję".

#### Przykłady z rozszerzeniami stylów zdefiniowanymi w pliku `design.css`

```
.scs-opainterview-interviewContent {
    background-color: beige;
}
.scs-opainterview-screenTitleBlock {
    background-color: bisque;
}
.scs-opainterview-screenTitle {
    font-style: italic;
    font-size: 20px;
}
.scs-opainterview-nextButton {
    color: darkgreen;
}
.scs-opainterview-backButton {
    color: crimson;
}
.scs-opainterview-question {
    color: green;
}
.scs-opainterview-control {
    background-color: cornflowerblue;
}
.scs-opainterview-label {
    color:aqua;
}
```

```
.scs-opainterview-textInput {  
  color: red;  
  cursor:crosshair;  
}  
.scs-opainterview-radioInput {  
  background-color: pink;  
}  
.scs-opainterview-checkboxInput {  
  cursor: pointer;  
}
```



## Omówienie użycia tła

Można określić kolor i obraz tła stron serwisu oraz poszczególnych gniazd na stronie. Tła dla stron, gniazd i składników układają się warstwami. Na przykład, jeśli dla gniazda zostanie określone tło, zostanie ono ułożone warstwowo na tle określonym dla strony.

- [Tła i motywy — informacje podstawowe](#)
- [W jaki sposób są implementowane tła](#)
- [Gdzie są przechowywane ustawienia](#)

### Tła i motywy — informacje podstawowe

Funkcja określania tła jest przeznaczona do użycia przede wszystkim na stronach serwisu i w gniazdach. Tła ustawiane dla stron i gniazd są konfigurowane przez użytkowników (nie programistów) na poziomie motywu.

Efekty związane z tłem można konfigurować w gniazdach zdefiniowanych w motywie; przez te efekty mogą zostać przesłonięte wszelkie efekty związane z tłem stron, ustawione podczas edytowania serwisu.

Funkcja określania tła nie modyfikuje plików motywu (w tym plików go tworzących). Motywu nie można zmodyfikować, zmieniając ustawienia tła strony; inna strona oparta na tym samym układzie nie odziedziczy żadnych z ustawień strony źródłowej. Aby dodać lub zmienić tło przez motyw, trzeba go zaktualizować.

Użytkownicy mogą konfigurować różne tła dla każdej ze stron serwisu, a także dla różnych stron używających tego samego układu. Nie byłoby to możliwe do osiągnięcia w motywie. Mimo że motyw może definiować tło strony, to specyfikacja ta może zostać przesłonięta przez ustawienia tła określone przez użytkowników. (Ustawienia tła mają zastosowanie tylko do konkretnej strony; w żaden sposób nie modyfikują motywu.)

Należy dołożyć starań, aby uniknąć przesłonięcia — przez motyw — stylów tła skonfigurowanych dla stron serwisu i gniazd przez użytkowników. Jest to możliwe w następujących sytuacjach:

- Ustawienia tła strony serwisu mogą zostać przesłonięte przez motyw, jeśli w sekcji `<body>` i w gniazdach będą używane atrybuty `"style"` oparte na elemencie.
- Ustawienia tła strony serwisu mogą zostać przesłonięte przez motyw, jeśli w pliku CSS motywu style tła zostaną oznaczone jako `"!important"`.

Zob. rozdział *Zmianie tła lub motywu* w podręczniku *Tworzenie serwisów za pomocą Oracle Content Management*.

### W jaki sposób są implementowane tła

W celu zaimplementowania funkcji tła strony i tła gniazda, Oracle Content Management dynamicznie tworzy arkusz stylów CSS w kodzie `<head>` strony.

Selektor stylów dla ustawień tła strony jest selektorem opartym na znacznikach (`body`). Dla ustawień tła gniazda i dla selektora opartego na ID jest używany ID gniazda.

Na przykład, ustawiając kolor tła strony, można uzyskać następujący kod CSS w kodzie `<head>` strony:

```
body
{
    background-color: #fa7c9d;
}
```

Analogicznie, ustawiając obraz jako tło gniazda, można uzyskać następujący kod CSS w kodzie `<head>` strony:

```
#PageFooter
{
    background-image: url("footer_image.png");
}
```

Taka implementacja oznacza, że ustawienia skonfigurowane w arkuszu stylów w kodzie `<head>`, mogą zostać przesłonięte przez style, które zostały określone bezpośrednio w atrybucie `"style"` znacznika `"body"` lub w elemencie gniazda.

#### ! Ważne:

Twórcy motywów powinni zadbać, aby ustawienia tła nie zostały przesłonięte przez style oparte na elementach.

Zob. rozdział *Zmianianie tła lub motywu* w podręczniku *Tworzenie serwisów za pomocą Oracle Content Management*.

## Gdzie są przechowywane ustawienia

Gdy strony są renderowane, ustawienia tła są dynamicznie zapisywane w znacznikach `"style"` w kodzie `<head>` strony.

Ustawienia tła są utrwalane w plikach modelu strony (na przykład `<pageid>.json`). Ustawienia tła stron są przechowywane w sekcji `properties.styles`, a ustawienia tła gniazd w sekcji `slots[<id_gniazda>].styles`.

Poza tym ustawienia tła są przechowywane w plikach JSON stron, konkretnie w obrębie wpisu `"styles"`, jak pokazano w przykładzie.

```
{
  "properties":
  {
    "pageLayout" : "oneslot.htm",
    "styles": [
      "background-image: url([!--$SCS_CONTENT_URL--]/
background_image.gif)",
      "background-position: center",
```

```

        "background-size: auto",
        "background-repeat: repeat",
        "background-origin: padding-box",
        "background-clip: border-box"
    ]
},
"slots":
{
    "slot100":
    {
        "components":
        [
            "dedda3a8-615d-44ad-ad71-51f2fa465cef",
            "95eb0fd6-bcfc-4e5e-ba67-a5c8c5d9c315"
        ],
        "grid": "<div class=\"scs-row\"><div class=\"scs-
col\" style=\"width: 50%;\">
<div id=\"dedda3a8-615d-44ad-ad71-51f2fa465ced\">
</div>
</div>
<div class=\"scs-col\" style=\"width: 50%;\">
<div id=\":95eb0fd6-bcfc-4e5e-ba67-a5c8c5d9c315\">
</div>
</div>
</div>\",
        "styles": [
            "background-image: url([!--$SCS_CONTENT_URL--]/
oracle-cloudworld.jpg)",
            "background-position: center",
            "background-size: cover",
            "background-repeat: no-repeat",
            "background-origin: padding-box",
            "background-clip: border-box",
            "background-color: transparent"
        ]
    }
},
"componentInstances":
{
    "dedda3a8-615d-44ad-ad71-51f2fa465cef":
    {
        "type": "scs-title",
        "data": {
            "alignment": "fill",
            "backgroundColor": "",
            "borderColor": "#808080",
            "borderRadius": 0,
            "borderStyle": "none",
            "borderWidth": 1,
            "fontColor": "#333333",
            "fontFamily": "'Helvetica Neue', Helvetica, Arial, sans-
serif",
            "fontSize": 24,

```

```

        "marginBottom": 5,
        "marginLeft": 5,
        "marginRight": 5,
        "marginTop": 5,
        "styleClass": "",
        "useStyleClass": "true",
        "userText": "<div>My Test Title</div>\n",
        "width": 400
    }
},
"95eb0fd6-bcfc-4e5e-ba67-a5c8c5d9c315":
{
    "type": "scs-image",
    "data": {
        "styleClass": "",
        "useStyleClass": "true",
        "imageUrl": "[!-$SCS_CONTENT_URL--]/example.jpg",
        "defaultImageUrl": "/components/comp/images/
default_image.png",
        "style": "",
        "imageWidth": 0,
        "borderStyle": "none",
        "borderWidth": 1,
        "borderColor": "black",
        "borderRadius": 0,
        "altText": "My Image",
        "title": "My Title",
        "caption": "My Caption",
        "imageAlignment": "center",
        "imageHref": "",
        "imageTarget": "_self",
        "marginTop": 0,
        "marginRight": 0,
        "marginBottom": 0,
        "marginLeft": 0,
        "linkType": "scs-link-no-link"
    }
}
}
}
}

```

## Ustawianie wyzwalaczy i czynności

Można skonfigurować komunikację między składnikami (w tym między składnikami renderowanymi w ramach wstawkowych), tak aby wyzwalacz w składniku wywoływał czynność w innym składniku.

- [Wyzwalacze i czynności — informacje podstawowe](#)
- [Ustawianie wyzwalaczy](#)
- [Ustawianie czynności](#)

### Wyzwalacze i czynności — informacje podstawowe

Można skonfigurować komunikację między składnikami (w tym między składnikami renderowanymi w ramach wstawkowych), tak aby **wyzwalacz** w składniku wywoływał **czynność** w innym składniku.

Wyzwalacze stanowią część mechanizmu komunikacji między składnikami Oracle Content Management. Każdy składnik może uaktywniać dowolną liczbę wyzwalaczy. Składnik może dostarczać do wyzwalacza ładunek, który jest przekazywany do czynności uruchamianej w chwili uaktywnienia wyzwalacza. Można wybrać czynności uruchamiane dla poszczególnych wyzwalaczy. Składniki, które są projektowane z myślą o ich współpracy, mogą automatycznie uaktywniać wyzwalacze w celu uruchamiania czynności w innym składniku bez interakcji użytkownika.

Podstawowy proces obejmuje:

1. Rejestrowanie wyzwalaczy
2. Uaktywnianie wyzwalaczy
3. Rejestrowanie czynności
4. Uruchamianie czynności w celu sprawdzenia konfiguracji

Na przykład używając składników "przycisk", można zlecić wykonywanie czynności, takich jak pokazywanie lub ukrywanie składników strony czy wyświetlanie komunikatów. W jednym składniku może być zawarta lista siedzib firm i — gdy użytkownik kliknie na siedzibie danej firmy — w innym składniku zostaną wyświetlone szczegóły lokalizacji.

W przypadku składnika dostosowywanego wyzwalacze i czynności są rejestrowane w ramach jego danych rejestracyjnych — nie są elementem implementacji składnika. W danych rejestracyjnych występuje wpis "triggers": [], i "actions": [], zawierający listę wyzwalaczy i czynności obsługiwanych przez składnik. Składnia jest identyczna z używaną dla składników lokalnych i odległych; różni się tylko lokalizacja i sposób jej pobierania.

Zob. rozdział *Używanie wyzwalaczy i czynności w podręczniku Tworzenie serwisów za pomocą Oracle Content Management*.

## Ustawianie wyzwalaczy

Składnik może zawierać wyzwalacze uruchamiające czynności w innych składnikach. Wyzwalacze, aby były uaktywniane przez składniki, trzeba zarejestrować.

Składnik dostarcza do wyzwalacza ładunek, który jest przekazywany do czynności uruchamianej w chwili uaktywnienia wyzwalacza. Można wybrać czynności uruchamiane dla poszczególnych wyzwalaczy. Składniki, które są projektowane z myślą o ich współpracy, mogą automatycznie uaktywniać wyzwalacze w celu uruchamiania czynności bez interakcji użytkownika.

### Rejestrowanie wyzwalaczy

W przypadku składnika niestandardowego wyzwalacze są rejestrowane w ramach jego danych rejestracyjnych. Aby dodać wyzwalacz, należy zaktualizować tablicę właściwości "triggers", dodając do niej obsługiwany wyzwalacz. Trzeba także określić ładunek obsługiwany przez wyzwalacz, aby mógł zostać utworzony interfejs użytkownika umożliwiający użytkownikom odwzorowywanie wartości z ładunku na właściwości obsługiwane przez czynność.

1. Otworzyć plik `appinfo.json` do edycji i sprawdzić wpis `"triggers":[]`:

```
"triggers": [{
  "triggerName": "helloWorldWhoAreYou",
  "triggerDescription": "Show Who I Am",
  "triggerPayload": [{
    "name": "whoAreYou",
    "displayName": "Who I Am"
  }]
}],
```

2. Zsynchronizować plik z serwerem serwisów.

W tym przykładowym wpisie wyzwalacza została zdefiniowana właściwość `triggerName` ("helloWorldWhoAreYou"). Wartość tej właściwości musi być unikatowa. Wyzwalacz ma przypisany opis ("Show Who I Am"), który jest używany w oknie dialogowym, w interfejsie użytkownika, do wyświetlenia wyzwalacza. Na koniec został zdefiniowany ładunek wartości dla wyzwalacza; użytkownicy będą mogli wybierać wpisy z tego ładunku i odwzorowywać je na pola używane w czynności.

Gdy wyzwalacz zostanie zarejestrowany, będzie można go zobaczyć i wybrać po przejściu na kartę **Łącze** w panelu ustawień składnika.

### Uaktywnianie wyzwalaczy

Wyzwalacze mogą być uaktywniane w dowolnej chwili przez składnik. Zazwyczaj wyzwalacz jest uaktywniany wskutek interakcji użytkownika, takiej jak naciśnięcie przycisku lub wybranie wiersza w tabeli. Składnik może uaktywnić wyzwalacz na podstawie dowolnych kryteriów — na przykład, gdy dane ulegną zmianie spowodowanej wywołaniem REST API. Gdy wyzwalacz zostanie uaktywniony, może być wykonywana dowolna liczba czynności.

Przykład uaktywniania wyzwalacza:

1. Otworzyć plik `render.js` do edycji i dodać w obiekcie `viewModel` funkcję JavaScript, która będzie wywoływać Sites SDK w celu uaktywnienia wyzwalacza.

```
self.raiseTrigger = function (triggerName) {  
  SitesSDK.publish(SitesSDK.MESSAGE_TYPES.TRIGGER_ACTIONS, {  
    'triggerName': 'helloWorldWhoAreYou',  
    'triggerPayload': { "whoAreYou": "This is " + self.whoAreYou() + "!" }  
  });  
};
```

2. Dodać w interfejsie użytkownika wpis wywołujący funkcję uaktywniającą wyzwalacz (`-edit template.html`) oraz poprzedzić składnik `</div>` przyciskiem.

```
<button data-bind="click raiseTrigger">Who Am I?</button>
```

3. Zsynchronizować (lub wysłać) plik `render.js` z serwerem swojej instancji Oracle Content Management.

W obiekcie `viewModel` utworzyliśmy funkcję JavaScript, która zostanie wywołana, gdy przycisk zostanie naciśnięty (`click`). Funkcja ta wywołuje Sites SDK i zleca mu uaktywnienie wszystkich czynności zdefiniowanych dla tego wyzwalacza "helloWorldWhoAreYou". Przekazuje także ładunek `triggerPayload` zawierający jedno pole "whoAreYou". Wartości "helloWorldWhoAreYou" i "whoAreYou" odpowiadają tym, które wprowadziliśmy na poprzednim etapie, rejestrując wyzwalacz.

#### Uwaga:

Nie ma określonej kolejności wykonywania czynności. Mimo że czynności są wywoływane w kolejności, w jakiej są wyszczególnione, to jednak nie ma okresu oczekiwania na ukończenie poprzedniej czynności — następną czynność jest wywoływana od razu. Jeśli czynność wykonuje wywołanie asynchroniczne, to jej działanie może nie zostać ukończone, zanim zostanie wywołana następną czynność.

## Ustawianie czynności

Można skonfigurować składnik tak, aby korzystał z zarejestrowanych czynności — gdy zostanie przeciągnięty na stronę, czynności te zostaną wykonane w obrębie składnika.

### Rejestrowanie czynności

Czynności są wywoływane dla składników, gdy zostaną uaktywnione odpowiednie wyzwalacze. Składnik może mieć zarejestrowaną dowolną liczbę czynności, przy czym można zdefiniować ładunek obsługiwany przez daną czynność. Użytkownik, wybierając czynność, może wypełnić danymi przekazywany do niej ładunek.

Podobnie jak w przypadku rejestrowania wyzwalaczy, można — używając danych rejestracyjnych `appinfo.json` dla danego motywu — rejestrować czynności obsługiwane przez składnik.

Przykład rejestrowania czynności:

1. Otworzyć do edycji plik `appinfo.json` składnika, po czym zaktualizować wpis `"actions": [ ]`,

```
"actions": [{
  "actionName": "helloWorldChangeWhoIAm",
  "actionDescription": "Change Who I Am",
  "actionPayload": [{
    "name": "whoAreYou",
    "description": "Who are you?",
    "type": {
      "ojComponent": {
        "component": "ojInputText"
      }
    },
    "value": ""
  }]
}]
```

2. Zarejestrowana czynność będzie od razu widoczna w oknie dialogowym czynności, które jest wywoływane, gdy użytkownik kliknie na wyzwalaczu na karcie **Łącze** w panelu ustawień składnika.

### Wykonywanie czynności

Mając zarejestrowane czynności, można przeciągać składniki na stronę, a czynności te będą wykonywane w obrębie składnika. Składnik, aby móc wykonać czynność, musi nasłuchiwać komunikatu `EXECUTE_ACTION`. W komunikacie tym jest także zawarty ładunek przekazany do czynności, z którego trzeba wyodrębnić oczekiwane wartości.

Na przykład, aby był prowadzony nasłuch komunikatu `EXECUTE_ACTION`, należy otworzyć plik `render.js` do edycji i zaktualizować w nim obiekt `ViewModel`, używając następującego kodu:

```
self.executeActionListener = function (args) {
  // get action and payload
  var payload = $.isArray(args.payload) ? args.payload[0] : {},
      action = args.action,
      actionName = action && action.actionName;

  // handle 'helloWorldChangeWhoIAm' actions
  if ((actionName === 'helloWorldChangeWhoIAm') && (payload.name ===
'whoAreYou')) {
    self.whoAreYou(payload.value);
  }
};
```

Kod ten tworzy funkcję JavaScript wykonywania czynności, a następnie używa biblioteki Sites SDK do wywołania tej funkcji, gdy tylko zostanie zgłoszony komunikat `EXECUTE_ACTION`.

Czynność ta będzie wywoływana przy każdym zgłoszeniu komunikatu `EXECUTE_ACTION`, a o tym, czy zostanie wykonana, decyduje sam składnik. Dlatego, aby mieć pewność, że jest to czynność obsługiwana, trzeba sprawdzić jej nazwę.



Ładunek dla czynności ma postać tablicy wartości. W tym przykładzie zostało przyjęte założenie, że wartością jest pierwszy wpis w tablicy. Zazwyczaj trzeba odnaleźć w tablicy odpowiednie wartości ładunku.

 **Uwaga:**

Ponieważ nasłuch funkcji działa w trybie wywołania zwrotnego, należy — aby mieć pewność, że w chwili uruchomienia funkcji jest możliwy dostęp do obiektu `ViewMode` 1 — użyć domknięcia (closure) JavaScript albo odpowiednio powiązać funkcję.

## Tworzenie szablonów

Pakiet tłumaczenia zawiera programistyczną wersję serwisu, motyw z układami strony, styl i nawigację oraz powiązane składniki używane w serwisie. Oracle Content Management udostępnia zbiór szablonów, których można używać do tworzenia serwisów oraz które mogą być używane jako punkt wyjścia do tworzenia szablonów niestandardowych.

- [Szablony — informacje podstawowe](#)
- [Podstawowa struktura szablonu](#)
- [Tworzenie szablonu](#)
- [Eksportowanie szablonu](#)
- [Importowanie szablonu](#)
- [Praca z szablonem początkowym](#)
- [Tworzenie szablonu serwisu z szablonu "Bootstrap" lub "Website Design"](#)
- [Tworzenie szablonów za pomocą usługi Developer Cloud Service](#)

### Szablony — informacje podstawowe

Szablon zawiera wszystkie elementy potrzebne do rozpoczęcia tworzenia serwisu, w tym serwis domyślny z przykładowymi stronami i zawartością, motyw ze stylami, nawigację, zasoby (takie jak obrazy) oraz powiązane składniki.

Oracle Content Management udostępnia różne szablony, których można używać podczas tworzenia serwisów. Szablony te są zazwyczaj instalowane przez administratora podczas inicjalizowania usługi. Zob. Konfigurowanie ustawień serwisów w podręczniku *Administrowanie Oracle Content Management*.

Za każdym razem, gdy jest tworzony nowy serwis, trzeba wybrać szablon. Szablony łączą motywy z serwisami i składnikami, udostępniając funkcje lub rozwiązania właściwe na przykład dla portalu partnerskiego lub kampanii marketingowej.

Lista gotowych szablonów jest dostępna pod hasłem [Omówienie szablonów](#).

Podczas gdy programiści i użytkownicy mogą tworzyć nowe szablony oraz modyfikować i zastępować już istniejące szablony, to jednym z głównych zadań programistów jest projektowanie nowych szablonów. Ten proces składa się zasadniczo z poniższych etapów.

1. Utworzyć nowy szablon, kopiując już istniejący (na przykład szablon początkowy JET). W wyniku tej czynności uzyskuje się również motyw powiązany z szablonem.

Na przykład:

```
cec create-template My_JET_Template -f JETStarterTemplate
```

2. Wyeksportować szablon z pliku `.zip` do środowiska programistycznego.
3. Otworzyć pliki w pakiecie szablonu, po czym wprowadzić zmiany.

4. Utworzyć zmieniony pakiet szablonu w pliku `.zip`.
5. Użyć interfejsu Oracle Content Management do zaimportowania nowego szablonu do instancji.
6. Udostępnić szablon, dzięki czemu inne osoby będą mogły z niego korzystać.

Można także użyć interfejsu Oracle Content Management do zmodyfikowania szablonu; wystarczy dodać, a następnie zmodyfikować układy strony i zasoby, rozwinąć strukturę serwisu, dodać składniki do stron serwisu, po czym dodać standardową zawartość, która ma być wyświetlana w każdym serwisie używającym motywu z tego szablonu.

Zob. także [Zarządzanie szablonami](#).

## Podstawowa struktura szablonu

Podstawowa struktura szablonu zawiera serwis (z zasobami, układami, stronami i zawartością), powiązany motyw oraz wszelkie składniki niestandardowe.

Tworząc serwis, trzeba wybrać szablon, zapewniający serwisowi strukturę i zawartość początkową, motyw z projektem i specyfikacjami układu oraz wszelkie składniki niestandardowe.

Szablon jest uporządkowany według określonej struktury, tak jak zostało to przedstawione w poniższym przykładzie pokazującym podstawowe foldery i pliki.

```
template_name
  components
    component_name
      assets
        render.js
        settings.html
      _folder.json
      _folder_icon.jpg
      appinfo.json
  template
    assets
    content
    layouts
    pages
      100.json
      200.json
      300.json
      400.json
    variants
      _folder.json
      _folder_icon.png
      componentsused.json
      controller.html
      siteinfo.json
      structure.json
  theme
    assets
      css
        main.css
      js
```

```
topnav.js
designs
  default
    design.css
    design.json
    facebook.png
    googleplus.png
    linkedin.png
    twitter.png
    youtube.pgn
  layouts
    index.html
responsepages
  404.html
_folder.json
_folder_icon.png
components.json
viewport.json
```

Foldery i pliki składników i motywów zostały opisane w innym rozdziale tego podręcznika. Zob. [Tworzenie składników — informacje podstawowe](#) i [Podstawowa struktura motywu](#).

#### Uwagi:

- Ogólnie motyw jest współużytkowany przez szablony, chyba że zostanie użyty szablon początkowy JET lub szablon początkowy, który korzysta z kopii motywu. Zob. [Praca z szablonem początkowym](#).
- Motyw nie zawiera już serwisu. Foldery i pliki serwisu są przechowywane w folderze `/template`.

Folder `nazwa_szablonu/template` zawiera foldery i pliki serwisu.

- `assets`: Zawiera obrazy, które są wyświetlane na stronie szczegółów szablonu w interfejsie użytkownika.
- `content`: Zawiera zarządzaną zawartość, która jest używana w serwisie.
- `layouts`: Obecnie nie jest używany.
- `pages`: Zawiera wszystkie pliki JSON strony wraz z danymi. Używa formatu `nnn.json`, gdzie `nnn` to ID strony.
- `variants`: Zawiera szczegóły wszystkich aktualizacji serwisu.
- `_folder.json`: Zawiera metadane szablonu, takie jak autor szablonu, nazwa serwisu, GUID elementu, krótki oraz długi opis serwisu.
- `_folder_icon.png`: Reprezentuje serwis w interfejsie użytkownika.
- `componentsused.json`: (Przestarzałe.) Zapisuje składniki niestandardowe, które są używane (jeśli takie istnieją), w obrębie serwisu. Plik ten jest utrzymywany wyłącznie w celu zapewnienia zgodności wstecznej.
- `controller.html`: Zawiera kod klucza, który wyświetla serwis w przeglądarce. Jeśli trzeba dokonać zmian w tym pliku, Oracle zaleca skorzystanie z ustawień serwisu

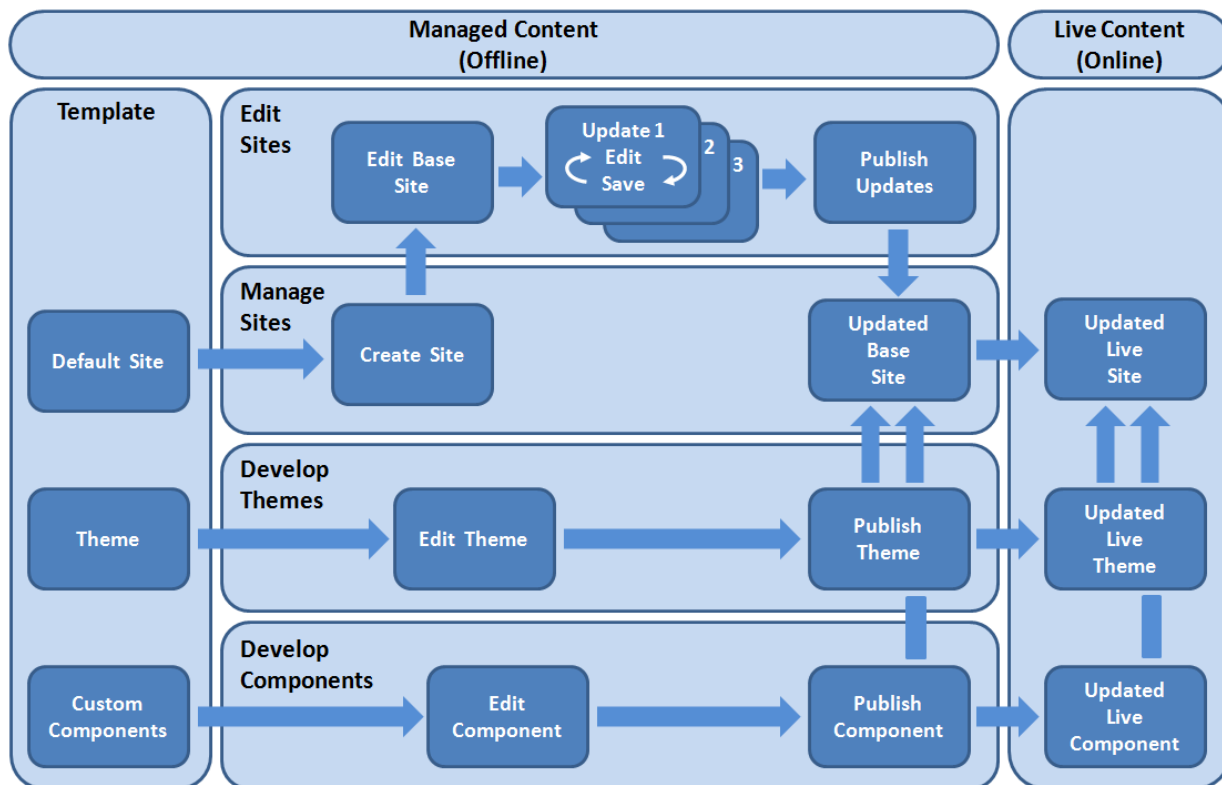
w interfejsie. Plik ten można modyfikować w trybie offline. Zob. [Dostosowywanie pliku kontrolera](#).

- `siteinfo.json`: Identyfikuje nazwę serwisu i nazwę powiązanego motywu wraz z innymi metadanymi serwisu. Tego pliku nie należy modyfikować.
- `structure.json`: Definiuje hierarchię stron w serwisie (stron nadrzędnych i podrzędnych). Render API można używać do tworzenia struktury drzewa podczas konfigurowania nawigacji dla serwisu. Zob. [Nawigacja serwisu](#) i [Render API — opis](#).

## Tworzenie szablonu

Mając serwis, który może zostać użyty jako punkt startowy dla innych serwisów, można z tego serwisu utworzyć szablon. Można także utworzyć nowy szablon, kopiując istniejący szablon i dokonując zmian w kopii.

Jeśli szablon jest tworzony z istniejącego serwisu, to w nowym szablonie — jako serwis domyślny — jest używana kopia tego serwisu. Szablon odwołuje się do motywu używanego przez serwis i do wszystkich składników niestandardowych, które są używane na stronach serwisu. Motyw i składniki niestandardowe nie są kopiowane do szablonu, lecz występują do nich odwołania (tak samo jak w przypadku serwisu). Szablon odzwierciedla serwis, który został użyty do utworzenia szablonu w chwili jego tworzenia. Dalsze zmiany, dokonywane w serwisie użytym do utworzenia szablonu, nie są odzwierciedlane w serwisie przechowywanym w szablonie.



Jeśli nowy szablon jest tworzony przez skopiowanie już istniejącego szablonu (i nadanie nowej nazwy), wszelkie zmiany są dokonywane w kopii. Należy pamiętać, że

gdy szablon jest kopiowany, nie są kopiowane informacje dotyczące jego udostępniania.

Dla szablonów, motywów, składników, serwisów ani stron serwisów nie wolno używać następujących nazw: authsite, content, pages, scstemplate\_\*, \_comps, \_components, \_compsdelivery, \_idcservice, \_sitescloud, \_sitesclouddelivery, \_themes, \_themesdelivery. Następujących nazw można używać dla stron serwisów, ale nie można używać dla szablonów, motywów, składników ani serwisów: documents, sites.

Jeśli użytkownik chce utworzyć szablon niestandardowy (z serwisem i motywem), zaleca się użycie szablonu początkowego JET lub szablonu początkowego, udostępnionego przez usługę Oracle Content Management, zawierającego podstawowe elementy dla serwisu i powiązanego motywu. Szablon początkowy zawiera informacje i instrukcje zapisane na stronach serwisu, ułatwiające projektowanie serwisu i motywu w szablonie niestandardowym.

Zob. [Zarządzanie szablonami](#) i [Praca z szablonem początkowym](#).

## Eksportowanie szablonu

Można wyeksportować szablon i zmodyfikować go offline, a następnie zaimportować go jako nowy szablon albo jako zamiennik istniejącego szablonu. Można także wyeksportować szablon, aby przenieść go (zaimportować) do innej instancji Oracle Content Management.


Eksportując szablon, kopiuje się go do folderu usługi Oracle Content Management jako plik .zip. Pakiet szablonu można pobrać bezpośrednio z tego folderu, po czym rozpakować pakiet i pracować z poszczególnymi plikami. Po ukończeniu pracy z plikami szablonu można utworzyć plik .zip zawierający pakiet szablonu i zaimportować go do Oracle Content Management, zastępując szablon pierwotny lub tworząc nowy.



### Uwaga:

Gdy szablon jest eksportowany, nie są dołączane informacje dotyczące jego udostępniania.

Aby wyeksportować szablon, należy:

1. Z nawigacji bocznej usługi Oracle Content Management wybrać opcję **Programista**.  
Zostanie wyświetlona strona **Programista**.
2. Nacisnąć przycisk **Wyświetl wszystkie szablony**.  
Zostanie wyświetlona lista istniejących szablonów.
3. Wybrać szablon, po czym wybrać z menu podręcznego opcję **Eksportuj** albo kliknąć na ikonie  (na pasku czynności).
4. Przejść do folderu albo utworzyć nowy folder: nacisnąć przycisk **Utwórz**, podać nazwę i opcjonalny opis, po czym nacisnąć przycisk **Utwórz**.  
Aby otworzyć folder, kliknąć na jego ikonie lub nazwie.
5. Wybrać folder, zaznaczając pole wyboru obok powiązanego folderu, po czym nacisnąć przycisk **OK**.

W wybranym folderze zostanie utworzony plik pakietu szablonu mający nazwę szablonu i rozszerzenie .zip.

## Importowanie szablonu

Można wyeksportować szablon i zmodyfikować go offline, a następnie zaimportować go jako nowy szablon albo jako zamiennik istniejącego szablonu. Można także wyeksportować szablon, aby przenieść go (zaimportować) do innej instancji Oracle Content Management.

Eksportując szablon, kopiuje się go do folderu usługi Oracle Content Management jako plik `.zip`. Pakiet szablonu można pobrać bezpośrednio z tego folderu, po czym rozpakować pakiet i pracować z poszczególnymi plikami. Po ukończeniu pracy z plikami szablonu można utworzyć plik `.zip` zawierający pakiet szablonu i zaimportować go do Oracle Content Management, zastępując szablon pierwotny lub tworząc nowy.

Aby zaimportować pakiet szablonu:

1. Z nawigacji bocznej usługi Oracle Content Management wybrać opcję **Programista**  
Zostanie wyświetlona strona **Programista**.
2. Nacisnąć przycisk **Wyświetl wszystkie szablony**.  
Zostanie wyświetlona lista istniejących szablonów.
3. Nacisnąć przycisk **Utwórz**, po czym wybrać opcję **Importuj pakiet szablonu**.
4. Mając wysłany pakiet szablonu, przejść do folderu zawierającego ten pakiet. Aby otworzyć folder, kliknąć na jego ikonie lub nazwie.

Jeśli pakiet szablonu nie został jeszcze wysłany:

- a. Przejść do folderu, do którego ma zostać wysłany pakiet szablonu, albo utworzyć folder: nacisnąć przycisk **Nowy**, podać nazwę i opcjonalny opis, po czym nacisnąć przycisk **Utwórz**.
- b. Nacisnąć przycisk **Wyślij**.
- c. Odszukać i wybrać pakiet szablonu, po czym nacisnąć przycisk **Otwórz**.

Na pasku postępu będzie pokazywana nazwa pliku i status operacji wysyłania.

5. Wybrać pakiet szablonu, zaznaczając pole wyboru obok nazwy pliku, po czym nacisnąć przycisk **OK**.

Jeśli nie występują żadne konflikty między zawartością importowanego szablonu a istniejącymi szablonami, motywami lub składnikami niestandardowymi, zostaną w Oracle Content Management utworzone nowe foldery dla szablonu, jego motywu i wszelkich składników niestandardowych.

6. Jeśli okaże się, że nazwa lub ID szablonu, motywu lub składnika niestandardowego już istnieje, pojawi się wezwanie do rozwiązywania konfliktu.

W zależności od charakteru konfliktu będzie dostępna opcja utworzenia nowego szablonu, motywu lub składnika niestandardowego albo — w niektórych przypadkach — opcja zastąpienia istniejącego szablonu, motywu lub składnika niestandardowego importowaną wersją.

## Praca z szablonem początkowym

Do utworzenia serwisu, wraz z motywem i składnikami niestandardowymi, można użyć kopii szablonu początkowego (startera), udostępnionego przez Oracle Content Management.

Szablon zawiera w jednym pakiecie wszystkie elementy (domyślny serwis, układ, nawigacja, przykładowa zawartość, motyw, powiązane elementy zawartości itd.) potrzebne do konstruowania serwisu internetowego. Szablon początkowy, w przeciwieństwie do innych szablonów udostępnianych przez Oracle Content Management, zapewnia łatwą w użyciu podstawową strukturę tworzenia nowego serwisu, w tym nowego motywu.

### Uwaga:

Motyw jest zazwyczaj współdzielony przez szablony, lecz szablon początkowy korzysta z kopii motywu.

W Oracle Content Management szablony początkowe, szablon "StarterTemplate" i szablon "JETStarterTemplate" są udostępniane wraz z innymi szablonami, jeśli administrator podczas instalowania i konfigurowania usługi włączy szablony. Szablon początkowy Oracle JET (JavaScript Extension Toolkit) zawiera najnowszy style JET dla szablonów, przykładową zawartość strony (jako składniki JET) oraz składniki początkowe do tworzenia szablonów i serwisów opartych na technologii JET.

### Proces podstawowy

Podstawowy proces pracy z szablonem początkowym wygląda następująco:

1. Utworzyć nowy serwis, po czym wybrać szablon początkowy. Zostanie utworzony nowy motyw początkowy wraz z nowym serwisem.

### Uwaga:

Nazwę serwisu należy starannie wybierać. Jest ona także używana jako nazwa nowego motywu, który stanie się widoczny dla użytkowników, gdy serwis zostanie opublikowany. Po utworzeniu motywu nie można zmienić jego nazwy.

Serwis i motyw początkowy zawierają zbiór folderów i plików, które są wymagane do rozpoczęcia tworzenia serwisu i motywu.

2. Zsynchronizować nowy motyw z aplikacją typu Desktop. Motyw można rozszerzyć i dostosować przy użyciu układów i zasobów statycznych. Ponieważ jest to kopia motywu, używająca nazwy właściwej dla szablonu, można wprowadzić zmiany w motywie, nie wpływając na motyw źródłowy.
3. Otworzyć serwis w konstruktorze serwisów. Za pomocą konstruktora serwisów można dodać elementy do struktury serwisu, dodać składniki i interakcje do stron oraz dodać zawartość, która stanie się częścią domyślnego serwisu nowego szablonu. Można ponownie użyć lub usunąć zawartość serwisu, udostępnioną w szablonie początkowym.
4. Gdy serwis i motyw będą gotowe, zsynchronizować zmiany z instancją Oracle Content Management, po czym z wybranego serwisu utworzyć nowy szablon.



5. Udostępnić szablon, dzięki czemu inne osoby będą mogły z niego korzystać.

#### Tworzenie szablonu przy użyciu szablonu początkowego

Aby użyć szablonu początkowego, należy:

1. Z nawigacji bocznej Oracle Content Management wybrać opcję **Serwisy**.  
Zostanie wyświetlona lista istniejących serwisów.
2. Nacisnąć przycisk **Utwórz**.
3. W oknie dialogowym "Tworzenie serwisu" wybrać szablon początkowy, który będzie używany jako podstawa serwisu.  
Nowy serwis używa motywu udostępnionego z szablonem początkowym, ze zmienioną nazwą odpowiadającą nazwie nowego serwisu.
4. W oknie dialogowym wpisać nazwę serwisu. Nazwa ta będzie używana w adresie URL serwisu. Może się ona składać tylko z liter, cyfr, znaków podkreślenia ( \_ ) i łączników ( - ). Użyta spacja zostanie automatycznie zastąpiona znakiem podkreślenia.

Dla szablonów, motywów, składników, serwisów ani stron serwisów nie wolno używać następujących nazw: authsite, content, pages, scsTEMPLATE\_\*, \_comps, \_components, \_compsdelivery, \_idcSERVICE, \_sitescloud, \_sitesclouddelivery, \_themes, \_themesdelivery. Następujących nazw można używać dla stron serwisów, ale nie można używać dla szablonów, motywów, składników ani serwisów: documents, sites.

#### Uwaga:

W ścieżce adresu URL serwisu Oracle Content Management jest uwzględniana wielkość liter. Wielkość liter w zapytaniach lub wartościach napisowych jest zarządzana przez programistów za pomocą ich niestandardowego kodu.

```
https://[host[:port]][/]path[?query][#fragment]
      |_____|_____|
      Location  Data
```

5. Opcjonalnie: podać opis serwisu.
6. Po ukończeniu nacisnąć przycisk **Utwórz**.

Na pasku postępu będzie pokazywana nazwa nowego serwisu i status operacji tworzenia. Gdy serwis zostanie utworzony, jego nazwa pojawi się na liście serwisów. Jego początkowy status to "offline".

Aby szybko odnaleźć na liście nowo utworzony serwis, wystarczy posortować ją według kolumny **Ostatnia aktualizacja**. Utworzony serwis pojawi się na samym początku listy.

 **Uwaga:**

Bieżącemu użytkownikowi jest — w odniesieniu do utworzonego przez niego serwisu — automatycznie przypisywana rola "Menedżer".

7. Aby zsynchronizować motyw z komputerem stacjonarnym, użyć aplikacji typu Desktop. Powinny być widoczne pliki oraz struktura hierarchiczna folderów motywu.
8. Motyw początkowy zawiera minimalny zestaw folderów i plików, na przykład:


```
theme
  assets
    css
      main.css
    js
      topnav.js
  designs
    default
      design.css
      design.json
      facebook.png
      googleplus.png
      linkedin.png
      twitter.png
      youtube.png
      x-close.png
  layouts
    index.html
  resonancepages
    404.html
  _folder.json
  _folder_icon.jpg
  components.json
  viewports.json
```

Plik `x-close.png` zawiera domyślną ikonę "Zamknij" dla wyskakującego okna wyrażenia zgody na pliki cookie.

W folderze `/layouts` znajduje się plik układu strony (`index.html`) z następującą zawartością:

- Zbiór znaczników HTML, które umożliwiają użycie pliku jako układu strony.
- Gniazdo zawierające standardowy tekst z instrukcjami, na przykład jak zsynchronizować motyw z aplikacją typu Desktop, jak dodać nowy układ strony, jak dodać składniki do układu strony, jak skonstruować hierarchiczną strukturę serwisu przy użyciu nowego układu strony.
- Prosty plik nawigacyjny JavaScript, stanowiący przykład użycia funkcji i obiektów JavaScript z Renderer API. Render API jest potrzebne do przechodzenia przez hierarchię serwisu i generowania wymaganych znaczników HTML, umożliwiających nawigację w obrębie serwisu.

Zob. także [Podstawowa struktura motywu](#).

9. Po zmodyfikowaniu motywu zsynchronizować foldery i pliki motywu z instancją Oracle Content Management. Aby zobaczyć, jak zmiany w motywie będą wyglądały i działały w serwisie, otworzyć serwis w konstruktorze serwisów. Prawdopodobnie będzie to proces iteracyjny.
10. Aby wyświetlić lub zmodyfikować serwis, wybrać serwis, po czym wybrać z menu podręcznego opcję **Otwórz** albo kliknąć na ikonie  (na pasku czynności).
11. Przełączyć tryb edytora do trybu **edycji**, w którym można dokonywać zmian w serwisie. Można modyfikować już istniejące strony oraz dodawać nowe, używając układu strony dostępnego w motywach początkowych.

 **Uwaga:**

Rozumiejąc strukturę i zastosowanie układu strony, można usunąć zbędne sekcje udostępnione przez szablon początkowy oraz przełączyć układ pozostawianych sekcji na jeden z nowych układów.

12. Po wprowadzeniu zmian w serwisie, **zapisać** serwis, po czym nacisnąć — aby scalić aktualizację z serwisem podstawowym — przycisk **Publikuj**.
13. Wybrać serwis, po czym utworzyć na jego podstawie szablon. Są wówczas ściągane zasoby i motyw dla nowego szablonu.
14. Udostępnić szablon tym współużytkownikom, którzy powinni mieć możliwość używania tego szablonu.

## Tworzenie szablonu serwisu z szablonu "Bootstrap" lub "Website Design"

Otwarta architektura Oracle Content Management oznacza, że można korzystać z prac wykonanych na innych platformach kodowania, takich jak Foundation lub Bootstrap. Dokonując kilku zmian, można przekształcić szablon Bootstrap w motyw i uczynić go częścią szablonu Oracle Content Management.

### Proces podstawowy

Następujące wymagania i etapy są przedstawione poniżej w kolejnych podrozdziałach:

1. [Wymagania wstępne](#)
2. [Tworzenie serwisu](#)
3. [Synchronizowanie folderu motywu](#)
4. [Konfigurowanie motywu podstawowego](#)
5. [Aktualizowanie stron serwisu](#)
6. [Aktualizowanie nawigacji](#)
7. [Aktualizowanie układów serwisu](#)
8. [Publikowanie serwisu](#)
9. [Tworzenie nowego szablonu](#)

### Wymagania wstępne

- Dostęp do skonfigurowanej i działającej aplikacji Oracle Content Management typu Desktop, potrzebnej do synchronizacji folderów i plików z komputera lokalnego.
- Pobrane do komputera lokalnego i przygotowane do użycia foldery, pliki oraz zawartość motywów szablonów Bootstrap.

### Tworzenie serwisu

Aby utworzyć serwis z szablonu początkowego Oracle Content Management, należy:

1. Z nawigacji bocznej wybrać opcję **Serwisy**.  
Zostanie wyświetlona lista istniejących serwisów.
2. Nacisnąć przycisk **Utwórz**.
3. W oknie dialogowym "Tworzenie serwisu" wybrać **szablon początkowy JET** lub **szablon początkowy**, który będzie używany jako podstawa serwisu.
4. W oknie dialogowym wpisać nazwę serwisu. Nazwa ta będzie używana w adresie URL serwisu. Może się ona składać tylko z liter, cyfr, znaków podkreślenia ( \_ ) i łączników (-). Użyta spacja zostanie automatycznie zastąpiona łącznikiem.

Dla szablonów, motywów, składników, serwisów ani stron serwisów nie wolno używać następujących nazw: authsite, content, pages, scstemplate\_\*, \_comps, \_components, \_compsdelivery, \_idcservice, \_sitescloud, \_sitesclouddelivery, \_themes, \_themesdelivery. Następujących nazw można używać dla stron serwisów, ale nie można używać dla szablonów, motywów, składników ani serwisów: documents, sites.

#### Uwaga:

W ścieżce adresu URL serwisu Oracle Content Management jest uwzględniana wielkość liter. Wielkość liter w zapytaniach lub wartościach napisowych jest zarządzana przez programistów za pomocą ich własnego kodu.

```
https://host[:port]][/]path[?query][#fragment]
      \_____/ \_____/
      Location      Data
```

5. Opcjonalnie: podać opis serwisu.
6. Po ukończeniu nacisnąć przycisk **Utwórz**.

Na pasku postępu będzie pokazywana nazwa nowego serwisu i status operacji tworzenia. Gdy serwis zostanie utworzony, jego nazwa pojawi się na liście serwisów. Jego początkowy status to "offline".

Aby szybko odnaleźć na liście nowo utworzony serwis, wystarczy posortować ją według kolumny **Ostatnia aktualizacja**. Utworzony serwis pojawi się na samym początku listy.

### Synchronizowanie folderu motywu

Gdy jest tworzony serwis z szablonu początkowego, jest tworzona kopia motywu z szablonu początkowego mająca nazwę serwisu z następującą niej nazwą motywu. Na przykład motyw `My_New_Site` ma postać `My_New_SiteTheme`.

Do synchronizacji folderów i plików motywu ze swoim komputerem lokalnym należy użyć aplikacji typu Desktop. Zob. Synchronizacja — pierwsze kroki w podręczniku *Praca zespołowa nad do*.

W komputerze lokalnym powinna być widoczna następująca struktura hierarchiczna folderów i plików motywu z szablonu. Na przykład:

```

theme_name
  assets
    css
      main.css
    js
      topnav.js
  designs
    default
      design.css
      design.json
      facebook.png
      googleplus.png
      linkedin.png
      twitter.png
      youtube.pgn
  layouts
    index.html
  responsepages
    404.html
  _folder.json
  _folder_icon.png
  components.json
  viewport.json

```

### Konfigurowanie motywu podstawowego

1. Skopiować pliki szablonu Bootstrap do synchronizowanych na komputerze lokalnym folderów motywu w celu przykrycia istniejących plików.
  - Pliki `html` idą do folderu `nazwa_motywu/layouts`
  - Pliki `css` idą do folderu `nazwa_motywu/assets/css`
  - Pliki `js` idą do folderu `nazwa_motywu/assets/js`
  - Pliki obrazów idą do folderu `nazwa_motywu/assets/images`, który może być podzielony na podfoldery zawierające obrazy tła, stopki, osób itp.
2. Zmodyfikować pliki `html` w folderze `layout`, aktualizując ścieżki względne i dodając wymagane elementy. W przypadku typowego motywu Bootstrap istnieje wiele ścieżek względnych odnoszących się do folderu `/assets` i dlatego trzeba je zmodyfikować tak, aby prowadziły do folderu motywu.

Poprawić ścieżki folderów `css`, `js` i `images`, tak aby były używane:

```
_scs_theme_root_/assets/css/  
_scs_theme_root_/assets/js/  
_scs_theme_root_/assets/images/
```

 **Uwaga:**

Gdy ten etap zostanie ukończony, część `_scs_theme_root` będzie automatycznie dostosowana do środowiska, w którym ten motyw jest używany.

3. Dla każdego z układów Oracle Content Management istnieją trzy wymagania:

a. Zawrzeć następujące znaczniki w znaczniku `<head>` pliku `html`:

- `<!--$SCS_RENDER_INFO-->`
- `<!--$SCS_SITE_HEADER-->`
- `<!--$SCS_PAGE_HEADER-->`

b. Dołączyć skrypt "renderer" na końcu plików układu, w znaczniku `<body>`. Obie te ścieżki są automatycznie dostosowywane w konstruktorze serwisów oraz w środowiskach wykonawczych.

```
<script data-main="/_sitescloud/renderer/renderer.js" src="/  
_sitescloud/renderer/require.js"></script>
```

c. Po zawarciu pliku `renderer.js` dodać następujące znaczniki:

- `<!--$SCS_PAGE_FOOTER-->`
- `<!--$SCS_SITE_FOOTER-->`


Sprawdzić w Oracle Content Management, czy zmodyfikowane pliki motywu zostały zsynchronizowane z serwisem.

#### Aktualizowanie stron serwisu

Otwierając serwis w konstruktorze serwisów, początkowo widzimy strony z serwisu domyślnego. Można usunąć zbędne strony oraz przełączyć układ stron, które będą zachowane, do jednego z nowych układów.


1. Z nawigacji bocznej wybrać opcję **Serwisy**.




Zostanie wyświetlona lista istniejących serwisów.

2. Wybrać serwis, po czym wybrać z menu podręcznego opcję **Otwórz** albo kliknąć na ikonie  (na pasku czynności).

3. Podać nazwę aktualizacji i opcjonalny opis, po czym nacisnąć przycisk **Utwórz**.

W nazwie aktualizacji można używać liter, cyfr, znaków podkreślenia (`_`) i łączników (`-`). Użyta spacja zostanie automatycznie zastąpiona łącznikiem.

Jeśli już istnieją aktualizacje dla serwisu, wybrać odpowiednią aktualizację z listy, po czym kliknąć na ikonie .

4. Konstruktor serwisów zostanie otwarty w trybie podglądu. Aby można było dokonywać zmian lub używać opcji nawigacji z paska bocznego, upewnić się, że przełącznik  jest ustawiony w pozycji **Edytuj**.
5. Aby edytować określoną stronę, wybrać ją, korzystając z drzewa serwisu (na pasku bocznym) lub własnej nawigacji serwisu.
6. Aby usunąć niepotrzebną stronę, wybrać ją, po czym kliknąć na ikonie .
7. Aby dodać nową stronę, nacisnąć przycisk **Dodaj stronę**. Położenie tej strony w drzewie serwisu można zmienić, przeciągając ją w nowe miejsce.
8. Aby zmienić układ powiązany ze stroną, wybrać stronę w drzewie serwisu, po czym wyświetlić jej ustawienia, klikając na ikonie .  
Z listy **Układ strony** wybrać inny układ. Liczba i typy układów strony zależą od motywu powiązanego z serwisem.
9. Aby zmiany zostały zapisane w bieżącej aktualizacji, nacisnąć przycisk **Zapisz**. Można od razu kontynuować pracę nad bieżącą aktualizacją lub — jeśli trzeba — utworzyć nowe aktualizacje.

#### Aktualizowanie nawigacji

Jeśli przyjrzymy się nawigacji w podglądzie serwisu, to zobaczymy, że nie odpowiada ona obecnej strukturze hierarchicznej, ponieważ jest trwale zakodowana w układzie z motywu Bootstrap.

Należy zaktualizować nawigację, zastępując trwały kod w motywie Bootstrap kodem dynamicznie wygenerowanym ze struktury hierarchicznej serwisu.

W tym celu należy edytować lokalne synchronizowane kopie plików motywu.

1. Usunąć trwały kod nawigacji z układów. Poniżej jest przedstawiony przykład trwale zakodowanej nawigacji, którą należałoby usunąć z typowego motywu Bootstrap. Jest to typowa sekcja "header" z logo, z elementami przełączania (toggle) nawigacji dla menu nawigacyjnego (gdy strona jest za wąska (część reaktywna)) oraz z trwale zakodowaną nawigacją dla innych stron.

```
<header id="header" class="header navbar-fixed-top">
  <div class="container">
    <h1 class="logo">
      <a href="index.html"><span class="text">Velocity</
span></a>
    </h1><!--//logo-->
    <nav class="main-nav navbar-right" role="navigation">
      <div class="navbar-header">
        <button class="navbar-toggle" type="button"
data-toggle="collapse" data-target="#navbar-collapse">
          <span class="sr-only">Toggle navigation</
span>
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
        </button><!--//nav-toggle-->
      </div><!--//navbar-header-->
      <div id="navbar-collapse" class="navbar-collapse
collapse">
```

```

        <ul class="nav navbar-nav">
          <li class="active nav-item"><a
href="index.html">Home</a></li>
          <li class="nav-item"><a
href="features.html">Features</a></li>
          <li class="nav-item"><a
href="pricing.html">Pricing</a></li>
          <li class="nav-item dropdown">
            <a class="dropdown-toggle" data-
toggle="dropdown" data-hover="dropdown" data-delay="0" data-close-
others="false" href="#">Pages <i class="fa fa-angle-down"></i></a>
            <ul class="dropdown-menu">
              <li><a href="download.html">Download
Apps</a></li>
              <li><a href="blog.html">Blog</a></li>
              <li><a href="blog-single.html">Blog
Single</a></li>
              <li><a href="blog-category.html">Blog
Category</a></li>
              <li><a href="blog-archive.html">Blog
Archive</a></li>
              <li><a href="about.html">About Us</a></li>
              <li><a
href="contact.html">Contact</a></li>
            </ul>
          </li><!--//dropdown-->
          <li class="nav-item"><a href="login.html">Log
in</a></li>
          <li class="nav-item nav-item-cta last"><a
class="btn btn-cta btn-cta-secondary" href="signup.html">Sign Up
Free</a></li>
        </ul><!--//nav-->
      </div><!--//navabr-collapse-->
    </nav><!--//main-nav-->
  </div><!--//container-->
</header><!--//header-->

```

2. Napisać kod JavaScript w celu przejścia przez informacje o strukturze serwisu i wygenerowania kodu nawigacyjnego, po czym zawrzeć ten kod JavaScript w układach, na przykład:

```

<script type="text/javascript" src="_scs_theme_root_/assets/js/
navbar.js"></script>

```

3. Zmodyfikować wynik działania pliku `topnav.js` tak, aby odpowiadał kodowi znacznikowemu, oczekiwanemu w konkretnym arkuszu stylów CSS.

#### Aktualizowanie układów serwisu

Serwis jest już funkcjonalny, ale w układach nie ma jeszcze edytowalnych obszarów (gniazd).

1. Odszukać lub dodać w układzie element DIV i wyznaczyć go na gniazdo.



**Gniazda** są to występujące w układzie elementy DIV, mające w atrybucie "class" wartość "scs-slot". Każde gniazdo musi mieć unikatowy atrybut id. Na przykład:

```
<div id="slot-content1" class="scs-slot scs-responsive"></div>
```

Aby gniazdo automatycznie się dostosowywało do portu widoku przeglądarki, należy dodać atrybut "scs-responsive" klasy.

2. Powtórzyć te czynności w celu utworzenia dodatkowych gniazd w tym układzie lub w innych układach.

#### Publikowanie serwisu

Po wprowadzeniu wszystkich zmian w serwisie, a następnie zapisaniu ich, trzeba **opublikować** serwis.

Gdy serwis jest publikowany, wszystkie zmiany z bieżącej aktualizacji są scalane z serwisem podstawowym i stają się zawartością nowego serwisu podstawowego.

#### Tworzenie nowego szablonu

Teraz, mając już funkcjonalny serwis z powiązaniem motywem, można z tego serwisu utworzyć szablon, który następnie można udostępnić innym użytkownikom, aby mogli tworzyć z niego własne serwisy.

1. Z nawigacji bocznej usługi Oracle Content Management wybrać opcję **Programista**.
2. Nacisnąć przycisk **Wyświetl wszystkie szablony**.
3. Nacisnąć przycisk **Utwórz**, po czym wybrać opcję **Z istniejącego serwisu**.
4. Wybrać nowy serwis, który został utworzony z szablonu początkowego i szablonu Bootstrap.
5. Podać nazwę nowego szablonu, po czym nacisnąć przycisk **Utwórz**.
6. Aby spakować szablon do użycia z innymi instancjami Oracle Content Management, wybrać go, a następnie wybrać opcję **Eksportuj** w celu utworzenia pliku .zip, który będzie można pobierać.

## Tworzenie szablonów za pomocą usługi Developer Cloud Service

Za pomocą usługi Developer Cloud Service można tworzyć szablony dla usługi Oracle Content Management.

Aby utworzyć szablon w usłudze Developer Cloud Service, następnie przetestować go lokalnie, po czym wyeksportować do usługi Oracle Content Management, należy wykonać następujące zadania:

1. [Konfigurowanie zestawu OCE Toolkit na komputerze lokalnym.](#)
2. [Logowanie do konsoli Developer Cloud Service dla Oracle Content Management.](#)
3. [Tworzenie projektu w usłudze Developer Cloud Service.](#)
4. [Dodawanie zestawu OCE Toolkit do kodu projektu w nowym repozytorium Git.](#)

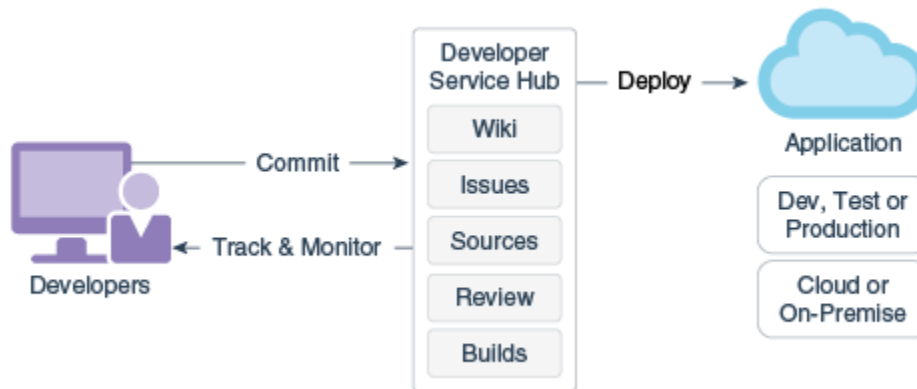
5. Tworzenie szablonu w usłudze Developer Cloud Service.  
Można utworzyć nowy szablon, [kopiując istniejący szablon w usłudze Developer Cloud Service](#) lub [importując szablon z usługi Oracle Content Management](#).
6. Testowanie szablonu przy użyciu lokalnego zestawu testów.
7. Scalanie zmian.
8. Eksportowanie szablonu z usługi Developer Cloud Service do Oracle Content Management.

## Logowanie do konsoli Developer Cloud Service dla Oracle Content Management

Korzystając z konsoli Developer Cloud Service, można rozpocząć tworzenie składników niestandardowych dla usługi Oracle Content Management.

Będąc administratorem usług Oracle Cloud, można — za pomocą interfejsu administracyjnego aplikacji "Moje usługi" — tworzyć usługi Cloud i zarządzać nimi. Będąc administratorem instancji usługi Oracle Content Management i administratorem usługi Standard Developer Service, można te usługi skonfigurować, a następnie ich używać:

1. Zalogować się do Oracle Cloud, używając informacji udostępnionych dla konta.
2. Aby utworzyć instancje usług Oracle Content Management i Standard Developer Service oraz nimi zarządzać, zalogować się do interfejsu administracyjnego aplikacji "Moje usługi".



3. Sprawdzić e-mail powiązany z usługą Oracle Developer Cloud Service.
4. Skonfigurować instancję Oracle Content Management, używając szczegółów subskrypcji usługi, po czym przejść pod adres URL instancji usługi Oracle Content Management.
5. Przejść pod adres URL usługi Standard Developer Service.
6. Zalogować się do konta Oracle Developer Cloud Service.

Uzyskać dostęp do adresu URL usługi Developer Cloud Service, po czym zalogować się do konsoli.

## Tworzenie projektu w usłudze Developer Cloud Service

Używając szablonu projektu "Content Experience Cloud", można utworzyć projekt w usłudze Developer Cloud Service. Można także utworzyć projekt z pustym repozytorium Git, a następnie zaimportować zestaw OCE Toolkit z instancji Oracle Content Management.

- [Tworzenie projektów Developer Cloud Service za pomocą szablonu usługi Oracle Content Management](#)
- [Tworzenie projektu w usłudze Developer Cloud Service za pomocą zestawu OCE Toolkit pobranego z Oracle Content Management](#)
- [Dodawanie zestawu OCE Toolkit do kodu projektu w nowym repozytorium Git](#)

## Tworzenie szablonów w usłudze Developer Cloud Service

Do tworzenia szablonów Oracle Content Management z dostępnych szablonów źródłowych można użyć polecenia `cec`, opartego na wierszu polecenia.

Aby utworzyć szablon Oracle Content and Experience z jednego z dostępnych szablonów źródłowych, należy użyć polecenia `cec create-template`. Wprowadzając w wierszu polecenia polecenie `cec create-template -h`, uzyskuje się wykaz dostępnych szablonów źródłowych.

Przykład tworzenia szablonu:

```
cec create-template CafeSupremoLite_yourname -f CafeSupremoLite
```

**Windows:** To polecenie tworzy dowiązanie symboliczne dla motywów renderowanych w zewnętrznym edytorze HTML WYSIWYG (np. `_scs_theme_root_`), gdy użytkownik tworzy szablony. Aby można było w systemie Windows utworzyć dowiązanie symboliczne, zazwyczaj trzeba uruchomić wiersz polecenia z uprawnieniami administratora. Jeśli do edytowania motywu nie jest używany edytor WYSIWYG, uprawnienia administratora nie są konieczne i można zignorować komunikat o błędzie tworzenia dowiązania symbolicznego.

W powyższym przykładzie jest tworzony szablon `CafeSupremoLite_yournameTheme`, a kod źródłowy jest udostępniany w katalogu `cec-components/src/main/`. W poniższej tabeli są pokazane lokalizacje kodu źródłowego, gdy szablon serwisu zostanie utworzony w usłudze Developer Cloud Service.

Kod źródłowy szablonu	Motyw	Składniki szablonu
<code>cec-components/src/main/templates</code>	<code>cec-components/src/main/themes</code>	<code>cec-components/src/main/components</code>

Lokalny zestaw testów także pokazuje składniki, z możliwością filtrowania ich na podstawie szablonu i typu.

Pliki motywów i składników można edytować za pomocą dowolnego edytora tekstu lub edytora kodu. Zob. [Testowanie przy użyciu lokalnego zestawu testów](#). Po przeprowadzonej edycji motywu lub składnika należy, aby zobaczyć dokonane zmiany, odświeżyć przeglądarkę.

**! Ważne:**

Kod źródłowy szablonów, motywów i składników jest przechowywany w katalogu `src/main/`. Nie należy modyfikować żadnych plików spoza folderu `src/main`, ponieważ zapewniają one poprawne funkcjonowanie lokalnego serwera Oracle Content Management.

## Kopiowanie szablonu w usłudze Developer Cloud Service

W usłudze Developer Cloud Service można skopiować już istniejący szablon Oracle Content Management.

Aby skopiować już istniejące szablony z folderu `src/main/templates`, należy użyć polecenia `cec copy-template`. Jeśli szablon zawiera zasoby z innych repozytoriów, można opcjonalnie podać odwzorowanie repozytorium; w przeciwnym razie zasoby te nie zostaną skopiowane.

W poniższym przykładzie jest kopiowany szablon `Temp1` do nowego szablonu o nazwie `Temp2`:

```
cec copy-template Temp1 -n Temp2
```

## Importowanie szablonu do usługi Developer Cloud Service

Szablony można importować z Oracle Content Management do Developer Cloud Service w celu dalszego ich rozwijania.

Mając utworzony plik zip szablonu z serwera Oracle Content Management, można zaimportować ten plik do usługi Developer Cloud Service w celu dalszego opracowywania szablonu (na przykład edytowania motywu lub składników). Należy użyć następującego polecenia:

```
cec import-template <location of the template zip file>
```

Należy określić folder w usłudze Oracle Content Management, w którym znajduje się plik zip.

Zob. [Szablony — informacje podstawowe](#) i [Eksportowanie szablonu](#).

## Scalanie zmian

Po utworzeniu składnika, szablonu lub układu zawartości bądź edycji kodu źródłowego na komputerze, należy w repozytorium Git projektu scalić nowe oraz zmienione składniki i szablony.

Aby scalić zmiany w repozytorium Git, należy w oknie terminala wprowadzić w podanej kolejności poniższe polecenia.

```
cd cec-components git pull
git add .
git status
git commit -a -m "komentarze" git pull
git push
```

## Eksportowanie szablonu z usługi Developer Cloud Service

Plik zip szablonu można wyeksportować z usługi Developer Cloud Service, a następnie użyć tego pliku do utworzenia serwisu w Oracle Content Management.

Gdy tworzenie szablonu zostanie ukończone, można wyeksportować szablon, używając poniższego polecenia. W odpowiedzi na polecenie jest wskazywane miejsce w Oracle Content Management, w którym został utworzony plik zip szablonu.

```
cec export-template CafeSupremoLite_yourname
```

Zob. [Szablony — informacje podstawowe](#) i [Importowanie szablonu](#).

## Tworzenie motywów

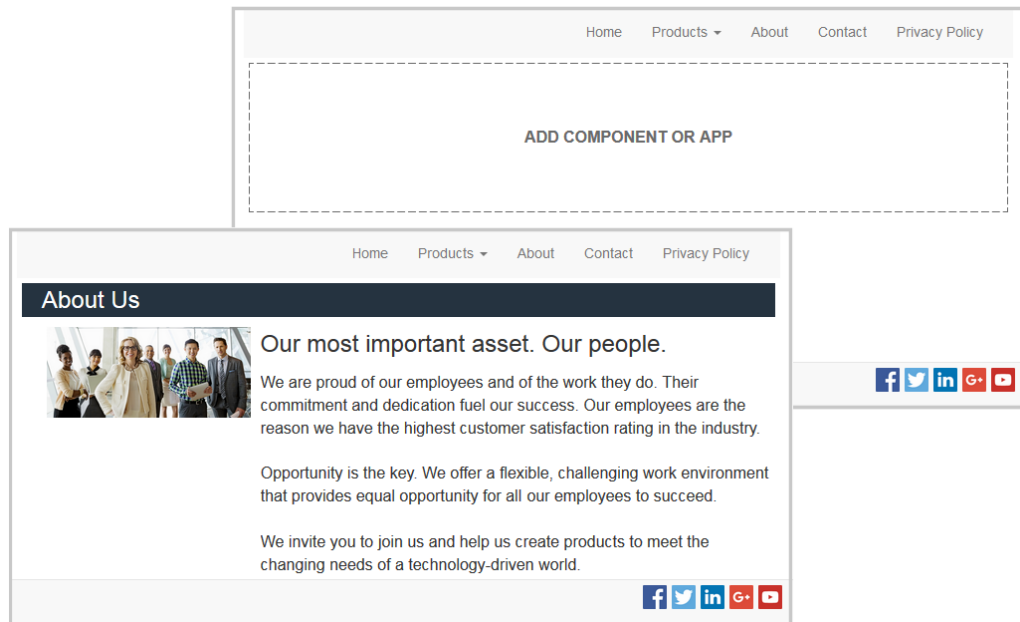
Motyw definiuje ogólny wygląd i styl serwisu, w tym schemat kolorów, rozmiar czcionek, typ czcionek i tła stron. Ponadto motywy zapewniają wizualną spójność stron serwisu. Można utworzyć unikatowe motywy i warianty motywów, określając projekt i przykładową zawartość, które następnie mogą być używane do tworzenia serwisów promujących markę i wizję firmy.

- [Motywy — informacje podstawowe](#)
- [Podstawowa struktura motywu](#)
- [Nawigacja serwisu](#)
- [Tworzenie motywu](#)
- [Wiązanie składników z motywami](#)
- [Sites Rendering API](#)

### Motywy — informacje podstawowe

Motyw definiuje ogólny wygląd i styl serwisu, w tym zawartość, wygląd i sposób funkcjonowania. Ponadto motywy zapewniają wizualną spójność stron serwisu.

Projektowanie nowego motywu oznacza określenie układu, stylu, przykładowej zawartości, nawigacji i wszystkich podstawowych informacji, które służą jako punkt początkowy dla nowego serwisu. Projektanci motywów określają, w jaki sposób serwis będzie wyglądał i działał. Motyw powinien być projektowany z myślą o sposobie używania go; na przykład większość użytkowników będzie uzyskiwała dostęp do serwisu za pomocą urządzenia mobilnego. Projektowanie motywów niestandardowych jest przydatne, jeśli jest grupa użytkowników, która będzie tworzyła wiele podobnych serwisów. Można zaprojektować motyw, używając układów strony dla wspólnych wzorców, które będą współużytkowane przez poszczególne motywy.



Motyw zawiera układy strony, które są używane do projektowania zawartości, wyglądu i sposobu funkcjonowania serwisów. Możemy zmienić projekt i ustawienia oraz dodać zawartość, tworząc serwis prezentujący nasz styl, naszą markę i naszą wizję.

Motyw zawiera:

- Zasoby dla obrazów tła lub inną zawartość, stanowiącą część układów strony (na przykład obrazu lub pliki JavaScript)
- Ustawienia stylu dla serwisu (arkusze CSS)
- Kilka układów strony (pliki HTML)
- Kod tworzący nawigację po serwisie (pliki JavaScript)
- Listę podstawowych stylów, które mogą zostać użyte dla składników (określona jest w plikach `design.css` i `design.json`)

Motyw może także zawierać *dane wypełniające*, które są używane do wypełniania nowych stron utworzonych z jednego z szablonów strony. Na przykład użytkownik tworzy nową stronę sekcji "Produkty" i wybiera układy strony o nazwie `new_product.html`. Jeśli motyw zawiera plik o nazwie `new_product-pageseed.json`, nowa strona zostanie — podczas jej pierwszego tworzenia — wypełniona zawartością pliku wypełniania stron. Podobnie jak w przypadku zawartości przykładowej, dane wypełniania mogą być modyfikowane i są przechowywane w motywie tylko po to, aby zapewnić punkt startowy do tworzenia stron.

Można utworzyć motyw, w którym jest używany podzbiór składników, przeznaczony do pracy z tym motywem. Gdy użytkownik wybierze motyw dla serwisu, zobaczy tylko te składniki, które są właściwe dla tego motywu. Zob. [Wiązanie składników z motywami](#).

Każdy serwis internetowy korzysta z motywu. Gdy serwis jest tworzony z szablonu, z szablonu tego jest dziedziczony motyw. Używany dla serwisu motyw można w dowolnej chwili zmienić. Oracle Content Management udostępnia różne, gotowe do użycia szablony, zawierające motywy.

Jeśli serwis używa nowego, nieopublikowanego motywu, to zostanie on automatycznie opublikowany przy pierwszym przełączeniu serwisu do trybu online. Jeśli dokonamy

zmian w motywie i zechcemy zaktualizować serwisy online, tak aby te zmiany zostały uwzględnione, musimy motyw opublikować jawnie. Może to zrobić tylko właściciel motywu lub użytkownik z uprawnieniami menedżera.

 **Uwaga:**

Gdy motyw zostanie opublikowany, dokonane zmiany zostaną odzwierciedlone we wszystkich korzystających z niego serwisach online. Na przykład, jeśli w motywie zmienimy czcionkę domyślną, a następnie ten motyw opublikujemy, we wszystkich serwisach korzystających z tego motywu będzie używana nowa czcionka domyślna.

Zob. także [Zarządzanie motywami](#).

## Podstawowa struktura motywu

Podstawowa struktura motywu zawiera projekt, nawigację i style określone w folderach przechowywanych w Oracle Content Management. Motyw jest częścią szablonu serwisu.

Gdy użytkownik wybierze szablon do utworzenia nowego serwisu, dane powiązanego motywu zostaną automatycznie załadowane. Jeśli jest używany szablon początkowy JET lub szablon początkowy, motyw jest automatycznie kopiowany (nie jest używane odwołanie do niego). Jeśli w pracach programistycznych jest używany szablon początkowy, programista potrzebuje własnej kopii motywu.

Motyw jest uporządkowany według określonej struktury folderów i plików, tak jak zostało to przedstawione w poniższym przykładzie, pokazującym podstawowe foldery i pliki:

```
theme
  assets
    css
      main.css
    js
      topnav.js
  designs
    default
      design.css
      design.json
  layouts
  publish
  responsepages
    404.html
  viewport.json
  _folder.json
  _folder_icon.png
  components.json
```

Niektóre foldery zawierają określone typy informacji:

- **assets:** Skrypty JavaScript, arkusze CSS (Cascading Style Sheet), obrazy i inne pliki pomocnicze, do których odwołują się układy.
- **designs:** Pliki `design.css` i `design.json`, które są używane do określania opcji stylów dla składników.



- `layouts`: Pliki HTML szablonów strony, które służą do wyświetlania stron serwisów.

 **Uwaga:**

Wszystkie pliki HTML muszą zawierać na początku pliku element DOCTYPE w następującej postaci: `<!DOCTYPE html>`

- `publish`: Gdy motyw zostanie użyty w opublikowanym serwisie, katalog ten będzie uwzględniany na listach i będzie zawierał kopie plików. Jeśli motyw został zsynchronizowany przy użyciu aplikacji typu Desktop, katalog ten jest widoczny, lecz nie jest uwzględniany, jeśli motyw wyeksportowano jako część szablonu.
- `responsepages`: Specjalna strona do obsługi błędów (404).

 **Uwaga:**

Jeśli strona w serwisie jest oznaczona jako strona błędu, zostanie zignorowany komunikat o błędzie 404 z motywu i zostanie użyta wyznaczona strona błędu.

- `viewport.json`: Określa ustawienia portu widoku motywu.
- `_folder.json`: Określa nazwę i GUID motywu. Na przykład:

```
{
  "themeName": "MarketingCampaignTheme",
  "itemGUID": "TB79D65F699B022AC4E11F4D4EE870070A1ADD86BBBB"
}
```

GUID jest tworzony przez Oracle Content Management, kiedy motyw jest importowany po raz pierwszy lub gdy jest kopiowany. Nazwa motywu jest przypisywana przez twórcę motywu podczas jego tworzenia.

- `components.json`: (Przestarzałe.) Zapisuje składniki niestandardowe, które są używane w obrębie motywu. Plik ten jest utrzymywany wyłącznie w celu zapewnienia zgodności wstecznej.

Istnieją dwa kluczowe pliki, z którymi twórca pracuje podczas tworzenia nowego motywu. Za pomocą tych plików ustawia się style składników:

- `design.css`
- `design.json`

Dobłą praktyką jest zawieranie informacji nawigacyjnych w jednym pliku JavaScript; na przykład w pliku o nazwie `nav.js`. Folder `/assets/js/` motywu jest dobrym miejscem do przechowywania tego typu pliku.

## Nawigacja serwisu

Hierarchiczna struktura serwisu jest przechowywana w pliku `structure.json` powiązonym z serwisem. JSCS.`structureMap`.

Narzędzie konstruktor serwisów odczytuje plik `structure.json` w celu narysowania drzewa serwisu. Kod stron serwisu będzie zawarty w pliku `structure.json`. Na przykład:

```
"pages": [ {
  "id": 100,
  "name": "Home",
  "parentId": null,
  "pageUrl": "index.html",
  "hideInNavigation": false,
  "linkUrl": "",
  "linkTarget": "",
  "children": [ 200,
                300,
                400,
                500 ],
  "overrideUrl":false
}

{
  "id":200,
  "name":"Products"
  "parentId":100,
  "hideInNavigation":false,
  "LinkUrl":"",
  "linkTarget":"",
  "children":[ 204, 205],
  "overrideUrl":false
}

{
  "id":204,
  "name":"Hiking Boots",
  "parentId":200,
  "pageUrl":"products/hiking_boots.html",
  "hideInNavigation":false,
  "linkUrl":"",
  "linkTarger":"",
  "children":[],
  "overrideUrl":false
}
```

Kod JavaScript nawigacji jest niezbędny w ramach stron serwisu także do odczytania struktury serwisu i wyznaczenia odpowiednich łączy. W szablonach, dostarczanych z Oracle Content Management, są zawarte przykładowe pliki JavaScript nawigacji pokazujące, jak to działa.

Plik `topnav.js`, używany w niektórych motywach dostarczanych z Oracle Content Management

, stanowi przykład prezentujący, w jaki sposób można używać obiektu `SCS.structureMap` wraz z wywołania `SCSRenderAPI.getPageLinkData`) do przechodzenia po strukturze serwisu i wyznaczania

kodu HTML, niezbędnego do renderowania — na stronie — menu nawigacji. Poniżej jest pokazany kod z przykładowego pliku `topnav.js`:

```
function renderNode(id, navBar)
{
    if (id >= 0)
    {
        var navNode = SCS.structureMap[id];
        if( navNode &&
            (
                ( typeof navNode.hideInNavigation != "boolean" ) ||
                ( navNode.hideInNavigation === false )
            ) )
        {
            var navItem = document.createElement("li");
            var navLink = document.createElement("a");
            var navText = document.createTextNode(navNode.name);

            var linkData = SCSRenderAPI.getPageLinkData(navNode.id) ||
        };

            if( linkData.href ) {
                navLink.href = linkData.href;
            }
            if( linkData.target ) {
                navLink.target = linkData.target;
            }

            navLink.appendChild(navText);
            navItem.appendChild(navLink);

            if (navNode.children.length > 0)
            {
                var navSub = document.createElement("ul");

                for (var c = 0; c < navNode.children.length; c++)
                {
                    renderNode(navNode.children[c], navSub);
                }

                navItem.appendChild(navSub);
            }
            navBar.appendChild(navItem);
        }
    }
}

function renderNav()
{
    var topnav = document.getElementById("topnav");           // expected
    to be an empty <div>

    if (topnav)
    {
        var navBar = document.createElement("ul");
```

```
        renderNode(SCS.navigationRoot, navBar);

        topnav.appendChild(navBar);
    }
}

// Must wait for all our script to be ready...
if (document.addEventListener)
{
    document.addEventListener('scsrenderstart', renderNav, false);
}
else if (document.attachEvent)
{
    document.documentElement.scsrenderstart = 0;
    document.documentElement.attachEvent("onpropertychange",
        function(event)
        {
            if (event && (event.propertyName == "scsrenderstart"))
            {
                renderNav();
            }
        }
    );
}
```

Korzystając z Render API, można generować łącza nawigacyjne, które będą funkcjonować w trybach edycji i podglądu serwisu, a także w opublikowanym serwisie online. Zob. [Render API — opis](#).

Dobłą praktyką jest umieszczanie informacji nawigacyjnych w jednym pliku JavaScript, takim jak `topnav.js`. Plik JavaScript zazwyczaj jest przechowywany w katalogu `/assets/js/` motywu, jak pokazano w przykładowych motywach dostarczanych z Oracle Content Management.

## Tworzenie motywu

Można utworzyć nowy motyw, kopiując istniejący motyw i dokonując zmian w kopii. Można także zaimportować zawartość Bootstrap i użyć jej w nowym motywie.

### Uwaga:

Bez względu na to, czy jest tworzony nowy motyw, czy są dokonywane aktualizacje istniejącego motywu, zawsze należy sporządzić kopię motywu i nad nią pracować. Następnie — aby się upewnić, że poprawnie działa — należy ją przetestować, używając przykładowego serwisu lub kopii swojego rzeczywistego serwisu. Należy pamiętać, że zmiany, dokonane w istniejącym motywie, zostaną zaimplementowane we wszystkich używających go serwisach, gdy tylko ten zmodyfikowany motyw zostanie opublikowany.


### Kopiowanie motywu

1. Na stronie startowej wybrać opcję **Programista**.

Zostanie wyświetlona strona **Programista**.

2. Nacisnąć przycisk **Wyświetl wszystkie motywy**.

Zostanie wyświetlona lista istniejących motywów. Można określić, jak motywy będą wyświetlane, klikając na ikonie "Widok", po czym wybierając odpowiednią opcję z listy.

3. Wybrać motyw, po czym wybrać z menu podręcznego opcję **Kopiuj** albo kliknąć na ikonie  (na pasku czynności).

Wszystkie foldery i pliki motywu są kopiowane, w tym wszystkie przykładowe strony i ich zawartość.

 **Uwaga:**

Gdy motyw jest kopiowany, nie są kopiowane informacje dotyczące jego udostępniania.

4. Należy podać nazwę kopii. Nie może to być nazwa już używana dla innego motywu.

W nazwie można używać liter, cyfr, znaków podkreślenia ( ) i łączników (-). Użyta spacja zostanie automatycznie zastąpiona łącznikiem.

Dla szablonów, motywów, składników, serwisów ani stron serwisów nie wolno używać następujących nazw: `authsite`, `content`, `pages`, `scstemplate_*`, `_comps`, `_components`, `_compsdelivery`, `_idcservice`, `_sitescloud`, `_sitesclouddelivery`, `_themes`, `_themesdelivery`. Następujących nazw można używać dla stron serwisów, ale nie można używać dla szablonów, motywów, składników ani serwisów: `documents`, `sites`.

5. Opcjonalnie: podać opis motywu.
6. Nacisnąć przycisk **Kopiuj**.

Na pasku postępu są pokazywane nowa nazwa motywu i status operacji kopiowania. Gdy motyw zostanie skopiowany, jego nazwa pojawi się na liście motywów. Foldery i pliki, wchodzące w skład motywu, można eksplorować po kliknięciu na nazwie motywu wyświetlanej na liście motywów.

7. Do synchronizacji folderów i plików motywu z systemem lokalnym należy użyć aplikacji Oracle Content Management typu Desktop. Dzięki temu można przechodzić do folderów lokalnych i pracować bezpośrednio nad plikami. Zmiany dokonywane w motywie są automatycznie synchronizowane. Zmian można dokonywać, używając swoich ulubionych narzędzi do edytowania kodu HTML, tekstu czy kodu programistycznego.

#### Określenie, czy style niestandardowe mogą być używane w serwisie

Administrator motywów może określać style, które są dostępne wraz z motywem, lub je dostosowywać.

Ustawienia w pliku `components.json` motywu określają, czy style niestandardowe można wprowadzać w konstruktorze serwisów. Dzięki temu osoba współtworząca serwis zachowuje styl w obrębie serwisu i nie używa — na przykład — innych czcionek lub kolorów.

Aby nie zezwolić na używanie niestandardowych stylów w serwisie, należy:

1. Ukryć w panelu **Ustawienia** dla składników opcję **Dostosuj**.  
W pliku `components.json` motywu dodać następujący obiekt wraz z innymi definicjami składników:

```
[
  {
    "showStyleClassOnly": "true"
  }
]
```

2. Dostosować grupy i przyciski z pasków narzędzi edytora CKEditor, uniemożliwiając w ten sposób stylizację niestandardową:

W pliku `components.json`

motywu można określić właściwości `toolbarGroups` i `removeButtons` edytora CKEditor w celu dostosowania grup i zawartych w nich przycisków dla składników "tytuł" (`title`) i akapit (`paragraph`).

Na przykład można dodać następujące obiekty wraz z definicjami innych składników, aby usunąć style, czcionki i kolory z grupy `Styles`:

```
[
  {
    "name": "",
    "list": [
      {
        "type": "scs-title",
        "id": "scs-title",
        "config": {
          "toolbarGroups": [
            {
              "name": "basicstyles",
              "groups": ["basicstyles"]
            },
            {
              "name": "styles",
              "groups": ["styles"]
            },
            {
              "name": "colors",
              "groups": ["colors"]
            },
            "/",
            {
              "name": "undo",
              "groups": ["undo"]
            },
            {
              "name": "links",
              "groups": ["links"]
            },
            {
              "name": "paragraph",
              "groups": ["list", "indent"]
            },
            {

```

```

        "name": "align",
        "groups": ["align"]
    },
    {
        "name": "cleanup",
        "groups": ["cleanup"]
    }
],
"removeButtons":
"Styles,Subscript,Superscript,Strike,Anchor,Blockquote,Link,Unlink,Font,TextColor,BGColor"
}
},
{
    "type": "scs-paragraph",
    "id": "scs-paragraph",
    "config": {
        "toolbarGroups": [
            {
                "name": "basicstyles",
                "groups": ["basicstyles"]
            },
            {
                "name": "styles",
                "groups": ["styles"]
            },
            {
                "name": "colors",
                "groups": ["colors"]
            },
            "/",
            {
                "name": "undo",
                "groups": ["undo"]
            },
            {
                "name": "links",
                "groups": ["links"]
            },
            {
                "name": "paragraph",
                "groups": ["list", "indent"]
            },
            {
                "name": "align",
                "groups": ["align"]
            },
            {
                "name": "insert",
                "groups": ["image", "table"]
            },
            {
                "name": "cleanup",
                "groups": ["cleanup"]
            }
        ]
    }
}

```

```
    ],  
    "removeButtons":  
    "Styles,Subscript,Superscript,Strike,Anchor,Blockquote,Link,Unlink,Font,TextColor,BGColor"  
  }  
]  
}
```

3. Zapobiec wklejaniu stylizacji do edytora CKEditor. Podczas wklejania jest zachowywana zgodność ze skonfigurowanymi stylami i przyciskami. W poprzednim przykładzie jest z wklejanych danych usuwane ustawienie czcionki `Font`.

Niezwolenie na style niestandardowe powoduje, że:

- We wszystkich wbudowanych składnikach jest na karcie **Styl** ukrywania opcja "Dostosuj".
- W edytorze CKEditor przyciski z paska narzędzi, służące do ustawiania stylów, są ukrywane, a specyfikacja rodziny czcionek, koloru czcionek itd. jest przesłaniania.
- W edytorze CKEditor nie jest możliwe ustawianie stylu przez wklejony tekst. Dozwolone jest stosowanie kursywy i pogrubienia.

#### Ukrywanie składników i układów sekcji motywu

Aby uniemożliwić twórcom serwisów używanie określonych składników i układów sekcji, można je ukryć, edytując odpowiednio plik `component.json` motywu. Niekiedy trzeba to zrobić w celu zapewnienia spójnego wyglądu. Zob. [Ukrywanie składników i układów sekcji motywu](#).

#### Weryfikowanie motywu z użyciem serwisu

Po ukończeniu edytowania serwisu trzeba sprawdzić, czy wszystkie jego elementy współpracują z nim zgodnie z oczekiwaniami.

1. Upewnić się, że skopiowane pliki i foldery zostały zsynchronizowane przy użyciu aplikacji Oracle Content Management typu Desktop oraz że wszystkie modyfikacje zostały zapisane.
2. Otworzyć w konstruktorze serwisów serwis (może to być serwis testowy lub istniejący serwis działający w trybie offline) i przełączyć go do używania nowego motywu.
3. Przetestować serwis z nowym motywem, dodając strony, korzystając z ustawień, wyświetlając obrazy, sprawdzając nawigację i wszystko inne, co zostało w tym motywie zmienione.
4. Sprawdzić, czy wszystko jest poprawnie renderowane, w tym wszystkie pliki i łącza.
5. Opublikować motyw, aby występujące w nim zmiany zostały zaimplementowane, po czym sprawdzić go w środowisku wykonawczym.

Zob. [Publikowanie motywów](#).

#### Korzystanie z motywu Bootstrap

Podobieństwo między motywami Oracle Content Management a motywami Bootstrap powoduje, że możliwe jest przekształcenie istniejących stron i zawartości z motywem Bootstrap do postaci pozwalającej ich użyć w motywie Oracle Content Management.



Bootstrap jest to bezpłatna kolekcja narzędzi typu "open-source", służących do tworzenia serwisów i aplikacji internetowych. Zawiera szablony projektowe oparte na standardzie HTML i CSS, przeznaczone dla składników interfejsu, oraz rozszerzenia JavaScript.

Tworząc motyw, postępujemy zgodnie z dotychczasowymi instrukcjami, lecz — w celu zaimportowania i edytowania stron oraz zawartości z motywu Bootstrap — pracujemy z kodem (używając dowolnego edytora).

Jest mało prawdopodobne, że zostanie użyty cały motyw Bootstrap ze wszystkimi jego stronami i całą zawartością. Zazwyczaj wybiera się niektóre części motywu do użycia w motywie Oracle Content Management, na przykład kilka szablonów stron i ich niektóre części. Na przykład motyw Bootstrap zawiera w szablonach kilka bloków, takich jak nagłówek, blok nawigacji, blok treści i blok stopki. Ze względu na sposób dostarczania motywów Bootstrap jako kolekcji plików, które można wyświetlać bezpośrednio z systemu plików (bez serwera internetowego), w obrębie jednej strony występuje wiele duplikatów (wynika to z tego, że nie jest używany żaden motor składania stron). Jeśli informacje te mają zostać w motywie Oracle Content Management, trzeba wydobyć te części i dodać je do szablonów stron Oracle Content Management, a następnie użyć Oracle Content Management do dynamicznego zestawienia stron z tych szablonów.


Przykładem dodania informacji do szablonów wielokrotnego użytku jest sekcja nawigacji. Nawigacja z typowego motywu Bootstrap jest powielana na wszystkich stronach, lecz — przechodząc do motywu Oracle Content Management — trzeba użyć kodu JavaScript do dynamicznego przejścia po hierarchicznej strukturze serwisu i wygenerowania struktur nawigacyjnych. Następnie skrypt ten należy uwzględnić na wszystkich stronach a uzyskają one wszystkie odpowiednią nawigację, która będzie się zmieniać w miarę dodawania lub usuwania stron do/z serwisu.

Zob. [Tworzenie szablonu serwisu z szablonu "Bootstrap" lub "Website Design"](#).

## Ukrywanie składników i układów sekcji motywu

Domyślnie wszystkie składniki i układy sekcji są dostępne dla wszystkich osób tworzących serwis. Niekiedy może wystąpić konieczność ukrycia składnika lub układu sekcji, tak aby był niedostępny, gdy jest używany konkretny motyw. Na przykład, chcąc zapewnić spójny wygląd, w którym nie będą używane żadne składniki "przycisk" ani układy sekcji suwaka, można je ukryć w motywie, edytując odpowiednio plik `components.json` motywu.

Aby ukryć składniki i układy sekcji w motywie, należy:

1. W usłudze Oracle Content Management wybrać opcję **Programista** (lewe menu nawigacyjne).
2. Na stronie "Programista" nacisnąć przycisk **Wyświetl wszystkie motywy**.
3. Na stronie "Motywy" wybrać motyw, który ma zostać zmodyfikowany, po czym wybrać z menu podręcznego opcję **Otwórz** albo kliknąć na ikonie  (na pasku czynności).
4. Wybrać plik `components.json`, a następnie nacisnąć przycisk **Pobierz**.
5. Otworzyć plik `components.json` w edytorze tekstu.
6. Dodać obiekty określające typ składnika i ID, po czym ustawić właściwość `hidden` na `true`.

Poniżej podano przykład wpisu ukrywającego składnik "przycisk":

```
"type": "scs-button",  
"id": "scs-button",  
"hidden": true
```

Analogicznie, poniżej podano przykład wpisu ukrywającego układ sekcji suwaka:

```
"type": "scs-sectionlayout",  
"id": "scs-sl-slider",  
"hidden": true
```

Pełny wpis w pliku components.json, pozwalający ukryć jednocześnie składnik "przycisk" i układ sekcji suwaka, wygląda następująco:

```
[  
  {  
    "name": "",  
    "list": [  
      {  
        "type": "scs-button",  
        "id": "scs-button",  
        "hidden": true  
      },  
      {  
        "type": "scs-sectionlayout",  
        "id": "scs-sl-slider",  
        "hidden": true  
      }  
    ]  
  },  
  {  
    "name": "Starter",  
    "list": [  
      {  
        "type": "component",  
        "id": "StarterComponent",  
        "themed": true  
      },  
      {  
        "type": "component",  
        "id": "StarterFooter",  
        "themed": true  
      }  
    ]  
  }  
]
```


7. Po wprowadzeniu odpowiednich zmian, wysłać plik components.json do motywu jako nową wersję.

## Ukrywanie opcji wyrównania, szerokości lub odstępów dla motywu

Domyślnie większość składników umożliwia osobom tworzącym serwisy ustawianie opcji wyrównania, szerokości i odstępów (opcje te są dostępne w oknie dialogowym "Ustawienia" danego składnika). Będąc programistą, można jednak ukryć te opcje w składniku niestandardowym w celu uniemożliwienia współtwórcom renderowania zawartości w sposób niespójny z motywem serwisu. Na przykład, aby zapewnić spójny wygląd, w którym przyciski są wyśrodkowane, a odstępy są ustawione na 30 pikseli, można ukryć te opcje w ustawieniach, edytując odpowiednio plik `components.json` motywu.

Niekiedy może także wystąpić potrzeba przesłonięcia początkowych wartości domyślnych składnika niestandardowego. Można to uczynić również, edytując plik `components.json`. Wszelkie zmiany w wartościach początkowych będą stosowane tylko do składników dodawanych do strony już po wysłaniu edytowanego pliku `components.json` jako nowej wersji.

Aby ukryć opcje ustawień wyrównania, szerokości lub odstępów dla składnika w motywie, należy:

1. W usłudze Oracle Content Management wybrać opcję **Programista** (lewe menu nawigacyjne).
2. Na stronie "Programista" nacisnąć przycisk **Wyświetl wszystkie motywy**.
3. Na stronie "Motywy" wybrać motyw, który ma zostać zmodyfikowany, po czym wybrać z menu podręcznego opcję **Otwórz** albo kliknąć na ikonie  (na pasku czynności).
4. Wybrać plik `components.json`, a następnie nacisnąć przycisk **Pobierz**.
5. Otworzyć plik `components.json` w edytorze tekstu.
6. Dodać obiekty określające typ składnika i ID, po czym ustawić właściwości, które mają zostać ukryte, na wartość `true`.  
Poniżej podano przykład wpisu ukrywającego w oknie dialogowym "Ustawienia" właściwości wyrównania, szerokości i odstępów składnika "przycisk":

```
"type": "scs-button",  
"id": "scs-button",  
"hideAlignmentAndWidth": true,  
"hideSpacing": true,
```

Chcąc przesłonić domyślne właściwości wyrównania i odstępów składnika "przycisk", można skorzystać z poniższego przykładu przedstawiającego, w jaki sposób określić wartości początkowe:

```
"initialData": {  
  "alignment": "center",  
  "marginTop": 30,  
  "marginBottom": 30
```

Poniżej przedstawiono pełny wpis z pliku `components.json`, określający wartości domyślne i ukrywający w oknie dialogowym "Ustawienia" opcje wyrównania, szerokości i odstępów składnika "przycisk", używanego w motywie:

```
[
  {
    "name": "",
    "list": [
      {
        "type": "scs-button",
        "id": "scs-button",
        "hideAlignmentAndWidth": true,
        "hideSpacing": true,
        "initialData": {
          "alignment": "center",
          "marginTop": 30,
          "marginBottom": 30
        }
      }
    ]
  },
  {
    "name": "Starter",
    "list": [
      {
        "type": "component",
        "id": "StarterComponent",
        "themed": true
      },
      {
        "type": "component",
        "id": "StarterFooter",
        "themed": true
      }
    ]
  }
]
```

7. Po wprowadzeniu odpowiednich zmian wysłać plik `components.json` do motywu jako nową wersję.

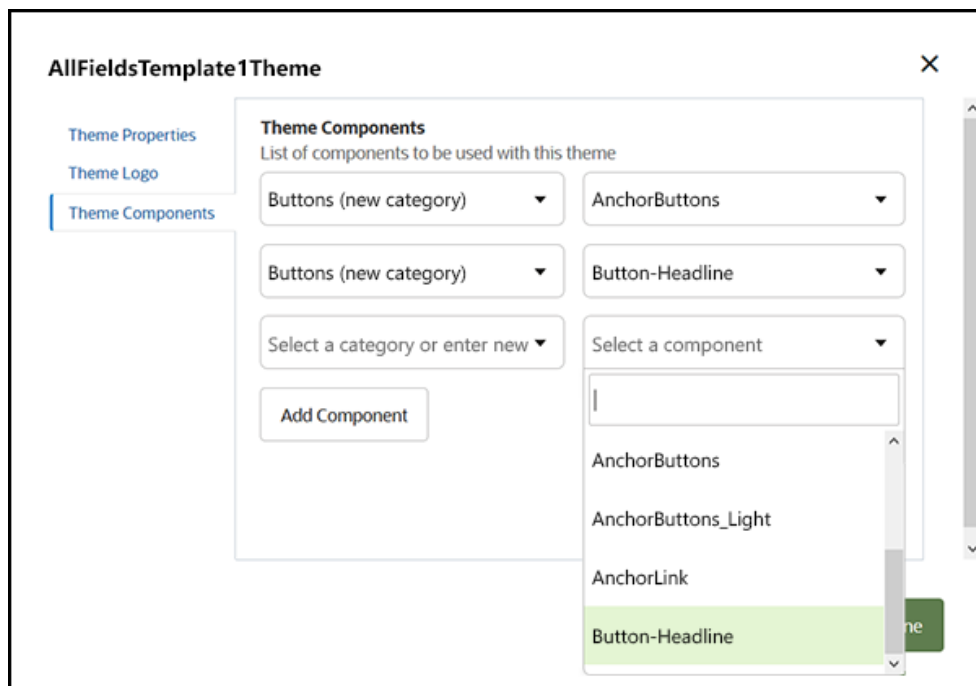
## Wiązanie składników z motywami

Aby można było używać określonego podzbioru składników z motywem, można z nim powiązać składniki, układy sekcji i grupy składników.

Będąc programistą, można tworzyć motywy i składniki dla szablonu, którego marketingowcy będą używać do tworzenia serwisów promujących produkty organizacji. Powiązanie składnika z motywem sprawia, że składnik ten staje się dostępny, gdy użytkownik wybiera w konstruktorze serwisów kartę **Składniki motywu**. Na karcie **Składniki motywu** użytkownik zobaczy tylko te składniki, które są powiązane z motywem. Powiązanie składnika z motywem pozwala uzyskać pewność, że składnik ten zostanie wyeksportowany wraz z szablonem serwisu, nawet jeśli nie zostanie użyty w serwisie.

Aby powiązać składnik z motywem, należy:

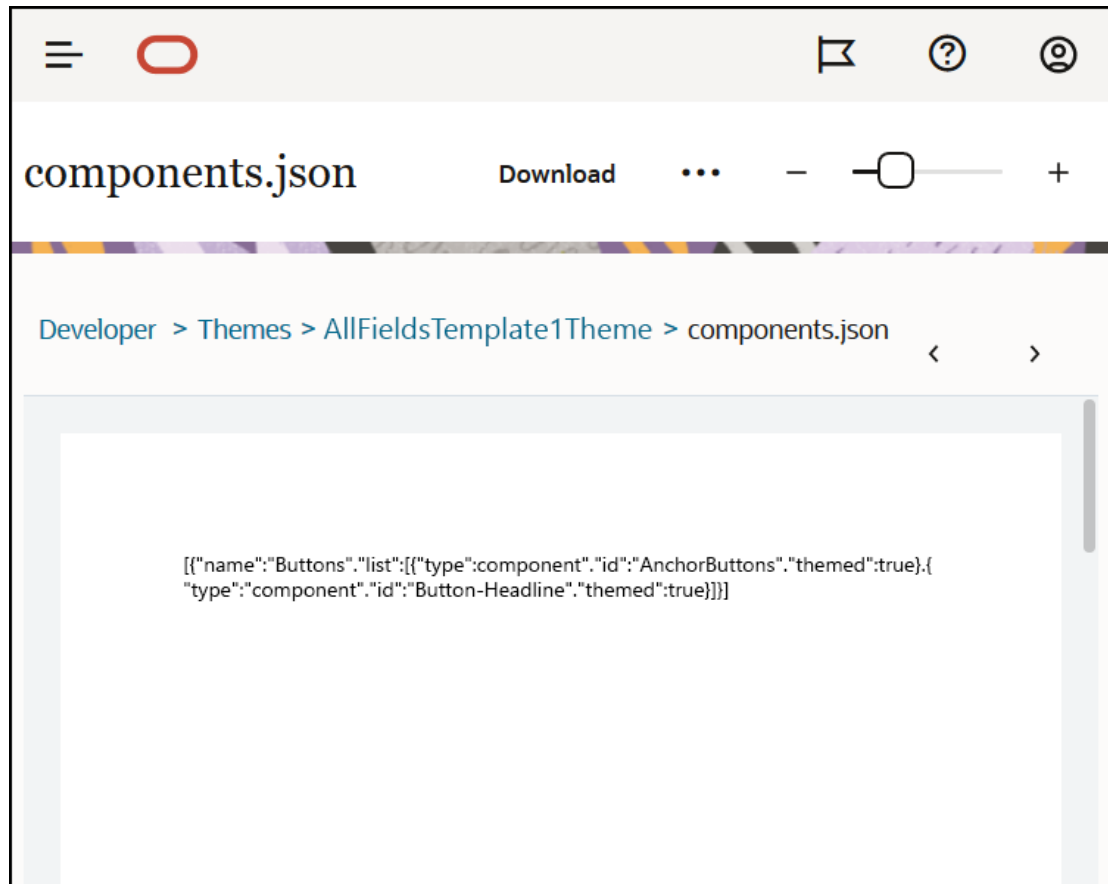
1. W usłudze Oracle Content Management wybrać opcję **Programista** (lewe menu nawigacyjne).
2. Na stronie **Programista** nacisnąć przycisk **Wyświetl wszystkie motywy**.
3. Na stronie "Motywy" wybrać motyw, po czym wybrać z paska czynności lub z menu podręcznego (kliknięcie prawym przyciskiem myszy) opcję **Właściwości**.
4. Aby otworzyć kartę **Składniki "motyw"**, nacisnąć przycisk **Składniki "motyw"**.
5. Wybrać jeden lub większą liczbę składników, które będą powiązane z motywem, a następnie sklasyfikować składniki:
  - a. Z rozwijanej listy **Wybór składnika** wybrać składnik niestandardowy.
  - b. Dla pierwszego wybranego składnika podać nazwę kategorii tworzonej dla składnika.
  - c. Dla każdego dodatkowo wybranego składnika wybrać już istniejącą kategorię albo utworzyć nową.
  - d. Nacisnąć przycisk **Dodaj składnik**.



6. Aby powiązać wybrane składniki z motywem, nacisnąć przycisk **Zapisz**.
7. Aby ponownie otworzyć kartę **Składniki "motyw"**, na stronie **Motywy** wybrać motyw, po czym nacisnąć przycisk "Właściwości".
8. Sprawdzić, czy wybrane składniki zostały zapisane.

Aby usunąć powiązany składnik z motywu, należy na karcie **Składniki "motyw"** kliknąć na ikonie **X**, znajdującej się obok nazwy składnika, po czym w oknie dialogowym **Usuwanie składnika** nacisnąć przycisk **Tak**.

Na stronie **Szczegóły** danego szablonu można zobaczyć listę składników powiązanych z motywem, których ten szablon używa. Odwołania do składników powiązanych z motywem są przechowywane w pliku `components.json` motywu.



Po skonfigurowaniu powiązania motywu, szablon można wprowadzić do środowiska wykonawczego Oracle Content Management, a następnie udostępnić go zespołowi marketingowców. Gdy marketingowcy wybiorą ten motyw dla serwisu, zobaczą na karcie **Składniki motywu** (w konstruktorze serwisów) tylko te składniki, które zostały dla tego motywu określone. Listy składników **niestandardowe** i **wszystkie** również obejmują powiązane składniki używane w motywie.

#### Wiązanie składnika z motywem za pomocą zestawu OCE Toolkit

Aby powiązać składnik z motywem za pomocą zestawu OCE Toolkit, należy użyć poniższego polecenia `cec`. Składnik zostanie wyświetlony na karcie **Składniki "motyw"** w konstruktorze serwisów, a także na listach składników **Niestandardowe** i **Wszystkie**.

```
cec add-component-to-theme <component>
```

Poniższe polecenie `cec` usuwa powiązanie składnika z motywem.

```
cec remove-component-from-theme <component>
```

Informacje dotyczące OCE Toolkit są dostępne w podręczniku Tworzenie za pomocą zestawu OCE Toolkit.

## Sites Rendering API

**Sites Rendering API** dla Oracle Content Management (SCSRenderAPI) jest globalnym obiektem "window" występującym na wszystkich stronach Oracle Content Management. Przede wszystkim jest odpowiedzialny za renderowanie gniazd i składników strony, a ponadto udostępnia interfejs dla kodu JavaScript występującego w układach motywów. Z tymi zdarzeniami i funkcjami mogą mieć do czynienia programiści zajmujący się motywami lub składnikami. SCSRenderAPI w **trybie wykonawczym** renderuje widok i podgląd, zaś w **trybie projektowania** — nawigację, elementy edycyjne i adnotacje.

W trybie wykonawczym **Sites Rendering API** ma trzy ogólne zastosowania:

- Wypełnianie gniazd na stronie składnikami i zawartością.
- Spełnianie informacyjnych żądań kodu JavaScript występującego w układzie strony.
- Zgłaszanie zdarzeń w całym cyklu życia procesu renderowania strony.

W trybie projektowym **Sites Rendering API** ma cztery ogólne zastosowania:

- Wypełnianie gniazd na stronie składnikami i zawartością.
- Spełnianie informacyjnych żądań kodu JavaScript występującego w układzie strony.
- Zgłaszanie zdarzeń w całym cyklu życia procesu renderowania strony.
- Współdziałanie z modułem "Konstruktor serwisów" w celu umożliwienia edytowania strony i wprowadzania adnotacji.

**Sites Rendering API** można załadować do internetowych stron Oracle Content Management, używając następującego znacznika "script" w układach motywów:

```
<script data-main="/_sitescloud/renderer/renderer.js" src="/_sitescloud/renderer/require.js"></script>
```

Znacznik ten zazwyczaj umieszcza się na dole układu. Występujące w tym znaczniku adresy URL będą automatycznie dostosowywane do odpowiedniego środowiska: projektowego lub wykonawczego. Obiekt SCSRenderAPI jest ładowany asynchronicznie; niestandardowy kod JavaScript może nasłuchiwać dostępności Sites Rendering API, obsługując zdarzenie scsrenderstart.

## Tworzenie układów

Układ wyznacza sposób rozmieszczenia zawartości na stronie oraz jest używany do tworzenia kodu HTML stron tworzących serwisy w Oracle Content Management.

- [Układy — informacje podstawowe](#)
- [Optymalizacja pod kątem wyszukiwarek \(SEO\)](#)
- [Omówienie pliku components.json i jego formatu](#)
- [Dostosowywanie grup narzędzi w konstruktorze serwisów](#)
- [Ograniczanie składników w gniazdach](#)
- [Czynienie zawartości układu edytowalną](#)
- [Tworzenie układu sekcji](#)
- [Tworzenie układu sekcji obsługującego skrypty Lazy Load](#)
- [Tworzenie niestandardowych układów sekcji za pomocą API](#)
- [Tworzenie układów zawartości](#)

### Układy — informacje podstawowe

Układ wyznacza sposób rozmieszczenia zawartości na stronie serwisu. Różne układy mogą zawierać różną liczbę nazwanych gniazd. Gniazdo jest obszarem rozciągającym się na szerokość strony, mogącym zawierać jeden lub więcej typów zawartości.

Każdy motyw zawiera kilka układów strony. Zob. [Motywy — informacje podstawowe](#).

Dodając stronę do serwisu, wybiera się jej układ. Każdy układ wyznacza obszary na stronie — określane terminem "gniazdo" — do których można przeciągać zawartość. To, jaka zawartość znajdzie się w tych gniazdach, zależy wyłącznie od nas. Może to być wszystko: od tytułów, tekstów i separatorów do multimediów, galerii i mediów społecznościowych.

Układ zawiera poprawne konstrukcje w języku HTML oraz specjalny kod znacznikowy rozumiany przez moduł renderujący Oracle Content Management. Układ musi się zaczynać instrukcją DOCTYPE mającą na celu skonfigurowanie przeglądarki do renderowania w trybie zgodnym z obowiązującymi standardami; na przykład: `<!DOCTYPE html>` (ta instrukcja jest wymagana przez niektóre składniki, aby zostały uzyskane możliwie najlepsze wyniki).

Poniżej jest przedstawiony przykładowy kod minimalnego układu:

```
1 <!DOCTYPE html>
3 <head>
4     <meta http-equiv="X-UA-Compatible" content="IE=edge">
5
6     <script src="/_themes/[!-$SCS_THEME_NAME--]/assets/js/
topnav.js"></script>
7     <link rel="stylesheet" type="text/css"
8         href="/_themes/[!-$SCS_THEME_NAME--]/assets/css/main.css">
9     <link rel="stylesheet" type="text/css"
```



```
10             href="/_themes/[!--$SCS_THEME_NAME--]/designs/  
[!--$SCS_DESIGN_NAME--]/design.css">  
11  
12         <!--$SCS_RENDER_INFO-->  
13         <!--$SCS_SITE_HEADER-->  
14         <!--$SCS_PAGE_HEADER-->  
15     </head>  
16     <body>  
17         <div id="topNavigation"></div>  
18         <div id="mainContentSlot" class="scs-slot scs-responsive"></  
div>  
19         <script data-main="/_sitescloud/renderer/renderer.js"  
20             src="/_sitescloud/renderer/require.js"></script>  
21         <!--$SCS_SITE_FOOTER-->  
22     </body>  
23 </html>
```

Różne tokeny są rozwijane, gdy strona jest renderowana w przeglądarce.

- [!--\$SCS\_THEME\_NAME--]  
Ten token rozwija się do nazwy motywu obecnie wybranego dla serwisu. Używając tego tokenu, umożliwia się kopiowanie motywu, ponieważ korzystające z tego tokenu adresy URL będą się odwoływać do bieżącego motywu.
- [!--\$SCS\_DESIGN\_NAME--]  
Ten token rozwija się do nazwy konstrukcji obecnie wybranej dla serwisu. Używając tego tokenu, umożliwia się używanie układu przez wiele konstrukcji zawartych w motywie.
- [!--\$SCS\_RENDER\_INFO--]  
Ten token rozwija się do znacznika "script", w którym jest przechowywana informacje dotyczące hierarchicznej struktury strony oraz renderowania składników. Token ten powinien zostać umieszczony w sekcji <head> układu.
- [!--\$SCS\_SITE\_HEADER--]  
Token ten rozwija się do wartości nagłówka serwisu, określonej w polu "Header" we właściwościach SEO (Search Engine Optimization). Można tu wprowadzić kod znacznikowy, który ma być umieszczany na wszystkich stronach serwisu. Zob. [Ustawienie właściwości dotyczących wyszukiwarek](#).
- [!--\$SCS\_PAGE\_HEADER--]  
Ten token rozwija się do wartości nagłówka (nagłówka strony) znalezionej w polu "Nagłówek strony" we właściwościach "Ustawienia strony" w konstruktorze serwisów. Można tu wprowadzić kod znacznikowy właściwy dla strony.
- [!--\$SCS\_SITE\_FOOTER--]  
Token ten rozwija się do wartości stopki serwisu, występującej w polu "Footer" we właściwościach SEO (Search Engine Optimization). Można tu wprowadzić kod znacznikowy, który ma być umieszczany na wszystkich stronach serwisu. Zob. [Ustawienie właściwości dotyczących wyszukiwarek](#).

 **Uwaga:**

Dla tokenów tych można także używać (jako ograniczników) prefiksu `<!--$` i sufiksu `-->`, zamiast `[!--$ i --]`.

Gdy serwis działa w trybie online, przedstawione poniżej tokeny (występujące w łączy) są zastępowane wartościami rzeczywistymi, zgodnymi z kontekstem, w którym są używane. Dzięki temu łączy może funkcjonować zarówno wtedy, gdy serwis jest edytowany, jak i w opublikowanym serwisie online.

- `/_sitescloud/` jest zastępowany przez `/_sitesclouddelivery/`
- `/_themes/` jest zastępowany przez `/_themesdelivery/`

**Gniazda** są to występujące w układzie elementy DIV, mające w atrybucie "class" wartość "scs-slot". Gniazdo jest to miejsce, w którym użytkownicy mogą dodawać składniki w celu wypełnienia zawartości serwisu. Na gniazda można wyznaczać wiele elementów DIV, przypisując im w atrybucie "class" wartość "scs-slot". Każde gniazdo musi mieć unikatowy atrybut id.

 **Uwaga:**

Gniazda nie można zagnieżdżać, lecz można to sugerować, używając kodu CSS powodującego ich nakładanie się. Chcąc użyć takiego sposobu, trzeba użyć układu "grupa składników" lub układu "sekcja". Zob. [Tworzenie układu sekcji](#).

Gniazda, dla których w atrybucie "class" została użyta wartość "scs-responsive", będą odświeżały swoją zawartość, gdy ulegnie zmianie rozdzielczość portu widoku przeglądarki. Dzięki temu gniazda mogą być renderowane odpowiednio zarówno na dużych monitorach, jak i w małych urządzeniach mobilnych.

Końcowy znacznik `<script>` (w przykładzie linia nr 19) ładuje kod renderowania Oracle Content Management. Kod ten jest odpowiedzialny za rysowanie składników na stronie, a ponadto umożliwia dostęp kodu niestandardowego do Render API. Bez tego znacznika `<script>`, strony oparte na tym układzie nie mogłyby być zmieniane przez konstruktor serwisów.

## Optymalizacja pod kątem wyszukiwarek (SEO)

Można określić słowa kluczowe pomagające wyszukiwarkom rozpoznawać zawartość serwisu.

### Ustawienia SEO

Ustawienia optymalizacji pod kątem wyszukiwarek (SEO — Search Engine Optimization) można definiować na poziomie serwisu i na poziomie strony. Tekst SEO jest wprowadzany do stopki wszystkich gotowych szablonów.

Zob. [Ustawienie właściwości dotyczących wyszukiwarek](#).

### Pliki cookie dla odwiedzających serwis

Odwiedzający serwis używają plików cookie w celu optymalizacji pod kątem wyszukiwarek — jeden plik cookie na każdą przeglądarkę i każdy serwis — w celach rozliczeniowych. Każdy plik cookie musi być odnawiany co godzinę lub po 24 godzinach.

Każdy serwis musi zawierać wyskakujące okno, informujące odwiedzających o plikach cookie.

Footer

Text entered above will be displayed in the Cookie consent message.

Oracle Content and Experience uses cookies for the purpose of identifying unique user visits. To comply with EU Cookie Regulations your site should display the Cookie consent message and include a separate page explaining use of cookies on the site to visitors. For more information about use of cookies in Oracle Content and Experience, see your Service Agreement.

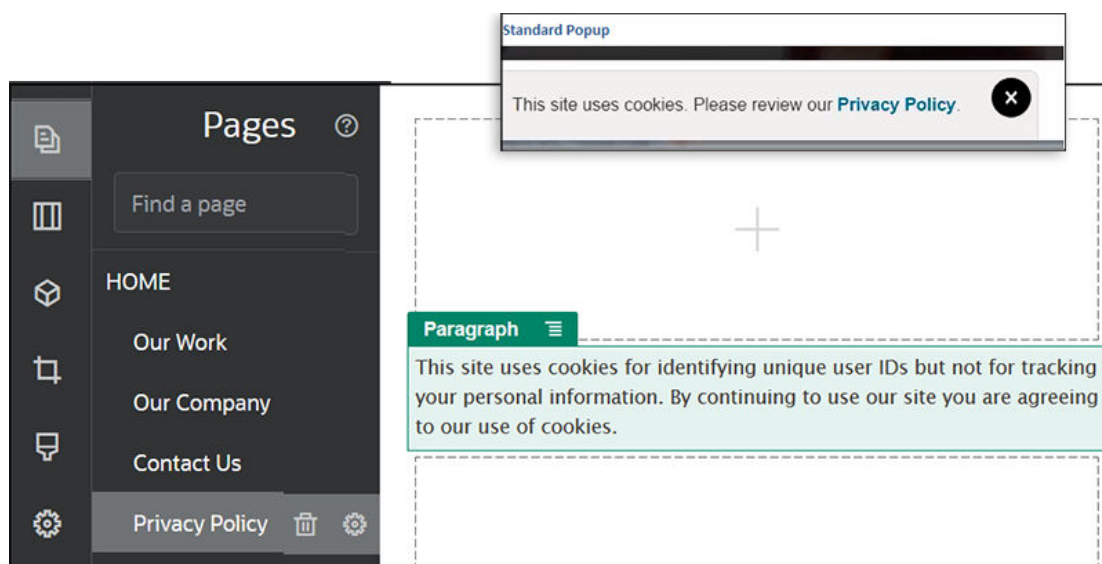
Okno to powinno zawierać następujące informacje:

- Tekst nagłówka SEO i stopki
- Zawarty w stopce element div z określonym ID, w którym to elemencie będzie wybierany i dołączany odpowiedni tekst

```
<div class="scs-cookie-consent" style="display: none;">This site uses cookies. Please review our  
<a href="[!-$SCS_PAGE-]500[!-$SCS_PAGE-]">Privacy Policy</a>. <a href="#" alt="close"  
class="scs-cookie-consent-close"></a></div>
```

#### Strona "Zasady ochrony prywatności"

Tekst ten będzie tworzył łącze do strony z zasadami ochrony prywatności, zawierającej tekst wyjaśniający.



### Wewnętrzny kod optymalizacji pod kątem wyszukiwarek (SEO)

Zapisując składnik w konstruktorze serwisów, można zapisać wewnętrzny kod HTML, który będzie używany, jeśli składnik będzie renderowany w trybie wykonawczym. Ten wewnętrzny kod HTML jest przechowywany w danych strony, wskutek czego dane mogą zostać wstawione zamiast składnika, gdy strona jest renderowana. Może to nastąpić na wczesnym etapie renderowania, dzięki czemu wyszukiwarka ma większą szansę na pomyślne przeszukanie strony.

## Omówienie pliku `components.json` i jego formatu

Motyw musi mieć plik `components.json` znajdujący się w katalogu `/NazwaMotywu` i określający składniki używane w tym motywie.

Plik musi zawierać poprawną strukturę JSON — przynajmniej pustą tablicę JSON `[]`.

W pliku `components.json` są wyszczególnione wszystkie składniki lokalne, przy czym składnia pliku w pełni obsługuje klasyfikację składników. (Składniki odległe są rejestrowane w Katalogu składników.)

Bez względu na to, które składniki zostały dodane do pliku `components.json` poziomu składnika (w tym brak składników), Oracle Content Management oferuje domyślny zestaw składników dostępnych dla użytkowników. Ten zestaw domyślny jest definiowany w kodzie źródłowym. Poniższy wykaz zawiera składniki lokalne oraz składniki renderowane w ramach wstawkowych. Ponadto w konstruktorze serwisów będą dostępne wszelkie składniki odległe, zarejestrowane na poziomie usługi i udostępnione użytkownikom instancji.

W Oracle Content Management są dostępne następujące składniki lokalne:

Nazwa	Typ	ID
Tytuł	<code>scs-title</code>	<code>scs-title</code>
Akapit	<code>scs-paragraph</code>	<code>scs-paragraph</code>
Obraz	<code>scs-image</code>	<code>scs-image</code>
Galeria	<code>scs-gallery</code>	<code>scs-gallery</code>

Nazwa	Typ	ID
Siatka galerii	<code>scs-gallerygrid</code>	<code>scs-gallerygrid</code>
Dokument	<code>scs-document</code>	<code>scs-document</code>
Przycisk	<code>scs-button</code>	<code>scs-button</code>
Mapa	<code>scs-map</code>	<code>scs-map</code>
Separator	<code>scs-divider</code>	<code>scs-divider</code>
Rozdzielacz	<code>scs-spacer</code>	<code>scs-spacer</code>
YouTube	<code>scs-youtube</code>	<code>scs-youtube</code>
Pasek społecznościowy	<code>scs-socialbar</code>	<code>scs-socialbar</code>
Wideo	<code>scs-video</code>	<code>scs-video</code>
Artykuł (składnik niestandardowy)	<code>scs-component</code>	<code>scs-comp-article</code>
Nagłówek (składnik niestandardowy)	<code>scs-component</code>	<code>scs-comp-headline</code>
Obraz i tekst (składnik niestandardowy)	<code>scs-component</code>	<code>scs-comp-image-text</code>

W Oracle Content Management są dostępne poniższe składniki renderowane w ramach wstawkowych. Wykaz ten nie obejmuje zarejestrowanych składników odległych.

Nazwa	Typ	ID
Konwersacja	<code>scs-app</code>	Konwersacja
Menedżer dokumentów	<code>scs-app</code>	Menedżer dokumentów
Lista folderów	<code>scs-app</code>	Lista folderów
Lista plików	<code>scs-app</code>	Lista plików
Facebook - Lubię to	<code>scs-app</code>	Facebook - Lubię to
Facebook - Rekomendacja	<code>scs-app</code>	Facebook - Rekomendacja
Twitter - Obserwuj	<code>scs-app</code>	Twitter - Obserwuj
Twitter - Udostępnij	<code>scs-app</code>	Twitter - Udostępnij

### Format ogólny

Ogólny format pliku `components.json` jest następujący:

- Właściwości składników są określane w ramach każdego z nich. Właściwości `"components"` i `"apps"` najwyższego poziomu są już nieużywane.
- Każdy składnik ma właściwość `"type"`. Składniki mogą mieć tylko określone wartości (wszystkie możliwe wartości są wymienione w tabeli składników domyślnych).
- Każdy składnik ma właściwość `"id"`, która musi mieć unikatową wartość. Właściwość ta służy do rozróżniania składników tego samego typu (o tej samej wartości właściwości `"type"`). Wcześniej aplikacje miały właściwość `"appName"`. Podczas gdy `"appName"` nadal działa, jeśli właściwość `"id"` jest niedostępna, to właściwość `"appName"` jest już nieużywana.

- Każdy składnik ma właściwość "name", określającą nazwę wyświetlaną w interfejsie użytkownika. Jeśli nie zostaną określone wartości rezerwowe, to dla składników wartością domyślną jest nazwa odpowiedniego składnika domyślnego, a dla składników odległych — ich ID.

Przykład pliku components.json:

```
[
  {
    "name": "COMP_CONFIG_TEXT_CATEGORY_NAME",
    "list": [
      {
        "type": "scs-title",
        "id": "my-headline",
        "name": "My Headline",
        ...
      },
      {
        ...
      },
      ...
    ],
    ...
  },
  {
    "name": "My own category name",
    "list": [ ... ]
  }
]
```

Ogólna struktura ma postać tablicy JSON obiektów "kategoria". Każdy obiekt "kategoria" ma właściwość "name" i właściwość "list". Właściwość "name" może być kluczem odwzorowywanym na zlokalizowany napis. Jeśli te kategorie domyślne są niewystarczające, można podać nazwę własnej kategorii, która to nazwa nie będzie lokalizowana. W poniższej tabeli wymieniono dostępne kategorie domyślne i odpowiadające im klucze.

Klucz	Nazwa kategorii (po angielsku)
COMP_CONFIG_CONTENT_CATEGORY_NAME	Content
COMP_CONFIG_CUSTOM_CATEGORY_NAME	Custom
COMP_CONFIG_MEDIA_CATEGORY_NAME	Media
COMP_CONFIG_SOCIAL_CATEGORY_NAME	Social
COMP_CONFIG_TEXT_CATEGORY_NAME	Text

Właściwość "list", występująca w każdym obiekcie "kategoria", zawiera tablicę obiektów składników. Każdy składnik lub obiekt musi mieć właściwości "type" i "id". Pozostałe właściwości są opcjonalne.

- Właściwość "type" musi dokładnie odpowiadać jednemu z typów występujących w składnikach domyślnych. Jeśli "type" jeszcze nie istnieje, składnik nie zostanie wyświetlony.
- Właściwość "id" musi być unikatowa w obrębie wszystkich składników. Jeśli okaże się, że "id" już istnieje, składnik nie zostanie wyświetlony.

- Właściwość "name" nazwą składnika wyświetlaną w interfejsie użytkownika. Zastępuje wcześniejszą właściwość "appName" dla aplikacji (obecnie składniki odległe).
- Wszystkie inne właściwości są traktowane tak samo jak w poprzednich wydaniach.

#### Dodawanie nowych składników do pliku components.json

Składników domyślnych nie można modyfikować. Można jednak utworzyć nowy składnik oparty na istniejącym składniku domyślnym. Na przykład można by utworzyć nowy składnik "scs-title" ustawiający pewien tekst domyślny. Minimalnym wymogiem — który trzeba spełnić, dodając składnik — jest określenie właściwości "type" i "id".

- Właściwość "type" musi dokładnie odpowiadać jednemu z typów występujących w składnikach domyślnych. Jeśli "type" jeszcze nie istnieje, składnik nie zostanie wyświetlony.
- Właściwość "id" musi być unikatowa w obrębie wszystkich składników. Jeśli okaże się, że "id" już istnieje, składnik nie zostanie wyświetlony.

Przykład kodu dodającego nowy składnik "title". Składnik ten będzie wyświetlany wraz z domyślnym składnikiem "title".

```
[
  {
    "name": "COMP_CONFIG_TEXT_CATEGORY_NAME",
    "list": [
      {
        "type": "scs-title",
        "id": "my-headline"
      }
    ]
  }
]
```

Przykład kodu dodającego nowy składnik "title" z wyświetlaną nazwą i tekstem domyślnym:

```
[
  {
    "name": "COMP_CONFIG_TEXT_CATEGORY_NAME",
    "list": [
      {
        "type": "scs-title",
        "id": "my-headline",
        "name": "My Headline",
        "initialData": {
          "userText": "This is a second title component"
        }
      }
    ]
  }
]
```

Składnik "title" przyjmuje wszystkie właściwości domyślnego składnika "title", przy czym w celu utworzenia nowego składnika są stosowane modyfikacje na poziomie motywu.

### Zgodność wsteczna

Pliki `components.json` w poprzednim formacie nadal będą odczytywane.

- Pliki z właściwościami "components" lub "apps" najwyższego poziomu.
- Jeśli plik zawiera właściwość "apps", występujące w niej zdefiniowane przez użytkownika składniki odległe będą nadal ładowane.
- Jeśli plik zawiera właściwość "apps" najwyższego poziomu, to należy przyjąć założenie, że wszystkie wymienione w niej składniki są typu "scs-app".
- Jeśli występuje właściwość "appName", należy ustawić właściwość "id" na wartość "appName". Wyświetlaną nazwą będzie "name" (o ile właściwość ta zostanie określona) albo wartość "id".

## Dostosowywanie grup narzędzi w konstruktorze serwisów

Dla składników niestandardowych można dostosować kopie gotowych (standardowych) grup narzędzi, wyświetlanych na pasku narzędzi konstruktora serwisów.

Można utworzyć własną wersję standardowego składnika, mającego ustawione ograniczenia i dostępnego w znaczniku "custom". Żadne dostosowania nie mają wpływu na standardowe składniki.

Można utworzyć własną wersję standardowego składnika, mającego ustawione ograniczenia i dostępnego w znaczniku "custom", lecz nie wpłynie ona na żaden standardowy składnik.

Grupy narzędzi decydują o tym, co zostanie wyświetlone na pasku narzędzi, gdy użytkownik kliknie na składniku "tytuł" lub "akapit", aby go edytować. Można usunąć grupy narzędzi obsługiwanych przez składnik "tytuł" lub "akapit", ale dodatkowe wtyczki nie są dozwolone.

Aby dostosować grupy narzędzi w konstruktorze serwisów, należy użyć tej samej składni, której używa CKEditor dla konfiguracji `toolbarGroups`. Zob. <https://ckeditor.com/latest/samples/toolbarconfigurator/index.html#basic>.

Składniki "tytuł" i "akapit" obsługują następujące grupy:

- "style podstawowe" — ograniczone do: pogrubienie, kursywa, podkreślenie; <https://ckeditor.com/cke4/addon/basicstyles>
- "style" — style czcionek; [https://docs.ckeditor.com/ckeditor4/latest/guide/dev\\_styles.html](https://docs.ckeditor.com/ckeditor4/latest/guide/dev_styles.html)
- "kolory" — kolory tekstu i tła; [https://docs.ckeditor.com/ckeditor4/latest/guide/dev\\_colorbutton.html](https://docs.ckeditor.com/ckeditor4/latest/guide/dev_colorbutton.html)
- "cofnięcie" — cofa/ponawia bieżące działanie w edytorze CKEditor
- "łącza" — niestandardowa wtyczka dialogowego okna łącza
- "akapit" — obsługa list wypunktowanych/numerowanych oraz wcięć
  - "lista"
  - "wcięcie"
- "wyrównanie" — do lewej, do prawej, wyśrodkuj
- "czyszczenie" — usuwa wszystkie style z wybranego tekstu



Składnik akapit obsługuje także wtyczki wstawiania obrazów i tabel

- "wstaw"
  - "obraz"
  - "tabela"

Dodatkowo można używać wpisu separatora wierszy:

- "/"

 **Uwaga:**

Jeśli w konfiguracji grupy (paska narzędzi) ustawiono dowolną inną wartość, to — przed utworzeniem paska narzędzi konstruktora serwisów — wartość ta zostanie usunięta. Nie można udostępnić "extraPlugins" (dodatkowych wtyczek). Obsługiwane są tylko konfiguracje "nazwa"/"grupa". Wszelkie wpisy "elementy" będą ignorowane.

Na przykład, aby uniemożliwić użytkownikom definiowanie czcionek, kolorów, stylów lub rozmiarów, można zaktualizować konfigurację, jak pokazano poniżej. Dla "id" trzeba określić własne wartości różniące się od wartości standardowych.

```
[{
  "name": "<category name>",
  "list": [{
    "type": "scs-title",
    "id": "<custom-value>",
    "config": {
      "toolbarGroups": [{
        "name": "basicstyles",
        "groups": ["basicstyles"]
      }, {
        "name": "undo",
        "groups": ["undo"]
      },
      "/", {
        "name": "links",
        "groups": ["links"]
      }, {
        "name": "paragraph",
        "groups": ["list", "indent"]
      },
      "/", {
        "name": "align",
        "groups": ["align"]
      },
      {
        "name": "insert",
        "groups": ["image", "table"]
      }, {
        "name": "cleanup",
        "groups": ["cleanup"]
      }
    ]
  }
}]
```

```

    ]
  }, {
    "type": "scs-paragraph",
    "id": "<custom-value>",
    "config": {
      "fontSize_sizes": "16/16px;24/24px;48/48px;"
    }
  }
}]

```

### Weryfikacja

Należy zastąpić swój plik `components.json` pokazanym wcześniej kodem, a następnie edytować swój serwis (jeśli już jest edytowany, należy odświeżyć przeglądarkę). Od tej pory, gdy będzie edytowany składnik "tytuł", nie będą już pokazywane możliwe do wybrania style ani kolory czcionek. Będą one nadal pokazywane dla składnika "akapit", lecz lista dostępnych rozmiarów czcionek będzie ograniczona do 16, 24 i 48.

### Domyślne grupy narzędzi

Domyślne grupy narzędzi "Tytuł" i "Akapit":

- Tytuł

```

[ {
  "name": "basicstyles",
  "groups": ["basicstyles"]
}, {
  "name": "styles",
  "groups": ["styles"]
}, {
  "name": "colors",
  "groups": ["colors"]
}, {
  "name": "undo",
  "groups": ["undo"]
},
"/", {
  "name": "links",
  "groups": ["links"]
}, {
  "name": "paragraph",
  "groups": ["list", "indent"]
}, {
  "name": "align",
  "groups": ["align"]
}, {
  "name": "cleanup",
  "groups": ["cleanup"]
}
]

```

- Akapit

```
[{
  "name": "basicstyles",
  "groups": ["basicstyles"]
}, {
  "name": "styles",
  "groups": ["styles"]
}, {
  "name": "colors",
  "groups": ["colors"]
}, {
  "name": "undo",
  "groups": ["undo"]
},
"/", {
  "name": "links",
  "groups": ["links"]
}, {
  "name": "paragraph",
  "groups": ["list", "indent"]
}, {
  "name": "align",
  "groups": ["align"]
}, {
  "name": "insert",
  "groups": ["image", "table"]
}, {
  "name": "cleanup",
  "groups": ["cleanup"]
}
]
```

## Ograniczanie składników w gniazdach

Dla każdego gniazda układu można określić ograniczenia dotyczące składników dozwolonych w danym gnieździe.

Jeśli takie ograniczenia zostaną określone dla gniazda, to każdy użytkownik, przeciągając składnik niedozwolony w tym gnieździe, zobaczy komunikat z ostrzeżeniem i nie będzie mógł dodać ani przenieść takiego składnika do tego gniazda.

Chcąc skonfigurować ograniczenie, trzeba edytować układy zawarte w motywie (na przykład plik układu `themes\nazwa_układu\layouts\oneslot.htm`) i dodać niestandardowe atrybuty "data" do elementu "div" gniazda.

Poniżej jest przedstawiony format atrybutów niestandardowych. Główna różnica w porównaniu z poprzednimi wersjami polega na tym, że użytkownicy muszą określić tylko `id` składnika. Poprzednio użytkownicy, aby ograniczyć możliwość użycia

składnika, musieli używać dosłownej składni "<type> <id>" (która nadal jest obsługiwana).

```
data-allowed-items='["<id>:", "<type>", "<type>:<id>", ...]'
data-disallowed-items='["<id>":"<type>", "<type>:<id>", ...]
```

#### Uwaga:

W wartościach atrybutów `data-allowed-items` i `data-disallowed-items` można używać znaków cudzysłowu (") lub apostrofów ('). W poniższym przykładzie dla atrybutu `data-allowed-items` tablica JSON jest ujęta w apostrofy, podczas gdy dla atrybutu `data-disallowed-items` jest ujęta w cudzysłów:

```
<div id="slot101"
  class="scs-slot"
  data-allowed-items='["scs-app", "scs-title"]'
  data-disallowed-items=["'File List', 'scs-map']">
</div>
```

W poniższej tabeli wymieniono nazwy, typy i identyfikatory (ID) składników udostępnionych przez Oracle Content Management. W celu zapobieżenia konfliktowi nazw *nie* należy poprzedzać żadnego dostosowanego ID składnika prefiksem `scs-` ani używać któregośkolwiek z typów lub identyfikatorów zamieszczonych w tej tabeli.

Nazwa	Typ	ID
Menedżer dokumentów	scs-app	Menedżer dokumentów
Facebook - Lubię to	scs-app	Facebook - Lubię to
Facebook - Rekomendacja	scs-app	Facebook - Rekomendacja
Lista plików	scs-app	Lista plików
Lista folderów	scs-app	Lista folderów
Twitter - Obserwuj	scs-app	Twitter - Obserwuj
Twitter - Udostępnij	scs-app	Twitter - Udostępnij
Przycisk	scs-button	scs-button
Artykuł (składnik niestandardowy)	scs-component	scs-comp-article
Nagłówek (składnik niestandardowy)	scs-component	scs-comp-headline
Obraz i tekst (składnik niestandardowy)	scs-component	scs-comp-image-text
Grupa składników	scs-componentgroup	scs-componentgroup
Wyszukiwanie zawartości	scs-contentsearch	scs-contentsearch
Lista zawartości	scs-contentlist	scs-contentlist
Element zastępczy zawartości	scs-component	scs-contentplaceholder
Element zawartości	scs-component	scs-contentitem

Nazwa	Typ	ID
Separator	scs-divider	scs-divider
Dokument	scs-document	scs-document
Galeria	scs-gallery	scs-gallery
Siatka galerii	scs-gallerygrid	scs-gallerygrid
Obraz	scs-image	scs-image
Mapa	scs-map	scs-map
Akapit	scs-paragraph	scs-paragraph
Pasek społecznościowy	scs-socialbar	scs-socialbar
Rozdzielacz	scs-spacer	scs-spacer
Tytuł	scs-title	scs-title
YouTube	scs-youtube	scs-youtube

Użytkownicy mogą tworzyć składniki lokalne lub odległe. Podana w tym przykładzie nazwa `My_Local_Component` jest identyfikatorem, którego można użyć do określenia tego składnika jako niedozwolonego w gniazdach.

### Create Local Component

✕

Provide a name for your component

My\_Local\_Component

Use only letters, numbers, hyphens, and underscores in component names.

Provide a description for your component (optional)

Add an optional description for this component

Component Type

Default ▼

Cancel

Create

## Czynienie zawartości układu edytowalną

Niektórą występującą w układzie zawartość graficzną lub tekstową można uczynić edytowalną dla użytkowników pracujących ze stronami opartymi na danym układzie.

Funkcji tej można użyć w dowolnym motywie, w tym w motywach Bootstrap. Modyfikacje automatycznie przyjmą styl oryginalnej strony.

Można dodać prosty kod znacznikowy do następujących znaczników używanych w układzie: `<p>`, `<h1>` do `<h6>`, `<div>` i `<img>`.

Opcje formatowania tekstu i obrazów obejmują:

- Tekst: użytkownicy mogą wprowadzać pogrubienie, kursywę i podkreślenie oraz powiązanie łączem.
- Obraz: użytkownicy mogą określać łącza prowadzące do obrazów. Mogą też zmieniać właściwości tytułu (wyświetlanego, gdy na obrazie zostanie umieszczony wskaźnik myszy) oraz dodawać tekst alternatywny w celu ułatwienia dostępu.

Modyfikując element typu tekst lub obraz, tak aby użytkownicy mogli go edytować, należy:

1. Dodać do atrybutu `class` właściwość `scs-editable`.
2. Dodać unikatowy atrybut `id`.

Gdy tylko układ zostanie zmodyfikowany, nowa funkcjonalność będzie dostępna dla wszystkich stron opartych na tym układzie (także dla nowych stron).

Jeśli strona zostanie skopiowana i wklejona, to modyfikacje zostaną wklejone na nowej stronie.

### Modyfikowanie znacznika nagłówka

Poniżej jest zamieszczony przykład, jak zmodyfikować występujący w układzie znacznik nagłówka, tak aby nagłówek ten mógł być edytowany przez użytkowników.

1. Zsynchronizować plik układu z komputerem lokalnym albo edytować źródłowy plik HTML w celu zmodyfikowania nagłówka `<h1>`.
2. Dodać do atrybutu `class` właściwość `scs-editable` oraz dodać w linii kodu nagłówka atrybut `id="test-heading"`, tak aby można było dołączyć edytor. Na przykład:

```
<h1 class="brand-heading scs-editable" id="test-heading">Sample Heading Value</h1>
```

3. Zapisać plik.
4. Przeprowadzić synchronizację z Oracle Content Management, po czym ponownie załadować przeglądarkę.

Gdy użytkownik umieści wskaźnik myszy na nagłówku, powinna się pojawić czarna ramka otaczająca nagłówek — w ten sposób jest sygnalizowane, że można go edytować. Gdy użytkownik kliknie na nagłówku, ramka zmieni kolor na zielony, sygnalizując że użytkownik może teraz edytować zawartość nagłówka.

5. Opublikować motyw, jeśli jeszcze nie zostało to zrobione. Jeśli motyw został opublikowany, zmiana będzie widoczna, gdy użytkownik odświeży zawartość w przeglądarce.

Gdy motyw zostanie opublikowany, użytkownicy serwisu mogą klikać na nagłówku i edytować go w konstruktorze serwisów.

Dla elementu `scs-editable` jest tworzony pseudoelement, dzięki czemu można go zmieniać w konstruktorze serwisów i przechowywać wraz z danymi strony. W trybie wykonawczym, zanim strona będzie renderowana, kontroler zastąpi znaczniki "scs-editable" wartościami określonymi w konstruktorze serwisów.

## Modyfikowanie znacznika obrazu

Procedura modyfikowania znacznika obrazu, używanego w układzie, tak aby obraz ten mógł być edytowany przez użytkowników, jest podobna do procedury modyfikowania znacznika nagłówka.

1. Dodać do atrybutu `class` właściwość `scs-editable`.
2. Dodać unikatowy `id` obrazu.

Użytkownicy mogą klikać na obrazie, a następnie zmieniać jego właściwości, tak aby był używany inny obraz.

Przykładowy kod, umożliwiający edycję obrazu przez użytkowników:

```

```

## Tworzenie układu sekcji

Tworząc układ sekcji, można розміścić zawartość w gnieździe na stronie serwisu.

Użytkownik firmowy może розміścić elementy zawartości w serwisie na podstawie układów sekcji, które zostały udostępnione przez programistę. Można tworzyć nowe układy sekcji z układu domyślnego.

Układ sekcji można wyeksportować i zmodyfikować offline, a następnie zaimportować go jako nowy układ sekcji albo jako zamiennik istniejącego układu sekcji. Układ sekcji można wyeksportować indywidualnie bądź w ramach pakietu szablonu, zawierającego składniki niestandardowe i układy.

Dostępne są następujące gotowe układy sekcji:


- Poziomy
- Dwie kolumny
- Trzy kolumny
- Pionowy
- Z kartami
- Suwak

Układów tych można użyć od razu w konstruktorze serwisów bez konieczności tworzenia czegokolwiek.

Pliki tych układów sekcji zawierają komentarze z dodatkowymi szczegółami dotyczącymi struktury plików układu sekcji. Aby wyświetlić komentarze, można utworzyć nowy układ sekcji na podstawie gotowego układu, a następnie wyeksportować nowy układ do edycji, zgodnie z poniższą procedurą.

Aby utworzyć układ sekcji, należy:

1. Na stronie startowej wybrać opcję **Programista**.
2. Nacisnąć przycisk **Wyświetl wszystkie składniki**.
3. Z rozwijanego menu **Utwórz** (po prawej stronie) wybrać opcję **Utwórz układ sekcji**.

4. W oknie dialogowym **Tworzenie układu sekcji** podać nazwę opis składnika "układ sekcji".
5. Aby wyeksportować układ sekcji do edycji, wybrać go, po czym wybrać z menu podręcznego opcję **Eksportuj** albo kliknąć na ikonie  (na pasku czynności).
  - a. Przejść do istniejącego folderu albo utworzyć nowy folder: nacisnąć przycisk **Utwórz**, a następnie podać nazwę i opcjonalny opis.
  - b. Zaznaczyć pole wyboru obok folderu, po czym nacisnąć przycisk **OK**.
  - c. Aby otworzyć folder, kliknąć na jego ikonie lub nazwie.

W wybranym folderze zostanie utworzony plik pakietu układu, mający nazwę układu sekcji i rozszerzenie `.zip`. Aby można było edytować pliki, należy pobrać plik do swojego środowiska programistycznego.

Informacje o zestawach API układów sekcji są dostępne pod hasłem [Tworzenie niestandardowych układów sekcji za pomocą API](#).

6. Zaimportować zmodyfikowane pliki jako nowy układ sekcji albo w celu zastąpienia już istniejącego układu sekcji.
  - a. Na stronie startowej kliknąć na karcie **Zawartość**, po czym wybrać opcję **Dokumenty**.
  - b. Wysłać zmodyfikowany układ sekcji do folderu (plik z rozszerzeniem `.zip`), zawierającego te same nazwy folderów i plików, które zostały wyeksportowane.
  - c. Na stronie **Programista** nacisnąć przycisk **Wyświetl wszystkie składniki**.
  - d. Z menu **Utwórz** wybrać opcję **Importuj składnik**.
  - e. Zaznaczyć pole wyboru obok wysyłanego pliku zip, zawierającego zmodyfikowany układ sekcji, po czym nacisnąć przycisk **OK**.

Zmodyfikowany układ sekcji zostanie zaimportowany do wybranego folderu.

Układ można także wyeksportować, aby skopiować go lub przenieść (zaimportować) do innej instancji Oracle Content Management.

## Tworzenie układu sekcji obsługującego skrypty Lazy Load

Lista zawartości może wywoływać układ sekcji z dodatkowymi składnikami, do których odnosi się zapytanie.

Gdy lista zawartości jest renderowana, można wybrać układ sekcji do renderowania wszystkich zwracanych elementów zawartości. Dzięki temu można dla elementów zawartości tworzyć różne układy, takie jak tabela, suwak czy układ ośmiokolumnowy. Te niestandardowe układy sekcji mogą także być wykorzystywane w bardziej zaawansowanych funkcjach paginacji.

Listy zawartości obsługują paginację typu:

- **Paginacja**
- **Ładowanie przy przewijaniu**
- **Ładowanie przy kliknięciu**

W przypadku standardowej funkcji paginacji układ sekcji nie musi nic wykonywać. Zostanie wyrenderowany z następnym zestawem elementów, gdy użytkownik kliknie na łączu następnej strony. Natomiast dla typów **Ładowanie przy przewijaniu** i **Ładowanie przy kliknięciu** nie następuje ponowne renderowanie sekcji, lecz do układu sekcji są dodawane



dotatkowe składniki. Rozwiązanie to jest używane głównie dla modelu z nieograniczonym przewijaniem, kiedy to jest ładowanych  $n$  pierwszych elementów, a następnie — gdy użytkownik przewija stronę — jest ładowany i renderowany następny zestaw elementów. Aby zapewnić obsługę typów **Ładowanie przy przewijaniu** i **Ładowanie przy kliknięciu**, układ sekcji musi:

1. `render.js`: Implementować API `addComponent()`. Interfejs ten będzie wywoływany z każdym nowym składnikiem, który ma zostać dodany do układu sekcji.

```
// dynamic API for adding additional components through "load more"
when used in a Content List
addComponent: function (parentObj, component) {
    // create the component div and add it to the parent
    object
        $
    (parentObj).append(this.createComponentDiv(component));
}
```

2. `appInfo.json`: Dodać poniższy kod informujący listę zawartości, że układ zawartości obsługuje api `addComponent()`.

```
"contentListData": {
  "addComponent": true
},
```

Gdy plik `appInfo.json` zostanie zaktualizowany, wówczas użytkownik — kiedy wybierze ten układ sekcji w panelu ustawień i przejdzie do ekranu paginacji — zobaczy opcje **Ładowanie przy kliknięciu** i **Ładowanie przy przewijaniu**.

## Tworzenie niestandardowych układów sekcji za pomocą API

W usłudze Oracle Content Management można tworzyć niestandardowe układy sekcji za pomocą Section Layout API, zawierającego Rendering API i Editing API.

Pliki z zestawu początkowego, którym warto się przyjrzeć, są omówione pod hasłem [Tworzenie układu sekcji](#). Pliki z zestawu początkowego, przeznaczone dla układów sekcji, zawierają szczegółowe komentarze opisujące strukturę plików układu sekcji.

### Rendering API

Rendering API, ładowane z modułu `render.js`, są używane w konstruktorze serwisów oraz w trybie wykonawczym.

Rendering API	Opis	Parametry wejściowe	Wynik po wykonaniu
<b>(Constructor)</b>	Inicjalizuje moduł renderowania układu sekcji.	<p>Obiekt JavaScript zawierający następujące właściwości:</p> <ul style="list-style-type: none"> <li>• <b>sectionlayoutData (obiekt)</b> : Dane układu sekcji wykryte w modelu strony.</li> <li>• <b>componentId (napis)</b> : Wartość componentId układu sekcji (zazwyczaj GUID).</li> <li>• <b>renderMode (napis; opcjonalne)</b>: Tryb renderowania podczas operacji renderowania.</li> <li>• <b>customSettingsData (obiekt)</b>: Kopia właściwości customSettingsData zawartej w danych sectionLayoutData</li> </ul>	Są inicjalizowane Rendering API służące do renderowania układu sekcji.
<b>render</b>	Emituje do strony elementy DOM odpowiednie dla układu sekcji, w tym elementy "container DIV" dla składników podrzędnych.	<b>container (element)</b> : Element DOM, w którym będzie renderowany kod znacznikowy układu sekcji.	Gdy ta metoda zostanie wykonana, będą renderowane składniki podrzędne. Składniki podrzędne można zidentyfikować, odnajdując podrzędne elementy div[id]
<b>addComponent</b>	Używana ze składnikami "lista zawartości" do dynamicznego dodawania składników podrzędnych do układu sekcji. Ta funkcja jest opcjonalna.	<p><b>container (element)</b> : Element DOM, w którym będzie renderowany nowy składnik.</p> <p><b>componentId (napis)</b>: ID nowego składnika dodawanego do układu sekcji.</p>	Gdy ta metoda zostanie wykonana, będzie renderowany element, którego ID jest zgodny z wejściową wartością componentId

### Editing API

Moduł `edit.js` jest ładowany, *jeśli* właściwość `hasEditHandlers` jest ustawiona na wartość `true` w pliku `appinfo.json` powiązany z układem sekcji.

Editing API są używane w konstruktorze serwisów.

Wszystkie funkcje w tym module, z wyjątkiem funkcji `Constructor`, są opcjonalne.

Editing API	Opis	Parametry wejściowe	Wynik po wykonaniu
<b>(Constructor)</b>	Inicjalizuje moduł edycji układu sekcji.	Obiekt JavaScript zawierający następującą właściwość: <ul style="list-style-type: none"><li>• <b>componentId (napis):</b> Wartość componentId układu sekcji (zazwyczaj GUID).</li></ul>	Są inicjalizowane Editing API służące do edytowania układu sekcji.

Editing API	Opis	Parametry wejściowe	Wynik po wykonaniu
<b>getCapabilities</b>	Zwraca obiekt opisujący możliwości edytowania układu sekcji.	<p>Obiekt JavaScript opisujący możliwości edytowania układu sekcji.</p> <p>Po wprowadzeniu, do funkcji zostaną przekazane możliwości domyślne. Jeśli trzeba, funkcja może modyfikować obiekt Capabilities.</p> <p>Obiekt Capabilities może zawierać następujące możliwości:</p> <ul style="list-style-type: none"> <li>• <b>title (napis):</b> Tytuł układu sekcji wyświetlany użytkownikowi.</li> <li>• <b>settingsTitle (napis):</b> Tytuł wyświetlany w oknie dialogowym "Panel ustawień".</li> <li>• <b>hasSettings (wartość logiczna):</b> Sygnalizuje, czy panel ustawień jest obsługiwany przez układ sekcji.</li> <li>• <b>allowMove (wartość logiczna):</b> Sygnalizuje, czy układ sekcji zezwala na przenoszenie elementów podrzędnych.</li> <li>• <b>allowDelete (wartość logiczna):</b> Sygnalizuje, czy układ sekcji zezwala na usuwanie elementów podrzędnych.</li> <li>• <b>isHidden (wartość logiczna):</b> Sygnalizuje, czy układ sekcji jest obecnie ukrywany w odpowiedzi na opcje użytkownika.</li> <li>• <b>dropTarget (wartość logiczna):</b> Sygnalizuje, czy układ sekcji jest celem dla operacji "przeciągnij i upuść".</li> <li>• <b>customMenuOptions (tablica):</b> Niestandardowe opcje menu dodawane do podręcznego menu układu sekcji. Każdy obiekt menu-option ma następujące właściwości: <ul style="list-style-type: none"> <li>– <b>label (etykieta) :</b> Wyświetlany tekst opcji menu.</li> <li>– <b>action (funkcja):</b> Funkcja wywoływana, gdy</li> </ul> </li> </ul>	(Obiekt): Możliwości układu sekcji.

Editing API	Opis	Parametry wejściowe	Wynik po wykonaniu
		<p>użytkownik wybierze tę opcję menu.</p> <ul style="list-style-type: none"> <li>– <b>disabled (wartość logiczna):</b> Sygnalizuje, czy opcja menu ma być wyświetlana w stanie wyłączenia.</li> <li>– <b>icon (etykieta) :</b> URL wyświetlany obok etykiety opcji menu. (Ta właściwość jest zarezerwowana do użycia w przyszłości.)</li> <li>– <b>checkmark (etykieta) :</b> Sygnalizuje, czy obok etykiety opcji menu powinien być wyświetlany znacznik.</li> <li>– <b>subMenuItems (tablica):</b> Opcje wyświetlane w podmenu.</li> </ul>	
<b>getCaptionContent</b>	Zwraca wyświetlaną nazwę układu sekcji, która to nazwa będzie się pojawiać w elementach UI.	Brak.	(Napis): Wyświetlana nazwa układu sekcji.
<b>filterCapabilities</b>	Umożliwia układowi sekcji modyfikowanie obiektu <code>Capabilities</code> , zanim menu zostaną wyświetlone użytkownikowi. Tego API można używać do dostosowywania a lub usuwania opcji menu. (Zob. także: <b>getCapabilities</b> .)	Obiekt JavaScript opisujący możliwości użytkownika układu sekcji. Po wprowadzeniu, do funkcji zostaną przekazane możliwości domyślne.	(Obiekt): Możliwości układu sekcji.

Editing API	Opis	Parametry wejściowe	Wynik po wykonaniu
<b>onDragOver</b>	Funkcja wywoływana podczas operacji "przeciągnij i upuść", służąca do zasygnalizowania, czy przeciągany element można przeciągnąć do układu sekcji.	<b>eventObject (obiekt event):</b> Obiekt event przechowujący informacje o zdarzeniu przeciągania. <b>dataTransfer (obiekt DataTransfer)</b> : Obiekt DataTransfer przechowujący informacje o elemencie przeciąganym do układu sekcji.	(Wartość logiczna): Wartość sygnalizująca, czy przeciągany element może zostać przyjęty przez układ sekcji. Jeśli przeciągany element może zostać przyjęty przez układ sekcji, jest zwracana wartość <i>true</i> ; w przeciwnym razie — wartość <i>false</i> .
<b>onDrop</b>	Funkcja wywoływana podczas czynności "upuść" z operacji "przeciągnij i upuść", służąca do zasygnalizowania, czy przeciągany element powinien zostać umieszczony w układzie sekcji.	<b>eventObject (obiekt event):</b> Obiekt event przechowujący informacje o zdarzeniu upuszczenia. <b>dataTransfer (obiekt DataTransfer):</b> Obiekt DataTransfer przechowujący informacje o elemencie upuszczanym do układu sekcji.	(Wartość logiczna): Wartość sygnalizująca, czy operacja upuszczania została obsłużona przez układ sekcji. Jeśli jest zwraca wartość <i>true</i> , logika domyślna jest pomijana.
<b>onAddComponent</b>	Powiadamia układ sekcji, że element został — wskutek operacji "przeciągnij i upuść" — dodany do układu sekcji.	<b>eventObject (obiekt event):</b> Obiekt event przechowujący informacje o zdarzeniu przeciągania. <b>dataTransfer (obiekt DataTransfer):</b> Obiekt DataTransfer przechowujący informacje o elemencie upuszczanym do układu sekcji. <b>componentId (napis)</b> : Wartość <code>componentId</code> nowo dodanego elementu.	Powiadomienie układu sekcji.

Editing API	Opis	Parametry wejściowe	Wynik po wykonaniu
<b>onMoveComponent</b>	Powiadamia układ sekcji, że element został — wskutek operacji "przeciągnij i upuść" — przeniesiony do układu sekcji.	<b>eventObject (obiekt event):</b> Obiekt event przechowujący informacje o zdarzeniu przeciągania. <b>dataTransfer (obiekt DataTransfer):</b> Obiekt DataTransfer przechowujący informacje o elemencie upuszczanym do układu sekcji. <b>componentId (napis)</b> : Wartość componentId przeniesionego elementu.	Powiadomienie układu sekcji.
<b>getSettingsData</b>	Umożliwia układowi sekcji zmianę danych dla ustawień, zanim zostanie wyświetlony panel ustawień.	<b>settingsData (obiekt):</b> Domyślne dane ustawień, obliczone dla układu sekcji.	(Obiekt): Dane ustawień układu sekcji.
<b>updateSettings</b>	Umożliwia układowi sekcji zmianę swoich ustawień po zamknięciu panelu ustawień. Ten API jest wywoływany tuż przed składowaniem ustawień w modelu strony.	<b>parameters (obiekt):</b> Obiekt "raw parameters" zwracany z panelu ustawień. <b>sectionLayoutData (obiekt) :</b> Dane układu sekcji, które będą składowane. Dane domyślne zostaną wygenerowane z metody <code>parameters (obiekt)</code> i przekazane w tym parametrze do funkcji.	(Obiekt): Dane układu sekcji do przechowania w modelu strony.

Editing API	Opis	Parametry wejściowe	Wynik po wykonaniu
<b>dispose</b>	Umożliwia modułowi edycyjnemu zwalnianie pamięci, odłączanie zdarzeń oraz dezalokację zasobów powiązanych z procedurami obsługi edycji.  Ten API jest wywoływany, gdy trzeba całkowicie odświeżyć układ sekcji, na przykład w przypadku operacji Cofnij/ Ponów.	Brak.	Odświeżenie układu sekcji jest włączone.

## Tworzenie układów zawartości

Układy zawartości ułatwiają użytkownikom wyświetlanie danych w elementach zawartości za pomocą składników "lista zawartości" lub "element zastępczy zawartości", używanych na stronach serwisów. Tworząc różne układy zawartości dla typu zawartości, można utworzyć różne widoki lub zaprezentować różne części elementu zawartości.

Na przykład typ zawartości "Blog-Post" może wymagać różnych układów zawartości w zależności od przewidywanego sposobu i miejsca używania zawartości "Blog-Post". Na stronie startowej serwisu może być wyświetlana lista elementów "Blog-Post" i — gdy użytkownik kliknie na którymś z nich — zostanie wyświetlona strona **Szczegóły** wpisu na blogu.

Na stronie startowej znajduje się listę zawartości skonfigurowana tak, aby wyświetlała — za pomocą układu zawartości "Blog-Post-Summary" służącego za widok elementów — elementy zawartości typu "Blog-Post".

Strona "Szczegóły" do wyświetlania obrazu i tytułu nagłówka używa układu zawartości "Blog-Post-Header" (w elemencie zastępczym zawartości). Dwukolumnowy układ sekcji obejmuje dwa elementy zastępcze zawartości (o szerokości 70% i 30%), używające układów zawartości "Blog-Post-Content" i "Blog-Post-Author". Jak widać, do wizualizacji tego samego typu zawartości można użyć czterech różnych układów zawartości.

Układ zawartości można utworzyć na dwa sposoby:

- W usłudze Oracle Content Management wybrać kolejno opcje **Programista > Wyświetl wszystkie składniki > Utwórz > Utwórz układ zawartości**.
- W projekcie OCE Toolkit użyć polecenia `cec create-contentlayout`.

Oracle Content Management tworzy domyślny układ zawartości dla typu zawartości. Aby zmodyfikować domyślny układ zawartości, należy zmodyfikować następujące pliki:



- `assets/layout.html`  
Należy edytować ten plik, aby zmienić widok HTML.
- `assets/design.css`  
Należy edytować ten plik, aby nadać styl układowi zawartości.
- `assets/render.js`  
Należy edytować ten plik, aby zmienić dane używane w pliku `layout.html` lub zapewnić dynamiczne działanie układu zawartości.

Składniki "układ zawartości" renderują elementy zawartości z serwera Oracle Content Management. Większość zasobów jest przechowywanych na serwerze Oracle Content Management. Niekiedy trzeba jednak użyć zasobu statycznego, który jest dostępny lokalnie w samym układzie zawartości, taki jak obraz używany na tło. Na przykład w poniższym układzie zawartości bezwzględny URL obrazu `images/background.jpg` można wygenerować w pliku `render.js`, a następnie użyć w pliku `layout.html`.

Najprostszym sposobem generowania bezwzględnego adresu URL jest użycie wbudowanego pliku `require.js`. Należy zdefiniować 'require' jako zależność, a następnie wygenerować URL za pomocą funkcji `require.toURL()`, jak pokazano w następującym kodzie:

```
define([
  'require',
  'jquery',
  'mustache',
  'text!./layout.html',
  'css!./design.css'
], function (require, $, Mustache, templateHtml, css) {
  'use strict';

  var imageURL = req.toUrl('./images/background.jpg');
```

W poniższych rozdziałach opisano, w jaki sposób tworzyć układy zawartości:

- [Tworzenie układów zawartości za pomocą Oracle Content Management](#)
- [Przekazywanie widoku układu do układu zawartości](#)
- [Generowanie adresu URL strony ze szczegółami serwisu przy użyciu API](#)
- [Tworzenie układów zawartości lokalnie za pomocą usługi Developer Cloud Service](#)
- [Rozwijanie makra w zapytaniach dotyczących listy zawartości](#)
- [Tworzenie odpornych układów zawartości](#)
- [Tworzenie przykładowego szablonu bloga](#)
- [Dodawanie odwzorowań układu zawartości do szablonów](#)
- [Testowanie układów zawartości przy użyciu lokalnego zestawu testów](#)
- [Importowanie szablonów z układami zawartości do Oracle Content Management](#)

#### Tematy pokrewne

Aby uzyskać informacje o zarządzaniu układami zawartości, zob. [Zarządzanie niestandardowymi składnikami i układami](#).

Informacje o używaniu zasobów cyfrowych i innych elementów zawartości w serwisie są dostępne w rozdziałach Korzystanie z zasobów i Zarządzanie zasobami cyfrowymi w podręczniku *Zarządzanie zasobami przy użyciu Oracle Content Management*.

## Tworzenie układów zawartości za pomocą Oracle Content Management

Można utworzyć układ zawartości służący do układania pól w elemencie zawartości wyświetlanym na stronie serwisu. Jeśli element zawartości zostanie dodany do strony, będzie on używał wybranego układu zawartości.

Użytkownik firmowy może tworzyć i korzystać z elementów zawartości na podstawie udostępnionych przez programistę typów zawartości i układów. Można tworzyć nowe układy zawartości z układu domyślnego. Wiele układów zawartości powiązanych z określonym typem zawartości umożliwia projektantowi serwisów wyświetlanie elementów zawartości w różnych kontekstach, bez zmieniania zawartości.

Jeśli zostanie użyty układ zawartości ze składnika "lista zawartości", układ ten zostanie powtórzony raz na element zawartości. Układy zawartości są następnie rozmieszczane przez układ sekcji.

Układ zawartości można wyeksportować i zmodyfikować offline, a następnie zaimportować go jako zamiennik istniejącego układu zawartości.

Aby utworzyć układ zawartości, należy:

1. Na stronie startowej Oracle Content Management nacisnąć przycisk **Programista** (boczna nawigacja).  
Zostanie wyświetlona strona **Programista**.
2. Nacisnąć przycisk **Wyświetl wszystkie składniki**.
3. Z rozwijanego menu **Utwórz** (po prawej stronie) wybrać opcję **Utwórz układ zawartości**.
4. W oknie dialogowym **Tworzenie układu zawartości** wybrać typy zawartości, które będą używały tego układu, wybrać wyświetlane pola oraz włączyć opcję **Dodaj obsługę ustawień niestandardowych, gdy są one używane w serwisach** (aby w kreatorach serwisów można było — podczas dodawania składników "element zawartości" i "lista zawartości" do strony — dodawać niestandardowe ustawienia i style).

### Create Content Layout ✕

**Choose a Content Type**  
*Create a content layout to display items of this type*

Choose a content type ▾

**Choose Fields to Display**

Overview ▾

Add support for custom settings when used in Sites

---

**Provide a name for your content layout**  
*Use only letters, numbers, hyphens, and underscores in content layout names.*

Add a name for this content layout

**Provide a description for your content layout (optional)**

Add an optional description for this content layout


Cancel Create

5. Podać nazwę i opis składnika "układ zawartości", po czym nacisnąć przycisk **Utwórz**.

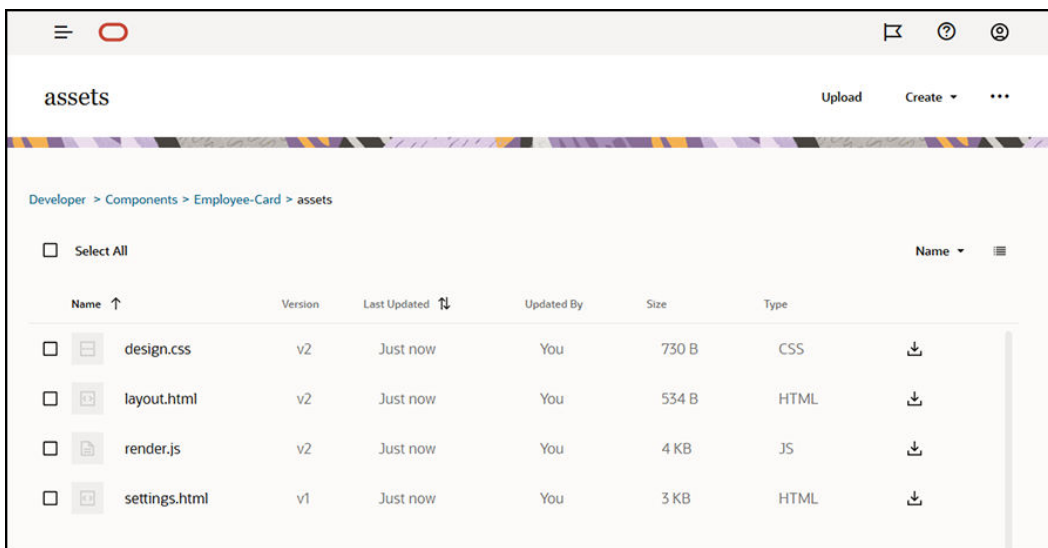
Układ zawartości zostanie dodany do składników.

 **Uwaga:**

Tytuły układów zawartości mogą zawierać tylko znaki alfanumeryczne, łączniki i znaki podkreślenia.

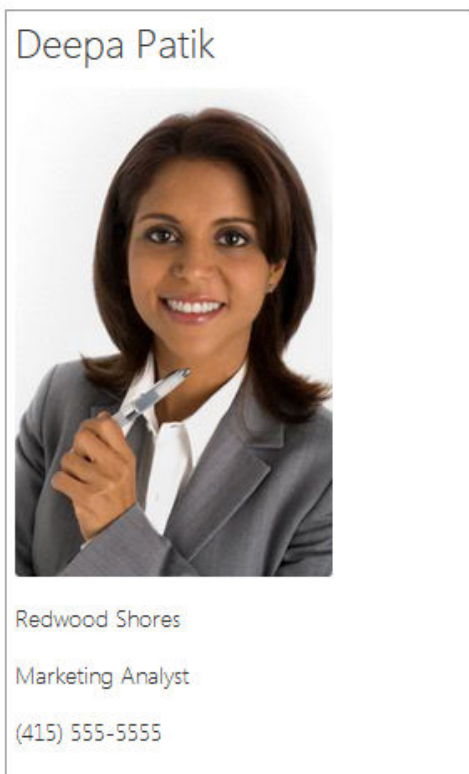
6. Aby wyeksportować układ zawartości do edycji, wybrać go, po czym wybrać z menu podręcznego opcję **Eksportuj** albo kliknąć na ikonie  (na pasku czynności).
  - a. Przejść do istniejącego folderu albo utworzyć nowy folder: wybrać kolejno opcje **Utwórz** i **Folder**, a następnie podać nazwę i opcjonalny opis.
  - b. Zaznaczyć pole wyboru obok folderu, po czym nacisnąć przycisk **OK**.
  - c. Aby otworzyć folder, kliknąć na jego ikonie lub nazwie.

W wybranym folderze zostanie utworzony plik pakietu układu mający nazwę układu zawartości i rozszerzenie `.zip`. Aby można było edytować pliki, należy pobrać plik do swojego środowiska programistycznego.

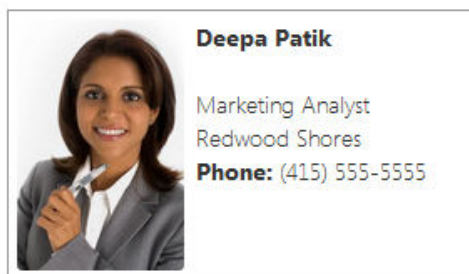


Pliki te kontrolują układ pól w elementach zawartości używających układu zawartości "Employee-Card". Jeśli zostanie włączona opcja **Dodaj obsługę ustawień niestandardowych, gdy są one używane w serwisach**, zostanie także utworzony dodatkowy plik o nazwie **settings.html**, udostępniający domyślny sposób renderowania elementu zawartości, dzięki czemu może być on wyświetlany.

## Default Layout



## Card Layout



7. Edytować pliki `design.css`, `layout.html` i `render.js`, tak aby uzyskać odpowiedni układ zawartości.

Na przykład poniższe pliki określają układ zawartości "Employee-Card":

- a. Edytować plik `design.css`:

```
.scs-tile-layout {
  font-family: 'Helvetica Neue', 'Segoe UI', sans-serif-
regular, Helvetica, Arial;
  font-size: 16px;
  margin: 0px;
  padding: 0px;
  font-style: normal;
  color: #333;
}

.scs-tile-layout li {
  list-style: none;
  font-size: 14px;
  font-style: normal;
  font-variant-caps: normal;
  font-weight: 200;
  margin: 0px;
}
```

```
.scs-tile-layout-img-container {
    height: 150px;
    width: 100px;
    float: left;
    margin: 0em 0.5em 0em 0em;
    padding: 0px;
    border-radius: 3px;
    overflow: hidden;
    position: relative;
}

.scs-tile-layout-img {
    position: absolute;
    left: -100%;
    right: -100%;
    top: -100%;
    bottom: -100%;
    margin: auto;
    height: 100%;
    min-width: 100%;
}

.scs-tile-layout p {
    margin: 0px;
}
```

**b.** Edytować plik `layout.html`:

```
{{#data.employee_profile_pictureURL}}
<div class="scs-tile-layout-img-container">
    
</div>
{{/data.employee_profile_pictureURL}}
<ul class="scs-tile-layout">
    <li>
        <p><b>{{name}}</b></p>
    </li>
    <li>&nbsp;</li>
    <li>
        <p>{{data.employee_job_title}}</p>
        <p>{{data.employee_location}}</p>
        <p><b>Phone: </b>{{data.employee_phone}}</p>
    </li>
    {{#scsData.detailPageLink}}
    <li>
        <a href="{{scsData.detailPageLink}}" title="Go to detail
        page"><span class="detail-page">Profile</span></a>
    </li>
    {{/scsData.detailPageLink}}
</ul>
```

c. Edytować plik `render.js`:

```
/* globals define */

define([
  'jquery',
  'mustache',
  'text!./layout.html',
  'css!./design.css'
], function($, Mustache, templateHtml, css) {
  'use strict';

  function ContentLayout(params) {
    this.contentItemData = params.contentItemData || {};
    this.scsData = params.scsData;
    this.contentClient = params.contentClient ||
      (params.scsData ? params.scsData.contentClient : null);
    // retrieve the custom settings
    this.customSettings = this.scsData.customSettingsData
      || {};
  }

  ContentLayout.prototype = {

    render: function(parentObj) {
      var template,
          content = $.extend({}, this.contentItemData),
          contentClient = this.contentClient,
          contentType,
          secureContent = false;

      if (this.scsData) {
        content = $.extend(content, { 'scsData':
this.scsData });
        contentType =
content.scsData.showPublishedContent === true ? 'published' :
'draft';
        secureContent = content.scsData.secureContent;
      }

      if (contentClient) {
        var params = {
          'itemGUID': typeof
content.data.employee_profile_picture === 'string' ?
          content.data.employee_profile_picture :
content.data.employee_profile_picture.id,
          'contentType': contentType,
          'secureContent': secureContent
        };
        content.data.employee_profile_pictureURL =
contentClient.getRenditionURL(params);
      }

      console.log(content);
    }
  };
}
```

```
        try {
            // Mustache
            template = Mustache.render(templateHtml, content);

            if (template) {
                $(parentObj).append(template);
            }
        } catch (e) {
            console.error(e.stack);
        }
    }
};

return ContentLayout;
});
```

Informacje o edytowaniu pliku `render.js` i innych plików są dostępne pod hasłem [Tworzenie składników](#).

8. Aby zastąpić istniejący układ zawartości, zaimportować zmodyfikowane pliki.
  - a. Na stronie startowej wybrać opcję **Dokumenty**.
  - b. Wysłać zmodyfikowany układ zawartości do folderu (plik z rozszerzeniem **.zip**), zawierającego te same nazwy folderów i plików, które zostały wyeksportowane.  
Aby zaimportować układ zawartości jako nowy, trzeba w pliku `_folder.json` zmienić GUID układu zawartości.
  - c. Na stronie startowej wybrać opcję **Programista**.  
Zostanie wyświetlona strona **Programista**.
  - d. Nacisnąć przycisk **Wyświetl wszystkie składniki**.
  - e. Z menu **Utwórz** wybrać opcję **Importuj składnik**.
  - f. Zaznaczyć pole wyboru obok przesłanego pliku zip, zawierającego zmodyfikowany składnik, po czym nacisnąć przycisk **OK**.

Zmodyfikowany układ zawartości zostanie zaimportowany do **składników**.

Można także wyeksportować układ zawartości, aby skopiować go lub przenieść (zaimportować) do innej instancji Oracle Content Management.

## Przekazywanie widoku układu do układu zawartości

Tworząc układ zawartości, można uzyskać właściwości używanego składnika, jeśli układ zawartości jest używany w serwisach.

Dla tego konkretnego przypadku można uzyskać widok układu zawartości wybranego dla typu zawartości z listy zawartości. Następnie można zmienić sposób renderowania składnika w zależności od jego kategorii. Nie mając dostępu do tej właściwości, należy utworzyć dwa układy zawartości, które będą identyczne.



Właściwość `contentLayoutCategory` jest dostępna dla układów zawartości renderowanych zarówno dla składników "element zawartości", jak i "lista zawartości". Dostęp do tej właściwości można uzyskać poprzez narzędzie Sites SDK:

```
scsData.SitesSDK.getProperty('contentLayoutCategory', function  
(layoutCategory)  
    { console.log(layoutCategory); });
```

Zob. Oracle Content Management SDK.

Ta właściwość jest dostępna tylko wtedy, gdy układ zawartości jest renderowany za pomocą składnika "element zawartości" lub "lista zawartości", znajdującego się w serwisie Oracle Content Management. Nie jest ona dostępna, gdy układ zawartości jest renderowany z innych aplikacji.

## Generowanie adresu URL strony ze szczegółami serwisu przy użyciu API

Renderując w układzie zawartości listę elementów zawartości uzyskaną za pomocą własnego zapytania, można utworzyć dla elementu zawartości łącze prowadzące do strony szczegółów. Do wygenerowania adresu URL strony **Szczegóły serwisu** można użyć API `SCSRenderAPI.getPageLinkData` z Sites SDK.

Właściwość `detailPageId` określa identyfikator strony szczegółów wybranej z elementu zawartości lub z listy zawartości. Jeśli ta wartość nie zostanie określona, zwracaną wartością jest pierwsza strona z obiektu `SiteStructureMap` mająca ustawioną właściwość `isDetailPage`.

Dostęp do tej właściwości można uzyskać za pomocą Sites SDK. Jest ona dostępna tylko wtedy, gdy układ zawartości jest używany dla serwisu Oracle Content Management. Można uzyskać do niej dostęp tylko przez `scsData`, przekazywane jako jeden z argumentów, gdy jest tworzony układ zawartości. Na przykład:

```
scsData.SitesSDK.getProperty('detailPageId', function (detailPageId)  
{ console.log(detailPageId); });
```

Mając `detailPageId`, można go użyć do skonstruowania łącza do strony szczegółów.

`SCSRenderAPI` zawiera funkcję `getPageLinkData()`, która przyjmuje `pageId` wraz z dodatkowymi opcjami i konstruuje wymagany URL dla strony przekazywanej przez opcje. Funkcja ta ma następującą postać:

```
SCSRenderAPI.getPageLinkData(pageId,  
    options);
```

Zawiera następujące parametry:

- **pageId:** Identyczny z `detailPageId` zwracanym z właściwości `detailPageId` z Sites SDK.
- **options:**
  - `contentType`
  - `contentId`

– contentName

Jest zwracany obiekt z następującymi właściwościami:

- hideInNavigation
- href
- href

Wszystko to jest przedstawione w poniższym przykładzie:

```
scsData.SitesSDK.getProperty('detailPageId', function (detailPageId) {
  var pageDetails = SCSRenderAPI.getPageLinkData(pageId, {
    'contentType': contentType,
    'contentId': contentId,
    'contentName': contentItemData.slug || contentItemData.name
  });
```

```
  // get the URL to the page
  console.log(pageDetails.href);
});
```

Wynikiem będzie: `"/sites/{site}/{detailPageName}/{contentType}/{contentId}/{contentSlug}"`

Jeśli `pageId` nie jest identyfikatorem strony szczegółów, wartości "content" nie są dodawane do adresu URL.

Zob. Oracle Content Management SDK.

## Tworzenie układów zawartości lokalnie za pomocą usługi Developer Cloud Service

Za pomocą usługi Developer Cloud Service można — poprzez zestaw OCE Toolkit — lokalnie tworzyć, edytować, konfigurować i testować układy zawartości dla usługi Oracle Content Management.

W celu przygotowania się do tworzenia układów zawartości przy użyciu usługi Developer Cloud Service, należy wykonać następujące zadania:

1. [Konfigurowanie zestawu OCE Toolkit na komputerze lokalnym.](#)
2. [Logowanie do konsoli Developer Cloud Service dla Oracle Content Management](#)
3. [Tworzenie projektu w usłudze Developer Cloud Service.](#)
4. [Dodawanie zestawu OCE Toolkit do kodu projektu w nowym repozytorium Git.](#)
5. [Tworzenie układu zawartości](#)

W poniższych tematach opisano, jak używać zestawu OCE Toolkit do tworzenia układu zawartości lokalnie za pomocą usługi Developer Cloud Service:

- [Tworzenie układu zawartości za pomocą usługi Developer Cloud Service](#)
- [Definiowanie modułu RequireJS](#)
- [Konfiguracja parametru funkcji Constructor](#)

- [Renderowanie układu zawartości](#)
- [Edytowanie układu zawartości w szablonie Mustache](#)
- [Dodawanie funkcji dynamicznej manipulacji modelem DOM](#)
- [Definiowanie stylów w pliku design.css](#)
- [Uzyskiwanie elementów referencyjnych](#)
- [Uzyskiwanie adresu URL zasobu multimedialnego](#)
- [Uaktywnianie wyzwalaczy](#)
- [Przechodzenie do strony wyszukiwania za pomocą zapytania wyszukiwania](#)
- [Rozwijanie makra i renderowanie tekstu formatowanego](#)
- [Łącze do strony "Szczegóły"](#)

Po utworzeniu układu zawartości za pomocą szablonu Developer Cloud Service, należy w repozytorium Git projektu [scalić zmiany](#).

## Tworzenie układu zawartości za pomocą usługi Developer Cloud Service

Używając polecenia `cec create-contentlayout`, można w projekcie usługi Developer Cloud Service utworzyć układ zawartości dla typu zawartości.

Aby utworzyć układ zawartości w projekcie Developer Cloud Service, należy:

1. W oknie terminala przejść do katalogu `cec-components`.
2. Wpisać polecenie `cec create-contentlayout` w celu wyświetlenia opcji i przykładów polecenia:

```
Usage: cec create-contentlayout <name>
```

```
Creates a content layout based on a content type from a local
template or from CEC server.
By default, an "overview" content layout is created. Optionally
specify -s <style>
to create in a different style.
```

Valid values for <style> are:

```
detail
overview
```

Options:

```
--contenttype, -c <contenttype> Content layout is based
on [required]

--template, -t <template> Content type is from
--server, -r flag to indicate the content type is from
server
--style, -s <style> Content layout style
--addcustomsettings, -a Add support for custom settings when
used in Sites
--help, -h Show
help [boolean]
```

Examples:

```
cec create-contentlayout Blog-Post-Overview-Layout -c Blog-Post -t
BlogTemplate
cec create-contentlayout Blog-Post-Detail-Layout -c Blog-Post -t
BlogTemplate -s detail
cec create-contentlayout Blog-Post-Overview-Layout -c Blog-Post -t
BlogTemplate -a
cec create-contentlayout Blog-Post-Overview-Layout -c Blog-Post -r
cec create-contentlayout Blog-Post-Overview-Layout -c Blog-Post -r -s
detail
```

3. Aby wyświetlić dostępne typy zawartości na serwerze, wpisać następujące polecenie:

```
cec list-server-content-types
```

4. Jak pokazano w akapicie "Usage", można utworzyć układ zawartości dla typu zawartości na serwerze Oracle Content Management albo dla typu zawartości z szablonach znajdujących się w folderze `cec-components/src/main/templates`. Na przykład za pomocą następującego polecenia można utworzyć na serwerze układ zawartości dla danego typu:

```
cec create-contentlayout Blog-Post-Overview-Layout -c Blog-Post -r
```

Aby zmodyfikować układ zawartości, można edytować następujące pliki:

- `assets/layout.html`

Ten plik określa widok HTML. Zob. [Edytowanie układu zawartości w szablonie Mustache](#).

- `assets/design.css`

Ten plik określa styl układu zawartości. Zob. [Definiowanie stylów w pliku design.css](#).

- `assets/render.js`

Ten plik określa dane używane w pliku `layout.html` oraz umożliwia dodawanie dynamicznego działania do układu zawartości. Zob. [Definiowanie modułu RequireJS](#).

Jeśli zostanie utworzony układ, który zezwala na ustawienia niestandardowe (przy użyciu opcji `-a` podczas tworzenia układu), zostanie także utworzony dodatkowy plik o nazwie `settings.html`, udostępniający domyślny sposób renderowania elementu zawartości, dzięki czemu może być on wyświetlany.

## Definiowanie modułu RequireJS

W pliku `render.js` można zdefiniować moduł RequireJS. System Sites ładuje zależności, takie jak JQuery, Mustache, wtyczka RequireJS "Text" i wtyczka RequireJS "CSS".

```
define([
    'jquery',
    'mustache',
    'text!./layout.html',
    'css!./design.css'
], function ($, Mustache, templateHtml, css) {
```

Do renderowania układów można także używać systemu szablonów Mustache.

Plik `assets/render.js` dla układu zawartości ma następujące właściwości:

- Powinien być modulem RequireJS.
- Powinien zwracać funkcję JavaScript "Constructor". System Sites wywołuje funkcję "Constructor", przekazując obiekt parametrów. Obiekt parametrów zawiera dane elementu zawartości oraz API wymagane do renderowania układu.
- Funkcja "Constructor" powinna mieć metodę `render(parentObj)` obsługującą układ zawartości. Powinna dołączać obiekt DOM układu zawartości do obiektu `parentObj` przekazywanego do metody `render()`.
- Moduł RequireJS może korzystać z zależności, takich jak JQuery, Mustache, wtyczka RequireJS "Text" i wtyczka RequireJS "CSS". Zależności te są ładowane przez serwisy. Można także używać innych bibliotek.

## Konfiguracja parametru funkcji Constructor

Gdy system Sites tworzy nową instancję funkcji Constructor, przekazuje — w celu ułatwienia tworzenia układu zawartości — parametr zawierający obiekty `contentItemData`, `scsData` i `contentClient`.

Przykład funkcji Constructor:

```
function ContentLayout(params) {
    this.contentItemData = params.contentItemData || {};
    this.scsData = params.scsData;
    this.contentClient = params.contentClient;
}
ContentLayout.prototype = {
    render: function (parentObj) {
        var content = {
            blogTitle: this.contentItemData.data['starter-blog-
post_title'],
        };

        if (this.scsData) {
            content = $.extend(content, {
                'scsData': this.scsData
            });
        }

    }
};
return ContentLayout;
```

Parametr funkcji Constructor zawiera następujące obiekty:

- **params.contentItemData:** Zawiera element zawartości, w tym jego nazwę, opis, ID i dane. Na przykład dostęp do pola `blogpost_title`, występującego w elemencie zawartości, można uzyskać używając `params.contentItemData.data['blogpost_title']`.
- **params.scsData:** Ten obiekt przekazuje informacje, gdy funkcja Constructor jest wywoływana w systemie Sites. Obiekt ten nie istnieje dla układów zawartości renderowanych w aplikacjach podmiotów trzecich. Zawiera obiekt Sites SDK, metodę `contentTriggerFunction` do uaktywniania wyzwalacza oraz łączy do strony **szczegółów**.

- **params.contentClient:** Jest to obiekt `contentClient` tworzony z pakietu Content SDK, używany do wywołania układu zawartości. Z tego powodu jest skonfigurowany z użyciem odpowiednich parametrów serwera zawartości. Jeśli trzeba skierować dodatkowe wywołania do serwera zawartości, można — zamiast tworzyć własny obiekt — użyć tego obiektu `contentClient`. Obiekt ten zawiera API klienta odnoszące się do zawartości. Za ich pomocą można kierować zapytania, wyszukiwać oraz uzyskiwać elementy zawartości i ich typy zawartości. Dostępne są także inne wspomagające API; na przykład `expandMacros()` do rozwijania makr używanych w tekście z formatowaniem.

## Renderowanie układu zawartości

Metoda `render(parameters)` z obiektu `ContentLayout` renderuje układ zawartości z szablonu. Dla układów zawartości domyślnie jest używany szablon Mustache, lecz można użyć szablonu opartego na innej dowolnej technologii.

Metoda `render(parameters)` z obiektu `ContentLayout` może użyć następującego kodu do renderowania szablonu z użyciem danych:

```
try {
  // Mustache
  template = Mustache.render(templateHtml, content);
  if (template) {
    $(parentObj).append(template);
  }

  // Dynamic DOM Manipulation can be done here
} catch (e) {
  console.error(e.stack);
}
```

Wymagane dane można dodać do obiektu `content` tworzonoego z `params.contentItemData`. Oracle zaleca scalenie — w tym obiekcie — właściwości z obiektu `params.scsData`, dzięki czemu szablon będzie mógł z nich korzystać. Renderowany szablon powinien zostać dołączony do nadrzędnego obiektu przekazywanego do metody `render()`.

## Edytowanie układu zawartości w szablonie Mustache

W pliku `assets/layout.html` można edytować domyślny szablon Mustache, który zawiera domyślny układ zawartości.

Przykład prostego szablonu do renderowania tytułu bloga:

```
<h1>
  {{blogTitle}}
</h1>
```

## Dodawanie funkcji dynamicznej manipulacji modelem DOM

Do pliku `render.js` można dodać — po wywołaniu metody `Mustache.render()` — funkcję dynamicznej manipulacji modelem DOM, a do obiektu nadrzędnego zostanie dołączony szablon.

Na przykład można, dynamicznie zmieniając styl, dołączyć nasłuch:

```
// Dynamic DOM Manipulation can be done here
$('hl').click(function (event) {
    alert('Title is : ' + $(this).text());
});
```

## Definiowanie stylów w pliku design.css

W pliku `design.css` można zdefiniować dowolny styl, który będzie używany w szablonie układu zawartości.

Plik `design.css` jest ładowany w definicji modułu przy użyciu wtyczki RequireJS CSS.

## Uzyskiwanie elementów referencyjnych

Można uzyskać element referencyjny dla typu zawartości z referencyjnym polem danych, które odwołuje się do innego typu zawartości.

I tak w poniższym przykładzie pole `Author` z typem zawartości "Blog-Post" odwołuje się do pola zawartości "Author". W układzie zawartości dla "Blog-Post" metoda `contentClient.getItems()` wiąże szczegóły referencyjnego elementu "Author" z bieżącym elementem "Blog-Post".

```
var author_id = this.contentItemData.data['starter-blog-
post_author'].id;
var ids = [author_id];
// Get the author reference item
contentClient.getItems({
    'ids': ids
}).then(function (items) {
    // Use the item
}, function (error) {
    // Handle error
});
```

## Uzyskiwanie adresu URL zasobu multimedialnego

Używając metody `contentClient.getRenditionURL()`, można uzyskać realizację domyślną zasobu cyfrowego, takiego jak obraz.

```
blogHeaderImage = contentClient.getRenditionURL({
    'id': data['blog-post_header_image'].id
}),
```

Jeśli jest potrzebna inne realizacja, taka jak miniatura (`thumbnail`), można uzyskać zasób cyfrowy, używając metody `contentClient.getItems()`, a następnie odwoływać się do niego za pomocą `item.data.renditions.default` i `item.data.renditions.thumbnail`.

## Uaktywnianie wyzwalaczy

Do uaktywnienia wyzwalacza z układu zawartości można użyć funkcji `scsData.contentTriggerFunction()`.

Poniżej jest zamieszczony przykład układu zawartości "autor" uaktywniającego wyzwalacz, gdy użytkownik kliknie na nazwie autora:

```
{{#data}}
<div class="author-container">
  <span class="author-name" onclick='{{scsData.contentTriggerFunction}}
("field:starter-blog-post_author>equals={{author_id}}")'>{{starter-blog-
author_name}}</span>
{{/data}}
```

Ładunkiem (payload) jest zapytanie wyszukiwania obecnie wybranego autora, którego to zapytania mogą nasłuchiwać inne elementy "lista zawartości" na stronie.

## Przechodzenie do strony wyszukiwania za pomocą zapytania wyszukiwania

Typowym przypadkiem użycia jest przechodzenie do strony wyszukiwania za pomocą zapytania wyszukiwania, gdy użytkownik kliknie na łączu w obrębie układu zawartości.

Na przykład założmy, że powinno nastąpić przejście do strony autorów, gdy użytkownik kliknie na łączu "Więcej artykułów tego autora" występującym w układzie zawartości i w ten sposób przekaże ładunek wyszukiwania. Służy temu poniższy kod. Warto zwrócić uwagę, że globalne obiekty `SCS` i `SCSRenderAPI` są dostępne do użycia w układzie zawartości, gdy kod ten jest wykonywany w obrębie strony serwisu.

```
$('.more-from-author').click($.proxy(function () {
  var childrenPages = SCS.structureMap[SCS.navigationRoot].children;

  if (!childrenPages) return; // No pages

  // Find the Authors page
  for (var i = 0; i < childrenPages.length; i++) {
    var page = SCS.structureMap[childrenPages[i]];
    if (page.name === 'Authors') {
      var linkData = SCSRenderAPI.getPageLinkData(page.id);
      if (linkData && linkData.href) {
        var href = linkData.href,
            searchPayload = content.author_id + '*',
            contentType = "Starter-Blog-Post";
        // if both the page URL and the search query exists,
        navigate to the page passing in the query
        if (href && searchPayload) {
          var queryStart = href.indexOf('?') === -1 ? '?' : '&';

          // add in the contentType and search parameters
          // contentType isn't a required URL parameter
          // Payload contains search string only. No parameter
          name.

          href += queryStart + (contentType ? 'contentType=' +
```



```

contentType + '& : ' + 'q=' + searchPayload;

        // navigate to the search results page
        window.location = href;
    }
}
}, this));

```

Spodziewając się, że ten sam układ zawartości może być używany więcej niż raz na jednej stronie, lepiej jest — zamiast selektora klasy, na przykład `$('.more-from-author').click(...)` — użyć unikatowego ID w selektorze CSS.

Na przykład:

```

template.html
    <div id="{{navigateId}}">...</div>

render.js
    content.navigateId = this.scsData.id + 'detailTrigger';
    $('#' + navigateId).click(...)

```

## Rozwijanie makra i renderowanie tekstu formatowanego

W tekście formatowanym, występującym w elemencie zawartości, można osadzić obraz cyfrowy.

Aby taki tekst był poprawnie renderowany w układzie zawartości, pola z tekstem formatowanym używają API `contentClient.expandMacros()`. Metoda ta rozstrzyga wszystkie odwołania do zasobów cyfrowych osadzonych w tekście formatowanym.

```

data["starter-blog-post_content"] =
    contentClient.expandMacros(data["starter-blog-post_content"]);

```

Jeśli do renderowania są używane szablony Mustache, to — w celu renderowania wartości z tekstu formatowanego — należy używać nawiasów klamrowych `{{{ }}}`, ponieważ w tekście formatowanym występuje kod HTML. Jeśli zmienna zostanie ujęta w `{{{ }}}`, to Mustache nie modyfikuje kodu HTML.

## Łącze do strony "Szczegóły"

Strona "Szczegóły" jest dostępna za pomocą funkcji `scsData.detailPageLink`.

Na przykład, chcąc przejść do strony "Szczegóły" w celu wyświetlenia szczegółów bloga (po kliknięciu na tytule bloga), można użyć łącza do strony "Szczegóły" w następujący sposób:

```

<a href="{{scsData.detailPageLink}}">
    <h2 class="post-title">
        {{blogTitle}}
    </h2>
</a>

```

## Rozwijanie makra w zapytaniach dotyczących listy zawartości

W zapytaniach dotyczących listy zawartości można — aby wyświetlać ostatnio zaktualizowaną zawartość — definiować wartości właściwości, które są obliczane, gdy strona jest uruchamiana.

Większość właściwości składników serwisu jest statyczna. Użytkownik wybiera lub wprowadza ustaloną wartość dla jednej z właściwości składnika i wartość ta nie ulega zmianie bez względu na to, czy strona jest czy nie jest uruchomiona. Można jednak zdefiniować wartości właściwości, które to wartości są obliczane, gdy strona jest uruchamiana. Jest to przydatne do wyświetlania — w zapytaniach dotyczących zawartości — zawartości, która ostatnio została zaktualizowana. Użytkownicy mogą wówczas wprowadzać takie daty, jak "w ostatnich 3 dniach".

Rozszerzenie Mustache JS można wstawiać w wielu właściwościach. Wartości, do których występują odwołania, są wyprowadzane z modelu, który jest wykonywany w chwili uruchomienia strony. Gotowy do użycia model obsługuje daty sformatowane dla wywołań Content REST API. Model ten można rozbudować o dodatkowe wartości, tak aby spełniał specyficzne wymagania użytkowników.

Na przykład wartość napisowa, którą można wprowadzić dla właściwości, może mieć postać:

```
Content List component:
  Additional Query String property:
    updatedDate gt "{{#content.date}}today - 3 days{{/
content.date}}"
```

Ten odnoszący się do daty wpis Mustache będzie obliczany w trybie wykonawczym, a zatem zwracana wartość będzie się zmieniać w zależności od tego, kiedy nastąpiło uruchomienie (rozwiija się do postaci `updatedDate gt "2220181002060000000"`). W ten sposób użytkownik może skonstruować dowolną złożoną wartość napisową daty bez konieczności wprowadzania predefiniowanej wartości.

### Obsługujące właściwości składników

Następujące właściwości obsługują składnię szablonu Mustache JS:

- Lista zawartości
  - Dodatkowy napis-zapytanie
  - Na przykład: `updatedDate gt "{{#content.date}}today - 3 days{{/content.date}}"`
- Tytuł/akapit/tekst
  - Formatowany tekst wprowadzany za pomocą edytora CKEditor
  - Na przykład: `"Format Content REST API dla daty: {{#content.date}}now{{/content.date}}"`

 **Uwaga:**

Bez własnego modelu dla szablonu Mustache, rozwijanie w tytułach/akapitach/tekście jest mało przydatne. Jest jednak bardzo przydatne do weryfikowania, czy to, co zostało wprowadzone w dodatkowym napisie-zapytaniu, będzie obliczane przy przełączaniu między trybem edycji a trybem widoku i czy będzie od razu widoczne.

**Obsługiwana składnia składników**

Obiekt `content.date` jest od razu obsługiwany. Przyjmuje dwa główne parametry: `today` i `now`.

Wartość `today` przyjmuje bieżącą datę i godzinę z przeglądarki, przekształca tę wartość na dzisiejszą północ, a następnie tę wartość przekształca w godzinę UTC.

- `{{#content.date}}today{{/content.date}}` rozwija wartość z przeglądarki z dzisiejszą północ przekształconą w wartość UTC i sformatowaną do formatu daty Content REST API. Na przykład:

```
2220181008065959999
```

- Następnie wartość tę można powiększyć o:

```
today +/- [day | week | month | year]
```

- Wartość `today` zachowuje się inaczej, gdy dni są do niej dodawane lub są od niej odejmowane. Jeśli dni są odejmowane, jest używana godzina poranna. Jeśli dni są dodawane, jest używana godzina o północy. Na przykład:

- `{{#content.date}}today - 1 day{{/content.date}}` rozwija się na "wczoraj, na początku dnia".
- `{{#content.date}}today + 2 days{{/content.date}}` rozwija się na "pojutrze o północy".

Wartość `now` przyjmuje bieżącą datę i godzinę z przeglądarki i przekształca ją w godzinę UTC bez żadnych dostosowań.

- `{{#content.date}}now{{/content.date}}` rozwija bieżącą datę i godzinę z przeglądarki w wartość UTC sformatowaną do formatu daty Content REST API.
- `now` można także uzupełnić o `hour`. Można użyć:

```
now +/- [hour | day | week | month | year]
```

- Na przykład:

- `{{#content.date}}now + 2 hours{{/content.date}}` — dwie godziny od teraz przekształcone w wartość UTC sformatowaną do formatu daty Content REST API.
- `{{#content.date}}now - 1 day{{/content.date}}` — wczoraj o terażniejszej godzinie z przeglądarki przekształcone w wartość UTC sformatowaną do formatu daty Content REST API.

### Korzystanie z obsługiwanej składni składników

Aby użyć rozwijania makra w dodatkowym napisie-zapytaniu i na przykład uzyskać wszystko z ostatnich trzech tygodni, można by użyć:

```
updatedAt gt "{{#code.date}}today - 3 weeks{/code.date}}"
```

Aby zwrócona data działała w wywołaniu Content REST API, są do niej — gdy jest konstruowany napis-zapytanie — dodawane znaki cudzysłowu, tak jak miałyby to miejsce podczas wprowadzania wartości statycznej.

### MustacheJS

Składnia szablonów Mustache JS jest opisana na stronie "mustache.github.io/mustache.5.html".

W instancji Mustache, uruchamianej w celu rozwijania wartości napisowych, trzeba dokonać jednej zmiany. Mustache udostępnia rozwinięcie `text`, w którym są używane `{{ }}`, oraz rozwinięcie `html`, w którym są używane `{{{ }}`. Różnica między nimi polega na tym, że rozwinięcie "text" nie wprowadza kodowania HTML; oznacza to, że jeśli wartość jest rozwijana do `a < b`, to wynikiem będzie `a &lt; b`. To nie jest oczekiwane podczas konstruowania napisów tworzących adresy URL. Można by poinformować użytkownika o konieczności stosowania rozwijania HTML, lecz zwiększyłoby to nakłady pracy i stworzyłoby dodatkowe problemy, takie jak wyjaśnienie, dlaczego trzeba używać znaków `{{{ }}`.

Aby tego uniknąć Mustache skonfigurowano tak, aby nie modyfikował wartości, gdy są używane `{{ }}`. Oznacza to, że `{{ }}` i `{{{ }}` działają identycznie. Wskutek tego może być wymagane dodatkowe kodowanie przez użytkownika.

### Gotowy (standardowy) model Mustache

Do szablonu, aby nastąpiło rozwinięcie, trzeba zastosować model Mustache. W poprzednim przykładzie obiekt `{{#content.date}}` jest od razu zdefiniowany, podczas gdy może okazać się konieczne dodanie — przez programistę — nowego obiektu, takiego jak `{{person}}`. Jeśli użytkownik wprowadzi w szablonie Mustache wartość, której nie ma w modelu, to wynikiem będzie napis pusty. A zatem, jeśli programista nie doda obiektu `person` do modelu, to `Hello {{person}}` rozwinie się do samego `Hello`.

Obiekt "model" jest używany jako obiekt globalny o nazwie `SCSMacros`. Programista może dodawać do tego obiektu dowolne wpisy. Obiekt zostanie przekazany do systemu Mustache podczas przetwarzania szablonu.

Obecnie przez gotowy obiekt "model" jest obsługiwany tylko obiekt `content.date`:

```
{
  content: {
    date: <lambda implementation>
  }
}
```

### Niestandardowy model Mustache

Programista może wprowadzić dodatkowe obsługiwane obiekty, zgodnie z wymaganiami. Na przykład może wprowadzić obiekt `lastTwoDays` i uprościć rozwijanie przez tylko `{{lastTwoDays}}`.

Chcąc rozszerzyć model, tak aby obsługiwał na przykład `Hello {{person}}` z poprzedniego przykładu, należy dodać do `SCSMacros` obiekt `person`. Trzeba to zrobić przed uruchomieniem strony. Można to zrobić z układu strony, dodając na jego początku znacznik "script". Na przykład:

```
<script type="text/javascript">
window.SCSMacros = window.SCSMacros || {}; // define/get the SCSMacros
object
window.SCSMacros.person = "World";
</script>
```

Gdy zmiana ta zostanie dokonana, `Hello {{person}}` rozwinie się do `Hello World`.

Zamierzając przekazywać wartości do obiektu (na przykład `Hello {{#person}}personId{{/person}}`), trzeba zaimplementować interfejs `mustache lambda`, a następnie izolować i rozwijać wartość w obrębie tej implementacji.

Na przykład:

```
<script type="text/javascript">
window.SCSMacros = window.SCSMacros || {}; // define/get the SCSMacros
object
//implement "person" as a lambda
window.SCSMacros.person = function () {
  var people = { '111': { firstName: 'Small', lastName: 'World'},
'222': { firstName: 'Big', lastName: 'Universe' } };
  return function (text, render) {
    var expandedText = render(text);
    var chosenPerson = people[expandedText] || people['111'];
    return chosenPerson.firstName;
  }
};
</script>
```

Gdy zmiana ta zostanie dokonana, szablon `Hello {{#person}}111{{/person}}` rozwinie się do `Hello Small`, a `Hello {{#person}}222{{/person}}` — do `Hello Big`.

#### Uwaga:

Rozwijanie `Mustache` jest wykonywane synchronicznie. Jeśli trzeba pobierać wartości asynchroniczne, należy je ustalać w obrębie modelu jeszcze przed próbą wykonania rozwinięcia `Mustache`, a to jeszcze nie jest obecnie obsługiwane, aczkolwiek są możliwe implementacje opracowywane na zamówienie.

## Tworzenie odpornych układów zawartości

Układy zawartości muszą być odporne na trzy typy danych z odpowiedzi uzyskiwanych z wywołań REST dotyczących zawartości:

- Element zawartości z parametrem `expand=all` będzie zawierał rozwinięte odwołania i duże pola tekstowe.
- Element zawartości bez parametru `expand=all` nie będzie zawierał rozwiniętych odwołań, lecz będzie zawierał duże pola tekstowe.
- Zapytania dotyczące zawartości nie będą miały rozwijanych odwołań ani dużych pól tekstowych.

W celu poprawy wydajności wywołanie REST dotyczące zawartości, pobierające dane zasobów, nie zawiera już parametru `expand=all` (zaczynając od wydania Oracle Content Management 19.2.3). Parametr `expand` zleca usłudze Oracle Content Management drażenie w dół i pobieranie wszystkich elementów, do których występuje odwołanie, a także bieżącego elementu z odpowiedzi. Niestandardowe układy zawartości, bazujące na danych pobieranych przy użyciu parametru `expand`, trzeba zaktualizować, tak aby były obsługiwane przypadki, kiedy to pobierane dane nie zawierają wartości pól, do których występują odwołania.

## Renderowanie elementów zawartości

Układy zawartości są używane do renderowania elementów zawartości. Przyjmują dane elementu zawartości, renderują je do formatu HTML i wstawiają na stronie.

Domyślnie układy zawartości używają szablonów Mustache do renderowania elementów zawartości, aczkolwiek mogą być one implementowane w dowolnej technologii JavaScript. Szablon Mustache, aby móc renderować, oczekuje danych w określonym formacie. Plik `renderer.js` układów zawartości musi zapewnić zgodność modelu, przekazywanego do szablonu, z tym formatem.

Układy zawartości są renderowane w kilku przypadkach użycia:

- Gdy są używane w UI zarządzania zasobami Oracle Content Management — dane mogą pozostawać w stanie "edytowano" w celu umożliwienia użytkownikowi podgląd zmian przed ich zapisaniem.
- Gdy są używane na liście zawartości lub w elemencie zawartości w serwisie Oracle Content Management — dane są uzupełniane o dodatkowe informacje dotyczące serwisu, w którym układ zawartości jest używany.
- Gdy są używane przy użyciu wywołania `contentClient.renderLayout()` z pakietu Content SDK — kiedy użytkownik pakietu Content SDK przekazuje do układu zawartości dowolne dane.

Z punktu widzenia wydajności istnieje ogólny kompromis między tworzeniem jednego zapytania zwracającego wszystkie dane a kilkoma zapytaniami (powodującymi, że struktura jest renderowana możliwie szybko) z szybkim zapytaniem inicjującym i obszarami wypełnianymi przy użyciu kolejnych zapytań. Wybór odpowiedniego modelu zależy od danych i przypadku użycia.

Dodatkowo dane przekazywane do układu zawartości mogą mieć różny format wynikający ze sposobu ich pobierania. Na przykład, jeśli jest używane wywołanie REST (dotyczące zawartości) z rozwijanym parametrem, to mogą zostać także zwrócone odniesienia pól do innych elementów — zarówno indywidualnie, jak i grupowo. Jeśli elementy zawartości, do których występuje odniesienie, nie zostaną uwzględnione, trzeba użyć dodatkowych wywołań REST.

Programista układów zawartości, aby zapewnić obsługę wszystkich przypadków użycia, powinien elastycznie traktować format otrzymywanych danych. Ponadto, gdy jest to potrzebne, programista może pobierać dodatkowe dane i wymuszać przekształcanie ich do formatu oczekiwanego przez szablon renderowania.

## Standaryzacja struktury danych dla układu zawartości

Programista tworzący układ zawartości powinien ustandaryzować strukturę danych otrzymywanych przez układ zawartości.

Jeśli wszystkie dane są obecne, układ zawartości może bezproblemowo wyrenderować składnik. Jeśli nie wszystkie dane są obecne, może wystąpić wykonania dodatkowych zapytań. We wszystkich przypadkach układ zawartości nie powinien nigdy zakładać określonego formatu danych, a zamiast tego wymuszać dane w formacie umożliwiającym renderowanie.

Trzeba zapewnić obecność wszystkich oczekiwanych danych. Jeśli pewne dane nie istnieją, trzeba wykonać dodatkowe zapytania. W danych może potencjalnie brakować:

- Wpisu "fields" dla pól, do których występuje odwołanie
- Dużych pól tekstowych

Ponieważ układy zawartości są projektowane dla określonych typów zawartości, programista układu wie, które pola są wymagane. Aby układ zawartości mógł zostać wyrenderowany, dla każdego z tych pól muszą zostać pobrane dane. Są dostępne dwa rozwiązania: pobrać brakujące dane i renderować układ z użyciem kompletnych danych albo renderować od razu, a następnie pobrać brakujące dane w celu wypełnienia pustych miejsc.

### Rozwiązanie 1: Pobieranie brakujących danych i renderowanie z użyciem kompletnych danych

Utworzyć obiekt "Promise" do pobrania wymaganych danych, a następnie kontynuować renderowanie, gdy zostaną zwrócone wszystkie dane "Promise".

Na przykład mamy następujące typy zawartości z odpowiadającymi im polami:

- starter-blog-author
  - pola
    - \* starter-blog-author\_name — pole tekstowe
    - \* starter-blog-author\_bio — pole tekstowe
- starter-blog-post
  - pola
    - \* starter-blog-post\_title — pole tekstowe
    - \* starter-blog-post\_content — duże pole tekstowe
    - \* starter-blog-post\_author — odwołanie do elementu starter-blog-author

Układ zawartości ma następujący szablon do renderowania wartości tych oczekiwanych pól:

```
{{#fields}}  
<div class="blog_container">  
  <div class="blog-post-title">{{starter-blog-post_title}}</div>  
  {{#starter-blog-post_author.fields}}  
  <div class="blog-author-container">  
    <div class="blog-author-details">
```

```

        <div class="blog-author-name">{{starter-blog-author_name}}</div>
        <div class="blog-author-bio">{{starter-blog-author_bio}}</div>
        <span class="more-from-author">More articles from this author</
span>
    </div>
</div>
{{/starter-blog-post_author.fields}}
<div class="blog-post-content">{{starter-blog-post_content}}</div>
</div>
{{/fields}}

```

Układ zawartości można wywołać z danymi z następujących zapytań:

- Zapytanie (o element) z właściwością "expand" — dostarczone wszystkie dane
  - `/content/published/api/v1.1/items/{id}?expand=fields.starter-blog-post_author&channelToken=8dd714be0096ffaf0f7eb08f4ce5630f`
  - Jest to format danych wymaganych do pomyślnego wypełnienia wszystkich wartości w szablonie. Jeśli którekolwiek z zapytań zostanie użyte, wymagane jest pobranie danych i przekształcenie ich do tego formatu.
  - "fields": {
 

```

            "starter-blog-post_title": "...",
            "starter-blog-post_summary": "...",
            "starter-blog-post_content": "...",
            "starter-blog-post_author": {
              "id": "CORE386C8733274240D0AB477C62271C2A02",
              "type": "Starter-Blog-Author"
              "fields": {
                "starter-blog-author_bio": "...",
                "starter-blog-author_name": "..."
              }
            }
          
```
- Zapytanie (o element) bez właściwości "expand" — brak pól elementu, do których występuje odwołanie "**starter-blog-post\_author.fields**":
  - `/content/published/api/v1.1/items/{id}?channelToken=8dd714be0096ffaf0f7eb08f4ce5630f`
  - "fields": {
 

```

            "starter-blog-post_title": "...",
            "starter-blog-post_summary": "...",
            "starter-blog-post_content": "...",
            "starter-blog-post_author": {
              "id": "CORE386C8733274240D0AB477C62271C2A02",
              "type": "Starter-Blog-Author"
            }
          
```
- Zapytanie SCIM — brak dużego pola tekstowego "**starter-blog-post\_content**" oraz brak pól "**starter-blog-post\_author.fields**" elementu, do których występuje odwołanie:
  - `/content/published/api/v1.1/items?q=(type eq "Starter-Blog-Post")&fields=ALL&channelToken=8dd714be0096ffaf0f7eb08f4ce5630f`



```
- "fields": {
  "starter-blog-post_title": "...",
  "starter-blog-post_summary": "...",
  "starter-blog-post_author": {
    "id": "CORE386C8733274240D0AB477C62271C2A02",
    "type": "Starter-Blog-Author"
  }
}
```

Aby można było spójnie renderować z użyciem dowolnego z tych zapytań, `render.js` z układu zawartości musi się upewnić, że pola, do których występują odwołanie, zostały rozwinięte oraz że są obecne duże pola tekstowe.

Jeśli tak nie jest, musi ponownie uruchomić zapytanie, poprawić dane, po czym renderować z użyciem kompletnych danych.

Przykładowa funkcja `render()`:

```
render: function (parentObj) {
  var self = this,
      template,
      contentClient = self.contentClient,
      content = self.contentItemData;

  var getRefItems = function (contentClient, ids) {
    // Calling getItem() with no "ids" returns all items.
    // If no items are requested, just return a resolved Promise.
    if (ids.length === 0) {
      return Promise.resolve({});
    } else {
      return contentClient.getItems({
        "ids": ids
      });
    }
  };

  var fetchIDs = [], // list of items to fetch
      referredFields = ['starter-blog-post_author'], // names of
reference fields
      largeTextFields = ['starter-blog-post_content'], // large text
fields in this asset
      fieldsData = content.fields;
  // See if we need to fetch any referenced fields
  referredFields.forEach(function (fieldName) {
    if(fieldsData[fieldName] && fieldsData[fieldName].fields) {
      // got data already, nothing else to do
    } else {
      // fetch this item
      fetchIDs.push(fieldsData[fieldName].id);
    }
  });

  // See if we need to fetch any large text fields
  for(var i = 0; i < largeTextFields.length; i++) {
    if(!fieldsData[largeTextFields[i]]) {
```

```

        // need to fetch this content item directly to get all the large
text fields
        fetchIDs.push(content.id);
        break;
    }
}
// now we have the IDs of all the content items we need to fetch, get
them all before continuing
getRefItems(contentClient, fetchIDs).then(function (referenceData) {
    var items = referenceData && referenceData.items || [];

    // add the data back in
    items.forEach(function (referencedItem){
        // check if it's the current item
        if(referencedItem.id === content.id) {
            // copy across the large text fields
            largeTextFields.forEach(function (fieldName) {
                fieldsData[fieldName] = referencedItem.fields[fieldName];
            });
        } else{
            // check for any referenced fields
            for (var i = 0; i < referedFields.length; i++) {
                if(referencedItem.id === fieldsData[referedFields[i]].id)
                {
                    // copy across the fields values
                    fieldsData[referedFields[i]].fields =
referencedItem.fields;
                    break;
                }
            }
        }
    });

    // now data is fixed up, we can continue as before
    try{
        // Mustache
        template = Mustache.render(templateHtml, content);

        if(template) {
            $(parentObj).append(template);
        }

    } catch (e) {
        console.error(e.stack);
    }
});
}

```

### Rozwiązanie 1: Natychmiastowe renderowanie i późniejsze pobranie brakujących danych w celu wypełnienia pustych miejsc

Wydajność operacji można poprawić, oddzielając elementy, których może nie być, i renderując je w drugim przebiegu. W tym celu są potrzebne dwa szablony Mustache: pierwszy do wykonania pierwszego renderowania z pozostawieniem pustych miejsc, które

następnie zostaną wypełnione podczas drugiego renderowania, kiedy będą obecne wszystkie dane.

Wymaga to skonfigurowania szablonu Mustache do obsługi więcej niż jednego przebiegu przez używanie albo osobnych szablonów dla pustych miejsc, albo modelu z makrami zwracającymi szablon zamiast faktycznych wartości. W obu przypadkach trzeba ukryć puste miejsca do chwili pobrania danych, wstawienia ich i pokazania za pomocą odpowiedniej animacji UI (aby uniknąć zbytecznego „przeskakiwania”).

## Tworzenie przykładowego szablonu bloga

W przykładzie "BlogTemplate" zademonstrowano funkcje układu zawartości.

Można utworzyć szablon w usłudze Developer Cloud Service, przetestować układy zawartości w szablonie, a następnie przetestować funkcje wyszukiwania:

1. Utworzyć szablon typu BlogTemplate:

```
cec create-template MyBlogTemplate -f BlogTemplate
```

Nowy szablon MyBlogTemplate zostanie utworzony poprzez skopiowanie szablonu BlogTemplate .

2. Otworzyć stronę <http://localhost:8085/>, wybrać opcję **Szablony**, po czym kliknąć na szablonie **MyBlogTemplate**.
3. Zostanie wyświetlona lista wpisów na blogu. Kliknąć na jednym z tych wpisów. Zostanie wyświetlona strona szczegółów, która do renderowania używa trzech różnych układów zawartości.
4. Nacisnąć przycisk **More articles from this author** (Więcej artykułów tego autora). Zostanie wyświetlona strona wyszukiwania.
5. Kliknąć na nazwie autora. Ten układ zawartości uaktywnia wyzwalacz oraz wyświetla artykuły po prawej stronie.
6. Ponownie przejść do strony startowej, po czym przetestować wyszukiwanie.

## Dodawanie odwzorowań układu zawartości do szablonów

Po utworzeniu układu zawartości można go dodać do lokalnych szablonów w projekcie Developer Cloud Service, dodając odwzorowanie układu zawartości.

Odwzorowanie układu zawartości można dodać dla szablonu, używając polecenia `cec add-contentlayout-mapping`. Typ zawartości (`-c`), na którym jest oparty układ zawartości, oraz szablon (`-t`), dla którego jest to odwzorowanie, są wartościami wymaganymi. Na przykład:

```
cec add-contentlayout-mapping Blog-Post-Detail-Layout -c Blog-Post -t BlogTemplate
```

Domyślnym odwzorowanie układu zawartości jest styl "Default" dla wersji stacjonarnej. Można użyć opcji `-s <styl_układu>` w celu określenia (za pomocą nazwy) innego stylu układu, na przykład "Overview" lub "Details":

```
cec add-contentlayout-mapping Blog-Post-Detail-Layout -c Blog-Post -t
BlogTemplate -s Details
```

Można także ustawić odwzorowanie dla wersji mobilnej, używając opcji `-m`:

```
cec add-contentlayout-mapping Blog-Post-Detail-Layout -c Blog-Post -t
BlogTemplate -m
```

## Testowanie układów zawartości przy użyciu lokalnego zestawu testów

Po dodaniu (do szablonu) typów zawartości i odwzorowań układów zawartości można przetestować układy zawartości przy użyciu lokalnego zestawu testów.

Zob. [Testowanie przy użyciu lokalnego zestawu testów](#).

## Testowanie przy użyciu lokalnego zestawu testów

Przed zaimportowaniem składników niestandardowych, szablonów i układów zawartości do usługi Oracle Content Management należy uruchomić je przy użyciu lokalnego zestawu testów.

Aby uruchomić lokalny zestaw testów, należy:

1. W oknie terminala wpisać polecenie `cd cec`.
2. Wpisać polecenie `cec develop &` lub `cec develop --server <nazwa_serwera> &`
3. Aby wyświetlić składniki, szablony i układy zawartości, działające w lokalnym zestawie testów, otworzyć przeglądarkę, a następnie stronę `http://localhost:8085`.
4. Składniki, szablony, motywy itp. można znaleźć w następujących katalogach:
  - `cec/src/main/components`
  - `cec/src/main/templates`
  - `cec/src/main/themes`

## Importowanie szablonów z układami zawartości do Oracle Content Management

Po utworzeniu i przetestowaniu układów zawartości można wyeksportować z projektu Developer Cloud Service szablony, który zawiera układy zawartości, a następnie zaimportować ten szablon do Oracle Content Management.

Zob. [Eksportowanie szablonu](#).

## Tworzenie składników

Można tworzyć składniki przeznaczone do użycia w serwisach Oracle Content Management.

- [Składniki — informacje podstawowe](#)
- [Tworzenie składników — informacje podstawowe](#)
- [Tworzenie składnika](#)
- [Tworzenie składników niestandardowych za pomocą usługi Developer Cloud Service](#)
- [Tworzenie możliwych do przetłumaczenia składników dla serwisów wielojęzycznych](#)
- [Tworzenie składnika H1 za pomocą panelu "Ustawienia"](#)
- [Tworzenie prostego składnika z użyciem szablonów HTML](#)
- [Porównanie składników lokalnych ze składnikami odległymi](#)
- [Renderowanie ustawień składnika](#)
- [Implementacja składnika lokalnego](#)
- [Klasy stylów składników](#)
- [Jak nadawać styl składnikom wbudowanym](#)
- [Ustawianie właściwości składnika](#)
- [Składniki renderowane w ramach wstawkowych](#)
- [ID instancji i struktura składników renderowanych w ramach wstawkowych — informacje podstawowe](#)
- [Zabezpieczenia dla składników odległych](#)
- [Rejestrowanie składnika odległego](#)
- [Usuwanie składnika](#)
- [Sites SDK](#)

## Składniki — informacje podstawowe

Składnik jest określonym typem zawartości, który można dodać do strony w serwisie. W Oracle Content Management składniki zawierają elementy, takie jak akapit, tytuł, obrazy, separatory itp.

Oracle Content Management obsługuje następujące typy składników:

- Składnik lokalny: pliki są przechowywane w Oracle Content Management
- Składnik odległy: pliki są przechowywane na serwerze odległym

Składniki lokalne można ustawić, tak aby były renderowane bezpośrednio w obrębie strony lub renderowane w ramce wstawkowej na stronie. Składniki odległe są zawsze renderowane w ramce wstawkowej.

Oracle Content Management udostępnia z każdym z szablonów domyślny zbiór składników, zawierający również motywy i serwisy. Można tworzyć nowe składniki oraz używać tych domyślnych składników we własnych składnikach. Po dodaniu składnika do serwisu można edytować ustawienia właściwości składnika, tak aby spełniał stawiane mu wymagania; można określić zawartość strony, czcionki i rozmiar czcionki, obramowanie i rozmieszczenie obrazów oraz inne style. To, które ustawienia można edytować, zależy od typu zawartości.

Składniki te są zawarte w usłudze Oracle Content Management.

Nazwa	Typ	ID
Tytuł	scs-title	scs-title
Akapit	scs-paragraph	scs-paragraph
Tekst	scs-text	scs-text
Obraz	scs-image	scs-image
Galeria	scs-gallery	scs-gallery
Siatka galerii	scs-gallerygrid	scs-gallerygrid
Dokument	scs-document	scs-document
Przycisk	scs-button	scs-button
Mapa	scs-map	scs-map
Separator	scs-divider	scs-divider
Rozdzielacz	scs-spacer	scs-spacer
YouTube	scs-youtube	scs-youtube
Pasek społecznościowy	scs-socialbar	scs-socialbar
Artykuł (składnik niestandardowy)	scs-component	scs-comp-article
Nagłówek (składnik niestandardowy)	scs-component	scs-comp-headline
Lista folderów	scs-app	Lista folderów
Lista plików	scs-app	Lista plików
Menedżer dokumentów	scs-app	Menedżer dokumentów
Facebook - Lubię to	scs-app	Facebook - Lubię to
Twitter - Obserwuj	scs-app	Twitter - Obserwuj
Twitter - Udostępnij	scs-app	Twitter - Udostępnij
Facebook - Rekomendacja	scs-app	Facebook - Rekomendacja
Obraz i tekst (składnik niestandardowy)	scs-component	scs-comp-image-text

Zob. rozdział Rozmieszczanie zawartości strony w podręczniku *Tworzenie serwisów za pomocą Oracle Content Management*.

## Tworzenie składników — informacje podstawowe

Tworzenie własnych składników niestandardowych pozwala na tworzenie elementów złożonych, które można osadzać na stronach serwisu przy użyciu dowolnej technologii

dla stron. Takie działanie skutecznie umożliwia rozszerzenie listy składników udostępnionych przez usługę Oracle Content Management.

Wpisy wszystkich zarejestrowanych składników są przechowywane w Katalogu składników, będącym przeznaczonym do tego celu folderem w Oracle Content Management.

Dla szablonów, motywów, składników, serwisów ani stron serwisów nie wolno używać następujących nazw: `authsite`, `content`, `pages`, `scstemplate_*`, `_comps`, `_components`, `_compsdelivery`, `_idcservice`, `_sitescloud`, `_sitesclouddelivery`, `_themes`, `_themesdelivery`. Następujących nazw można używać dla stron serwisów, ale nie można używać dla szablonów, motywów, składników ani serwisów: `documents`, `sites`.

### Typy składników

Katalog składników obsługuje następujące typy składników:

- Składnik lokalny
- Składnik lokalny renderowany w ramce wstawkowej
- Składnik odległy

Typ składnika jest przechowywany jako atrybut rozszerzenia (`xScsApType`) folderu składnika. Poniżej są podane poprawne wartości.

Typ	Opis
Składnik lokalny	Wszystkie zależności powinny być lokalne.
Składnik lokalny renderowany w ramce wstawkowej	Składnik jest obsługiwany z tej samej domeny, z której jest obsługiwana usługa Oracle Content Management.
Składnik odległy	Składnik jest obsługiwany z lokalizacji odległej.

### Struktura pliku składnika

Podczas tworzenia typu składnika w katalogu składników, każdy z typów składników ma inny zestaw plików zależny od sposobu implementacji.

#### Składnik lokalny:

```
/Components/component-name
  appinfo.json
  _folder_icon.jpg
  assets
    settings.html
    render.js
```

#### Składnik lokalny używający ramki wstawkowej:

```
/Components/component-name
  appinfo.json
  _folder_icon.jpg
  assets
    settings.html
    render.js
  js
    sites.min.js
```

```
knockout.min.js
jquery.min.js
```

### Składnik odległy:

```
/Components/component-name
  appinfo.json
  _folder_icon.jpg
  keys.json
```

### Metadane folderu

Dane rejestracji składnika są przechowywane w metadanych folderu. Poniższe właściwości umożliwiają unikatowe identyfikowanie składnika i jego typu.

Właściwość	Opis
app name	Nazwa folderu, która wyznacza nazwę składnika.
app description	Opis folderu.
app guid	Każdy składnik jest powiązany z identyfikatorem GUID i jest przechowywany jako atrybut rozszerzenia <code>xScsItemGUID</code> . GUID jest generowany przez serwer podczas tworzenia składnika.
app type	Właściwość wyznacza typ składnika. Jest przechowywany jako atrybut rozszerzenia <code>xScsAppType</code> .

#### Uwaga:

Właściwość `iconUrl`, która jest przechowywana jako atrybut rozszerzenia `xScsAppIconUrl`, jest już nieużywana.

### Plik `appinfo.json`

Plik rejestracji `appinfo.json` dla każdego typu składnika zawiera tylko te dane, które nie są dostępne w metadanych folderu. Właściwości składnika, które są zdefiniowane w metadanych folderu, nie są powielane w pliku `appinfo.json`.

### Składnik lokalny:

```
{
  "settingsData":{
    "settingsHeight":80,
    "settingsRenderOption": "dialog",
    "settingsWidth":300,
    "componentLayouts":[],
    "triggers":[],
    "actions":[]
  },
  "initialData":{
    "customSettingsData":[],
```



```

    }
  }
}

```

**Składnik lokalny renderowany w ramce wstawkowej:**

```

{
  "endpoints": {
    "settings": {
      "height": "300",
      "width": "400"
    }
  }
  "initialData": {
    "customSettingsData": {}
  }
}

```

**Składnik odległy:**

```

{
  "endpoints": {
    "widget": {
      "url": "http://www.externaldomain.com/app/render.html"
    }
    "settings": {
      "url": "http://www.externaldomain.com/app/
settings.html",
      "height": "300",
      "width": "400"
    }
  }
  "initialData": {
    "customSettingsData": {}
  }
}

```

W przypadku składników lokalnych nie ma żadnych rejestrowanych wartości "url". Składniki te używają kilku plików, których zawartość można edytować, lecz nie można zmienić ich lokalizacji ani nazwy.

- Składniki lokalne używają plików `assets/render.js` i `assets/settings.html`.
- Składniki lokalne, renderowane w ramce wstawkowej, używają plików `assets/render.html` i `assets/settings.html`.
- Składniki odległe używają określonych wartości "url".

## Tworzenie składnika

Można tworzyć niestandardowe składniki przeznaczone do użycia w Oracle Content Management.

Wprawdzie Oracle Content Management udostępnia różne predefiniowane składniki, które mogą być używane do tworzenia serwisów, to jednak będąc programistą można tworzyć

składniki niestandardowe z dostępem do tych samych funkcji i możliwości. Są dostępne następujące typy składników: lokalne, lokalne renderowane w ramach wstawkowych oraz odległe renderowane w ramach wstawkowych.

#### Składnik lokalny

Gdy jest tworzony składnik lokalny, jest udostępniany w pełni funkcjonalny składnik przykładowy, używany jako podstawa nowego składnika. Aby utworzyć wersję Knockout składnika, należy wybrać opcję **Domyślne**; aby utworzyć wersję opartą na technologii Mustache — opcję **Szablon**. Zob. rozdział Tworzenie składników lub układów lokalnych w podręczniku *Tworzenie serwisów za pomocą Oracle Content Management*.

#### Składnik lokalny z ramką wstawkową

Gdy jest tworzony składnik lokalny z ramką wstawkową, jest udostępniany w pełni funkcjonalny składnik przykładowy, używany jako podstawa nowego składnika. Należy użyć tych samych instrukcji, jak w przypadku tworzenia składnika lokalnego. Ponadto — aby utworzyć wersję składnika (przechowywanego lokalnie) w ramce wstawkowej — należy wybrać opcję **W środowisku Sandbox**. Zob. [Składniki renderowane w ramach wstawkowych](#).

#### Składnik odległy

Gdy jest tworzony składnik odległy, który będzie używał ramki wstawkowej, należy wybrać opcję **W środowisku Sandbox**

Następnie należy skopiować utworzone pliki do odległego serwera, po czym zarejestrować składnik odległy. Przed dołączeniem składnika do opublikowanego serwisu, należy składnik ten przetestować.

Zob. [Składniki renderowane w ramach wstawkowych](#) i [Rejestrowanie składnika odległego](#).

## Tworzenie składników niestandardowych za pomocą usługi Developer Cloud Service

Za pomocą usługi Oracle Developer Cloud Service można tworzyć szablony, motywy i składniki niestandardowe dla Oracle Content Management.

W wyniku integracji usługi Developer Cloud Service z Oracle Content Management jest udostępniany szablon z narzędziami służącymi do tworzenia szablonów i składników. Są także udostępniane przykładowe testy jednostkowe, od których można zacząć. Integracja obejmuje repozytorium Git i narzędzia pomagające opracowywać szablony i składniki oraz lokalny zestaw testów służący do szybkiego, interaktywnego tworzenia szablonów, motywów i składników niestandardowych.

Developer Cloud Service pomaga przy wykonywaniu następujących zadań:

- Konfigurowanie lokalnego środowiska programistycznego pod kątem używania instancji Oracle Content Management do lokalnego tworzenia i testowania szablonów, motywów i składników
- Tworzenie szablonów i składników (na podstawie przykładów lub elementów początkowych) oraz uruchamianie ich w zestawie testów i eksplorowanie, a także

tworzenie szablonów, motywów i składników w środowisku Developer Cloud Service

- Importowanie — z Oracle Content Management do środowiska Developer Cloud Service — szablonów lub składników, które zostały utworzone na potrzeby zarządzania źródłami i przyszłych potrzeb programistycznych
- Eksportowanie — z środowiska Developer Cloud Service — szablonów lub składników, które można zaimportować do Oracle Content Management w celu użycia ich w serwisach internetowych.
- Kopiowanie już istniejącego składnika
- Tworzenie testów jednostkowych
- Optymalizowanie składników
- Wprowadzanie składników do środowiska wykonawczego Oracle Content Management

W poniższych tematach opisano, jak korzystać z integracji Developer Cloud Service z usługą Oracle Content Management :

- [Tworzenie składnika niestandardowego dla Oracle Content Management](#)
- [Optymalizowanie \(pomniejszanie\) składników dla lepszej wydajności](#)
- [Uruchamianie zleceń integracji ciągłej](#)

## Tworzenie składnika niestandardowego dla Oracle Content Management

Korzystając z usługi Developer Cloud Service i komputera lokalnego, można tworzyć składniki niestandardowe dla usługi Oracle Content Management.

W poniższych tematach są opisane etapy tworzenia i testowania składników Oracle Content Management przy użyciu usługi Developer Cloud Service:

1. [Konfigurowanie zestawu OCE Toolkit na komputerze lokalnym.](#)
2. [Logowanie do konsoli Developer Cloud Service dla Oracle Content Management.](#)
3. [Tworzenie projektu w usłudze Developer Cloud Service.](#)
4. [Dodawanie zestawu OCE Toolkit do kodu projektu w nowym repozytorium Git.](#)
5. [Tworzenie składnika niestandardowego](#)
6. [Testowanie przy użyciu lokalnego zestawu testów](#)
7. [Pisanie i uruchamianie testów jednostkowych](#)

## Tworzenie składnika niestandardowego

Używając narzędzia cec opartego na wierszu polecenia można utworzyć nowy składnik, opracować go lokalnie, a następnie wyeksportować do Oracle Content Management.

### Tworzenie składnika

Chcąc utworzyć i opracować nowy składnik lokalnie, należy użyć następującego polecenia:

```
cec create-component <component-name> -f <source>
```

Jako *source* należy wybrać jedną z następujących wartości:

- JET-CCA-Demo-Card

- local
- local-iframe
- Sample-Facebook-Share
- Sample-News-API
- Sample-Stocks-Embedded
- Sample-To-Do
- Sample-Text-With-Image
- Sample-Weather-Embedded
- Sample-Folder-List
- Sample-File-List
- Sample-Documents-Manager
- Sample-Process-Start-Form
- Sample-Process-Task-List
- Sample-Process-Task-Details
- SimpleHTML

Przykład:

```
cec create-component MyLocalComponent1 -f local
```

Składnik zostaje utworzony w repozytorium Git, w katalogu `cec-components/src/main/components`.

Katalog `src/main/components` jest wypełniany składnikiem `Sample-To-Do`. Wszystkie składniki, tworzone przez użytkownika, idą do tego katalogu.

#### Kopiowanie składnika

Składnik można skopiować w usłudze Developer Cloud Service, używając następującego polecenia `cec copy-component`:

```
cec copy-component <source> [<destination>]
```

Polecenie to kopiuje istniejący składnik, określony jako `<source>`, do miejsca docelowego, określonego jako `<destination>`.

#### Eksportowanie (lub wdrażanie) składnika do Oracle Content Management

Składnik, opracowany i przetestowany na serwerze lokalnym, można wyeksportować, używając następującego polecenia: Zostaje utworzony plik zip składnika. Ten plik zip składnika można ręcznie zaimportować do Oracle Content Management.

```
cec export-component <component name>
```

Alternatywnie można składnik wdrożyć w Oracle Content Management bezpośrednio z usługi Developer Cloud Service, używając następującego polecenia:

```
cec deploy <component name>
```

Polecenie `cec deployAll` droży wszystkie składniki z `src/main/components`.

#### Importowanie składników do usługi Developer Cloud Service

Mając utworzony plik zip składnika z serwera Oracle Content Management, można zaimportować ten plik do usługi Developer Cloud Service w celu dalszego opracowywania składnika. Należy użyć następującego polecenia:

```
cec import-component <location of the component zip file>
```

#### ! Ważne:

Kod źródłowy składników znajduje się w katalogu `src/main/components`. Nie należy modyfikować żadnych plików spoza folderu `src/main/components`, ponieważ zapewniają one poprawne funkcjonowanie lokalnego serwera Oracle Content Management.

## Pisanie i uruchamianie testów jednostkowych

Pisanie i uruchamianie testów swoich niestandardowych składników Oracle Content Management warto rozpocząć od zapoznania się z testem przykładowym.

### Pierwsze kroki — przykładowy test jednostkowy

W przykładowych materiałach z usługi Developer Cloud Service są dostępne następujące pliki dla Oracle Content Management, pomocne przy pisaniu testów jednostkowych:

- **src/test/unit:** Zawiera test jednostkowy dla składnika `Sample-To-Do`. Test ten służy jako przykład pomocny przy pisaniu testów jednostkowych dla kodu JavaScript składników, zawierającego moduły RequireJS. Do jednostkowego testowania kodu JavaScript są używane struktury Mocha i Chai.
- **index.html:** Uruchamia test jednostkowy. Ładuje Mocha, Chai i główny plik testu `test-main.js`.
- **test-main.js:** Ładuje moduł `Sample-To-Do-Test` testu jednostkowego i uruchamia testy jednostkowe Mocha.
- **Sample-To-Do-Test.js:**
  - Definiuje rzeczywiste testy. Ładuje kod źródłowy `components/Sample-To-Do/assets/render.js` z użyciem RequireJS.
  - Testuje metody `add()`, `delete()`, `title()` i `placeholder()`.

### Pisanie testów jednostkowych dla swoich składników

Aby napisać testy jednostkowe dla składnika niestandardowego, należy:

1. Napisać test jednostkowy, podobny do `Sample-To-Do-Test.js`.

2. Załadować napisany test do `test-main.js`.

#### Lokalne uruchamianie testu jednostkowego

Aby uruchomić lokalnie test jednostkowy, należy:

1. Sklonować lokalnie repozytorium Git.
2. Na komputerze lokalnym wprowadzić w oknie terminala polecenie `npm install`.
3. Wprowadzić polecenie `npm start`.
4. Wprowadzić polecenie `npm test` albo otworzyć w przeglądarce adres `http://localhost:8085/unit/`.

## Optymalizowanie (pomniejszanie) składników dla lepszej wydajności

Można poprawić wydajność składników oraz zmniejszyć rozmiar pobierania, optymalizując kod JavaScript za pomocą usługi Developer Cloud Service.

#### Optymalizowanie składników

Przykład opcjonalnej optymalizacji kodu źródłowego składnika jest przedstawiony dla składnika `Sample-Text-With-Image`. Używając optymalizatora RequireJS, można pomniejszyć i połączyć kod źródłowy. Pomniejszanie składnika polega na usunięciu zbędnych odstępów w kodzie JavaScript, wskutek czego poprawia się wydajność i zmniejsza rozmiar pobierania. Na optymalizację składają się następujące czynności:

- Pomniejszenie kodu JavaScript.
- Skompresowanie kodu CSS.
- Połączenie kodu JavaScript, HTML i CSS w jeden plik `render.js`.

Wskutek tej optymalizacji ulega zmniejszeniu rozmiar pobierania oraz poprawia się wydajność.

#### Włączanie optymalizacji składnika

Optymalizacja składnika `Sample-Text-With-Image` pomniejszy i połączy wszystkie pliki oznaczone w pliku `render.js` jako zależności (`template.html`, `data-defaults.js`, and `design.css`) w jeden plik `render.js`. Poniżej są wymienione pozostałe składniki obsługujące optymalizację:

- `Sample-Folder-List`
- `Sample-File-List`
- `Sample-Documents-Manager`
- `Sample-Process-Start-Form`
- `Sample-Process-Task-List`
- `Sample-Process-Task-Details`

Aby włączyć optymalizację dla swojego składnika lokalnego, należy wykonać następujące czynności (wymagane):

1. Skopiować do swojego składnika plik `gulpfile.js` ze składnika `Sample-Text-With-Image`.

2. Poprawić nazwy modułów w pliku `gulpfile.js` zgodnie z zależnościami ze swojego pliku `render.js`. Należy przy tym korzystać z komentarzy z pliku `gulpfile.js`.

#### Uwaga:

Mając istniejący składnik z folderem `css` zawartym w folderze `assets`, należy — aby uniknąć błędów podczas optymalizowania — zmienić nazwę folderu `css` na `styles`.

## Uruchamianie zleceń integracji ciągłej

W celu zapewnienia aktualności serwisu można uruchomić zlecenia integracji ciągłej.

Więcej informacji jest dostępnych na stronie Wiki "Getting Started with CEC Custom Components Development" w usłudze Developer Cloud Service. Dla integracji ciągłej można skonfigurować zlecenie kompilacji.

## Tworzenie możliwych do przetłumaczenia składników dla serwisów wielojęzycznych

Twórcy składników niestandardowych mogą określać, które napisy ze składnika niestandardowego mogą występować w tłumaczeniach stron dla serwisów wielojęzycznych (MLS).

W tym celu trzeba, zapisując dane w obiekcie `customSettingsData`, wprowadzić na poziomie głównym właściwość `nls`.

Na przykład:

```
SitesSDK.setProperty('customSettingsData', {
  'nls': {
    linkText: 'More...'
  }
});
```

Gdy zostanie utworzone zlecenie tłumaczenia, Oracle Content Management sprawdzi właściwości (właściwości głównego poziomu) obiektu `customSettingsData` i wyeksportuje cały obiekt `nls` dla każdego wystąpienia niestandardowego składnika na stronie. Przetłumaczone wartości będzie można zaimportować ponownie do serwisu.

Gdy tłumaczenia serwisu zostaną zaimportowane, w obiekcie `customSettingsData` (dla ustawień narodowych tłumaczenia) zostanie zwrócona właściwa wersja obiektu `nls`.

Na przykład, jeśli serwis został przetłumaczony na francuski, a następnie strona będzie renderowana z francuskimi ustawieniami narodowymi, wartość obiektu `nls` w obiekcie `customSettingsData` przekazany do składnika niestandardowego zostanie zaktualizowana do postaci:

```
{
  'nls': {
    'linkText': 'Plus...'
  }
}
```

```
    },  
  }  
}
```

Format obiektu `nls` w obiekcie `customSettingsData` ogranicza się do par nazwa/wartość. Pomaga to przy tłumaczeniu i gwarantuje, że — gdy strona będzie renderowana z użyciem ustawień narodowych tłumaczenia — przetłumaczone wartości zostaną poprawnie zastosowane do wartości podstawowych w serwisie.

## Tworzenie składnika H1 za pomocą panelu "Ustawienia"

Można utworzyć minimalny składnik Oracle Content Management, zawierający prosty szablon HTML i arkusz CSS. Ten składnik H1 zawiera prosty panel ustawień oraz wpis motywu w pliku `design.json`, umożliwiającą innym użytkownikom Oracle Content Management wybór — podczas używania składnika w edytorze — jednego z trzech wbudowanych stylów.

Użytkownik, tworząc nowy składnik, uzyskuje zestaw standardowych, od razu działających plików. Pliki standardowe obejmują większą część funkcji składników w obrębie produktu. Zmieniając standardowy kod, można utworzyć własny składnik; aby uzyskać pożądany wynik końcowy, wystarczy tylko niewielki podzbiór standardowego kodu.

Składnik H1 można utworzyć za pomocą panelu "Ustawienia", wykonując następujące zadania:

1. [Tworzenie nowego składnika lokalnego](#)
2. [Tworzenie podstawowego składnika H1](#)
3. [Dodawanie stylów CSS dla swojego składnika](#)
4. [Dodawanie panelu ustawień umożliwiającego zmianę tekstu nagłówka](#)
5. [Aktualizacja motywu umożliwiająca użytkownikom wybór stylu składnika H1](#)

## Tworzenie nowego składnika lokalnego

Korzystając z Oracle Content Management można utworzyć składnik lokalny i od razu umieścić go na stronie. Jest to punkt startowy do tworzenia każdego nowego składnika.

Aby utworzyć nowy składnik lokalny, należy:

1. Z nawigacji bocznej wybrać opcję **Programista**, po czym nacisnąć przycisk **Wyświetl wszystkie przed**.
2. Wybrać opcje **Utwórz > Utwórz składnik lokalny**.
3. Podać nazwę (na przykład `H1_Component`) i opcjonalny opis.
4. Aby utworzyć nowy składnik, nacisnąć przycisk **Utwórz**.

Pomyślnie utworzony składnik jest widoczny na liście składników na stronie "Składniki" oraz na palecie składników (**Dodaj > Niestandardowe**) dla każdego tworzonych serwisu. Aby zweryfikować utworzenie składnika (punkt kontrolny 1), należy:

1. Utworzyć nowy szablon, używając dowolnego standardowego szablonu; na przykład utworzyć serwis `ComponentTest`, używając szablonu `StarterTemplate`.



2. Wybrać opcję **Edytuj**, po czym utworzyć aktualizację serwisu, aby otworzyć ją w edytorze.
3. Edytować stronę w utworzonym serwisie.
4. Nacisnąć przycisk **Dodaj (+)** na lewym pasku, po czym wybrać opcję **Niestandardowe** — zostanie wyświetlona lista składników niestandardowych.
5. Wybrać z palety składników niestandardowych składnik **H1\_Component** i przeciągnąć go na stronę.  
Zostaje wyświetlone domyślne renderowanie utworzonego składnika lokalnego.
6. Wybrać menu podręczne dla składnika.
7. Wybrać z rozwijanego menu opcję **Ustawienia**.  
Można zmienić ustawienia, aby zobaczyć, jak zmieni się standardowe renderowanie składnika.  
Pliki standardowe można modyfikować w celu utworzenia nowego składnika niestandardowego.

## Tworzenie podstawowego składnika H1

Aby utworzyć składnik H1, można usunąć większą część zawartości standardowych plików. Składnik H1 wyświetla tekst nagłówka, który jest wstępnie określany podczas tworzenia modelu `viewModel`. Później można wprowadzić ustawienia i style dla składnika.

Aby sprawdzić strukturę składnika lokalnego, należy:

1. Używając aplikacji synchronizującej typu Desktop Oracle Content Management, zlokalizować składnik, po czym zsynchronizować go z systemem plików.
  - W najnowszej wersji aplikacji synchronizującej typu Desktop wybrać opcję **Rozpocznij synchronizację** lub **Wybierz foldery do synchronizacji**.
  - Nie mając aplikacji synchronizującej typu Desktop, można na karcie **Składniki** usługi Oracle Content Management wybrać składnik, a następnie — aby wyświetlić pliki — drażyć w dół.
2. Pod składnikiem można zobaczyć następujące pliki:
  - Pliki składnika w folderze `assets`:
    - `render.js`
    - `settings.html`
  - `appinfo.json`: Plik JSON zawierający opis składnika.  
Zob. [Tworzenie składników — informacje podstawowe](#).
  - `folder_icon.jpg`: Ikona, która jest wyświetlana w katalogu składników.

Aby utworzyć składnik H1, należy:

1. Otworzyć plik `appinfo.json`, po czym zastąpić jego zawartość następującymi wierszami kodu:

```
{
  "id": "h1-component-id",
```

```

"settingsData": {
  "settingsHeight": 90,
  "settingsWidth": 300,
  "settingsRenderOption": "inline",
  "componentLayouts": [ ],
  "triggers": [ ],
  "actions": [ ]
},
"initialData": {
  "componentId": "h1-component-id",
  "customSettingsData": {
    "headingText": "Heading 1"
  },
  "nestedComponents": [ ]
}
}

```

2. Z folderu `assets` otworzyć plik `render.js` w preferowanym edytorze tekstów.
3. Zmienić zawartość pliku `render.html`, wprowadzając:

```

/* globals define */
define(['knockout', 'jquery', 'text!./render.html'],
function(ko, $, template) {
  'use strict';
  // -----
  // Define a Knockout ViewModel for your template
  // -----
  var SampleComponentViewModel = function(args) {
    var SitesSDK = args.SitesSDK;
    // create the observables -- this allows updated settings to
    // automatically update the HTML on the page
    this.headingText = ko.observable();

    //
    // Handle property changes from the Settings panel
    //
    this.updateCustomSettingsData = $.proxy(function(customData) {
      this.headingText(customData && customData.headingText);
    }, this);
    this.updateSettings = function(settings) {
      if (settings.property === 'customSettingsData') {
        this.updateCustomSettingsData(settings.value);
      }
    };
    // Register your updateSettings listener to receive
    // SETTINGS_UPDATED events

    SitesSDK.subscribe(SitesSDK.MESSAGE_TYPES.SETTINGS_UPDATED, $.proxy(
      this.updateSettings, this));

    //
    // Get the initial settings data for the component and apply it
    //
    SitesSDK.getProperty('customSettingsData',

```

```

this.updateCustomSettingsData);
    };

    // -----
    // Create a knockout based component implementation
    // -----
    var SampleComponentImpl = function(args) {
        // Initialize the custom component
        this.init(args);
    };
    // initialize all the values within the component from the given
argument values
    SampleComponentImpl.prototype.init = function(args) {
        this.createViewModel(args);
        this.createTemplate(args);
        this.setupCallbacks();
    };
    // create the viewModel from the initial values
    SampleComponentImpl.prototype.createViewModel = function(args) {
        // create the viewModel
        this.viewModel = new SampleComponentViewModel(args);
    };
    // create the template based on the initial values
    SampleComponentImpl.prototype.createTemplate = function(args) {
        // create a unique ID for the div to add, this will be passed to the
callback
        this.contentId = args.id + '_content_' + args.viewMode;
        // create a hidden custom component template that can be added to the
DOM
        this.template = '<div id="' + this.contentId + '"' +
            template +
            '</div>';
    };
    //
    // SDK Callbacks
    // setup the callbacks expected by the SDK API
    //
    SampleComponentImpl.prototype.setupCallbacks = function() {
        //
        // callback - render: add the component into the page
        //
        this.render = $.proxy(function(container) {
            var $container = $(container);
            // add the custom component template to the DOM
            $container.append(this.template);
            // apply the bindings
            ko.applyBindings(this.viewModel, $('#' + this.contentId)[0]);
        }, this);
        //
        // callback - dispose: cleanup after component when it is removed
from the page
        //
        this.dispose = $.proxy(function() {
            // nothing required for this sample since knockout disposal will

```

```

automatically clean up the node
    }, this);
  };
  // -----
  // Create the factory object for your component
  // -----
  var sampleComponentFactory = {
    createComponent: function(args, callback) {
      // return a new instance of the component
      return callback(new SampleComponentImpl(args));
    }
  };
  return sampleComponentFactory;
});

```

4. W folderze `assets` utworzyć nowy plik `render.html`, który będzie zwykłym szablonem HTML składnika.
5. W pliku `render.html` użyć następującej zawartości:

```

<h1 data-bind="text: headingText()">
</h1>

```

Folder `assets` składnika zawiera teraz trzy pliki.

- `render.html`
- `render.js`
- `settings.html`

Następnie należy dodać do strony nowy składnik H1 (punkt kontrolny 2).

## Dodawanie stylów CSS dla swojego składnika

Można dodać arkusz CSS określający styl domyślny składnika.

Aby dodać arkusz CSS, należy:

1. Dodać w folderze `assets` składnika plik `design.css` z następującą zawartością:

```

.h1-component-default-style .scs-component-content {
  font-family: "Helvetica Neue", "Helvetica", "Arial", sans-serif;
  font-size: 24px;
  color:red;
  font-weight: normal; }

```

2. Dodać do pliku `appinfo.json` deklarowany prefiks klasy stylu, która będzie używana dla składnika. Jeśli zostanie dodana właściwość `styleClassName` składnika `h1-component`, to — gdy składnik zostanie przeciągnięty na stronę — stylem domyślnym będzie `h1-component-default-style`. Nowa zawartość pliku `appinfo.json` przedstawia się następująco:

```

{
  "id": "h1-component-id",

  "settingsData": {

```

```

        "settingsHeight": 90,
        "settingsWidth": 300,
        "settingsRenderOption": "inline",
        "componentLayouts": [ ],
        "triggers": [ ],
        "actions": [ ]
    },
    "initialData": {
        "componentId": "h1-component-id",
        "styleClassName": "h1-component",
        "customSettingsData": {
            "headingText": "Heading 1"
        },
        "nestedComponents": [ ]
    }
}

```

3. Zaktualizować plik `render.js`, tak aby łądował plik CSS; w tym celu należy zastąpić linię 2 (`define(['knockout', 'jquery', 'text!./render.html'], function(ko, $, template) {})` następującą linią:

```

define(['knockout', 'jquery', 'text!./render.html', 'css!./design.css'],
function(ko, $, template, css) {

```

#### Uwaga:

Ponieważ plik `appinfo.json` został zmieniony, trzeba odświeżyć zawartość wyświetlaną w przeglądarce, po czym — aby wychwycić zmianę — ponownie dodać składnik do strony.

Sprawdzić, czy składnik pobiera teraz swoje dane domyślne z pliku `appinfo.json`.

## Dodawanie panelu ustawień umożliwiającego zmianę tekstu nagłówka

Aby dodać panel ustawień umożliwiający zmianę tekstu składnika H1, należy zmodyfikować plik `settings.html`.

Aby dodać panel ustawień umożliwiający zmianę tekstu nagłówka, należy:

1. Zmodyfikować plik `settings.html`, tak aby zawierał:

```

<!DOCTYPE html>
<html lang="en">

<head>
  <!-- only allow embedding of this iFrame in SCS -->
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <title>H1 Component</title>
  <!-- include sample apps styling -->
  <link href="/_sitescloud/renderer/app/sdk/css/app-styles.css"
rel="stylesheet">
  <!-- include supporting files -->
  <script type="text/javascript" src="/_sitescloud/renderer/app/apps/js/

```

```
knockout.min.js"></script>
  <script type="text/javascript" src="/_sitescloud/renderer/app/
apps/js/jquery.min.js"></script>
  <!-- include the Sites SDK -->
  <script type="text/javascript" src="/_sitescloud/
renderer/app/sdk/js/sites.min.js"></script>
</head>

<body data-bind="visible: true" style="display:none; margin:0px;
padding:0px;background:transparent;background-image:none;">
  <div class="scs-component-settings">
    <div>
      <!-- Heading Text -->
      <label id="headingTextLabel" for="headingText"
class="settings-heading" data-bind="text: 'Heading Text'"></label>
      <input id="headingText" data-bind="value: headingText"
placeholder="Heading" class="settings-text-box">
    </div>
  </div>
  <script type="text/javascript">
    // define the viewModel object
    var SettingsViewModel = function() {
      var self = this;

      // create the observables for passing data
      self.headingText = ko.observable();

      // create rest of viewModel
      self.saveData = false;

      // Get custom settings
      SitesSDK.getProperty('customSettingsData', function(data) {
        // update observable
        self.headingText(data.headingText);

        // now that viewModel is initialized and can start saving
data
        self.saveData = true;
      });

      // save whenever any updates occur, but don't save if data
hasn't been loaded into the form yet
      self.save = ko.computed(function() {
        var saveconfig = {
          'headingText': self.headingText()
        };
        // save data in page
        if (self.saveData) {
          SitesSDK.setProperty('customSettingsData', saveconfig);
        }
      }, self);
    };

    // apply the bindings
    ko.applyBindings(new SettingsViewModel());
```

```

</script>
</body>

```

- Wybrać swój składnik w konstruktorze serwisów, po czym nacisnąć przycisk **Ustawienia**.

Panel ustawień ma małe rozmiary; jest osadzany na górze panelu ustawień składnika. Należy poszukać sekcji zatytułowanej "Heading Text".

Aby sprawdzić, czy można zmienić tekst nagłówka (punkt kontrolny 4), należy:

Gdy w panelu ustawień zmieniamy tekst nagłówka, następuje aktualizacja składnika i jest wyświetlany nowy tekst.

## Aktualizacja motywu umożliwiająca użytkownikom wybór stylu składnika H1

Wraz z motywem można zarejestrować style składnika, dzięki czemu użytkownicy mogą je przełączać, korzystając z panelu ustawień lub karty **Styl** tab.

Aby zaktualizować motyw, umożliwiając użytkownikom wybór stylu składnika, należy:

- Dodać style do pliku `design.css` składnika. Poprzedzić każdy styl zarejestrowaną wartością `styleClassName` składnika, zdefiniowaną w pliku `appinfo.json`. Dla tego składnika prefiksem jest `h1-component`.

Zostały dodane dwa dodatkowe style: `h1-component-gothic-style` i `h1-component-courier-style`.

Nowa zawartość pliku `design.css` będzie teraz wyglądać następująco:

```

.h1-component-default-style .scs-component-content {
  font-family: "Helvetica Neue", "Helvetica", "Arial", sans-serif;
  font-size: 24px;
  color:red;
  font-weight: normal; }
.h1-component-gothic-style .scs-component-content {
  font-family: "Century Gothic","CenturyGothic","AppleGothic",sans-serif;
  font-size: 32px;
  font-weight: bold; }
.h1-component-courier-style .scs-component-content {
  font-family: "Courier";
  font-size: 32px;
  font-weight: bold; }

```

- Zarejestrować swoje style w pliku `design.json`. Plik ten można znaleźć w motywie używanym przez serwis. Należy przejść do plików motywów w folderze "designs", następnii do folderu "defaults", po czym dodać w pliku `design.json` sekcję dla swojego składnika.

Przedstawiony poniżej **pogrubiony** tekst stanowi przykład tego, co trzeba dodać:

```

"news-article": {
  "styles": [{
    "name": "News Article 1",
    "class": "news-article-default-
style"
  }],

```

```

        {
            "name": "News Article 2",
            "class": "news-article-
style-1"
        }
    ],
    "h1-component": {
        "styles": [{
            "name": "Plain",
            "class": "h1-component-default-style"
        }, {
            "name": "Courier",
            "class": "h1-component-courier-style"
        }, { "name": "Gothic",
            "class": "h1-component-gothic-style"
        }
    ]
    },
    "componentIcons": {
        "scs-socialbar": {
            "icons": [

```

Nazwy "Plain", "Courier", "Gothic", które zostały dodane w pliku `design.json`, pojawią się panelu ustawień składnika na karcie "Styl", jak poniżej. Wybór nazwy spowoduje zastosowanie odpowiedniego stylu ("*h1-component-default-style*", "*h1-component-courier-style*", "*h1-component-gothic-style*") do składnika.

## Tworzenie prostego składnika z użyciem szablonów HTML

Można utworzyć składnik Oracle Content Management używający — do renderowania swojej zawartości na podstawie danych wprowadzanych w panelu ustawień — kodu HTML i CSS z szablonami Mustache JS.

Udostępniany panel może zezwalać na edytowanie danych, które będą renderowane w szablonie HTML.

W przykładach dla tej procedury są używane szablony Mustache JS.

Prosty składnik z użyciem szablonów HTML można utworzyć, wykonując następujące dwie procedury:

1. [Tworzenie składnika HTML](#)
2. [Dodawanie danych, dostarczanych przez użytkownika, do składnika HTML](#)

## Tworzenie składnika HTML

W katalogu składników Oracle Content Management można utworzyć nowy składnik lokalny, po czym skonstruować składnik HTML, dodając, a następnie zmieniając pliki w folderze zasobów składnika.

Aby utworzyć składnik HTML, należy:

1. Utworzyć składnik, po czym nazwać go HTML Component.



Zob. [Tworzenie nowego składnika lokalnego](#).

2. W składniku HTML dodać plik `mustache.min.js` do folderu `assets`.
3. W folderze `assets` utworzyć nowy plik o nazwie `render.html`, który będzie zawierał kod HTML z treścią składnika. W przypadku poniższego przykładu utworzyć plik `render.html` z następującą zawartością:

```
<ul class="wrapper">
  <li class="box">
    <h1 class="title">One</h1>
    <p class="text"> Lorem ipsum dolor sit amet, consectetur adipiscing
elit, sed do eiusmod tempor incididunt ut labore et dolore magna
aliqua.</p>
  </li>
  <li class="box">
    <h1 class="title">Two</h1>
    <p class="text"> Lorem ipsum dolor sit amet, consectetur adipiscing
elit, sed do eiusmod tempor incididunt ut labore et dolore magna
aliqua.</p>
  </li>
  <li class="box">
    <h1 class="title">Three</h1>
    <p class="text">Lorem ipsum dolor sit amet, consectetur adipiscing
elit, sed do eiusmod tempor incididunt ut labore et dolore magna
aliqua.</p>
  </li>
  <li class="box">
    <h1 class="title">Four</h1>
    <p class="text">Lorem ipsum dolor sit amet, consectetur adipiscing
elit, sed do eiusmod tempor incididunt ut labore et dolore magna
aliqua.</p>
  </li>
</ul>
```

4. W folderze `assets` utworzyć nowy plik o nazwie `design.css`, który będzie zawierał arkusz CSS dla składnika. W przypadku poniższego przykładu dodać jako zawartość pliku `design.css` następujące wiersze:

```
.wrapper {
  text-align: center;
}
.box {
  display: inline-block;
  position: relative;
  width: 200px;
  height: 200px;
  padding: 0px 10px 10px 10px;
  background: transparent;
  border-width: 1px;
  border-style: solid;
  border-radius: 5px;
  border-color: #CCCCCC;
  z-index: 0;
  margin: 2px 2px 2px 2px;
```

```

    transition: all .15s ease-in-out;
  }
  .box:hover {
    background: #9CC;
    z-index: 100;
    transform: scale(1.2,1.2);
    box-shadow: 0 5px 10px 0 rgba(0,0,0,.2);
  }
  .title {
    color:red;
  }
  .text {
    color:#555555;
  }

```

5. Z folderu `assets` otworzyć plik `render.js`, po czym zmienić jego zawartość na poniższą. Niezależnie od tego, który kod HTML i CSS był używany w etapach poprzedzających, poniższy plik `render.js` będzie renderował kod HTML i CSS na stronie:

```

/* globals define */
define(['jquery', './mustache.min', 'text!./render.html', 'css!./
design.css'], function($, Mustache, template, css) {
  'use strict';

  // -----
  // Create a Mustache-based component implementation
  // -----
  var SampleComponentImpl = function(args) {
    this.SitesSDK = args.SitesSDK;

    // Initialize the custom component
    this.createTemplate(args);
    this.setupCallbacks();
  };
  // create the template based on the initial values
  SampleComponentImpl.prototype.createTemplate = function(args) {
    // create a unique ID for the div to add, this will be passed
to the callback
    this.contentId = args.id + '_content_' + args.viewMode;
    // create a hidden custom component template that can be added
to the DOM
    this.template = '<div id="' + this.contentid + '">' +
      template +
      '</div>';
  };
  SampleComponentImpl.prototype.updateSettings = function(settings)
{
  if (settings.property === 'customSettingsData') {
    this.update(settings.value);
  }
};
  SampleComponentImpl.prototype.update = function(data) {
    this.data = data;
    this.container.html(Mustache.to_html(this.template, this.data));

```

```

};
//
// SDK Callbacks
// setup the callbacks expected by the SDK API
//
SampleComponentImpl.prototype.setupCallbacks = function() {
  //
  // callback - render: add the component into the page
  //
  this.render = $.proxy(function(container) {
    this.container = $(container);

    this.SitesSDK.getProperty('customSettingsData', $.proxy(this.update,
    this));
    }, this);
  //
  // callback - SETTINGS_UPDATED: retrieve new custom data and re-render
  the component
  //

  this.SitesSDK.subscribe(this.SitesSDK.MESSAGE_TYPES.SETTINGS_UPDATED, $.pr
  oxy(this.updateSettings, this));
  //
  // callback - dispose: cleanup after component when it is removed
  from the page
  //
  this.dispose = $.proxy(function() {
    // nothing required
  }, this);
};
// -----
// Create the factory object for your component
// -----
var sampleComponentFactory = {
  createComponent: function(args, callback) {
    // return a new instance of the component
    return callback(new SampleComponentImpl(args));
  }
};
return sampleComponentFactory;
});

```

Aby sprawdzić, czy składnik HTML został skonstruowany (punkt kontrolny 1), należy:

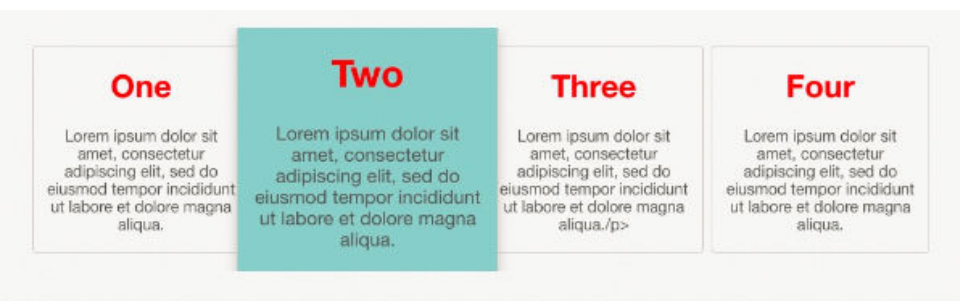
1. Sprawdzić, czy w folderze `assets` składnika znajduje się pięć plików pokazanych poniżej.
  - `design.css`
  - `mustache.min.js`
  - `render.html`
  - `render.js`
  - `settings.html`

- W serwisie testowym dodać do strony nowy składnik HTML. W edytorze — w trybie edycji i podglądu — powinien zostać wyświetlony na stronie składnik w następujący sposób:

Tryb edycji



Tryb podglądu



## Dodawanie danych, dostarczanych przez użytkownika, do składnika HTML

Można dodać panel **Ustawienia niestandardowe** dostarczający pól dla wszystkich elementów danych definiowanych w składniku HTML. Użytkownik, korzystając z panelu **Ustawienia niestandardowe**, może wprowadzić wszystkie wartości dla instancji składnika na stronie.

Aby dodać dane, dostarczane przez użytkownika, do składnika HTML, należy:

- Zaktualizować szablon HTML w swoim pliku `render.html`, tak aby zawierał wszystkie oczekiwane elementy danych. W poniższym przykładzie jest używany język szablonów Mustache JS, tak że elementy danych muszą być dodawane przy użyciu składni `{{ i }}`, jak pokazano:

```
<ul class="wrapper">
  <li class="box">
    <h1 class="title">{{title1}}</h1>
    <p class="text">{{text1}}</p>
  </li>
  <li class="box">
    <h1 class="title">{{title2}}</h1>
    <p class="text">{{text2}}</p>
  </li>
</ul>
```

```

</li>
<li class="box">
  <h1 class="title">{{title3}}</h1>
  <p class="text">{{text3}}</p>
</li>
<li class="box">
  <h1 class="title">{{title4}}</h1>
  <p class="text">{{text4}}</p>
</li>
</ul>

```

2. Określić wartości domyślne dla następujących pól danych, zmieniając w swoim składniku HTML zawartość pliku `appinfo.json`:

```

{
  "id": "html-component-id",

  "settingsData": {
    "settingsHeight": 600,
    "settingsWidth": 300,
    "settingsRenderOption": "dialog",
    "componentLayouts": [],
    "triggers": [],
    "actions": []
  },
  "initialData": {
    "styleClassName": "html-component",
    "customSettingsData": {
      "title1": "One",
      "title2": "Two",
      "title3": "Three",
      "title4": "Four",
      "text1": "Lorem ipsum dolor sit amet, consectetur adipiscing elit,
sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.",
      "text2": "Lorem ipsum dolor sit amet, consectetur adipiscing elit,
sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.",
      "text3": "Lorem ipsum dolor sit amet, consectetur adipiscing elit,
sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.",
      "text4": "Lorem ipsum dolor sit amet, consectetur adipiscing elit,
sed do eiusmod tempor incididunt ut labore et dolore magna aliqua."
    },
    "nestedComponents": []
  }
}

```

3. Dodać panel **Ustawienia niestandardowe** wyszukujący wartości danych w pliku `render.html` szablonu HTML w celu utworzenia ustawień, które użytkownik będzie mógł zmienić w edytorze. Zmienić zawartość pliku `settings.html`, wprowadzając:

```

<!DOCTYPE html>
<html lang="en">
<head>
  <!-- only allow embedding of this iFrame in SCS -->
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

```

```

<title>H1 Mustache Component</title>

<!-- include sample apps styling -->
<link href="/_sitescloud/renderer/app/sdk/css/app-styles.css"
rel="stylesheet">

<!-- include supporting files -->
<script type="text/javascript" src="/_sitescloud/renderer/app/
apps/js/knockout.min.js"></script>
<script type="text/javascript" src="/_sitescloud/renderer/app/
apps/js/jquery.min.js"></script>

<!-- include the Sites SDK -->
<script type="text/javascript" src="/_sitescloud/
renderer/app/sdk/js/sites.min.js"></script>
</head>
<body data-bind="visible: true" style="display:none; margin:0px;
padding:0px;background:transparent;background-
image:none;">
  <!-- ko if: initialized() -->
  <div class="scs-component-settings">
    <div>
      <!-- Width -->
      <label id="headingTextLabel"
for="headingText" class="settings-heading" data-bind="text:
'Heading Text'"></label>
      <input id="headingText" data-bind="value:
headingText" placeholder="Heading" class="settings-
text-box">
    </div>
  </div>
  <div data-bind="setSettingsHeight: true"></div>
  <!-- /ko -->
  <!-- ko ifnot: initialized() -->
  <div data-bind="text: 'waiting for initialization to
complete'"></div>
  <!-- /ko -->
  <script type="text/javascript">

    // set the iFrame height when we've fully rendered
    ko.bindingHandlers.scsCompComponentImpl = {
      init: function (element, valueAccessor,
allBindings, viewModel, bindingContext) {
        var body = document.body,
            html =
document.documentElement;

        SitesSDK.setHeight(Math.max(
            body.scrollHeight,
            body.offsetHeight,
            html.clientHeight,
            html.scrollHeight,
            html.offsetHeight));
      }
    }
  </script>

```

```

    };

    // define the viewModel object
    var SettingsViewModel = function () {
        var self = this;

        // create the observables for passing data
        self.headingText = ko.observable('Heading 1');

        // create rest of viewModel
        self.initialized = ko.observable(false);
        self.saveData = false;

        // Get custom settings
        SitesSDK.getProperty('customSettingsData',

function (data) {
            //update observable
            self.headingText(data.headingText);

            // note that viewModel is initialized
            self.initialized(true);
            self.saveData = true;
        });

        // save whenever any updates occur
        self.save = ko.computed(function () {
            var saveconfig = {
                'headingText':
self.headingText()
            };

            // save data in page
            if (self.saveData) {

                SitesSDK.setProperty('customSettingsData', saveconfig);
            }
        }, self);
    };

    // apply the bindings
    ko.applyBindings(new SettingsViewModel());
</script>
</body>

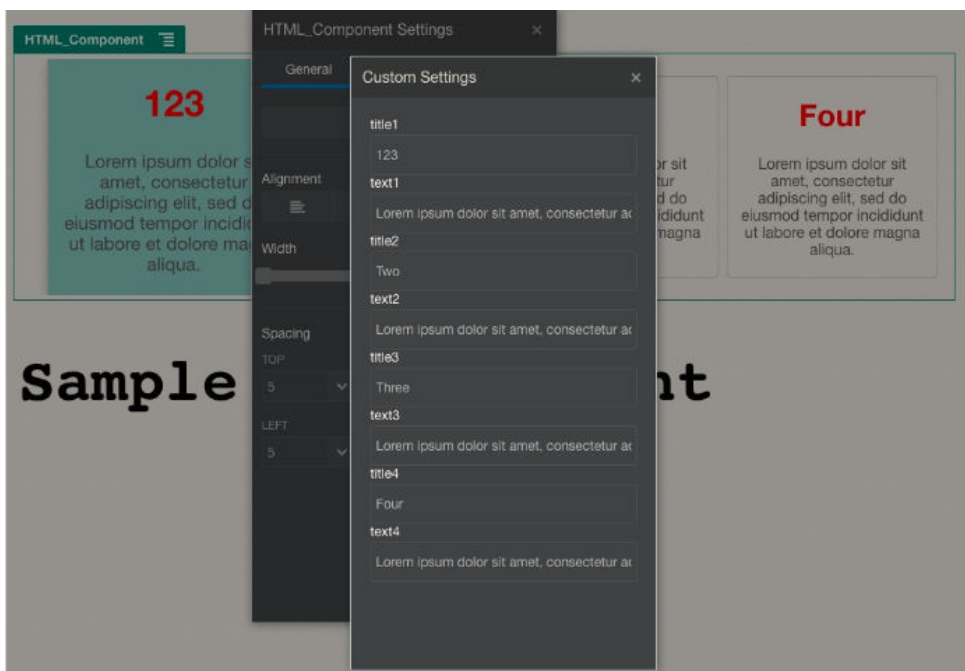
```

 **Uwaga:**

Ponieważ plik `appinfo.json` został zmieniony, trzeba odświeżyć zawartość wyświetlaną w przeglądarce, po czym — aby wychwycić zmianę — ponownie dodać składnik do strony.

Aby sprawdzić, czy składnik pobiera teraz swoje dane domyślne z pliku `appinfo.json` i czy można zmienić **Ustawienia niestandardowe** (punkt kontrolny 2):

1. Gdy plik `appinfo.json` zostanie zmieniony, należy pamiętać by odświeżyć zawartość wyświetlaną w przeglądarce, po czym — aby wychwycić zmianę — ponownie dodać składnik do strony.
2. Otworzyć panel **Ustawienia niestandardowe** w składniku HTML, po czym wprowadzić dane.



3. Sprawdzić, czy składnik Oracle Content Management używa — do renderowania swojej zawartości na podstawie danych wprowadzanych w panelu **Ustawienia niestandardowe** — kodu HTML i CSS z szablonami Mustache JS.

## Porównanie składników lokalnych ze składnikami odległymi

Różni się sposób implementacji składników lokalnych i składników odległych.

W poniższej tabeli wymieniono różnice w sposobie implementacji tych składników. Może okazać się ona pomocna w ustaleniu, czy ma zostać użyty składnik lokalny czy składnik odległy renderowany w ramce wstawkowej.

Implementacja składnika lokalnego	Implementacja składnika odległego
Integruje złożony interfejs użytkownika, ukierunkowany na zawartość, z usługą Oracle Content Management	Integruje algorytm aplikacji z usługą Oracle Content Management
Wykonuje skrypt JavaScript do renderowania zawartości na stronie	Używa znacznika HTML <code>&lt;iframe&gt;</code> do renderowania zawartości na stronie
Używa dowolnego stosu technologii JavaScript	Może używać dowolnej technologii, nie tylko JavaScript



Implementacja składnika lokalnego	Implementacja składnika odległego
Korzysta ze stosu JavaScript z Oracle Content Management oraz może wielokrotnie używać składników Oracle Content Management	Nie integruje się ze stosem JavaScript z Oracle Content Management
Dodaje zależności zasobów z dokumentów z folderu <code>/assets</code> , przechowywanych na serwerze Oracle Content Management	Wymaga adresów URL HTTPS oraz odpowiednich certyfikatów, zapewniających dostęp do wszystkich zasobów
Składnik jest hostowany z szablonem i jest publikowany przez usługę Oracle Content Management niezależnie od motywu	Wymaga hostowanego serwera warstwy pośredniej dla punktów końcowych adresu URL
Ponieważ kod jest wykonywany w obrębie strony, to w przypadku awarii kodu może nastąpić awaria strony	Strona jest renderowana niezależnie od zawartości ramek wstawkowych, dzięki czemu strona nie wystąpi awaria strony, nawet jeśli składnik zostanie uszkodzony
Jeśli składnik jest powoli renderowany, może to spowodować renderowanie strony	Strona jest ładowana niezależnie od zawartości ramek wstawkowych, dzięki czemu strona jest ładowana możliwie szybko, a następnie jest wczytywana zawartość wszelkich ramek wstawkowych

 **Uwaga:**

Wprawdzie sposób renderowania składników lokalnych i składników odległych różni się, lecz implementacja panelu "Ustawienia" w konstruktorze serwisów wygląda tak samo. Panel "Ustawienia" dla obu typów składników jest renderowany przy użyciu ramek wstawkowych. Panel ten używa tej samej biblioteki JavaScript SDK, umożliwiającej (w obu przypadkach) komunikację między składnikami i między domenami.

## Renderowanie ustawień składnika

Do renderowania ustawień składnika można używać adresu URL ustawień składnika i opcji renderowania ustawień składnika.

### URL ustawień składnika

URL ustawień składnika jest renderowany w ramce wstawkowej i jest wywoływany z użyciem parametrów określających konkretne ustawienia składnika umieszczanego na stronie. Format adresu URL ustawień jest następujący:

```
{Component Settings URL}?instance=<app-  
instance>&width=<width>&currCompId=<id of the app associated with the  
settings panel>&locale=<locale>
```

Nazwa	Typ	Opis
Component Settings URL	URL	URL ustawień składnika

Nazwa	Typ	Opis
width	Liczba	Szerokość ramki wstawkowej ustawień w pikselach
currCompId	Napis	Bieżący ID składnika edytowanego przy użyciu panelu "Ustawienia"
locale	Napis	Bieżące ustawienia narodowe serwisu-hosta (konstruktor serwisów). Format ma postać <język>_<dialekt>. Przykład: En_us.

### Opcje renderowania ustawień składnika lokalnego

W panelu "Ustawienia" są dla składnika lokalnego dostępne trzy opcje renderowania go w ramce wstawkowej, bazujące na rozmiarze i stopniu złożoności ramki wstawkowej. Każda z tych opcji jest określana we właściwości `settingsRenderOption`.

#### Uwaga:

Opcje te są dostępne tylko dla składników lokalnych. Panele "Ustawienia" składników odległych są zawsze renderowane z użyciem opcji "dialog".

- inline
  - Opcji tej należy używać, jeśli użytkownik ma wprowadzać niewielką liczbę właściwości.
  - Ramka wstawkowa zostanie wstawiona na karcie **Ogólne** w miejscu zwykłego przycisku używanego do przechodzenia do panelu "Ustawienia".
  - Zapewnia najbardziej zintegrowane rozwiązanie, wymagając najmniejszej liczby działań użytkownika, lecz ma ograniczone miejsce.
- panel
  - Opcji tej należy używać, gdy lista właściwości jest dłuższa, lecz nadal można je sensownie wyświetlić w 300-pikselowym obszarze standardowego panelu "Ustawienia".
  - Ramka wstawkowa wsunie się do widoku i pojawi się przycisk "Wstecz" pozwalający użytkownikowi wrócić na kartę "Ogólne".
  - Zapewnia zintegrowane rozwiązanie, umożliwiające interakcję z ramką wstawkową na stronie.
- dialog
  - Tryb domyślny do obsługi bardziej ogólnych układów ustawień, wymagających złożonego interfejsu użytkownika.
  - Wyświetla ramkę wstawkową w modalnym oknie dialogowym na stronie.

## Implementacja składnika lokalnego

Instancja składnika jest tworzona przy użyciu obiektu "Component Factory".

Wpisem obiektu "Component Factory" w pliku rejestracji składnika jest `initData.componentFactory`. Stanowi on odwołanie do pliku JavaScript używanego przez RequireJS do załadowania obiektu "Component Factory". Musi zwracać obiekt JavaScript implementujący interfejs `componentFactory`.

Obiekt "Component Factory" musi implementować interfejs API `customComponentFactory.createComponent(arg)`, tworzący poszczególne instancje składnika.

W `arg` są zawarte następujące ustawienia:

- `SitesSDK`: Oracle Content Management Sites SDK.
- `id`: Unikatowy ID (GUID) składnika dodawanego do strony.
- `viewMode`: Bieżący tryb renderowania strony. Jeśli strona jest edytowana, wartością jest "Edit". Jeśli jest wyświetlany podgląd strony, wartością jest "Navigate". W trybie wykonawczym, gdy serwis został opublikowany, wartość ta jest niezdefiniowana. Można udostępniać różne implementacje, bazując na funkcjonalności eksponowanej dla poszczególnych trybów. Na przykład, gdy strona działa w trybie edycji, łącza nie powinny być aktywne.

#### Uwaga:

Używanie bibliotek JQuery lub Knockout dla składnika nie jest wymagane, ale — zamierzając korzystać z funkcji Oracle Content Management, takich jak zagnieżdżanie składników — trzeba używać biblioteki Knockout w wersji udostępnianej przez Oracle Content Management. Wersja ta zapewnia rozbudowaną rejestrację składników oraz rozszerzone procedury obsługi, które w przeciwnym razie byłyby niedostępne.

W zakresie samego składnika biblioteka SDK jest przekazywana, gdy jest tworzona instancja składnika, dzięki czemu składnik może się komunikować z funkcjami cyklu życia strony. Funkcje cyklu życia muszą zostać zaimplementowane przez składnik — są wywoływane przez Oracle Content Management w celu wyrenderowania składnika na stronie.

Są udostępniane wymagane i opcjonalne API umożliwiające implementację składnika.

#### Wymagane API

`customComponent.render(container)`: Zleca składnikowi, aby sam się wstawił do udostępnianego elementu kontenera DOM.

- `container`: Element kontenera DOM dla kodu HTML składnika niestandardowego.

#### Opcjonalne API

`customComponent.dispose()`: Wywoływane, gdy składnik jest usuwany ze strony. Umożliwia składnikowi usunięcie wszystkich zasobów, które już nie są potrzebne.

## Klasy stylów składników

Można utworzyć zdefiniowaną listę stylów, które mogą być stosowane przez użytkowników do składników.

Utrzymywanie predefiniowanej listy stylów dla składnika opiera się na tym samym modelu, co definiowanie klas stylów dla składników dostarczanych przez Oracle Content

Management przy użyciu plików projektu motywu. Klasy stylów niestandardowych określa się w pliku `appinfo.json` dla składnika.

Dodatkowe style definiuje się w plikach `design.css` i `design.json`. Plik `json` zawiera odwzorowania na `css`, zaś plik `css` dostarcza szczegóły poszczególnych klas.

Plik `design.json` ma następującą strukturę dla składników:

```
{
  "componentStyles": {
    "scs-image": {
      "styles": []
    },
    "scs-map": {
      "styles": []
    },
    "scs-title": {
      "styles": []
    },
    "scs-paragraph": {
      "styles": []
    },
    "scs-txt": {
      "styles": []
    },
    "scs-divider": {
      "styles": []
    },
    "scs-button": {
      "styles": []
    },
    "scs-app": {
      "styles": []
    },
    "scs-spacer": {
    },
    "scs-gallery": {
      "styles": []
    },
    "scs-youtube": {
      "styles": []
    },
    "scs-socialbar": {
      "styles": []
    },
    "scs-document": {
      "styles": []
    }
  }
}
```

Każdy z wpisów `"styles": []` może zawierać listę stylów dla konkretnego składnika. Wpis `"name"` może się odwoływać do wbudowanego zlokalizowanego napisu lub do

określonej wartości, która ma zostać użyta. Na przykład składnik "tytuł" udostępnia następujące style domyślne:

```
{
  "styles": [{
    "name": "COMP_STYLE_FLAT",
    "class": "scs-title-default-style"
  },
  {
    "name": "COMP_STYLE_HIGHLIGHT",
    "class": "scs-title-style-2"
  },
  {
    "name": "COMP_STYLE_DIVIDER",
    "class": "scs-title-style-3"
  }
]
}
```

Wartości "name" są odwzorowywane na wyrazy wyświetlane w interfejsie użytkownika, jak poniżej:

```
"COMP_STYLE_FLAT": "Flat",
"COMP_STYLE_HIGHLIGHT": "Highlight",
"COMP_STYLE_DIVIDER": "Divider",
```

Plik `css` udostępnia definicje wartości "class".

```
.scs-title-default-style {
  color: #333333;
  display: block;
  font-family: "Helvetica Neue", "Helvetica", "Arial", sans-serif;
  font-size: 24px;
  font-weight: normal;
}
```

Na przykład w pliku `design.json` motywu można dodać wpisy dla składnika, oparte na wartości `initialData.componentId` zdefiniowanej w pliku `components.json`:

```
"componentId": "news-article"
```

Odpowiadającymi wpisami w pliku `design.json` mogą być następujące:

```
"componentStyles": {
  "news-article": {
    "styles": [{
      "name": "News Article 1",
      "class": "news-article-default-style"
    },
    {
      "name": "News Article 2",
      "class": "news-article-style-1"
    }
  ]
}
```

```
    },  
  } ]
```

Odpowiadającymi wpisami w pliku `design.css` mogą być następujące:

```
.news-article-default-style .scs-image {...}  
.news-article-style-1 .scs-image {...}
```

## Jak nadawać styl składnikom wbudowanym

Można utworzyć własny styl dla składników wbudowanych w Oracle Content Management, przesłaniając i rozszerzając style wbudowane.

Składniki wbudowane uzyskują swoją wizualną stylistykę z dwóch miejsc. Są to:

- `comp.css` — wbudowany plik CSS określający *podstawowy* wygląd poszczególnych składników
- `design.css` — plik CSS będący częścią motywu używanego przez serwis

W pliku `design.css` można przesłaniać i rozszerzać wbudowane style `comp.css`, tworząc nowy wygląd. W motywie plik `design.css` znajduje się w katalogu `designs/default`.

W następujących tematach opisano klasy z pliku `comp.css` wspólne dla wszystkich wbudowanych składników oraz przedstawiono procedurę definiowania motywu:

- [Stosowanie stylów do składników — podstawy](#)
- [Stosowanie stylów do określonych składników](#)
- [Ustawianie właściwości składnika](#)

## Stosowanie stylów do składników — podstawy

Wszystkie wbudowane składniki platformy Sites mają podobną strukturę klas CSS.

Każdy ze składników ma trzy następujące klasy CSS stosowane do najbardziej zewnętrznego składnika `<div>`:

```
scs-component scs-type design-style
```

Klasa `type` określa typ składnika (na przykład `image`, `gallery` lub `divider`). Klasa `design-style` jest wybraną klasą stylu dla składnika, zdefiniowaną w pliku motywu.

### Style motywu — podstawy

Plik `design.json` motywu zawiera wykaz wszystkich stylów (`frame`, `shadow`, `highlighted` itd.), które mogą zostać zastosowane do poszczególnych typów składników (takich jak `button` lub `image`). Każdy styl ma wyświetlaną nazwę (`name`) i nazwę klasy (`class`). Wyświetlana nazwa jest pokazywana na karcie **Styl** panelu **Ustawienia**

. Nazwa klasy odwołuje się do selektora CSS, zawartego w pliku `design.css` motywu. Na przykład wpis dla składnika "Przycisk" (`button`) może mieć następującą postać:

```
"scs-button": {
  "styles": [{
    "name": "COMP_STYLE_ALTA_SMALL",
    "class": "scs-button-default-style"
  },
  {
    "name": "COMP_STYLE_ALTA_LARGE",
    "class": "scs-button-style-2"
  },
  {
    "name": "COMP_STYLE_SIMPLE",
    "class": "scs-button-style-3"
  }
]
},
```

Nazwy wbudowanych składników można tłumaczyć, a zatem jest pokazywany klucz pozwalający uzyskać nazwę stylu z pakietu zasobów. Jeśli do strony dodamy składnik `button`, a następnie **Ustawienia > Styl styl Prosty**, to plik `design.json` powiąże wyświetlaną nazwę **Prosty** (klucz `COMP_STYLE_SIMPLE`) z nazwą klasy `scs-button-style-3`. Przycisk będzie renderowany z użyciem następujących klas:

```
scs-component scs-button scs-button-style-3
```

Jeśli dla danego składnika nie wybierzemy żadnego stylu, zostanie użyty styl domyślny `scs-type-default-style`. W poprzednim przykładzie przycisk będzie renderowany z użyciem następujących klas:

```
scs-component scs-button scs-button-default-style
```

### Styl `scs-component-content`

Dla każdego wbudowanego składnika, zawartego we wspomnianym wcześniej składniku `scs-component` <div>, istnieje składnik `content` <div> z klasą CSS `scs-component-content`. Inaczej mówiąc:

```
scs-component scs-type design-style
scs-component-content
```

W pliku `design.css` klasa `scs-component-content` jest często używana do stylizacji ramki otaczającej składnik (na przykład wyświetlenia jej krawędzi lub wprowadzenia cienia).

Warto zauważyć, że we wbudowanym pliku `comp.css` wspólna klasa `scs-component-content` jest zdefiniowana z użyciem m.in. właściwości CSS `position: relative` i `display: inline-block`.

Podczas gdy styl `scs-component-content` jest przydatny do stylizacji ramki otaczającej dany składnik, to — aby przeprowadzić pełną stylizację składnika — trzeba użyć właściwych dla niego klas. Zob. [Stosowanie stylów do określonych składników](#).

## Stosowanie stylów do określonych składników

Style można stosować do składników "obraz", "przycisk", "dokument", "akapit", "tytuł", "mapa" i innych.

### Składnik "obraz"

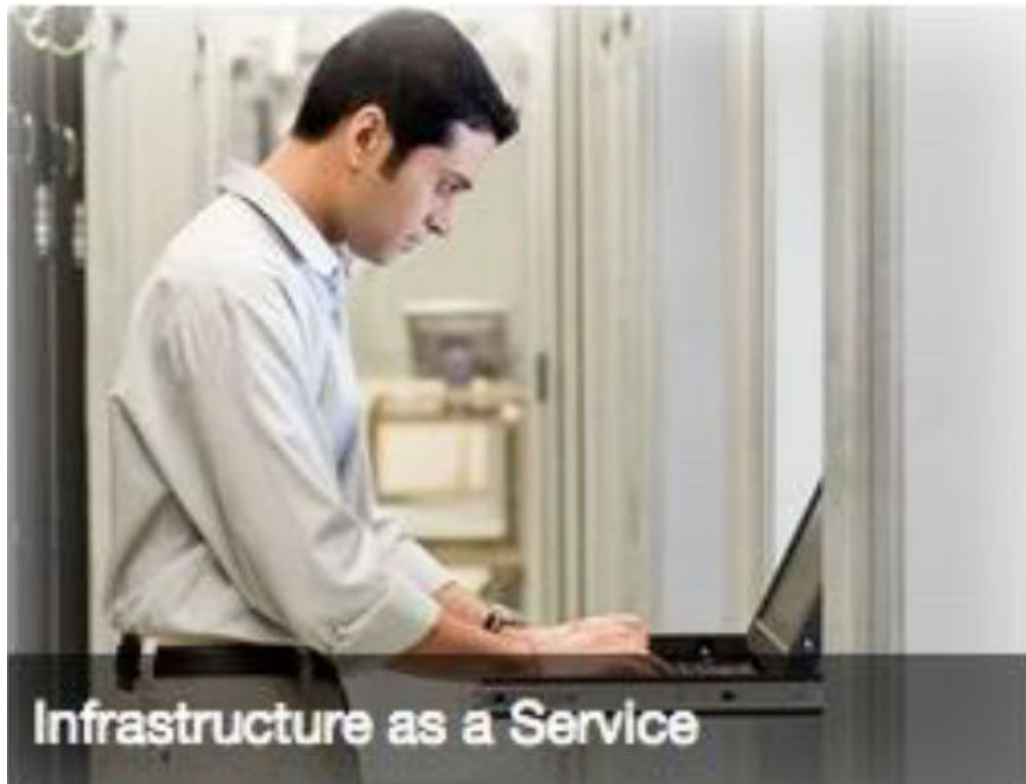
Składnik "obraz" ma pod klasą `scs-component-content` następującą strukturę klas CSS:

```
scs-image-container  
scs-image-link  
scs-image-image  
scs-image-caption
```

Klasa `scs-image-image` jest stosowana do samego znacznika `<img>`. Klasa `scs-image-caption` służy do przypisania stylu do podpisu (nazwy opisowej), jeśli podpis występuje.

Klasa `scs-image-link` występuje tylko wtedy, jeśli do obrazu jest dołączone łącze. Zazwyczaj klasa ta ani klasa `scs-image-container` nie wymaga stosowania stylów niestandardowych.

Domyślnie podpis obrazu jest renderowany jako półprzezroczysta nakładka, rozciągająca się na dole obrazu.





```
.scs-image .scs-image-caption {  
  position: absolute;  
  left: 0px;  
  bottom: 0px;  
  right: 0px;  
  background-color: rgba(0, 0, 0, 0.54);  
  padding: 0.5em;  
  color: #FFFFFF;  
}
```

Chcąc umieścić podpis na górze obrazu i zmienić kolory, należy dodać dla składnika "obraz" własny styl w pliku `design.json`, a następnie zdefiniować kod CSS w pliku `design.css`.



```
.scs-image-style-17 .scs-image-caption {  
  position: absolute;  
  top: 0px;  
  height: 35px;  
  font-weight: bold;  
  background-color: rgba(122, 213, 256, 0.54);  
  color: #515151;  
}
```

## Składnik "przycisk"

Składnik "przycisk" ma następującą strukturę klas:

```
scs-button-button
  scs-button-text
```

Klasa `scs-button-button` to klikalny element `<div>`, stylizowany na przycisk. Klasa `scs-button-text` służy do stosowania stylów do tekstu zawartego w przycisku.

Na przykład można przetestować modyfikowanie postaci składnika "przycisk" przez dodanie do niego dodatkowego stylu w pliku `design.json`, a następnie zdefiniowanie kodu CSS w pliku `design.css`.



```
.design-style .scs-button-button {
  background-image: linear-gradient(
    to top, #E3E7E9 0%, #E7EBED 50%, #F1F3F3 100%);
  border: 1px solid #c4ced7;
  color: #000000;
}

.design-style .scs-button-button:hover {
  background: #f7f8f9;
  border: 1px solid #c4ced7;
  color: #0572ce;
}

.design-style .scs-button-button:active {
  background: #0572ce;
  border: 1px solid #0572ce;
  color: #ffffff;
}
```



```
.scs-button-style-4 .scs-button-button {
  background-image: radial-gradient(
    red, yellow, green
```

```
);  
border: 1px solid #c4ced7;  
color: #000000;  
}  
  
.scs-button-style-4 .scs-button-button:hover {  
background: #f7f8f9;  
border: 1px solid #c4ced7;  
color: #0572ce;  
}  
  
.scs-button-style-4 .scs-button-button:active {  
background: #0572ce;  
border: 1px solid #0572ce;  
color: #ffffff;  
}
```

### Dokument

Składnik "dokument" ma następującą strukturę klas:

```
scs-document-container  
  scs-document-cap  
    scs-document-title  
    scs-document-desc
```

Klasa `scs-document-container` otacza przeglądarkę dokumentów i zazwyczaj nie jest do niej stosowany żaden styl.

### Galeria

Składnik "galeria" ma jedną klasę otaczającą używany składnik "suwak JSSOR":

```
scs-gallery-container
```

Suwak JSSOR używa kilku klas, do których można stosować style:

```
jssorb14 (navigator)  
jssora02l (left arrow)  
jssora02r (right arrow)  
jssort07 (thumbnails)
```

### Siatka galerii

Klasy używane dla składnika "siatka galerii" zależą od układu i przycinania, wybranych w panelu **Ustawienia**:

```
scs-gallerygrid-container scs-gallerygrid-layout  
  scs-gallerygrid-cell  
    scs-image (multiple)
```

W zależności od ustawień przycinania i układu, wybranych dla składnika "siatka galerii", wartością właściwości `layout` będzie `stretch` (rozciąganie), `crop` (przycinanie), `fit` (dopasowanie) lub `flowing` (przepływ).

Klasa `scs-gallerygrid-cell` występuje tylko dla układów "kolumna".

### Pasek społecznościowy

Składnik "pasek społecznościowy" ma następującą strukturę klas:

```
scs-socialbar-container
  scs-socialbar-icon
```

Klasa `scs-socialbar-icon` jest stosowana do każdego ze znaczników `<img>` w elemencie "pasek społecznościowy".

### Akapit

Składnik "akapit" ma tylko jedną klasę otaczającą faktyczny tekst akapitu:

```
scs-paragraph-text
```

Na przykład, aby tekst prezentowany za pomocą składnika "akapit" miał postać tekstu grawerowanego na metalu, należy dodać dodatkową klasę stylu w pliku `design.json`, a następnie zdefiniować dla niej kod CSS w pliku `design.css`.



```
.scs-paragraph-style-7 {
  font-size: 24px;
  font-family: Arial, Helvetica, sans-serif;
  font-weight: 700;
  padding: .3em;
  color: #000000;
  background: #666666;
  text-shadow: 0px 1px 1px #ffffff;
}
```

Chcąc zaś uzyskać bardziej fantazyjny efekt, można użyć na przykład kodu z poniższego przykładu..



```
.scs-paragraph-style-8 {  
  padding: 20px;  
  margin: 10px;  
  background: #ff0030;  
  color: #fff;  
  font-size: 21px;  
  font-weight: bold;  
  line-height: 1.3em;  
  border: 2px dashed #fff;  
  border-radius: 10px;  
  box-shadow: 0 0 0 4px #ff0030, 2px 1px 6px 4px rgba(10, 10, 0, 0.5);  
  text-shadow: -1px -1px #aa3030;  
  font-weight: normal;  
}
```

### Tytuł

Składnik "tytuł" także ma tylko jedną klasę otaczającą faktyczny tekst:

```
scs-title-text
```

### Mapa

Składnik "mapa" ma tylko jedną klasę otaczającą realizację mapy:

```
scs-map-content
```

To klasy tej zazwyczaj nie stosuje się własnych stylów.

### Niestandardowy składnik lokalny

Składnik "niestandardowy składnik lokalny" ma tylko jedną klasę otaczającą faktyczny składnik:

```
scs-custom-component-wrapper
```

Użytkownik ma pełną kontrolę nad stylami CSS, które mają zostać użyte do renderowania widoku niestandardowego składnika lokalnego. Składnik lokalny jest renderowany

wstawkowo; oznacza to, że można do niego bezpośrednio stosować style zdefiniowane w motywie lub w pliku `design.css`.

### Niestandardowy składnik odległy

Składnik "niestandardowy składnik odległy" ma jedną klasę otaczającą jego ramkę `iFrame`:

```
scs-app-iframe-wrapper
```

Oprócz stosowanie stylów CSS zdefiniowanych w niestandardowym składniku odległym, można użyć pakietu Sites SDK do pobrania pliku `design.css` z serwisu hosta.

```
// fetch current theme design from host site and then add it to the page
SitesSDK.getSiteProperty('theme',function(data){
  // check if we got a url back
  if ( data.url && typeof data.url === 'string' ) {
    if ( data.url !== '' ) {
      // theme is loaded, so dynamically inject theme
      SitesSDK.Utils.addSiteThemeDesign(data.url);
    }
  }
});
```

Tym samym składnik będzie mógł odziedziczyć style ze stylu używanego na hoście.

### Separator

Mimo że nie ma klas specyficznych dla składnika "separator", to jednak można stosować style w samym znaczniku `<hr>`.

Na przykład można utworzyć separator kropkowy:

```
.design-style .scs-divider hr {
border-top: 1px dotted #333333;
}
```

### Wideo, YouTube, rozdzielacz


Nie ma klas specyficznych dla składników "wideo", "YouTube" ani "rozdzielacz"

## Ustawianie właściwości składnika

Można skonfigurować właściwości składnika, który będzie używany w serwisie.

Składniki są poszczególnymi częściami serwisu internetowego, takimi jak teksty, tytuły, obrazy, przyciski, separatory, mapy, galerie czy filmy wideo. Tworząc projekt motywu, trzeba także określić domyślne ustawienia dla każdego typu składnika. Każdy składnik ma swoje ustawienia (takie jak rozmiar, wyrównanie, odstępy, kolory i krawędzie) definiujące jego wygląd i sposób działania. Ustawienia te zmieniają się w zależności od składnika. Można także zdecydować, czy właściwości składnika mogą być zmieniane przez użytkownika, gdy składnik stanie się dostępny w serwisie w nowym motywie.

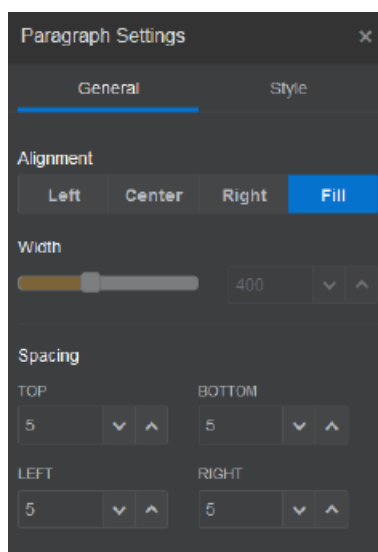
Na przykład poniższe etapy ilustrują sposób konfigurowania ustawień dla składnika "akapit":


1. Mając otwarty serwis w trybie edycji, otworzyć stronę, na której znajduje się składnik "akapit", lub dodać składnik "akapit".
2. Kliknąć na składniku "akapit", kliknąć na ikonie  znajdującej się w rogu składnika, po czym wybrać opcję **Ustawienia**.

Zostanie wyświetlony panel **Ustawienia**, w którym można wybrać ustawienia składnika.

 **Uwaga:**

Opcje ustawień są specyficzne dla danego typu składnika. Jeśli użytkownik pracuje z niestandardowymi składnikami lokalnymi lub odległymi, zostanie wyświetlone łącze **Ustawienia niestandardowe**.



3. Na przykład można zmienić ustawienia pod kątem wyrównania, szerokości, odstępów, czcionek, stylu, koloru i rozmiaru. Wygląd składnika zmienia się zgodnie z nowymi ustawieniami.
4. Po zakończeniu kliknąć na ikonie . Ustawienia zostaną zastosowane do strony.

## Składniki renderowane w ramach wstawkowych

Składniki, które są renderowane w ramach wstawkowych, mogą być określane na stronach Oracle Content Management poprzez zarejestrowanie, a następnie dodanie składników z serwerów zewnętrznych (składników określanych jako **składniki odległe**), a także poprzez wybranie opcji ramki wstawkowej podczas tworzenia składnika lokalnego. Ten typ składnika może rozszerzyć funkcje serwisów, na przykład poprzez możliwość dodania składnika społecznościowego lub składnika "koszyk z zamówieniami".

Dla składnika odległego trzeba określić URL punktu końcowego. W przypadku składnika lokalnego renderowanego w ramce wstawkowej adresy URL są wyprowadzane z nazwy składnika, przechowywanego w katalogu składników.

- URL punktu końcowego: Zawartość widżetu jest pobierana spode tego adresu URL i jest osadzana w obrębie ramki wstawkowej.
- URL ustawień: Ten URL jest renderowany w ramce wstawkowej, tak aby można było skonfigurować składnik zaraz po umieszczeniu go na stronie.

Składnik może renderować statyczne lub dynamiczne wizualizacje danych oraz wyświetlać formularz bądź inny interaktywny interfejs użytkownika rozszerzający funkcje serwisu. Zob. [Renderowanie ustawień składnika](#).

### Składniki udostępniane przez Oracle Content Management

Zbiór składników, które są renderowane w ramach wstawkowych, jest udostępniany przez usługę Oracle Content Management.

Nazwa	Typ	ID
Lista folderów	scs-app	Lista folderów
Lista plików	scs-app	Lista plików
Menedżer dokumentów	scs-app	Menedżer dokumentów
Facebook - Lubię to	scs-app	Facebook - Lubię to
Twitter - Obserwuj	scs-app	Twitter - Obserwuj
Twitter - Udostępnij	scs-app	Twitter - Udostępnij
Facebook - Rekomendacja	scs-app	Facebook - Rekomendacja

### Rejestracja składnika

Zanim w serwisie będzie można użyć odległego składnika renderowanego w ramce wstawkowej, trzeba zarejestrować — za pomocą protokołu HTTPS — URL punktu końcowego. Informacja ta jest przechowywana w katalogu składników. Punkt końcowy musi zezwalać adresowi URL na wyświetlanie w ramce wstawkowej; nie należy ustawiać nagłówka `X-Frame-Options="sameorigin"`. W przypadku składnika lokalnego kryterium punktu końcowego jest automatycznie spełniony, ponieważ pliki są przechowywane na serwerze Oracle Content Management.

Po zarejestrowaniu składnika renderowanego w ramce wstawkowej zostanie wygenerowany nowy GUID reprezentujący składnik. Jeśli ten składnik został zarejestrowany w instancji Oracle Content Management więcej niż raz, uzyska więcej niż jeden GUID, ponieważ identyfikatory te reprezentują każdą rejestrację składnika. Gdy składnik zostanie zarejestrowany, w Oracle Content Management przechowywany będzie tylko opis powiązany z identyfikatorem GUID; składnik nadal będzie uruchamiany ze swojego odległego punktu końcowego. Zob. [Rejestrowanie składnika odległego](#).

Po zarejestrowaniu składnika renderowanego w ramce wstawkowej zostanie również wygenerowany ID instancji. Ten ID instancji reprezentuje składnik zarejestrowany przy użyciu określonej dzierżawy Oracle Content Management. Ten sam składnik, zarejestrowany więcej niż raz w obrębie tej samej dzierżawy Oracle Content Management lub przy użyciu innej dzierżawy Oracle Content Management, będzie miał różne identyfikatory (ID) instancji.



### Utrwalanie ustawień składnika odległego

Po wyrenderowaniu (w panelu ustawień) adresu URL składnika renderowanego w ramce wstawkowej, są udostępniane ID instancji i ID składnika. Dzięki temu składnik może wybrać utrwalanie dowolnych ustawień na swoim własnym serwerze, indeksując je według ID instancji i ID składnika. Alternatywnie można użyć zestawu Sites SDK z Oracle Content Management i zezwolić na przechowywanie — z uwzględnieniem ID składnika — danych JSON (maksymalnie 1,5 KB) w modelu strony serwisu.

Używanie zestawu Sites SDK do utrwalania ustawień niesie dla składników renderowanych w ramach wstawkowych dwie korzyści:

- Składnik może z łatwością uczestniczyć w wersjach stron, aktualizacjach stron i modelu publikowania serwisu.
- Składnik może obejmować punkty końcowe HTML uruchamiane w przeglądarce (w przeciwieństwie do uruchamiania w systemie zaplecza).

Zob. Oracle Content Management SDK.

## ID instancji i struktura składników renderowanych w ramach wstawkowych — informacje podstawowe

ID instancji składnika jest unikatowym identyfikatorem składnika renderowanego w ramce wstawkowej w obrębie serwisu.

Gdy użytkownik przeciąga składnik (renderowany w ramce wstawkowej) z Katalogu składników i "upuszcza" ten składnik na stronie serwisu, do usługi Oracle Sites Cloud Service jest kierowane wywołanie zlecające wygenerowanie nowego unikatowego ID instancji składnika. Unikatowość tego ID jest gwarantowana, a wszystkie instancje tego składnika, udostępniane w tej samej usłudze, będą miały ten sam ID instancji.

ID instancji składnika zawiera dodatkowe informacje, których można użyć do zabezpieczenia ustawień oraz do zabezpieczenia użycia składnika renderowanego w ramce wstawkowej, dzięki czemu składnik ma pewność, że aktualizacja ustawień przychodzi z zaufanego miejsca.

Parametr "ID instancji" umożliwia programistom identyfikację serwisu i podmiotu wywołującego. Podmiot wywołujący jest identyfikowany przez sprawdzenie podpisu cyfrowego, który jest generowany przy użyciu tajnego klucza składnika. Tajny klucz jest generowany w trakcie procesu rejestracji składnika.

Instancja składnika składa się z dwóch części rozdzielonych kropką (.); są to: dane i struktura.

#### Instancja składnika: dane

Część "dane" instancji składnika renderowanego w ramce wstawkowej ma postać napisu w notacji JSON Base64. Struktura napisu JSON przedstawia się następująco:

```
{
  "instanceid": "BBDC7614F693B75110D811E6C0B77C935FAEC5112E5E",
  "permissions": "",
  "entitlements": "",
  "signdate": "1435426735293",
```



Oracle Content Management, aby urzeczywistnić zawartość na stronie serwisu, wywołuje zarejestrowane punkty końcowe składnika. Ponieważ te punkty końcowe są eksponowane w publicznym Internecie, programiści powinni sprawdzać, czy zarejestrowane punkty końcowe składnika odległego są wywoływane z Oracle Content Management. W celu sprawdzenia autentyczności wywołującego jest do zarejestrowanych punktów końcowych URL dostarczany podpisany token. Podmiot wywołujący jest identyfikowany przez sprawdzenie podpisu cyfrowego, osadzonego w podpisanym tokenie wraz z tajnym kluczem składnika odległego, który to klucz został dostarczony w trakcie procesu rejestracji.

Format tokenu ma następującą postać:

```
{base64 encoded serialized JSON data}.{base64 encoded signature}
```

Poniżej jest pokazany przykładowy token przekazywany do zarejestrowanych punktów końcowych aplikacji:

```
eyJpbmN0YW5jZWlkIjoiQTRGOTE3REY5OTZEN0Q3ODBCMjUzODZFOTFEMDA3ODJGMjVBRjY2Rjc3O  
TIIiLCJzaWduZGF0ZSI6IjE0NDU2MzcwNTk5MTciLCJzaXRlZG9tYWluIjoic2Vydm1jZTEtdGVuYW  
50MS51cy5vcnFjbGUuY29tIiwicGVybnWlzc2lvdnMiOiJTSVRFX09XTkVSIiwizW50aXRszW11bnR  
zIjoiIn0=.5p3of7t110wuy5F3zpm+YgICSHH8C/BHczdbVZx2VH8=
```

Token składa się z dwóch części rozdzielonych kropką (.); są to: dane i podpis.

Programiści z zasady powinni zawsze — przed udzieleniem dostępu do zarejestrowanych punktów końcowych składnika odległego — identyfikować token w trybie edycji lub podglądu. Ponadto, identyfikując podmiot wywołujący w punkcie końcowym ustawień, programiści powinni zawsze zadbać o sprawdzanie wartości `SITE_OWNER` w polu **permissions** tokenu. Pole **permissions** tokenu pokazuje wartość `SITE_OWNER` tylko w trybie edycji. Wygenerowany w trakcie sesji edycji token nigdy nie jest utrwalany w modelu strony i jest wyłączany za pomocą tokenu wykonawczego mającego w polu **permissions** wartość `NULL`.

### Dane

Część "dane" instancji ma postać napisu w notacji JSON Base64. Struktura napisu JSON przedstawia się następująco:

```
{
  "instanceid": "BBDC7614F693B75110D811E6C0B77C935FAEC5112E5E",
  "permissions": "",
  "entitlements": "",
  "signdate": "1435426735293",
  "sitedomain": "service1-tenant4.localhost"
}
```

Nazwa pola	Opis
instanceid	Unikatowy identyfikator składnika dla dzierżawy Oracle Content Management.
signdate	Data wygenerowania podpisu.
sitedomain	Nazwa domeny instancji Oracle Content Management.
permissions	Zestaw uprawnień współużytkownika serwisu. W trybie edycji będzie mieć wartość <code>SITE_OWNER</code> ; w przeciwnym razie nie będzie mieć przypisanej żadnej wartości.

Nazwa pola	Opis
entitlements	Lista funkcji "premium" nabytych przez właściciela serwisu.

### Podpis

Część "dane" instancji składnika odległego jest serializowana, zanim instancja zostanie podpisana za pomocą tajnego klucza `APP_SECRET_KEY`. Tajny klucz musi zostać wygenerowany i pokazany programiście podczas rejestracji składnika. Podpis jest obliczany przez wygenerowanie skrótu (hash) części "dane" składnika instancji (serializowana struktura JSON) przy użyciu tajnego klucza, jak pokazano poniżej:

```
$signature = HMAC (serialized JSON structure, APP_SECRET_KEY)
```

Algorytmem haszowania, używanym do wygenerowania podpisu, jest SHA256. Token stanowi wówczas połączenie serializowanej struktury JSON i wygenerowanego składnika podpisu, jak pokazano poniżej:

```
$instance = {base64encoded serialized JSON structure}.  
{base64encoded $signature}
```

## Rejestrowanie składnika odległego

Aby można było użyć w serwisie składnika odległego, trzeba najpierw zarejestrować go w Oracle Content Management.

Można rejestrować składniki odległe pochodzące od innych dostawców, a także samodzielnie utworzone.

Aby zarejestrować składnik odległy, przeznaczony do użycia w instancji Oracle Content Management, należy:

1. Wybrać kolejno opcje **Programista** i **Wyświetl wszystkie składniki**.
2. Nacisnąć przycisk **Utwórz**, po czym wybrać opcję **Zarejestruj składnik odległy**.
3. W oknie rejestracji składnika odległego wpisać lub wybrać następujące informacje:
  - **Nazwa:** Nazwa składnika widziana przez użytkowników.
  - **Opis:** Opis składnika widziany przez użytkowników.
  - **URL składnika:** Punkt końcowy, używany w ramce iFrame do renderowania zawartości składnika na stronie. Musi być oparty na protokole HTTPS.
  - **URL ustawień:** Punkt końcowy, używany w ramce iFrame do renderowania ustawień składnika odległego, dodawanego do strony. Musi być oparty na protokole HTTPS.
  - **Szerokość ustawień:** Określa szerokość domyślną (w pikselach) panelu ustawień składnika.
  - **Wysokość ustawień:** Określa wysokość domyślną (w pikselach) panelu ustawień składnika.
  - **Klucz:** 192-bitowy klucz AES, powiązany ze składnikiem odległym, używany — gdy składnik jest udostępniany — do utworzenia podpisanego tokenu

haszowanego skrótu. Służy do szyfrowania i zapewnienia bezpiecznego odczytu i zapisu ustawień składnika.

#### 4. Nacisnąć przycisk **Zarejestruj**.

Gdy składnik odległy zostanie utworzony, jego nazwa pojawi się na liście składników. Pliki, użyte do zarejestrowania składnika, można eksplorować, klikając na nazwie składnika na liście składników.

Informacje dotyczące rejestracji składnika są przechowywane w katalogu używanym przez serwisy utworzone w tej samej instancji Oracle Content Management, lecz składnik pozostaje usługą odległą.

Ikona składnika jest dodawana w konstruktorze serwisów do panelu "Składniki niestandardowe", z nazwą przypisaną do składnika. Składnik można udostępnić innym użytkownikom; będą go w konstruktorze serwisów widzieć w panelu "Składniki niestandardowe".

## Usuwanie składnika

Mając odpowiednie uprawnienia, można składnik usunąć (korzystając z menedżera składników), tak że nie będzie już dostępny do użycia. Gdy składnik zostanie usunięty, jego folder oraz wszystkie powiązane foldery i pliki są przenoszone do Kosza.


Składnik może zostać usunięty, przy użyciu menedżera składników, przez jego twórcę (właściciela składnika) lub osobę, której ten składnik został udostępniony i której została przypisana rola menedżera.



### Uwaga:

Nie można usunąć składnika, który jest używany przez dowolny serwis lub dowolną aktualizację, w tym serwisy i aktualizacje znajdujące się w Koszu.

Aby usunąć składnik, należy:

1. Na stronie startowej wybrać opcję **Programista**.
2. Nacisnąć przycisk **Wyświetl wszystkie składniki**. Zostają wyświetlone wszystkie obecnie zarejestrowane składniki.
3. Wybrać nazwę składnika, po czym wybrać z menu podręcznego opcję **Usuń** albo kliknąć na ikonie  (na pasku czynności). Jest wyświetlane wezwanie do potwierdzenia tej czynności.
4. Aby potwierdzić decyzję o usunięciu, nacisnąć przycisk **Tak**. Aby anulować decyzję o usunięciu, nacisnąć przycisk **Nie**. Jeśli decyzja o usunięciu zostanie potwierdzona, składnik (oraz wszystkie powiązane z nim pliki i foldery) zostanie przeniesiony do Kosza.

Usunięty folder składnika pozostaje w Koszu, aż:

- Folder zostanie przywrócony.
- Folder zostanie trwale usunięty.
- Zostanie osiągnięty przydział miejsca na Kosz.

- Kosz zostanie automatycznie opróżniony zgodnie z interwałem ustawionym przez administratora serwisu. Wartością domyślną jest 90 dni.

## Sites SDK

Składniki opracowane dla Oracle Content Management są w serwisie renderowane jako składniki, które można przeciągać w dowolne miejsce na stronie serwisu.

Sites SDK obsługuje całą komunikację między stroną a składnikiem.

- `Sites.Settings.getProperty(propertyName, callbackFunction)`: Udostępnia wywołanie zwrotne pozwalające pobrać żadaną właściwość dla instancji niestandardowego składnika.
- `Sites.Settings.setProperty(propertyName, propertyValue)`: Przechowuje żadaną wartość dla instancji niestandardowego składnika.

Dla wszystkich właściwości składnika niestandardowego należy użyć metody `Sites.Settings.getProperty` lub `Sites.Settings.setProperty`, a następnie użyć metod `SitesSDK.publish` i `SitesSDK.subscribe` w celu nasłuchiwania zdarzeń komunikatów.

Zob. [Sites SDK — opis](#).

## Dostosowywanie pliku kontrolera

W serwisach Oracle Content Management plik kontrolera jest używany do wyświetlania poszczególnych stron serwisu. Gdy przeglądarka wysyła żądanie dotyczące strony internetowej, serwer odpowiada za pomocą kopii pliku kontrolera. Dla każdej żądanej strony jest dostarczana z serwera ta sama kopia pliku kontrolera.

- [Plik kontrolera — informacje podstawowe](#)
- [Modyfikowanie domyślnego pliku controller.html](#)
- [Obiekt SCS — informacje podstawowe](#)
- [Sekcje pliku kontrolera, których nie należy dostosowywać](#)
- [Używanie tokenów do zapewnienia przenośności niestandardowego pliku kontrolera](#)
- [Przykłady niestandardowych plików kontrolera](#)

### Plik kontrolera — informacje podstawowe

Plik kontrolera to niewielka strona HTML, która dynamicznie inicjuje sekwencję renderowania reszty strony. Jest to pierwsze miejsce, w którym można stosować dostosowania, wpływające na działanie każdej strony serwisu.

Głównym zadaniem pliku kontrolera jest udostępnianie i hostowanie środowiska wykonawczego JavaScript kontrolera. Kod JavaScript kontrolera następnie ładuje i wyświetla stronę. Dostosowania w pliku kontrolera zapewniają możliwość przesłaniania operacji JavaScript kontrolera i oddziaływania na nie.

### Plik domyślny kontrolera

Po utworzeniu pierwszego serwisu Oracle Content Management zostaje powiązany z nim plik domyślny kontrolera (controller.html).

```
<!DOCTYPE html>
<html>
<head>
<!-- The following meta tag is used for Internet Explorer browsers. It
indicates that the browser should use the latest rendering mode to display
the web page. -->
<meta http-equiv="X-UA-Compatible" content="IE=edge">

<!-- The following meta tag is used for browsers on mobile devices to set
the initial viewport scale to the full page. -->
<meta name="viewport" content="initial-scale=1">

<!-- The following script initially defines the SCS object. The SCS object
must be present, and this variable name is reserved for use by Oracle
Content Management.. -->
<script type="text/javascript">
```

```
var SCS = { sitePrefix: '/SampleSite/' };
</script>

<!-- The following script loads the full controller JavaScript, which
is used to display the web page. -->
<script src="/SampleSite/_sitesclouddelivery/renderer/controller.js"></
script>
</head>

<!-- The body tag of the controller must have the id scsControllerBody.
This identifier is used by the Controller JavaScript. If JavaScript is
not enabled on the browser, the noscript tag content is displayed.-->
<body id="scsControllerBody"><noscript>This site requires JavaScript to
be enabled.</noscript>

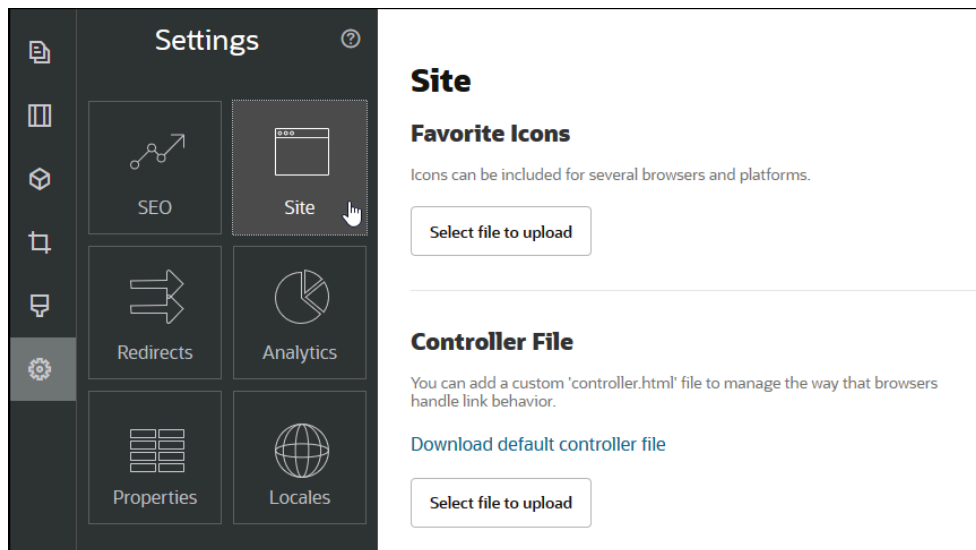
<!-- The following image tag displays an animated circle by default if
it takes too long for a page to display. The wait image must have the
id scsWaitImage. -->

</body></html>
```

## Modyfikowanie domyślnego pliku controller.html

Można pobrać plik controller.html serwisu, a następnie ten plik zmodyfikować, wpływając w ten sposób na funkcjonowanie poszczególnych stron serwisu. Aby pobrać plik controller.html serwisu, należy:

1. Otworzyć w konstruktorze serwisów serwis, który będzie modyfikowany, po czym ustawić go w trybie **edycji**.
2. Można wybrać istniejącą aktualizację lub utworzyć nową.
3. Z menu nawigacji bocznej wybrać opcję **Ustawienia**, a następnie **Serwis**.



4. W sekcji "Plik kontrolera" kliknąć na łączu **Pobierz domyślny plik kontrolera**.
5. Zapisać plik controller.html na dysku lokalnym, po czym ten plik edytować.



6. Po ukończeniu edytowania wrócić na stronę "Ustawienia" serwisu, po czym w sekcji "Plik kontrolera" nacisnąć przycisk **Wybór pliku do wysłania**.
7. Przejść do zmodyfikowanego pliku kontrolera, po czym go wysłać.

 **Uwaga:**

Dostosowania zawarte w pliku kontrolera są używane tylko w serwisach działających w trybie online. Nie są używane, gdy serwis jest edytowany lub gdy jest wyświetlany jego podgląd. Aby zmiany były uwzględniane (gdy plik `controller.html` zostanie wysłany do serwisu), aktualizacja serwisu musi zostać zatwierdzona, a sam serwis musi zostać opublikowany.

## Obiekt SCS — informacje podstawowe

Kontroler JavaScript używa — w celu wyrenderowania strony internetowej — zmiennych i wartości zdefiniowanych w globalnym obiekcie SCS. Ten globalny obiekt JavaScript musi zostać zdefiniowany w pliku kontrolera przed dołączeniem pliku `controller.js`. W obiekcie SCS istnieją dwie główne właściwości, które mogą być używane dla dostosowywania:

- [SCS.sitePrefix](#)
- [SCS.preInitRendering](#)
- [SCS.getDeviceInfo](#)

### SCS.sitePrefix

Zmienna `SCS.sitePrefix` definiuje prefiks ścieżki serwisu online. Zazwyczaj jest to domyślnie `site/<nazwaSerwisu>/`.

Kontroler JavaScript używa tej wartości do ustalenia strony internetowej, która ma zostać wyświetlona. Na przykład, jeśli przeglądarka żąda strony `/site/SampleSite/products/index.html`, to `sitePrefix` o wartości `/site/SampleSite/` umożliwi kontrolerowi JavaScript ustalenie, że powinna zostać wyświetlona strona `products/index.html`. Należy pamiętać, że `sitePrefix` musi być wartością napisową, zaczynającą się i kończącą znakiem `"/`.

Jeśli są używane serwer proxy lub inna infrastruktura (na przykład Akamai) i reguły odwzorowywania adresu URL, to — modyfikując tę zmienną — można dostosowywać prefiks serwisu. Na przykład ustawiając `sitePrefix` na `/intranet/ExampleSite/` umożliwia się dostarczanie serwisu z tą ścieżką w przeglądarce (zamiast domyślnego prefiksu `/site/SampleSite/`).

 **Uwaga:**

Wewnętrznie, jeśli URL w przeglądarce nie jest zgodny z prefiksem ścieżki, kontroler domyślny JavaScript używa jako zmiennej `sitePrefix` znaku `"/`. Dzięki temu serwis może być udostępniany — bez dostosowywania — przy użyciu zindywidualizowanej domeny.

## SCS.preInitRendering

Funkcja `SCS.preInitRendering` (jeśli zdefiniowano) jest wywoływana przez kod JavaScript kontrolera przed wykonaniem jakiegokolwiek logiki obliczeniowej. Jest to funkcja przydatna do definiowania (w niestandardowych plikach kontrolera) zdolności przesłania podstawowych operacji JavaScript kontrolera.

`SCS.preInitRendering` musi być funkcją (jeśli zdefiniowano). Jest wywoływana bez żadnych argumentów. Nie jest oczekiwana ani przetwarzana żadna wartość zwrrotna.

## SCS.getDeviceInfo

Funkcja `SCS.getDeviceInfo` umożliwia dostosowywanie — wewnątrz pliku kontrolera JavaScript — logiki wykrywania urządzenia. Funkcja wykrywania jest używana do określania, czy powinna być dostarczana mobilna czy responsywna wersja strony.

Funkcja `SCS.getDeviceInfo` nie przyjmuje żadnych argumentów; zwraca obiekt JavaScript, mający dwie właściwości:

- **isMobile** — właściwość logiczna wskazująca, że bieżące urządzenie jest klientem mobilnym, takim jak smartfon.
- **isIOS** — właściwość logiczna wskazująca, że bieżące urządzenie pracuje w systemie operacyjnym opartym na iOS.

Jeśli kontroler niestandardowy nie przesłania funkcji `SCS.getDeviceInfo`, jest używana wbudowana implementacja domyślna.

## Sekcje pliku kontrolera, których nie należy dostosowywać

Niektóre sekcje pliku kontrolera są przetwarzane przez serwer Oracle Content Management, tak aby można było używać sieci CDN oraz definiować segmenty buforowania adresów URL. Sekcje te są pokazane w poniższym przykładzie.

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="initial-scale=1">
<script type="text/javascript">
<!-- The global variable SCSCacheKeys will be inserted at this location
-->

var SCS = { sitePrefix: '/SampleSite/' };
<!-- A variable denoting the location of the CDN will be injected at
this location. -->
<!-- Additionally, the prefix "/site" will be inserted before the site
name segment. This is to support legacy controller files. -->

</script>
<script src="/SampleSite/_sitesclouddelivery/renderer/controller.js"></script>
```

```

<!-- The src value will be updated to use the CDN if this syntax is used. -->

</head>
<body id="scsControllerBody"><noscript>This site requires JavaScript to be
enabled.</noscript>
<!-- The id of the <body> should not be changed. -->


<!-- The id of the wait image tag should not be changed -->

</body></html>

```

## Używanie tokenów do zapewnienia przenośności niestandardowego pliku kontrolera

Gdy serwis zostanie utworzony, jego nazwa jest używana w pliku kontrolera. Wiąże to plik kontrolera z serwisem i uniemożliwia używanie tego pliku dla innych serwisów.

Jeśli plik kontrolera zostanie użyty dla serwisu o innej nazwie, to zawarte w tym pliku ustawienia nie będą działać. Dotyczy to także serwisów utworzonych z szablonu zawierającego niestandardowy plik kontrolera. Poniższe dynamicznie obliczane tokeny umożliwiają używanie pliku kontrolera z więcej niż jednym serwisem.

Token	Opis
[!-\$SCS_SITE_PREFIX-]	Token ten reprezentuje prefiks bieżącego serwisu. Przykładowe wartości:  /site/MySite/  lub  /site/authsite/MySecureSite/
[!-\$SCS_SITE_PATH-]	Token ten reprezentuje lokalizację CDN bieżącego produktu. Wartość nie będzie zawierać końcowego ukośnika "/". Przykładowe wartości:  /site/MySite/_cache_0000  lub  /site/authsite/MySecureSite

Token	Opis
[!-\$SCS_PRODUCT_PATH--]	Token ten reprezentuje URL bieżącego produktu, w tym klucz pamięci podręcznej (jeśli ma zastosowanie). Wartość nie będzie zawierać końcowego ukośnika "/". Przykładowe wartości:  <code>https://www.example.com/cdn/cec/v21.1.2.23</code>  lub  <code>https://www.example.com/cdn/cec/v21.1.3.18</code>

## Przykłady niestandardowych plików kontrolera

Omówiono tu przykłady dla następujących niestandardowych plików kontrolera:

- [Zmianie prefiksu serwisu](#)
- [Dostosowywanie grafiki oczekiwania](#)
- [Dostosowywanie ikon "favicon"](#)
- [Dostosowywanie znaczników <noscript> i <meta> dla przeszukiwarek innych niż JavaScript](#)
- [Pobieranie wstępne plików JavaScript](#)
- [Weryfikowanie praw własności serwisu za pomocą dodatkowego kodu znacznikowego](#)
- [Wspomaganie wykrywania urządzeń](#)
- [Używanie tokenów do zwiększania przenośności pliku controller.htm](#)

### Zmianie prefiksu serwisu

Poniższy przykładowy plik kontrolera definiuje funkcję `preInitRendering` w celu umożliwienia dostarczania serwisu dla różnych prefiksów.

#### Uwaga:

Aby można było skorzystać z tego przykładu, sieć CDN (lub inny serwer proxy) musi być skonfigurowana do reagowania na zdefiniowane prefiksy. Ponadto, jeśli żaden z dodatkowych prefiksów (zdefiniowanych w funkcji) nie jest zgodny z adresem URL przeglądarki, jest stosowane działanie domyślne.

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="initial-scale=1">
```

```
<script type="text/javascript">
var SCS = { sitePrefix: '/SampleSite/' };

SCS.preInitRendering = function() {
    // List additional site prefixes here. All sitePrefix values MUST start
    and end with a '/' character.
    var additionalSitePrefixes = [
        '/corporate/intranet/SampleSite/',
        '/marketing/preflight/',
        '/qa/'
    ];

    // Determine if the actual browser URL matches one of the additional
    site prefixes
    var pageUrl = decodeURI(window.location.pathname);
    var i, prefix;
    for (i = 0; i < additionalSitePrefixes.length; i++) {
        prefix = additionalSitePrefixes[i];
        if (pageUrl.startsWith(prefix) || (pageUrl === prefix.slice(0,
-1))) {
            // If we find a match, set the global site prefix variable
            SCS.sitePrefix = prefix;
            break;
        }
    }
};
</script>
<script src="/SampleSite/_sitesclouddelivery/renderer/controller.js"></
script>
</head>
<body id="scsControllerBody"><noscript>This site requires JavaScript to be
enabled.</noscript>

</body></html>
```

## Dostosowywanie grafiki oczekiwania

Poniższy przykładowy plik kontrolera dostosowuje grafikę oczekiwania, która jest wyświetlana, gdy rendering trwa dłużej wskutek opóźnień sieci.

### Uwaga:

W tym przykładzie — zamiast stosować osobne żądanie w trakcie renderingu — użyto adresu URL danych dla obrazu.

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="initial-scale=1">
<script type="text/javascript">
```

```
var SCS = { sitePrefix: '/SampleSite/' };
</script>
<script src="/SampleSite/_sitesclouddelivery/renderer/controller.js"></script>
</head>
<body id="scsControllerBody"><noscript>This site requires JavaScript to be enabled.</noscript>

</body></html>

```

## Dostosowywanie ikon "favicon"

Poniższy przykładowy plik kontrolera definiuje niestandardową ikonę "favicon" dla serwisu. Jest to przydatne, gdy przeglądarki nie ładują dynamicznie ikony "favicon" z modelu obiektu dokumentu renderowanej strony.

```

<!DOCTYPE html>
<html>
<head>
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="initial-scale=1">
<link rel="shortcut icon" href="/SampleSite/favicon.ico" />
<script type="text/javascript">
var SCS = { sitePrefix: '/SampleSite/' };
</script>
<script src="/SampleSite/_sitesclouddelivery/renderer/controller.js"></script>
</head>
<body id="scsControllerBody"><noscript>This site requires JavaScript to be
enabled.</noscript>

</body></html>

```

## Dostosowywanie znaczników <noscript> i <meta> dla przeszukiwarek innych niż JavaScript

Poniższy przykładowy plik kontrolera dostosowuje komunikat <noscript> i znaczniki <meta> dla przeszukiwarek, które nie przetwarzają kodu JavaScript.

### Uwaga:

Ponieważ plik kontrolera jest używany dla każdej strony serwisu internetowego, na każdej z nich pojawi się ten sam znacznik "meta".

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="initial-scale=1">
<meta name="description" content="A site with interesting content">
<script type="text/javascript">
var SCS = { sitePrefix: '/SampleSite/' };
</script>
<script src="/SampleSite/_sitesclouddelivery/renderer/controller.js"></script>
</head>
<body id="scsControllerBody"><noscript>Please enable JavaScript to view this site properly.</noscript>

</body></html>
```

## Pobieranie wstępne plików JavaScript

Niektóre przeglądarki umożliwiają zadeklarowanie zasobów potrzebnych w bieżącej nawigacji lub kolejnych. Kod znacznikowy w pliku kontrolera może określać zasoby, które należy wstępnie pobrać lub wstępnie załadować.

### Uwaga:

Ponieważ każda strona internetowa OCE używa plików require.js i renderer.js, są one dobrymi kandydatami do zastosowania technik wstępnego ładowania i wstępnego pobierania. Jeśli funkcja href jest użyta jako pierwsza w znaczniku <link>, to jej wartość — aby dostawa nastąpiła z sieci CDN — jest ustalana przez serwer OCE.

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="X-UA-Compatible" content="IE=edge">
```



```
<meta name="viewport" content="initial-scale=1">
<link href="/SampleSite/_sitesclouddelivery/renderer/require.js"
rel="preload" as="script">
<link href="/SampleSite/_sitesclouddelivery/renderer/renderer.js"
rel="preload" as="script">
<script type="text/javascript">
var SCS = { sitePrefix: '/SampleSite/' };
</script>
<script src="/SampleSite/_sitesclouddelivery/renderer/controller.js"></
script>
</head>
<body id="scsControllerBody"><noscript>Please enable JavaScript to view this
site properly.</noscript>

</body></html>
```

## Weryfikowanie praw własności serwisu za pomocą dodatkowego kodu znacznikowego

Procesy przeszukujące lub wyszukiwarki innego podmiotu mogą wymagać dodatkowego kodu znacznikowego do weryfikacji praw własności serwisu. Jeśli trzeba, kod ten może zostać wstawiony do pliku kontrolera.

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="initial-scale=1">
<meta name="search-site-verification" content="your verification string">
<script type="text/javascript">
var SCS = { sitePrefix: '/SampleSite/' };
</script>
<script src="/SampleSite/_sitesclouddelivery/renderer/controller.js"></
script>
</head>
<body id="scsControllerBody"><noscript>Please enable JavaScript to view this
site properly.</noscript>

</body></html>
```

## Wspomaganie wykrywania urządzeń

Rozpoznawanie nowych urządzeń mobilnych lub dostosowywanie identyfikatorów użytkownika-agenta urządzenia może być dokonywane poprzez dostosowanie w kontrolerze kodu wykrywania urządzenia.

 Uwaga:

- Funkcja `getDeviceInfo` jest wywoływana przez kontroler w celu określenia, czy jest używane urządzenie mobilne. Ten przykład przesłania domyślne wywołanie funkcji `"getDeviceInfo"`.
- Właściwość `"isMobile"` określa, który układ ma zostać użyty podczas renderowania strony.
- Właściwość `"isIOS"` decyduje, który dostawca ma zostać użyty dla składnika `"mapa"`.
- Wbudowany test domyślny właściwości `isMobile`:

```
/Mobi|iPhone|iPod|BlackBerry|IEMobile|Opera Mini/  
i.test(userAgent) && !/iPad/i.test(userAgent)
```

- Wbudowany test domyślny właściwości `isIOS`:

```
/iPad|iPhone|iPod/i.test(userAgent) && !window.MSStream
```

```
<!DOCTYPE html>  
<html>  
<head>  
<meta http-equiv="X-UA-Compatible" content="IE=edge">  
<meta name="viewport" content="initial-scale=1">  
<meta name="search-site-verification" content="your verification  
string">  
<script type="text/javascript">  
var SCS = { sitePrefix: '/SampleSite/' };  
  
SCS.getDeviceInfo = function() {  
    // Return an object with two Boolean properties, isMobile and isIOS.  
    var userAgent = navigator.userAgent;  
    return {  
        isMobile: /Mobi|iPhone/i.test(userAgent) && !/iPad/  
i.test(userAgent),  
        isIOS: /iPad|iPhone|iPod/i.test(userAgent)  
    };  
};  
</script>  
<script src="/SampleSite/_sitesclouddelivery/renderer/controller.js"></  
script>  
</head>  
<body id="scsControllerBody"><noscript>Please enable JavaScript to view  
this site properly.</noscript>  
  
</body></html>
```

## Używanie tokenów do zwiększania przenośności pliku controller.htm

Domyślnie nazwa serwisu jest trwale zakodowana w plikach controller.html, przez co zmiana nazwy serwisu lub ponowne użycie identycznych niestandardowych kontrolerów w różnych serwisach może okazać się trudne. W rozwiązaniu tego problemu mogą pomóc tokeny. W poniższym przykładzie pokazano, w jaki sposób można użyć tokenów ułatwiających przenośność.



### Uwaga:

Znaczniki `<link>` zostały w tym przykładzie użyte tylko w celach pokazowych. Właściwy plik controller.html nie będzie ich zawierał.

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="initial-scale=1">
<script type="text/javascript">
var SCS = { sitePrefix: '[!--$SCS_SITE_PREFIX--]' };
</script>
<script src="[!--$SCS_PRODUCT_PATH--]/_sitesclouddelivery/renderer/
controller.js"></script>
<link rel="shortcut icon" href="[!--$SCS_SITE_PATH--]/favicon.ico">
<link rel="preload" href="[!--$SCS_PRODUCT_PATH--]/_sitesclouddelivery/
renderer/require.js" as="script">
<link rel="preload" href="[!--$SCS_PRODUCT_PATH--]/_sitesclouddelivery/
renderer/renderer.js" as="script">
</head>
<body id="scsControllerBody"><noscript>Please enable JavaScript to view this
site properly.</noscript>

</body></html>
```

# Część V

## Tworzenie aplikacji dla serwisów za pomocą innych narzędzi

Oracle Content Management udostępnia kilka sposobów interakcji z aplikacjami i zasobami Experience tworzonymi w usłudze Oracle Content Management lub poza usługą Oracle Content Management przy użyciu narzędzi innych podmiotów.

### Korzystanie z zestawu OCE Toolkit

Oracle Content Management Toolkit i pakiety SDK pomagają tworzyć niestandardowe aplikacje korzystające z zawartości zarządzanej w repozytorium Oracle Content Management. Aplikacje te można tworzyć w Oracle Content Management lub za pomocą narzędzi innych podmiotów.

OCE Toolkit pomaga przy wykonywaniu następujących zadań:

- Konfigurowanie lokalnego środowiska programistycznego pod kątem używania instancji Oracle Content Management do lokalnego tworzenia i testowania składników, szablonów, motywów i układów zawartości
- Tworzenie składników, szablonów serwisów i układów zawartości z przykładów, uruchamianie w zestawie testów tych elementów, eksplorowanie ich oraz opracowywanie składników, szablonów, motywów i układów zawartości w środowisku Developer Cloud Service
- Importowanie — z Oracle Content Management do projektu Developer Cloud Service — składników i szablonów serwisów, które zostały utworzone na potrzeby zarządzania źródłami i przyszłych potrzeb programistycznych
- Eksportowanie składników, szablonów lub układów zawartości z środowiska Developer Cloud Service do usługi Oracle Content Management, w której będą używane
- Kopiowanie już istniejących składników, szablonów lub układów zawartości
- Tworzenie testów jednostkowych
- Optymalizowanie składników
- Wprowadzanie składników i szablonów do środowiska wykonawczego Oracle Content Management

Dodatkowe informacje można znaleźć na stronie <https://github.com/oracle/content-and-experience-toolkit#readme>.

### Orkiestracja zasobów Experience

Jeśli do tworzenia zasobów Experience są używane narzędzia inne niż Oracle Content Management, można do tych zasobów podłączyć repozytoria Oracle Content Management, tak aby twórcy zawartości mogli podczas pracy uzyskiwać podgląd zmian w serwisie i aby — gdy zawartość ulegnie zmianie lub zostanie opublikowana — mogły być uaktywniane automatyczne kompilacje. Orkiestracja zasobów Experience automatyzuje proces Workflow, w którym uczestniczą dostawcy zawartości i programiści serwisów, upraszczając zarządzanie zasobami Experience i ich publikowanie.

## Tworzenie za pomocą zestawu OCE Toolkit

OCE Toolkit pomaga tworzyć szablony, motywy, składniki niestandardowe i układy zawartości serwisów dla Oracle Content Management.

Używając zestawu OCE Toolkit, można pracować we własnym środowisku programistycznym oraz korzystać z repozytoriów zasobów, z plików oraz z folderów dostępnych w Oracle Content Management. OCE Toolkit zawiera narzędzia pozwalające tworzyć i rozwijać niestandardowe składniki i szablony serwisów, w tym motywy i układy zawartości. Zawiera oprogramowanie do wykonywania testów lokalnych, ułatwiające szybkie tworzenie iteracyjne, a także przykładowe testy jednostkowe, od których można zacząć.

W poniższych tematach opisano, jak skonfigurować zestaw OCE Toolkit i używać go na komputerze lokalnym lub jako projektu usługi Developer Cloud Service:

- [Konfigurowanie zestawu OCE Toolkit na komputerze lokalnym](#)
- [Uaktualnienie do jQuery 3.5.x](#)
- [Tworzenie różnych elementów dla Oracle Content Management za pomocą usługi Developer Cloud Service](#)
- [Propagacja zmian ze środowiska testowego do produkcyjnego za pomocą zestawu OCE Toolkit](#)
- [Tworzenie serwisu z szablonu i zachowywanie tych samych identyfikatorów GUID zawartości](#)
- [Importowanie i eksportowanie taksonomii](#)
- [Importowanie i eksportowanie rekomendacji](#)
- [Tworzenie i rozwijanie niestandardowych edytorów pól za pomocą zestawu OCE Toolkit](#)
- [Przenoszenie lub aktualizowanie serwisu z jednego serwera do innego](#)
- [Indeksowanie stron serwisu za pomocą zestawu OCE Toolkit](#)
- [Indeksowanie serwisu wielojęzycznego za pomocą zestawu OCE Toolkit](#)
- [Tworzenie uproszczonego składnika, łatwiejszego do dalszego rozwijania](#)
- [Kompilowanie serwisu w celu poprawy wydajności stron serwisu w trybie wykonawczym](#)
- [Tworzenie na serwerze Oracle Content Management nowego serwisu lub nowego zlecenia tłumaczenia zasobu](#)
- [Tłumaczenie serwisu przez dostawcę usług tłumaczeniowych \(LSP\)](#)

## Konfigurowanie zestawu OCE Toolkit na komputerze lokalnym

Zestaw OCE Toolkit można pobrać — do komputera lokalnego — z serwisu GitHub.

Aby skonfigurować zestaw OCE Toolkit, należy postępować zgodnie z instrukcjami dostępnymi na stronie

<https://github.com/oracle/content-and-experience-toolkit/blob/master/README.md>

Ukończyć konfigurację i przygotowania do używania zestawu OCE Toolkit:

1. Instalowanie zależności za pomocą narzędzia npm
2. Korzystanie z narzędzia cec opartego na wierszu polecenia
3. Testowanie przy użyciu lokalnego zestawu testów

## Instalowanie zależności za pomocą narzędzia npm

Za pomocą narzędzia npm (node package manager) można zainstalować zależności serwisów dla projektu.

Jeśli jest używany proxy do uzyskiwania dostępu do Internetu, należy ustawić proxy dla narzędzia npm, używając polecenia `npm config`. Zob. <https://docs.npmjs.com/misc/config>. Aby ustawić proxy dla pakietu Bower, zob. <https://bower.io/docs/config/>.



### Uwaga:

Należy się upewnić, że na komputerze lokalnym zainstalowano środowisko Node.js w wersji 8.0.0 lub nowszej (<https://nodejs.org/>).

## Korzystanie z narzędzia cec opartego na wierszu polecenia

Oparte na wierszu polecenia międzyplatformowe narzędzie `cec` udostępnia polecenia służące do tworzenia szablonów i składników oraz zarządzania nimi.

Przed przystąpieniem do używania narzędzia `cec` należy w swoim lokalnym środowisku rozwojowym utworzyć źródło dla poleceń wymagających połączenia z serwerem Oracle Content Management.

Zintegrowana pomoc udostępnia informacje dotyczące uruchamiania poleceń wraz z przykładami. Aby wyświetlić zintegrowaną pomoc `cec`, można w wierszu poleceń wpisać polecenia `cec`:

- W oknie terminala przejść do katalogu `cec`.
- Aby wyświetlić polecenia `cec`, wpisać `cec`.

```
cec
-----
Usage: cec <command> [options]
```

```
Run cec <command> -h' to get the detailed help for the command.
```

```
Commands:
```

Documents	
cec create-folder <name> folder hierarchy on OCM server.	Creates a folder or [alias: cfd]
cec share-folder <name> users and groups on OCM server.	Shares folder with [alias: sfd]
cec unshare-folder <name> access to a shared folder on OCM server.	Deletes user or group [alias: usfd]
cec list-folder <path> hierarchy on OCM server.	Displays folder [alias: lfd]
cec download-folder <path> OCM server.	Downloads folder from [alias: dlfd]
cec upload-folder <path> server.	Uploads folder to OCM [alias: ulfd]
cec delete-folder <path> server.	Deletes folder on OCM [alias:     ]
cec download-file <file> from OCM server.	Downloads file <file> [alias: dlf]
cec upload-file <file> OCM server.	Uploads file <file> to [alias: ulf]
cec delete-file <file> server.	Deletes file on OCM [alias:     ]
Components	
cec create-component <name> <name>.	Creates the component [alias: cc]
cec copy-component <source> [<destination>] component named <source> to <destination>.	Copies an existing [alias: cpc]
cec import-component <zip> from <zip>.	Imports a component [alias: ic]
cec export-component <name> <name> as a zip file.	Exports the component [alias: ec]
cec download-component <names> components <names> from the OCM server.	Downloads the [alias: dlcp]
cec upload-component <names> <names> to the OCM server.	Uploads the components [alias: ulcp]
cec control-component <action> <action> on components on OCM server.	Performs action [alias: ctcp]
cec share-component <name> users and groups on OCM server.	Shares component with [alias: sc]
cec unshare-component <name> access to a component on OCM server.	Deletes user or group [alias: usc]
Templates	
cec create-template <name> <name>.	Creates the template [alias: ct]
cec create-template-from-site <name> <name> from site <site> on the OCM server.	Creates the template [alias: ctfs]
cec download-template <name> <name> from the OCM server.	Downloads the template [alias: dlt]
cec compile-template <source> within the template.	Compiles the site [alias: cmpt]
cec copy-template <source> [<destination>] template named <source> to <destination>.	Copies an existing [alias: cpt]
cec import-template <zip>	Imports a template from

<zip>.	[alias: it]
cec export-template <name>	Exports the
template <name> as a zip file.	[alias:
et]	
cec upload-template <name>	Uploads the
template <name> to the OCM server.	[alias:
ult]	
cec delete-template <name>	Deletes the
template <name> on the OCM server.	
[alias: ]	
cec share-template <name>	Shares template
with users and groups on OCM server.	[alias: stm]
cec unshare-template <name>	Deletes user or
group access to a template on OCM server.	[alias: ustm]
cec update-template <action>	Performs action on
a local template.	[alias: ut]
cec describe-template <name>	Describes the
template <name> package.	[alias:
dst]	
cec create-template-report <name>	Generates an asset
usage report for the template <name> package.	[alias: cctr]
Themes	
cec add-component-to-theme <component>	Adds a component
to a theme.	[alias: actt]
cec remove-component-from-theme <component>	Removes
a component from a theme.	
[alias: rcft]	
cec control-theme <action>	Performs action
<action> on theme on OCM server.	[alias: ctt]
cec share-theme <name>	Shares theme with
users and groups on OCM server.	[alias: sth]
cec unshare-theme <name>	Deletes user or
group access to a theme on OCM server.	[alias: usth]
Sites	
cec create-site <name>	Creates Enterprise
Site <name>.	[alias: cs]
cec copy-site <name>	Copies Enterprise
Site <name>.	[alias: cps]
cec update-site <name>	Update Enterprise
Site <name>.	[alias: us]
cec transfer-site <name>	Transfers a site
from one OCM server to another.	[alias: ts]
cec transfer-site-content <name>	Creates scripts to
transfer site content from one OCM server to another.	[alias: tsc]
cec validate-site <name>	Validates site
<name>.	[alias: vs]
cec control-site <action>	Performs action
<action> on site on OCM server.	[alias: cts]
cec share-site <name>	Shares site with
users and groups on OCM server.	[alias: ss]
cec unshare-site <name>	Deletes user or
group access to a site on OCM server.	[alias: uss]
cec get-site-security <name>	Gets site security



on OCM server.	[alias: gss]
cec set-site-security <name>	Sets site security on
OCM server.	[alias: sss]
cec index-site <site>	Index the page content
of site <site> on OCM server.	[alias: is]
cec create-site-map <site>	Creates a site map for
site <site> on OCM server.	[alias: csm]
cec create-rss-feed <site>	Creates RSS feed for
site <site> on OCM server.	[alias: crf]
cec create-asset-report <site>	Generates an asset
usage report for site <site> on OCM server.	[alias: car]
cec upload-static-site-files <path>	Uploads files to render
statically from a site on OCM server.	[alias: ulss]
cec download-static-site-files <site>	Downloads the static
files from a site on OCM server.	[alias: dlss]
cec delete-static-site-files <site>	Deletes the static
files from a site on OCM server.	[alias: ]
cec refresh-prerender-cache <site>	Refreshes pre-render
cache for a site on OCM server.	[alias: rpc]
cec migrate-site <site>	Migrates a site from
OCI IC server to EC server.	[alias: ms]
Assets	
cec download-content	Downloads content from
OCM server.	[alias: dlc]
cec upload-content <name>	Uploads local content
to a repository on OCM server.	[alias: ulc]
cec control-content <action>	Performs action
<action> on channel items on OCM server.	[alias: ctct]
cec transfer-content <repository>	Creates scripts to
transfer content from one OCM server to another.	[alias: tc]
cec list-assets	Lists assets on OCM
server.	[alias: la]
cec create-digital-asset	Creates digital
asset	[alias: cda]
cec update-digital-asset <id>	Updates digital
asset	[alias: uda]
cec copy-assets <repository>	Copies assets to
another repository on OCM server.	[alias: ca]
cec create-asset-usage-report <assets>	Generates an asset
usage report for assets on OCM server.	[alias: caur]
Content	
cec create-repository <name>	Creates a repository on
OCM server.	[alias: cr]
cec control-repository <action>	Performs action
<action> on repositories on OCM server.	[alias: ctr]
cec share-repository <name>	Shares repository with
users and groups on OCM server.	[alias: sr]
cec unshare-repository <name>	Deletes user or group
access to a repository on OCM server.	[alias: usr]
cec create-channel <name>	Creates a channel on
OCM server.	[alias: cch]
cec share-channel <name>	Shares channel with
users and groups on OCM server.	[alias: sch]

cec unshare-channel <name> group access to a channel on OCM server.	Deletes user or [alias: usch]
cec create-localization-policy <name> a localization policy on OCM server. [alias: clp]	Creates
cec list-server-content-types types from server.	Lists all content [alias: lsct]
cec share-type <name> users and groups on OCM server.	Shares type with [alias: st]
cec unshare-type <name> group access to a type on OCM server.	Deletes user or [alias: ust]
cec download-type <name> from OCM server.	Downloads types [alias: dltp]
cec upload-type <name> OCM server.	Uploads types to [alias: ultp]
cec update-type <action> <action> on a type	Performs action [alias: utp]
cec download-recommendation <name> a recommendation from the OCM server. [alias: dlr]	Downloads
cec upload-recommendation <name> a recommendation to the OCM server. [alias: ulr]	Uploads
cec create-contentlayout <name> layout based on a content type.	Creates a content [alias: ccl]
cec add-contentlayout-mapping <contentlayout> type and content layout mapping.	Creates content [alias: aclm]
cec remove-contentlayout-mapping <contentlayout> layout mapping.	Removes a content [alias: rclm]
cec add-field-editor <name> editor to a field in a content type. afe]	Adds a field [alias:
cec remove-field-editor <name> editor from a field in a content type.	Removes a field [alias: rfe]
cec migrate-content <name> from OCI IC server to EC server.	Migrates content [alias: mc]
cec compile-content content items generating HTML renditions. cmppc]	Compiles the [alias:
cec upload-compiled-content <path> compiled content to OCM server. ulcc]	Uploads the [alias:
 Taxonomies	
cec download-taxonomy <name> a taxonomy from OCM server. dltx]	Downloads [alias:
cec upload-taxonomy <taxonomy> to OCM server.	Uploads a taxonomy [alias: ultx]
cec control-taxonomy <action> taxonomy on OCM server.	Performs action on [alias: cttx]
 Translation	
cec list-translation-jobs jobs.	Lists translation [alias: ltj]

cec create-translation-job <name> job <name> for a site on OCM server.	Creates a translation [alias: ctj]
cec download-translation-job <name> job <name> from OCM server.	Downloads translation [alias: dtj]
cec submit-translation-job <name> <name> to translation connection <connection>.	Submits translation job [alias: stj]
cec refresh-translation-job <name> job <name> from translation connection.	Refreshes translation [alias: rtj]
cec ingest-translation-job <name> <name> from translation connection and ingest.	Gets translated job [alias: itj]
cec upload-translation-job <name> <name> to OCM server.	Uploads translation job [alias: utj]
cec create-translation-connector <name> connector <name>.	Creates translation [alias: ctc]
cec start-translation-connector <name> connector <name>.	Starts translation [alias: stc]
cec register-translation-connector <name> connector.	Registers a translation [alias: rtc]

#### Groups

cec create-group <name> OCM server.	Creates an OCM group on [alias: cg]
cec delete-group <name> OCM server.	Deletes an OCM group on [alias: ]
cec add-member-to-group <name> to an OCM group on OCM server.	Adds users and groups [alias: amtg]
cec remove-member-from-group <name> groups from an OCM group on OCM server.	Removes users and [alias: rmfg]

#### Local Environment

cec create-encryption-key <file> key to encrypt/decrypt password for servers.	Create an encryption [alias: cek]
cec register-server <name> server.	Registers a OCM [alias: rs]
cec set-oauth-token <token> a registered server.	Set OAuth token for [alias: sot]
cec list resources.	Lists local or server [alias: l]
cec execute-get <endpoint> request to a REST API endpoint on OCM server	Makes an HTTP GET [alias: exeg]
cec install tree.	Creates source [alias: i]
cec develop server.	Starts a test [alias: d]
cec sync-server server.	Starts a sync [alias: scs]
cec webhook-server server.	Starts a webhook [alias: whs]

#### Options:

```
--version, -v Show version number [boolean]
--help, -h Show Help
```

```
-----
cec create-folder
```

```
-----
Usage: cec create-folder <name>
```

Create a folder or folder hierarchy on OCM server. Specify the server with `-s <server>` or use the one specified in `cec.properties` file.

```
Options:
  --help, -h    Show Help
  --server, -s  <server> The registered OCM server
```

```
Examples:
  cec create-folder Projects           Creates folder Projects under the
  Home folder
  cec create-folder Projects/Blogs     Creates folder Projects under the
  Home folder and folder Blogs under Projects
  cec create-folder Projects -s UAT    Creates folder Projects under the
  Home folder on the registered server UAT
```

```
-----
cec share-folder
-----
Usage: cec share-folder <name>
```

Shares folder with users and groups on OCM server and assign a role. Specify the server with `-s <server>` or use the one specified in `cec.properties` file. The valid roles are

```
  manager
  contributor
  downloader
  viewer
```

```
Options:
  --help, -h    Show Help
  --users, -u   The comma separated list of user names
  --groups, -g  The comma separated list of group names
  --role, -r    The role [manager | contributor | downloader | viewer]
  to assign to the users or groups [required]
  --server, -s  <server> The registered OCM server
```

```
Examples:
  cec share-folder Projects/Blogs -u user1,user2 -r manager
  Share folder Projects/Blogs with user user1 and user2 and assign
  Manager role to them
  cec share-folder Projects/Blogs -u user1,user2 -g group1 -r manager
  Share folder Projects/Blogs with user user1, user2 and group group1 and
  assign Manager role to them
  cec share-folder Projects/Blogs -g group1,group2 -r manager
  Share folder Projects/Blogs with group group1 and group2 and assign
  Manager role to them
  cec share-folder Projects/Blogs -u user1,user2 -r manager -s UAT
  Share folder Projects/Blogs with user user1 and user2 and assign
  Manager role to them on the registered server UAT
```

```
-----  
cec unshare-folder  
-----
```

Usage: cec unshare-folder <name>

Deletes user or group access to a shared folder on OCM server. Specify the server with -s <server> or use the one specified in cec.properties file.

Options:

```
--help, -h    Show Help  
--users, -u   The comma separated list of user names  
--groups, -g  The comma separated list of group names  
--server, -s  <server> The registered OCM server
```

Examples:

```
cec unshare-folder Projects/Blogs -u user1,user2  
cec unshare-folder Projects/Blogs -g group1,group2  
cec unshare-folder Projects/Blogs -u user1,user2 -g group1,group2  
cec unshare-folder Projects/Blogs -u user1,user2 -s UAT
```

```
-----  
cec list-folder  
-----
```

Usage: cec list-folder <path>

Displays folder and all its content on OCM server. Specify the server with -s <server> or use the one specified in cec.properties file.

Options:

```
--help, -h    Show Help  
--server, -s  The registered OCM server
```

Examples:

```
cec list-folder Releases/1  
cec list-folder Releases/1 -s UAT  
cec list-folder site:blog1  
cec list-folder theme:blog1Theme  
cec list-folder component:Comp1/assets
```

```
-----  
cec download-folder  
-----
```

Usage: cec download-folder <path>

Downloads folder and all its content from OCM server. Specify the server with -s <server> or use the one specified in cec.properties file. Optionally specify -f <folder> to save the folder on the local system.

Options:

```
--help, -h    Show Help  
--folder, -f  <folder> Local folder to save the folder on OCM server  
--server, -s  <server> The registered OCM server
```

Examples:

```
cec download-folder Releases/1                                Downloads folder
```

```

Releases/1 from OCM server and save to local folder src/documents/
  cec download-folder / Downloads
all documents from OCM server and save to local folder src/documents/
  cec download-folder Releases/1 -s UAT Downloads
folder Releases/1 from the registered server UAT and save to local
folder src/documents/
  cec download-folder Releases/1 -f ~/Downloads Downloads
folder Releases/1 from OCM server and save to local folder ~/Download/
  cec download-folder Releases/1 -f . Downloads
folder Releases/1 from OCM server and save to the current local folder
  cec download-folder site:blog1 -f ~/Downloads/blog1Files Downloads
all files of site blog1 and save to local folder ~/Download/blog1Files
  cec download-folder theme:blog1Theme Downloads
all files of theme blog1Theme and save to local folder src/documents/
blog1Theme/
  cec download-folder component:Comp1/assets Downloads
all files in folder assets of component Comp1 and save to local folder
src/documents/Comp1/assets/

```

```

-----
cec upload-folder
-----

```

Usage: cec upload-folder <path>

Uploads folder and all its content to OCM server. Specify the server with -s <server> or use the one specified in cec.properties file. Optionally specify -f <folder> to set the parent folder on OCM server.

#### Options:

```

--help, -h    Show Help
--folder, -f  <folder> The parent folder on OCM server
--server, -s  <server> The registered OCM server

```

#### Examples:

```

cec upload-folder ~/Downloads/docs
Uploads all content from ~/Downloads/docs to folder docs on the server
cec upload-folder ~/Downloads/docs/
Uploads all content from ~/Downloads/docs to the Home folder on the
server
cec upload-folder ~/Downloads/docs -f Mydoc
Uploads all content from ~/Downloads/docs to folder Mydoc/docs on the
server
cec upload-folder ~/Downloads/docs/ -f Mydoc
Uploads all content from ~/Downloads/docs to folder Mydoc on the server
cec upload-folder ~/Downloads/docs -s UAT
Uploads all content from ~/Downloads/docs to folder docs on the
registered server UAT
cec upload-folder ~/Downloads/docs/ -f site:blog1/settings/misc
Uploads all content from ~/Downloads/docs to folder settings/misc of
site blog1
cec upload-folder ~/Downloads/docs -f theme:blog1Theme
Uploads all content from ~/Downloads/docs to folder docs of theme
blog1Theme
cec upload-folder ~/Downloads/docs -f component:Comp1
Uploads all content from ~/Downloads/docs to folder docs of component

```

Comp1

```
-----
cec delete-folder
-----
```

Usage: cec delete-folder <path>

Deletes folder and all its content on OCM server. Specify the server with `-s <server>` or use the one specified in `cec.properties` file. Optionally specify `-p` to permanently delete the folder.

Options:

```
--help, -h      Show Help
--server, -s    <server> The registered OCM server
--permanent, -p Delete the folder permanently
```

Examples:

```
cec delete-folder Import/docs
cec delete-folder Import/docs -s UAT
cec delete-folder Import/docs -p
cec delete-folder site:blog1/docs
cec delete-folder theme:blog1Theme/docs
cec delete-folder component:Comp1/docs
```

```
-----
cec download-file
-----
```

Usage: cec download-file <file>

Downloads file <file> from OCM server. Specify the server with `-s <server>` or use the one specified in `cec.properties` file. Optionally specify `-f <folder>` to save the file on the local system.

Options:

```
--help, -h      Show Help
--folder, -f    <folder> Local folder to save the file
--server, -s    <server> The registered OCM server
```

Examples:

```
cec download-file Releases/Projects.pdf Downloads
the file from OCM server and save to local folder src/documents/
cec download-file Releases/Projects.pdf -s UAT Downloads
the file from the registered server UAT and save to local folder src/
documents/
cec download-file Releases/Projects.pdf -f ~/Downloads Downloads
the file from OCM server and save to local folder ~/Download/
cec download-file Releases/Projects.pdf -f . Downloads
the file from OCM server and save to the current local folder
cec download-file site:blog1/siteinfo.json Downloads
the file from folder blog1 and save to local folder src/documents/blog1
cec download-file theme:blog1Theme/designs/default/design.css Downloads
the css file from folder designs/default of theme blog1Theme and save to
local folder src/documents/blog1Theme/designs/default/
cec download-file component:Comp1/assets/render.js Downloads
the js file from folder assets of component Comp1 and save to local folder
```

```
src/documents/Compl/assets/
```

```
-----  
cec upload-file  
-----
```

```
Usage: cec upload-file <file>
```

Uploads file <file> to OCM server. Specify the server with `-s <server>` or use the one specified in `cec.properties` file. Optionally specify `-f <folder>` to set the parent folder on OCM server.

Options:

```
--help, -h      Show Help  
--folder, -f    <folder> The parent folder on OCM server  
--server, -s    <server> The registered OCM server
```

Examples:

```
cec upload-file ~/Documents/  
Projects.pdf           Uploads the file to the  
Home folder  
cec upload-file ~/Documents/Projects.pdf -s  
UAT                   Uploads the file to the Home folder on  
the registered server UAT  
cec upload-file ~/Documents/Projects.pdf -f Doc/  
Plan                 Uploads the file to folder Doc/Plan  
cec upload-file ~/Documents/Projects.pdf -f site:blog1/settings/  
misc                Uploads the file to folder settings/misc of site blog1  
cec upload-file ~/Documents/style1.css -f theme:blog1Theme/designs/  
default             Uploads the css file to folder designs/default of theme  
blog1Theme  
cec upload-file ~/Documents/compl.js -f component:Compl/  
assets               Uploads the js file to folder assets of component  
Compl
```

```
-----  
cec delete-file  
-----
```

```
Usage: cec delete-file <file>
```

Deletes file on OCM server. Specify the server with `-s <server>` or use the one specified in `cec.properties` file. Optionally specify `-p` to permanently delete the file.

Options:

```
--help, -h      Show Help  
--server, -s    <server> The registered OCM server  
--permanent, -p Delete the file permanently
```

Examples:

```
cec delete-file docs/Projects.pdf  
cec delete-file docs/Projects.pdf -s UAT  
cec delete-file docs/Projects.pdf -p  
cec delete-file site:blog1/docs/Projects.pdf  
cec delete-file theme:blog1Theme/docs/Projects.pdf  
cec delete-file component:Compl/docs/Projects.pdf
```



```
-----  
cec create-component  
-----
```

Usage: cec create-component <name>

Creates the component <name>. By default, it creates a local component. Optionally specify -f <source> to create from a different source.

Valid values for <source> are:

```
local  
local-template  
local-iframe  
remote  
sectionlayout  
Sample-File-List  
Sample-Folder-List  
Sample-Document-Manager  
Sample-Process-Start-Form  
Sample-Process-Task-List  
Sample-Process-Task-Details  
Sample-Stocks-Embedded  
Sample-Text-With-Image  
Sample-To-Do  
Anchor  
ContentForm  
Document-Search  
JET-CCA-Demo-Card  
MapFieldEditor  
Sample-OPA-Interview  
SimpleHTML  
SliderFieldEditor  
TextFieldEditor
```

Options:

```
--help, -h Show Help  
--from, -f <from> Source to create from
```

Examples:

```
cec create-component Comp1  
cec create-component Comp2 -f Sample-File-List
```

```
-----  
cec copy-component  
-----
```

Usage: cec copy-component <source> [<destination>]

Copies an existing component named <source> to <destination>. <source> is a folder name from src/components

Options:

```
--help, -h Show Help
```

Examples:

```
cec copy-component Sample-To-Do Compl Copies Sample-To-Do to Compl.
```

```
-----  
cec import-component  
-----
```

```
Usage: cec import-component <zip>
```

Imports a component from <zip>. Specify the absolute path of the zip file. The zip file name will be used as the component name.

Options:

```
--help, -h Show Help
```

Examples:

```
cec import-component /home/Compl.zip Imports the component Compl.
```

```
-----  
cec export-component  
-----
```

```
Usage: cec export-component <name>
```

Exports the component <name> as a zip file.

Options:

```
--help, -h Show Help
```

Examples:

```
cec export-component Sample-To-Do Exports the component Sample-To-Do.
```

```
-----  
cec download-component  
-----
```

```
Usage: cec download-component <names>
```

Downloads the components <names> from the Content Management server. Specify the server with -s <server> or use the one specified in cec.properties file.

Options:

```
--help, -h Show Help  
--server, -s <server> The registered OCM server
```

Examples:

```
cec download-component Sample-To-Do  
cec download-component Sample-To-Do,Sample-To-Do2  
cec download-component Sample-To-Do -s UAT
```

```
-----  
cec upload-component  
-----
```

```
Usage: cec upload-component <names>
```

Uploads the components <names> to the Content Management server. Specify the server with -s <server> or use the one specified in cec.properties file. Optionally specify -p to publish the component

after deploy. Optionally specify `-f <folder>` to set the folder to upload the component zip file.

Options:

```
--help, -h      Show Help
--folder, -f    <folder> Folder to upload the component zip file
--publish, -p   Publish the component
--server, -s    <server> The registered OCM server
```

Examples:

```
cec upload-component Sample-To-Do           Uploads the
component Sample-To-Do to the server specified in cec.properties.
cec upload-component Sample-To-Do -s UAT     Uploads the
component Sample-To-Do to the registered server UAT.
cec upload-component Sample-To-Do -p        Uploads and
publishes the component Sample-To-Do.
cec upload-component Sample-To-Do,Sample-To-Do2 Uploads component
Sample-To-Do and Sample-To-Do2.
cec upload-component Sample-To-Do -f Import/Components Uploads file
Sample-To-Do.zip to folder Import/Components and imports the component
Sample-To-Do.
```

```
-----
cec control-component
-----
```

Usage: `cec control-component <action>`

Perform `<action>` on components on OCM server. Specify the components with `-c <components>`. Specify the server with `-s <server>` or use the one specified in `cec.properties` file. The valid actions are

`publish`

Options:

```
--help, -h      Show Help
--components, -c <components> The comma separated list of components
[required]
--server, -s    <server> The registered OCM server
```

Examples:

```
cec control-component publish -c Comp1       Publish component
Comp1 on the server specified in cec.properties file
cec control-component publish -c Comp1 -s UAT Publish component
Comp1 on the registered server UAT
cec control-component publish -c Comp1,Comp2 -s UAT Publish component
Comp1 and Comp2 on the registered server UAT
```

```
-----
cec share-component
-----
```

Usage: `cec share-component <name>`

Shares component with users and groups on OCM server and assign a role. Specify the server with `-s <server>` or use the one specified in

cec.properties file. The valid roles are

```
manager
contributor
downloader
viewer
```

Options:

```
--help, -h    Show Help
--users, -u    The comma separated list of user names
--groups, -g   The comma separated list of group names
--role, -r     The role [manager | contributor | downloader | viewer]
to assign to the users or groups [required]
--server, -s  <server> The registered OCM server
```

Examples:

```
cec share-component Comp1 -u user1,user2 -r manager
Share component Comp1 with user user1 and user2 and assign Manager role
to them
cec share-component Comp1 -u user1,user2 -g group1,group2 -r manager
Share component Comp1 with user user1 and user2 and group group1 and
group2 and assign Manager role to them
cec share-component Comp1 -u user1,user2 -r manager -s UAT
Share component Comp1 with user user1 and user2 and assign Manager role
to them on the registered server UAT
```

```
-----
cec unshare-component
-----
```

Usage: cec unshare-component <name>

Deletes user or group access to a component on OCM server. Specify the server with -s <server> or use the one specified in cec.properties file.

Options:

```
--help, -h    Show Help
--users, -u    The comma separated list of user names
--groups, -g   The comma separated list of group names
--server, -s  <server> The registered OCM server
```

Examples:

```
cec unshare-component Comp1 -u user1,user2
cec unshare-component Comp1 -u user1,user2 -g group1,group2
cec unshare-component Comp1 -u user1,user2 -s UAT
```

```
-----
cec create-template
-----
```

Usage: cec create-template <name>

Creates the template <name>. By default, it creates a StarterTemplate. Optionally specify -f <source> to create from different source.

Valid values for <source> are:

```

CafeSupremoLite
JETStarterTemplate
StarterTemplate
BlogTemplate
VBCSSamplesTemplate
search_template

```

To create template based on a site on OCM server, specify `-s <site>` and specify the server with `-r <server>` or use the one specified in `cec.properties` file.

Options:

```

--help, -h           Show Help
--from, -f           <source> Source to create from
--site, -s           <site> Site to create from
--excludecontent, -x Exclude content
--enterprisetemplate, -e Enterprise template
--server, -r         <server> The registered OCM server

```

Examples:

```

cec create-template Temp1
cec create-template Temp2 -f CafeSupremoLite
cec create-template Temp1 -s Site1           Create template
Temp1 based on site Site1 on OCM server
cec create-template Temp1 -s Site1 -x       Create template
Temp1 based on site Site1 on OCM server and exclude the content in the site
cec create-template Temp1 -s Site1 -r UAT   Create template
Temp1 based on site Site1 on the registered server UAT
cec create-template EnterpriseTemp1 -s StandardSite1 -e Create enterprise
template EnterpriseTemp1 based on standard site StandardSite1 on OCM server

```

```

-----
cec create-template-from-site
-----

```

Usage: `cec create-template-from-site <name>`

Creates the template `<name>` from site `<site>` on the Content Management server. Specify the server with `-r <server>` or use the one specified in `cec.properties` file. Optionally specify `<includeunpublishedassets>` to include unpublished content items and digital assets in your template.

Options:

```

--help, -h           Show Help
--site, -s           <site> Site to create from [required]
--includeunpublishedassets, -i flag to indicate to include unpublished
content items and digital assets in your template
--enterprisetemplate, -e Enterprise template
--server, -r         <server> The registered OCM server

```

Examples:

```

cec create-template-from-site BlogTemplate -s BlogSite
cec create-template-from-site BlogTemplate -s BlogSite -r UAT
cec create-template-from-site BlogTemplate -s BlogSite -i -r UAT
cec create-template-from-site EnterpriseTemplate -s StandardSite -e

```

```
-----
cec download-template
-----
```

Usage: cec download-template <name>

Downloads the template <name> from the Content Management server. Specify the server with -s <server> or use the one specified in cec.properties file.

Options:

```
--help, -h      Show Help
--server, -s    <server> The registered OCM server
```

Examples:

```
cec download-template BlogTemplate
cec download-template BlogTemplate -s UAT
```

```
-----
cec compile-template
-----
```

Usage: cec compile-template <source>

Compiles all the pages within the site of the template and places the compiled pages under the sites assets folder.

Optionally specify -s <server> to make content queries against this server (requires channelToken).

Optionally specify -c <channelToken> to use this channelToken when generating any content URLs.

Optionally specify -t <contentType> [draft | published] content to retrieve from the server type, defaults to published.

Optionally specify -p <pages> the set of pages to compile.

Optionally specify -d <debug> to start the compilation with --inspect-brk flag.

Optionally specify -r <recurse> recurse through all child pages of specified pages.

Optionally specify -l <includeLocale> include default locale when creating pages.

Optionally specify -a <targetDevice> [desktop | mobile] target device type when using adaptive layouts.

Optionally specify -v <verbose> to display all warning messages during compilation.

Optionally specify -i <ignoreErrors> ignore compilation errors when calculating the exit code for the process.

Options:

```
--help, -h      Show Help
--server, -s    The registered OCM server
--channelToken, -c The channel access token to use for
content URLs
--type, -t      The type of content to retrieve from
the serve [published | draft]
--pages, -p     The list of pages to compile
--recurse, -r   Compile all child pages of those
specified in the page list
```

```

--debug, -d                Start the compiler with "--inspect-brk"
option to debug compilation
--noDetailPages, -e        Do not generate compiled detail pages
--noDefaultDetailPageLink, -o Do not generate compiled detail page for
items/content lists that use the default detail page
--targetDevice, -a        The target device type when using adaptive
layouts [desktop | mobile]
--siteName, -n            The target site name to use when compiling
the template
--secureSite, -u          The target site is a secure site
--includeLocale, -l       Include default locale when creating pages
--verbose, -v            Run in verbose mode to display all warning
messages during compilation.
--ignoreErrors, -i       Ignore compilation errors when calculating
the exit code for the process.

```

## Examples:

```

cec compile-template Templ           Compiles the
site in template Templ using content stored in the template.
cec compile-template Templ -c channelToken      Compiles the
site in template Templ using the given channelToken for any content URLs.
cec compile-template Templ -c channelToken -s UAT -t draft Compiles the
site in template Templ retrieving draft content from the specified server.
cec compile-template Templ -p 104,112,183 -r    Compiles the
specified pages in the site in template Templ including all child pages.
cec compile-template Templ -d                Waits for the
debugger to be attached. Once attached, compiles the site in template Templ.

```

```

-----
cec copy-template
-----

```

```
Usage: cec copy-template <source> [<destination>]
```

Copies an existing template named <source> to <destination>. <source> is a folder name from src/templates

## Options:

```
--help, -h Show Help
```

## Examples:

```
cec copy-template Templ Temp2 Copies Templ to Temp2.
```

```

-----
cec import-template
-----

```

```
Usage: cec import-template <zip>
```

Imports a template from <zip>. Specify the absolute path of the zip file. The zip file name will be used as the template name.

## Options:

```
--help, -h Show Help
```

## Examples:

```
cec import-template /home/Templ.zip Imports the template Templ.
```

```
-----  
cec export-template  
-----
```

Usage: cec export-template <name>

Exports the template <name> as a zip file and provides the location of the zip file.

Options:

--help, -h Show Help  
--optimize, -o Optimize the template

Examples:

cec export-template Templ Exports the template Templ.

```
-----  
cec upload-template  
-----
```

Usage: cec upload-template <name>

Uploads the template <name> to the Content Management server. Specify the server with -s <server> or use the one specified in cec.properties file. Optionally specify -f <folder> to set the folder to upload the template zip file. Optionally specify -p to publish theme and components after import.

Options:

--help, -h Show Help  
--folder, -f <folder> Folder to upload the template zip file  
--server, -s <server> The registered OCM server  
--optimize, -o Optimize the template  
--excludecontenttemplate, -x Exclude content template  
--excludecomponents, -e Exclude components  
--publish, -p Publish theme and components

Examples:

cec upload-template StarterTemplate Uploads the template StarterTemplate.  
cec upload-template StarterTemplate -s UAT Uploads the template StarterTemplate to the registered server UAT.  
cec upload-template StarterTemplate -f Import/Templates Uploads file StarterTemplate.zip to folder Import/Templates and imports the template StarterTemplate.  
cec upload-template StarterTemplate -p Publish the theme and all components in StarterTemplate.zip after import  
cec upload-template StarterTemplate -o Optimizes and uploads the template StarterTemplate.  
cec upload-template StarterTemplate -x Exclude the "Content Template" from the template upload. "Content Template" upload can be managed independently.  
cec upload-template StarterTemplate -e Exclude all components from the template upload. Components can be uploaded independently.



```
-----  
cec delete-template  
-----  
Usage: cec delete-template <name>
```

Deletes the template <name> on the Content Management server. Specify the server with `-s <server>` or use the one specified in `cec.properties` file. Optionally specify `-p` to permanently delete the template.

Options:

- `--help, -h` Show Help
- `--server, -s <server>` The registered OCM server
- `--permanent, -p` flag to indicate to permanently delete the template

Examples:

```
cec delete-template BlogTemplate  
cec delete-template BlogTemplate -p  
cec delete-template BlogTemplate -s UAT
```

```
-----  
cec share-template  
-----  
Usage: cec share-template <name>
```

Shares template with users and groups on OCM server and assign a role. Specify the server with `-s <server>` or use the one specified in `cec.properties` file. The valid roles are

```
manager  
contributor  
downloader  
viewer
```

Options:

- `--help, -h` Show Help
- `--users, -u` The comma separated list of user names
- `--groups, -g` The comma separated list of group names
- `--role, -r` The role [manager | contributor | downloader | viewer] to assign to the users or groups [required]
- `--server, -s <server>` The registered OCM server

Examples:

```
cec share-template Templatel -u user1,user2 -r manager  
Share template Templatel with user user1 and user2 and assign Manager role to them  
cec share-template Templatel -u user1,user2 -g group1,group2 -r manager  
Share template Templatel with user user1 and user2 and group group1 and group2 and assign Manager role to them  
cec share-template Templatel -u user1,user2 -r manager -s UAT  
Share template Templatel with user user1 and user2 and assign Manager role to them on the registered server UAT
```

```
cec unshare-template
```

```
-----  
Usage: cec unshare-template <name>
```

Deletes user or group access to a template on OCM server. Specify the server with `-s <server>` or use the one specified in `cec.properties` file.

Options:

```
--help, -h      Show Help  
--users, -u     The comma separated list of user names  
--groups, -g    The comma separated list of group names  
--server, -s    <server> The registered OCM server
```

Examples:

```
cec unshare-template Templat1 -u user1,user2  
cec unshare-template Templat1 -u user1,user2 -g group1,group2  
cec unshare-template Templat1 -u user1,user2 -s UAT
```

```
-----  
cec update-template
```

```
-----  
Usage: cec update-template <action>
```

Performs action `<action>` on a local template. Optionally specify `-c` for other local content. The valid actions are

```
rename-asset-id
```

Options:

```
--help, -h      Show Help  
--template, -t  The template [required]  
--content, -c   The comma separated list of local content
```

Examples:

```
cec update-template rename-asset-id -t Templat1  
cec update-template rename-asset-id -t Templat1 -c Content1,Content2
```

```
-----  
cec describe-template
```

```
-----  
Usage: cec describe-template <name>
```

Describes the template `<name>` package such as theme, components and content types.

Options:

```
--help, -h Show Help
```

Examples:

```
cec describe-template StarterTemplate Describes the template  
StarterTemplate package
```

```
-----  
cec create-template-report
```

```

-----
Usage: cec create-template-report <name>

Generates an asset usage report for the template <name> package. Optionally
specify -o to save the report to a json file.

Options:
  --help, -h          Show Help
  --includepagelinks, -i  Include validating page links
  --output, -o         Output the report to a JSON file

Examples:
  cec create-template-report StarterTemplate
  cec create-template-report StarterTemplate -
o                               The report will be saved to
StarterTemplateAssetUsage.json at the current local location
  cec create-template-report StarterTemplate -o ~/
Documents                       The report will be saved to ~/
Documents/StarterTemplateAssetUsage.json
  cec create-template-report StarterTemplate -o ~/Documents/
StarterTemplateReport.json       The report will be saved to ~/Documents/
StarterTemplateReport.json
  cec create-template-report StarterTemplate -
i                               Include validating page links

-----
cec add-component-to-theme
-----
Usage: cec add-component-to-theme <component>

Adds a component to a theme. Optionally specify -c <category> to set the
component category.

Options:
  --help, -h          Show Help
  --theme, -t         <theme> Theme [required]
  --category, -c     <category> component category

Examples:
  cec add-component-to-theme Sample-To-Do -t BlogTheme
  cec add-component-to-theme Sample-To-Do -t BlogTheme -c Samples

-----
cec remove-component-from-theme
-----
Usage: cec remove-component-from-theme <component>

Removes a component from a theme.

Options:
  --help, -h          Show Help
  --theme, -t         <theme> Theme [required]

Examples:
  cec remove-component-from-theme Sample-To-Do -t BlogTheme

```

```
-----  
cec control-theme  
-----
```

Usage: cec control-theme <action>

Perform <action> on theme on OCM server. Specify the theme with -t <theme>. Specify the server with -s <server> or use the one specified in cec.properties file. The valid actions are

publish

Options:

--help, -h Show Help  
--theme, -t <theme> The theme [required]  
--server, -s <server> The registered OCM server

Examples:

```
cec control-theme publish -t Theme1          Publish theme Theme1 on  
the server specified in cec.properties file  
cec control-theme publish -t Theme1 -s UAT  Publish theme Theme1 on  
the registered server UAT
```

```
-----  
cec share-theme  
-----
```

Usage: cec share-theme <name>

Shares theme with users and groups on OCM server and assign a role. Specify the server with -s <server> or use the one specified in cec.properties file. The valid roles are

manager  
contributor  
downloader  
viewer

Options:

--help, -h Show Help  
--users, -u The comma separated list of user names  
--groups, -g The comma separated list of group names  
--role, -r The role [manager | contributor | downloader | viewer]  
to assign to the users or groups [required]  
--server, -s <server> The registered OCM server

Examples:

```
cec share-theme Theme1 -u user1,user2 -r manager  
Share theme Theme1 with user user1 and user2 and assign Manager role to  
them  
cec share-theme Theme1 -u user1,user2 -g group1,group2 -r manager  
Share theme Theme1 with user user1 and user2 and group group1 and  
group2 and assign Manager role to them  
cec share-theme Theme1 -u user1,user2 -r manager -s UAT
```

Share theme Themel with user user1 and user2 and assign Manager role to them on the registered server UAT

```
-----
cec unshare-theme
-----
```

Usage: cec unshare-theme <name>

Deletes user or group access to a theme on OCM server. Specify the server with -s <server> or use the one specified in cec.properties file.

Options:

```
--help, -h      Show Help
--users, -u     The comma separated list of user names
--groups, -g    The comma separated list of group names
--server, -s    <server> The registered OCM server
```

Examples:

```
cec unshare-theme Themel -u user1,user2
cec unshare-theme Themel -u user1,user2 -g group1,group2
cec unshare-theme Themel -u user1,user2 -s UAT
```

```
-----
cec create-site
-----
```

Usage: cec create-site <name>

Create Enterprise Site on OCM server. Specify the server with -s <server> or use the one specified in cec.properties file.

Options:

```
--help, -h          Show Help
--template, -t      <template> Template [required]
--repository, -r    <repository> Repository, required for enterprise
site
--localizationPolicy, -l <localizationPolicy> Localization policy
--defaultLanguage, -d <defaultLanguage> Default language, required for
enterprise site
--description, -p    <description> Site description
--sitePrefix, -x     <sitePrefix> Site Prefix
--update, -u        Keep the existing id for assets
--server, -s        <server> The registered OCM server
```

Examples:

```
cec create-site Site1 -t StandardTemplate
Creates a standard site
cec create-site Site1 -t Templatel -r Repository1 -l L10NPolicy1 -d en-US
Creates an enterprise site with localization policy L10NPolicy1
cec create-site Site1 -t Templatel -r Repository1 -d en-US
Creates an enterprise site and uses the localization policy in Templatel
cec create-site Site1 -t Templatel -r Repository1 -d en-US -s UAT
Creates an enterprise site on server UAT
cec create-site Site1 -t Templatel -u -r Repository1 -d en-US -s UAT
Creates an enterprise site on server UAT and keep the existing id for assets
```

```

-----
cec copy-site
-----
Usage: cec copy-site <name>

Copy Enterprise Site on OCM server. Specify the server with -s <server>
or use the one specified in cec.properties file. If the site uses more
than one repository, only the assets from the default repository will
be copied.

Options:
  --help, -h          Show Help
  --target, -t        Target site [required]
  --repository, -r    Repository, required for enterprise site
  --description, -d   Site description
  --sitePrefix, -x    Site Prefix
  --server, -s        The registered OCM server

Examples:
  cec copy-site Site1 -t Site1Copy           Copies
a standard site
  cec copy-site Site1 -t Site1Copy -r Repository1   Copies an
enterprise site
  cec copy-site Site1 -t Site1Copy -r Repository1 -x sitelc   Copies an
enterprise site and sets the site prefix to sitelc
-----

cec update-site
-----
Usage: cec update-site <name>

Update Enterprise Site on OCM server using the content from the
template. Specify the server with -s <server> or use the one specified
in cec.properties file.

Options:
  --help, -h          Show Help
  --template, -t      <template> Template [required]
  --excludecontenttemplate, -x  Exclude content template
  --server, -s        <server> The registered OCM server

Examples:
  cec update-site Site1 -t Templatel         Updates site Site1 using the
content from template Templatel
  cec update-site Site1 -t Templatel -x      Updates site Site1 using the
content from template Templatel excluding the "Content Template"
-----

cec transfer-site
-----
Usage: cec transfer-site <name>

Transfers a site from one OCM server to another. By default all assets
are transferred, optionally specify -p to transfer only published
assets. Specify the source server with -s <server> and the destination

```

server with `-d <destination>`. If the site contains assets from other repositories, optionally provide the repository mapping otherwise those assets will not be transferred.

Options:

<code>--help, -h</code>	Show Help
<code>--server, -s</code>	The registered OCM server the site is from
[required]	
<code>--destination, -d</code>	The registered OCM server to create or update the site [required]
<code>--repository, -r</code>	Repository, required for creating enterprise site
<code>--localizationPolicy, -l</code>	Localization policy, required for creating enterprise site
<code>--sitePrefix, -f</code>	Site prefix
<code>--publishedassets, -p</code>	The flag to indicate published assets only
<code>--repositorymappings, -m</code>	The repositories for assets from other repositories
<code>--excludecontent, -x</code>	Exclude content
<code>--excludecomponents, -e</code>	Exclude components
<code>--excludetheme, -c</code>	Exclude theme
<code>--excludetype, -t</code>	Exclude content types
<code>--includestaticfiles, -i</code>	Include site static files
<code>--suppressgovernance, -g</code>	Suppress site governance controls

Examples:

```
cec transfer-site Site1 -s DEV -d UAT -r Repository1 -l L10NPolicy1
Creates site Site1 on server UAT based on site Site1 on server DEV
cec transfer-site Site1 -s DEV -d UAT -r Repository1 -l L10NPolicy1 -p
Creates site Site1 on server UAT based on site Site1 on server DEV with published assets
cec transfer-site Site1 -s DEV -d UAT -r Repository1 -l L10NPolicy1 -x
Creates site Site1 on server UAT based on site Site1 on server DEV without content
cec transfer-site Site1 -s DEV -d UAT -r Repository1 -l L10NPolicy1 -e
Creates site Site1 on server UAT based on site Site1 on server DEV without transferring
components to server UAT
cec transfer-site Site1 -s DEV -d UAT -r Repository1 -l L10NPolicy1 -e -c
Creates site Site1 on server UAT based on site Site1 on server DEV without transferring
components and theme to server UAT
cec transfer-site Site1 -s DEV -d UAT -r Repository1 -l L10NPolicy1 -m
"Shared Images:Shared Images,Shared Video:Shared Video"
Creates site Site1 on server UAT based on site Site1 on server DEV and transfer the assets
from repository Shared Images and Shared Video
cec transfer-site Site1 -s DEV -d UAT -r Repository1 -l L10NPolicy1 -i
Creates site Site1 on server UAT based on site Site1 on server DEV with static files from
DEV
cec transfer-site Site1 -s DEV -d UAT
Updates site Site1 on server UAT based on site Site1 on
server DEV
cec transfer-site StandardSite1 -s DEV -d
```

UAT

Creates standard site on server UAT based on site StandardSite1 on server DEV

```
-----
cec transfer-site-content
-----
```

Usage: cec transfer-site-content <name>

Creates scripts to transfer Enterprise Site content from one OCM server to another. This command is used to transfer large number of content items and the items are transferred in batches. By default the scripts will not be executed by this command. By default all assets are transferred, optionally specify `-p` to transfer only published assets. Specify the source server with `-s <server>` and the destination server with `-d <destination>`. Optionally specify `-n` for the number of items in each batch, defaults to 500. If the site contains assets from other repositories, optionally provide the repository mapping otherwise those assets will not be transferred.

Options:

<code>--help, -h</code>	Show Help
<code>--server, -s</code>	The registered OCM server the site is from [required]
<code>--destination, -d</code>	The registered OCM server to transfer the content [required]
<code>--repository, -r</code>	The site repository [required]
<code>--publishedassets, -p</code>	The flag to indicate published assets only
<code>--addtositecollection, -l</code>	Add assets to the site collection
<code>--repositorymappings, -m</code>	The repositories for assets from other repositories
<code>--number, -n</code>	The number of items in each batch, defaults to 500
<code>--execute, -e</code>	Execute the scripts

Examples:

```
cec transfer-site-content Site1 -s DEV -d UAT -r
Repository1
Generate script Site1_downloadcontent and Site1_uploadcontent
cec transfer-site-content Site1 -s DEV -d UAT -r Repository1 -
e
Generate
script Site1_downloadcontent and Site1_uploadcontent and execute them
cec transfer-site-content Site1 -s DEV -d UAT -r Repository1 -n
200
Set batch size
to 200 items
cec transfer-site-content Site1 -s DEV -d UAT -r Repository1 -
p
Only the
published assets will be transferred
cec transfer-site-content Site1 -s DEV -d UAT -r Repository1 -
l
The assets
from the site repository will be added to site default collection on
destination server
cec transfer-site-content Site1 -s DEV -d UAT -r Repository1 -m
"Shared Images:Shared Images,Shared Video:Shared Video"
```



```
-----
cec validate-site
-----
```

```
Usage: cec validate-site <name>
```

Validates site <name> on OCM server before publish or view publishing failure. Specify the server with -s <server> or use the one specified in cec.properties file.

Options:

```
--help, -h      Show Help
--server, -s    <server> The registered OCM server
```

Examples:

```
cec validate-site Site1           Validate site Site1 on the server
specified in cec.properties file
cec validate-site Site1 -s UAT    Validate site Site1 on the registered
server UAT
```

```
-----
cec control-site
-----
```

```
Usage: cec control-site <action>
```

Perform <action> on site on OCM server. Specify the site with -s <site>. Specify the server with -r <server> or use the one specified in cec.properties file. The valid actions are

```
publish
unpublish
bring-online
take-offline
```

Options:

```
--help, -h      Show Help
--site, -s      <site> Site [required]
--usedcontentonly, -u Publish used content only
--compilesite, -c Compile site after publish
--staticonly, -t Only publish site static files
--fullpublish, -f Do a full publish
--server, -r    <server> The registered OCM server
```

Examples:

```
cec control-site publish -s Site1           Publish site Site1 on the
server specified in cec.properties file
cec control-site publish -s Site1 -u       Publish the site and all
assets added to the site's pages
cec control-site publish -s Site1 -c      Compile and publish site
Site1
cec control-site publish -s Site1 -t      Only publish the static
files of site Site1
cec control-site publish -s Site1 -f      Do a full publish of Site1
cec control-site publish -s Site1 -r UAT  Publish site Site1 on the
registered server UAT
```

```

    cec control-site unpublish -s Site1 -r UAT    Unpublish site Site1
on the registered server UAT
    cec control-site bring-online -s Site1 -r UAT  Bring site Site1
online on the registered server UAT
    cec control-site take-offline -s Site1 -r UAT  Take site Site1
offline on the registered server UAT

```

```

-----
cec share-site
-----

```

Usage: cec share-site <name>

Shares site with users and groups on OCM server and assign a role. Specify the server with -s <server> or use the one specified in cec.properties file. The valid roles are

```

manager
contributor
downloader
viewer

```

Options:

```

--help, -h    Show Help
--users, -u   The comma separated list of user names
--groups, -g  The comma separated list of group names
--role, -r    The role [manager | contributor | downloader | viewer]
to assign to the users or groups [required]
--server, -s  <server> The registered OCM server

```

Examples:

```

    cec share-site Site1 -u user1,user2 -r manager
Share site Site1 with user user1 and user2 and assign Manager role to
them
    cec share-site Site1 -u user1,user2 -g group1,group2 -r manager
Share site Site1 with user user1 and user2 and group group1 and group2
and assign Manager role to them
    cec share-site Site1 -u user1,user2 -r manager -s UAT
Share site Site1 with user user1 and user2 and assign Manager role to
them on the registered server UAT

```

```

-----
cec unshare-site
-----

```

Usage: cec unshare-site <name>

Deletes user or group access to a site on OCM server. Specify the server with -s <server> or use the one specified in cec.properties file.

Options:

```

--help, -h    Show Help
--users, -u   The comma separated list of user names
--groups, -g  The comma separated list of group names
--server, -s  <server> The registered OCM server

```

## Examples:

```
cec unshare-site Site1 -u user1,user2
cec unshare-site Site1 -u user1,user2 -g group1,group2
cec unshare-site Site1 -u user1,user2 -s UAT
```

```
-----
cec get-site-security
-----
```

Usage: cec get-site-security <name>

Gets site security on OCM server. Specify the server with `-s <server>` or use the one specified in `cec.properties` file.

## Options:

```
--help, -h      Show Help
--server, -s    <server> The registered OCM server
```

## Examples:

```
cec get-site-security Site1
cec get-site-security Site1 -s UAT
```

```
-----
cec set-site-security
-----
```

Usage: cec set-site-security <name>

Makes the site publicly available to anyone, restrict the site to registered users, or restrict the site to specific users. Specify the server with `-r <server>` or use the one specified in `cec.properties` file. Optionally specify `-a <access>` to set who can access the site. The valid group names are

```
Cloud users
Visitors
Service users
Specific users
```

## Options:

```
--help, -h      Show Help
--signin, -s    If require sign in to access site: yes | no [required]
--access, -a    The comma separated list of group names
--addusers, -u  The comma separated list of users to access the site
--deleteusers, -d The comma separated list of users to remove access from
the site
--server, -r    <server> The registered OCM server
```

## Examples:

```
cec set-site-security Site1 -s no
make the site publicly available to anyone
cec set-site-security Site1 -s no -r UAT
make the site publicly available to anyone on server UAT
cec set-site-security Site1 -s yes
Require everyone to sign in to access this site and any authenticated user
can access
cec set-site-security Site1 -s yes -a "Visitors,Service users"
```

```
Require everyone to sign in to access this site and all service
visitors and users can access
  cec set-site-security Site1 -s yes -a "Specific users" -u
user1,user2 Require everyone to sign in to access this site and only
user1 and user2 can access
  cec set-site-security Site1 -s yes -d
user1 Remove user1's access from the site
```

```
-----
cec index-site
-----
Usage: cec index-site <site>
```

Creates content item for each page with all text on the page. If the page index content item already exists for a page, updated it with latest text on the page. Specify `-c <contenttype>` to set the page index content type. Optionally specify `-p` to publish the page index items after creation or update. Specify the server with `-s <server>` or use the one specified in `cec.properties` file.

Options:

- `--help, -h` Show Help
- `--contenttype, -c <contenttype>` page index content type
- `--publish, -p` publish page index items
- `--server, -s <server>` The registered OCM server

Examples:

```
cec index-site Site1 -c PageIndex
cec index-site Site1 -c PageIndex -p
cec index-site Site1 -c PageIndex -s UAT
```

```
-----
cec create-site-map
-----
Usage: cec create-site-map <site>
```

Creates a site map for site on OCM server. Specify the server with `-s <server>` or use the one specified in `cec.properties` file. Optionally specify `-p` to upload the site map to OCM server after creation. Optionally specify `-c <changefreq>` to define how frequently the page is likely to change. Optionally specify `-t <toppagepriority>` as the priority for the top level pages. Also optionally specify `<file>` as the file name for the site map.

The valid values for `<changefreq>` are:

```
always
hourly
daily
weekly
monthly
yearly
never
auto
```

Options:

--help, -h	Show Help
--url, -u	<url> Site URL [required]
--changefreq, -c	How frequently the page is likely to change.
--file, -f	Name of the generated site map file
--languages, -l	<languages> The comma separated list of languages used to create the site map
--publish, -p	Upload the site map to OCM server after creation
--toppagepriority, -t	Priority for the top level pages, a decimal number between 0 and 1
--server, -s	<server> The registered OCM server
--newlink, -n	Generate new 19.3.3 detail page link
--noDefaultDetailPageLink, -o	Do not generate detail page link for items/content lists that use the default detail page

## Examples:

```
cec create-site-map Site1 -u http://www.example.com/site1
cec create-site-map Site1 -u http://www.example.com/site1 -s UAT
cec create-site-map Site1 -u http://www.example.com/site1 -t 0.9
cec create-site-map Site1 -u http://www.example.com/site1 -f sitemap.xml
cec create-site-map Site1 -u http://www.example.com/site1 -p
cec create-site-map Site1 -u http://www.example.com/site1 -c weekly -p
cec create-site-map Site1 -u http://www.example.com/site1 -l de-DE,it-IT
```

```
-----
cec create-rss-feed
-----
```

Usage: cec create-rss-feed <site>

Creates RSS feed for site <site> on OCM server. Specify the server with -s <server> or use the one specified in cec.properties file. Optionally specify -x <template> to specify the RSS template. Optionally specify -p to upload the RSS feed to OCM server after creation.

## Options:

--help, -h	Show Help
--url, -u	<url> Site URL [required]
--query, -q	Query for content items [required]
--limit, -l	The limit of the items returned from the query [required]
--orderby, -o	The order by for the query [required]
--language, -i	The language for the query
--template, -x	The RSS xml template
--javascript, -j	Javascript file that contains functions to process Mustache data
--title, -t	The RSS feed title
--description, -d	The RSS feed description
--ttl	How long the data will last in number of minutes
--file, -f	Name of the generated RSS feed file
--publish, -p	Upload the RSS feed to OCM server after creation
--server, -s	<server> The registered OCM server
--newlink, -n	Generate new 19.3.3 detail page link

Examples:

```
cec create-rss-feed Site1 -u http://www.example.com/site1 -q 'type eq
"BlogType"' -l 10 -o name:asc -t "Blog RSS"
cec create-rss-feed Site1 -u http://www.example.com/site1 -q 'type eq
"BlogType"' -l 10 -o name:asc -t "Blog RSS" -x ~/Files/RSSTemplate.xml
cec create-rss-feed Site1 -u http://www.example.com/site1 -q 'type eq
"BlogType"' -l 10 -o name:asc -t "Blog RSS" -x ~/Files/RSSTemplate.xml -
i fr-FR -f rssfrFR.xml
```

```
-----
cec create-asset-report
-----
```

Usage: cec create-asset-report <site>

Generates an asset usage report for site <site> on OCM server. Specify the server with -s <server> or use the one specified in cec.properties file. Optionally specify -o to save the report to a json file.

Options:

```
--help, -h      Show Help
--output, -o    Output the report to a JSON file
--server, -s    The registered OCM server
```

Examples:

```
cec create-asset-report Site1
cec create-asset-report Site1 -s UAT
cec create-asset-report Site1 -o                               The
report will be saved to Site1AssetUsage.json at the current local
location
cec create-asset-report Site1 -o ~/Documents                   The
report will be saved to ~/Documents/Site1AssetUsage.json
cec create-asset-report Site1 -o ~/Documents/Site1Report.json The
report will be saved to ~/Documents/Site1Report.json
```

```
-----
cec upload-static-site-files
-----
```

Usage: cec upload-static-site-files <path>

Uploads files to render statically from a site on OCM server. Specify the site <site> on the server. Specify the server with -r <server> or use the one specified in cec.properties file.

Options:

```
--help, -h      Show Help
--site, -s      The site on OCM server [required]
--server, -r    The registered OCM server
```

Examples:

```
cec upload-static-site-files ~/Documents/localBlog -s BlogSite
cec upload-static-site-files ~/Documents/localBlog -s BlogSite -r UAT
```

```
-----
cec download-static-site-files
-----
```

Usage: cec download-static-site-files <site>

Downloads the static files from a site on OCM server. Specify the server with `-s <server>` or use the one specified in `cec.properties` file. Optionally specify `-f <folder>` to save the files on the local system.

Options:

```
--help, -h    Show Help
--folder, -f  <folder> Local folder to save the static files
--server, -s  The registered OCM server
```

Examples:

```
cec download-static-site-files BlogSite
Download the files and save to local folder src/documents/BlogSite/static
cec download-static-site-files BlogSite -f ~/Documents/BlogSite/static
Download the files and save to local folder ~/Documents/BlogSite/static
cec download-static-site-files BlogSite -s UAT
```

```
-----
cec delete-static-site-files
-----
```

Usage: cec delete-static-site-files <site>

Deletes the static files from a site on OCM server. Specify the server with `-s <server>` or use the one specified in `cec.properties` file.

Options:

```
--help, -h    Show Help
--server, -s  The registered OCM server
```

Examples:

```
cec delete-static-site-files BlogSite
cec delete-static-site-files BlogSite -s UAT
```

```
-----
cec refresh-prerender-cache
-----
```

Usage: cec refresh-prerender-cache <site>

Refreshes pre-render cache for a site on OCM server. Specify the server with `-s <server>` or use the one specified in `cec.properties` file.

Options:

```
--help, -h    Show Help
--server, -s  The registered OCM server
```

Examples:

```
cec refresh-prerender-cache BlogSite
cec refresh-prerender-cache BlogSite -s UAT
```

```
-----
cec migrate-site
-----
Usage: cec migrate-site <site>
```

Migrates a site from OCI IC server to EC server. Specify the IC server with `-s <server>` and the EC server with `-d <destination>`.

## Options:

```
--help, -h          Show Help
--server, -s        The registered IC server the site is from
--destination, -d  The registered EC server to create the site
[required]
--repository, -r   Repository [required]
--template, -t    The site template
--name, -n        Site name
--description, -p Site description
--sitePrefix, -x  Site Prefix
```

## Examples:

```
cec migrate-site Site1 -s ICServer -d ECServer -r
Repol           Migrates site Site1 from ICServer to
ECServer
cec migrate-site Site1 -s ICServer -d ECServer -r Repol -n
newSite        Migrates site Site1 from ICServer to ECServer and
rename to newSite
cec migrate-site Site1 -d ECServer -t ~/Documents/Site1Template.zip -
r Repol Migrates site Site1 to ECServer with template
Site1Template.zip from IC server
```

```
-----
cec download-content
-----
```

Usage: cec download-content

Downloads content from OCM server. By default all assets are downloaded, optionally specify `-p` to download only published assets. Specify the server with `-s <server>` or use the one specified in `cec.properties` file.

## Options:

```
--help, -h          Show Help
--publishedassets, -p The flag to indicate published assets only
--collection, -c    Collection name
--repository, -r    Repository name, required when <collection> is
specified
--query, -q         Query to fetch the assets
--assets, -a        The comma separated list of asset GUIDS
--assetsfile, -f    The file with an array of asset GUIDS
--name, -n          The name for this download, default to the
channel or repository name
--server, -s        The registered OCM server
```

## Examples:

```
cec download-content
Site1Channel
Download all assets in channel Site1Channel and save to local folder
src/content/Site1Channel
cec download-content Site1Channel -n
Site1Assets           Download all
```



```

assets in channel Site1Channel and save to local folder src/content/
Site1Assets
    cec download-content Site1Channel -p
                                                    Download published
assets in channel Site1Channel
    cec download-content Site1Channel -s
UAT
                                                    Download all assets
in channel Site1Channel on server UAT
    cec download-content Site1Channel -q 'fields.category eq
"RECIPE"'
                                                    Download assets from the channel
Site1Channel, matching the query, plus any dependencies
    cec download-content Site1Channel -r Repol -c
Collection1
                                                    Download assets from the
repository Repol, collection Collection1 and channel Site1Channel
    cec download-content Site1Channel -r Repol -c Collection1 -q
'fields.category eq "RECIPE"'
                                                    Download assets from repository Repol,
collection Collection1 and channel Site1Channel, matching the query, plus
any dependencies
    cec download-content -a
GUID1,GUID2
                                                    Download
asset GUID1 and GUID2 and all their dependencies
    cec download-content -r
Repol
                                                    Download
assets from the repository Repol

```

Please specify the channel, repository, query or assets

```

-----
cec upload-content
-----

```

Usage: cec upload-content <name>

Uploads local content from channel <name>, template <name> or local file <name> to repository <repository> on OCM server. Specify -c <channel> to add the template content to channel. Optionally specify -l <collection> to add the content to collection. Specify the server with -s <server> or use the one specified in cec.properties file.

Options:

```

--help, -h          Show Help
--repository, -r <repository> The repository for the types and items
[required]
--template, -t      Flag to indicate the content is from template
--file, -f          Flag to indicate the content is from file
--channel, -c       <channel> The channel to add the content
--collection, -l   <collection> The collection to add the content
--server, -s        <server> The registered OCM server
--update, -u        Update any existing content instead of creating new items
--types, -p         Upload content types and taxonomies only

```

Examples:

```

    cec upload-content Site1Channel -r Repol
content to repository Repol, creating new items, and add to channel
Site1Channel
    cec upload-content Site1Channel -r Repol -u
                                                    Upload
                                                    Upload

```

content to repository Repol, updating existing content to create new versions, and add to channel SiteChannel

```
cec upload-content SiteChannel -r Repol -l SiteCollection
```

Upload content to repository Repol and add to collection SiteCollection and channel SiteChannel

```
cec upload-content SiteChannel -r Repol -p
```

Upload content types from content SiteChannel to the server

```
cec upload-content SiteChannel -r Repol -s UAT
```

Upload content to repository Repol on server UAT and add to channel SiteChannel

```
cec upload-content Templatel -t -r Repol -c channel1
```

Upload content from template Templatel to repository Repol and add to channel channel1

```
cec upload-content ~/Downloads/content.zip -f -r Repol -c channel1
```

Upload content from file ~/Downloads/content.zip to repository Repol and add to channel channel1

```
-----
cec control-content
-----
```

Usage: cec control-content <action>

Performs action <action> on channel items on OCM server. Specify the channel with -c <channel>. Specify the server with -s <server> or use the one specified in cec.properties file. The valid actions are

```
publish
unpublish
add
remove
```

Options:

```
--help, -h          Show Help
--channel, -c       Channel
--repository, -r    Repository, required when <action> is add
--collection, -l    Collection
--query, -q         Query to fetch the assets
--assets, -a        The comma separated list of asset GUIDS
--server, -s        The registered OCM server
--date, -d          Date to publish items
--name, -n          Name of the scheduled publishing job to create
```

Examples:

```
cec control-content publish -c
Channel1
```

Publish all items in channel Channel1 on the server specified in cec.properties file

```
cec control-content publish -c Channel1 -a
GUID1,GUID2
```

Publish asset GUID1 and GUID2 in channel Channel1

```
cec control-content publish -c Channel1 -s
```

UAT

Publish all items in channel Channel1 on the registered server

UAT

```

    cec control-content unpublish -c Channel1 -s
UAT
    Unpublish all items in channel Channel1 on the registered server UAT
    cec control-content add -c Channel1 -r Repol -s
UAT
    Add all items in repository Repol to channel Channel1 on the registered
server UAT
    cec control-content add -c Channel1 -r Repol -q 'type eq "BlogType"' -s
UAT
    Add all items in
repository Repol, matching the query to channel Channel1 on the registered
server UAT
    cec control-content add -c Channel1 -r Repol -q 'channels co
"CHANNELF43508F995FE582EC219EFEF03076128932B9A3F1DF6"' -s UAT
    Add all items
in repository Repol and Channel2 to channel Channel1 on the registered
server UAT
    cec control-content add -c Channel1 -r Repol -a GUID1,GUID2 -s
UAT
    Add asset GUID1
and GUID2 in repository Repol to channel Channel1
    cec control-content remove -c Channel1 -s
UAT
    Remove all items in channel Channel1 on the registered server UAT
    cec control-content add -l Collection1 -r Repol -s
UAT
    Add
all items in repository Repol to collection Collection1 on the registered
server UAT
    cec control-content remove -l Collection -s
UAT
    Remove all items in collection Collection1 on the registered server UAT
    cec control-content publish -c C1 -r R1 -s UAT -d "2021/9/21 0:30:00 PST" -
n Name
    Create a publishing job
called Name to publish all items in channel C1 on the specified date.
Requires server version: 21.2.1

```

```

-----
cec transfer-content
-----

```

Usage: cec transfer-content <repository>

Creates scripts to transfer content from one OCM server to another. This command is used to transfer large number of content items and the items are transferred in batches. By default the scripts will not be executed by this command. By default all assets are transferred, optionally specify `-p` to transfer only published assets. Specify the source server with `-s <server>` and the destination server with `-d <destination>`. Optionally specify `-n` for the number of items in each batch, defaults to 200.

Options:

<code>--help, -h</code>	Show Help
<code>--server, -s</code>	The registered OCM server the content is from
[required]	
<code>--destination, -d</code>	The registered OCM server to transfer the content
[required]	
<code>--channel, -c</code>	The channel
<code>--publishedassets, -p</code>	The flag to indicate published assets only
<code>--number, -n</code>	The number of items in each batch, defaults to 200

```
--execute, -e          Execute the scripts
```

Examples:

```
cec transfer-content Repository1 -s DEV -d UAT
Generate script Repository1_downloadcontent and
Repository1_uploadcontent
cec transfer-content Repository1 -s DEV -d UAT -e
Generate script Repository1_downloadcontent and
Repository1_uploadcontent and execute them
cec transfer-content Repository1 -s DEV -d UAT -n 1000          Set
the number of items in each batch to 1000
cec transfer-content Repository1 -s DEV -d UAT -c Channel1
Transfer the items added to channel Channel1 in repository Repository1
cec transfer-content Repository1 -s DEV -d UAT -c Channel1 -p
Transfer the items published to channel Channel1 in repository
Repository1
```

```
-----
cec list-assets
-----
```

Usage: cec list-assets

Lists assets on OCM server. Optionally specify `-c <channel>`, `-r <repository>`, `-l <collection>` or `-q <query>` to query assets. Specify the server with `-s <server>` or use the one specified in `cec.properties` file.

Options:

```
--help, -h          Show Help
--channel, -c       Channel name
--collection, -l    Collection name
--repository, -r    Repository name, required when <collection> is
specified
--query, -q         Query to fetch the assets
--server, -s        The registered OCM server
```

Examples:

```
cec list-assets          List all assets
cec list-assets -s UAT   List all assets on
registered server UAT
cec list-assets -r Rep01 List all assets
from repository Rep01
cec list-assets -c Channel1 List all assets
from channel Channel1
cec list-assets -r Rep01 -l Collection1 List all assets
from collection Collection1 and repository Rep01
cec list-assets -q 'fields.category eq "RECIPE"' List all assets
matching the query
```

```
-----
cec create-digital-asset
-----
```

Usage: cec create-digital-asset

Creates digital asset on OCM server. Specify the server with `-s`

<server> or use the one specified in cec.properties file. Specify the asset attributes in JSON file, e.g.

```
{
  "imagetitle": "Logo",
  "copyright": "Copyright © 1995, 2021, Company and/or its affiliates"
}
```

Options:

```
--help, -h          Show Help
--from, -f          The digital asset source file [required]
--type, -t          The digital asset type [required]
--repository, -r    The repository to add the asset [required]
--slug, -l          The slug for the asset when create a single asset
--attributes, -a    The JSON file of asset attributes
--server, -s        The registered OCM server
```

Examples:

```
cec create-digital-asset -f ~/Documents/logo.jpg -t Image -r
Repol                               Create asset of type
Image
cec create-digital-asset -f ~/Documents/logo.jpg -t Image -r Repol -l
company-logo                         Create asset of type Image and
set slug to company-logo
cec create-digital-asset -f "~/Documents/demo.mp4,~/Documents/demo2.mp4" -
t Video -r Repol                     Create two assets of type Video
cec create-digital-asset -f ~/Documents/logo.jpg -t MyImage -r Repol -a ~/
Documents/logoattrs.json            Create asset of type MyImage with
attributes
cec create-digital-asset -f ~/Documents/logo.jpg -t MyImage -r Repol -l
company-logo -a ~/Documents/logoattrs.json Create asset of type MyImage
with slug and attributes
cec create-digital-asset -f ~/Documents/images -t Image -r
Repol                                 Create assets for
all images files from folder ~/Documents/images
Missing required arguments: from, type, repository
```

```
-----
cec update-digital-asset
-----
Usage: cec update-digital-asset <id>
```

Uploads a new version or updates attributes for a digital asset on OCM server. Specify the server with -s <server> or use the one specified in cec.properties file. Specify the asset attributes in JSON file, e.g.

```
{
  "imagetitle": "Logo2",
  "copyright": "Copyright © 1995, 2021, Company and/or its affiliates"
}
```

Options:

```
--help, -h          Show Help
--from, -f          The digital asset source file for the new version
--slug, -l          The slug for the asset
```

```
--attributes, -a The JSON file of asset attributes
--server, -s     The registered OCM server
```

## Examples:

```
cec update-digital-asset CORED129ACD36FCD42B1B38D22EEA5065F38 -l
company-logo
Update asset slug
cec update-digital-asset CORED129ACD36FCD42B1B38D22EEA5065F38 -f ~/
Documents/logo2.jpg
Upload a new version
cec update-digital-asset CORED129ACD36FCD42B1B38D22EEA5065F38 -f ~/
Documents/logo2.jpg -l company-logo -a ~/Documents/logoattrs2.json
Upload a new version and update slug and attributes
```

```
-----
cec copy-assets
-----
```

```
Usage: cec copy-assets <repository>
```

Copies assets to another repository on OCM server. Specify the server with `-s <server>` or use the one specified in `cec.properties` file.

## Options:

```
--help, -h           Show Help
--collection, -l     Collection name
--channel, -c        Channel name
--query, -q          Query to fetch the assets
--assets, -a         The comma separated list of asset GUIDS
--target, -t         The target repository [required]
--server, -s         The registered OCM server
```

## Examples:

```
cec copy-assets Repo1 -t
Repo2                               Copy all assets in
repository Repo1 to Repo2
cec copy-assets Repo1 -t Repo2 -s
UAT                                   Copy all assets in
repository Repo1 to Repo2 on server UAT
cec copy-assets Repo1 -a GUID1,GUID2 -t
Repo2                               Copy asset GUID1 and GUID2 and all
their dependencies in Repo1 to Repo2
cec copy-assets Repo1 -q 'fields.category eq "RECIPE"' -t
Repo2                               Copy assets from repository Repo1, matching the
query, plus any dependencies to Repo2
cec copy-assets Repo1 -c Channel1 -t
Repo2                               Copy assets from the repository
Repo1 and channel Channel1 to Repo2
cec copy-assets Repo1 -l Collection1 -t
Repo2                               Copy assets from the repository
Repo1 and collection Collection1 to Repo2
cec copy-assets Repo1 -c Channel1 -q 'fields.category eq "RECIPE"' -t
Repo2 Copy assets from repository Repo1, channel Channel1, matching
the query, plus any dependencies to Repo2
```

```
-----
```

```
cec create-asset-usage-report
```

```
-----
```

```
Usage: cec create-asset-usage-report <assets>
```

Generates an asset usage report for assets on OCM server. Optionally specify `-o` to save the report to a json file. Specify the server with `-s <server>` or use the one specified in `cec.properties` file.

Options:

```
--help, -h      Show Help
--output, -o    Output the report to a JSON file
--server, -s    The registered OCM server
```

Examples:

```
cec create-asset-usage-report GUID1
cec create-asset-usage-report GUID1 -s UAT
cec create-asset-usage-report GUID1 -o                               The report
will be saved to GUID1AssetUsage.json
cec create-asset-usage-report GUID1,GUID2 -o                       The report
will be saved to GUID1_GUID2AssetUsage.json
cec create-asset-usage-report GUID1,GUID2 -o ItemReport.json      The report
will be saved to ItemReport.json
```

```
-----
```

```
cec create-repository
```

```
-----
```

```
Usage: cec create-repository <name>
```

Creates a repository on OCM server. Specify the server with `-s <server>` or use the one specified in `cec.properties` file. Optionally specify `-d <description>` to set the description. Optionally specify `-t <contenttypes>` to set the content types. Optionally specify `-c <channels>` to set the publishing channels. Optionally specify `-l <defaultlanguage>` to set the default language. Optionally specify `-p <type>` to set the repository type. The valid repository types are

```
asset
business
```

Options:

```
--help, -h      Show Help
--description, -d  The description for the repository
--type, -p       The repository type [asset | business]. Defaults to
asset
--contenttypes, -t  The comma separated list of content types for the
repository
--channels, -c     The comma separated list of publishing channels to
use in this repository
--defaultlanguage, -l  The default language
--server, -s      The registered OCM server
```

Examples:

```
cec create-repository Repol
cec create-repository BusinessRepo -p business
```

```
cec create-repository Repol -d "Blog Repository" -t
BlogType,AuthorType -c channel1,channel2 -l en-US -s UAT
```

```
-----
cec control-repository
-----
```

Usage: cec control-repository <action>

Performs action <action> on repositories on OCM server. Specify the server with -s <server> or use the one specified in cec.properties file. The valid actions are

```
add-type
remove-type
add-channel
remove-channel
add-taxonomy
remove-taxonomy
```

Options:

```
--help, -h          Show Help
--repository, -r    The comma separated list of content repositories
[required]
--contenttypes, -t  The comma separated list of content types
--channels, -c      The comma separated list of publishing channels
--taxonomies, -x    The comma separated list of promoted taxonomies
--server, -s        The registered OCM server
```

Examples:

```
cec control-repository add-type -r Repol -t Blog,Author
cec control-repository add-type -r Repol,Repo2 -t Blog,Author
cec control-repository add-type -r Repol -t Blog,Author -s UAT
cec control-repository remove-type -r Repol -t Blog,Author
cec control-repository add-channel -r Repol -c channel1,channel2
cec control-repository remove-channel -r Repol -c channel1,channel2
cec control-repository add-taxonomy -r Repol -x Taxonomy1,Taxonomy2
cec control-repository remove-taxonomy -r Repol -x Taxonomy1,Taxonomy2
```

```
-----
cec share-repository
-----
```

Usage: cec share-repository <name>

Shares repository with users and groups on OCM server and assign a role. Specify the server with -s <server> or use the one specified in cec.properties file. Optionally specify -t to also share the content types in the repository with the users. Optionally specify -y <typerole> to share the types with different role. The valid roles for a repository are

```
manager
contributor
viewer
```



The valid roles for a type are

```
manager
contributor
```

Options:

```
--help, -h      Show Help
--users, -u     The comma separated list of user names
--groups, -g    The comma separated list of group names
--role, -r      The role [manager | contributor | viewer] to assign to the
users or groups [required]
--types, -t     Share types in the repository
--typerole, -y  The role [manager | contributor] to assign to the users or
groups for types
--server, -s    <server> The registered OCM server
```

Examples:

```
cec share-repository Repol -u user1,user2 -r manager
Share repository Repol with user user1 and user2 and assign Manager role to
them
cec share-repository Repol -u user1,user2 -g group1,group2 -r manager
Share repository Repol with user user1 and user2 and group group1 and group2
and assign Manager role to them
cec share-repository Repol -u user1,user2 -r manager -s UAT
Share repository Repol with user user1 and user2 and assign Manager role to
them on the registered server UAT
cec share-repository Repol -u user1,user2 -r manager -t
Share repository Repol and all the types in Repol with user user1 and user2
and assign Manager role to them
cec share-repository Repol -u user1,user2 -r manager -t -y contributor
Share repository Repol with user user1 and user2 and assign Manager role to
them, share all types in Repol with user user1 and user2 and assign
Contributor role to them
```

```
-----
cec unshare-repository
-----
```

Usage: cec unshare-repository <name>

Deletes user or group access to a repository on OCM server. Specify the server with `-s <server>` or use the one specified in `cec.properties` file. Optionally specify `-t` to also delete the user or group access to the content types in the repository.

Options:

```
--help, -h      Show Help
--users, -u     The comma separated list of user names
--groups, -g    The comma separated list of group names
--types, -t     Remove the user or group access to types in the repository
--server, -s    <server> The registered OCM server
```

Examples:

```
cec unshare-repository Repol -u user1,user2
cec unshare-repository Repol -u user1,user2 -g group1,group2
```

```
cec unshare-repository Repol -u user1,user2 -s UAT
cec unshare-repository Repol -u user1,user2 -t
```

```
-----
cec create-channel
-----
```

Usage: cec create-channel <name>

Creates a channel on OCM server. Specify the server with `-s <server>` or use the one specified in `cec.properties` file. Optionally specify `-t <type>` to set the channel type [public | secure], defaults to public. Optionally specify `-p <publishpolicy>` to set the publish policy [anythingPublished | onlyApproved], defaults to anythingPublished. Optionally specify `-l <localizationpolicy>` to set the localization policy.

Options:

<code>--help, -h</code>	Show Help
<code>--description, -d</code>	The description for the channel
<code>--type, -t</code>	The channel type [public   secure]
<code>--publishpolicy, -p</code>	The publish policy [anythingPublished   onlyApproved]
<code>--localizationpolicy, -l</code>	The localization policy for the channel
<code>--server, -s</code>	The registered OCM server

Examples:

```
cec create-channel channel1                                Create public
channel channel1 and everything can be published
cec create-channel channel1 -s UAT                        On registered
server UAT, reate public channel channel1 and everything can be
published
cec create-channel channel1 -l en-fr                       Create public
channel channel1 with localization policy en-fr and everything can be
published
cec create-channel channel1 -t secure -p onlyApproved    Create secure
channel channel1 and only approved items can be published
```

```
-----
cec share-channel
-----
```

Usage: cec share-channel <name>

Shares channel with users and groups on OCM server and assign a role. Specify the server with `-s <server>` or use the one specified in `cec.properties` file. The valid roles are

```
manager
contributor
viewer
```

Options:

<code>--help, -h</code>	Show Help
<code>--users, -u</code>	The comma separated list of user names
<code>--groups, -g</code>	The comma separated list of group names

```

--role, -r    The role [manager | contributor | viewer] to assign to the
users or groups [required]
--server, -s <server> The registered OCM server

```

## Examples:

```

cec share-channel Channel1 -u user1,user2 -r manager
Share channel Channel1 with user user1 and user2 and assign Manager role to
them
cec share-channel Channel1 -u user1,user2 -g group1,group2 -r manager
Share channel Channel1 with user user1 and user2 and group group1 and group2
and assign Manager role to them
cec share-channel Channel1 -u user1,user2 -r manager -s UAT
Share channel Channel1 with user user1 and user2 and assign Manager role to
them on the registered server UAT

```

```

-----
cec unshare-channel
-----

```

Usage: cec unshare-channel <name>

Deletes user or group access to a channel on OCM server. Specify the server with -s <server> or use the one specified in cec.properties file.

## Options:

```

--help, -h    Show Help
--users, -u   The comma separated list of user names
--groups, -g  The comma separated list of group names
--server, -s  <server> The registered OCM server

```

## Examples:

```

cec unshare-channel Channel1 -u user1,user2
cec unshare-channel Channel1 -u user1,user2 -g group1,group2
cec unshare-channel Channel1 -u user1,user2 -s UAT

```

```

-----
cec create-localization-policy
-----

```

Usage: cec create-localization-policy <name>

Creates a localization policy on OCM server. Specify the server with -s <server> or use the one specified in cec.properties file. Specify -r <requiredlanguages> to set the required languages. Specify -l <defaultlanguage> to set the default language. Optionally specify -o <optionallanguages> to set the optional languages. Optionally specify -d <description> to set the description.

## Options:

```

--help, -h    Show Help
--requiredlanguages, -r The comma separated list of required languages
for the localization policy [required]
--defaultlanguage, -l  The default language [required]
--optionallanguages, -o The comma separated list of optional languages
for the localization policy
--description, -d     The description for the repository
--server, -s         The registered OCM server

```

Examples:

```
cec create-localization-policy en-us -r en-US -l en-US
cec create-localization-policy en-fr -r en-US,fr-FR -l en-US
cec create-localization-policy multi -r en-US,fr-FR -l en-US -o zh-CN
-d "Policy for Blog" -s UAT
```

```
-----
cec list-server-content-types
-----
```

Usage: cec list-server-content-types

Lists all content types from server.

Options:

```
--help, -h    Show Help
--server, -s <server> The registered OCM server
```

Examples:

```
cec list-server-content-types
cec list-server-content-types -s UAT
```

```
-----
cec share-type
-----
```

Usage: cec share-type <name>

Shares type with users and groups on OCM server and assign a role. Specify the server with -s <server> or use the one specified in cec.properties file. The valid roles are

```
manager
contributor
```

Options:

```
--help, -h    Show Help
--users, -u   The comma separated list of user names
--groups, -g  The comma separated list of group names
--role, -r    The role [manager | contributor] to assign to the users
or groups [required]
--server, -s <server> The registered OCM server
```

Examples:

```
cec share-type BlogType -u user1,user2 -r manager
Share type BlogType with user user1 and user2 and assign Manager role
to them
cec share-type BlogType -u user1,user2 -g group1,group2 -r manager
Share type BlogType with user user1 and user2 and group group1 and
group2 and assign Manager role to them
cec share-type BlogType -u user1,user2 -r manager -s UAT
Share type BlogType with user user1 and user2 and assign Manager role
to them on the registered server UAT
```

```
cec unshare-type  
-----  
Usage: cec unshare-type <name>
```

Deletes user or group access to a type on OCM server. Specify the server with `-s <server>` or use the one specified in `cec.properties` file.

Options:

- `--help, -h` Show Help
- `--users, -u` The comma separated list of user names
- `--groups, -g` The comma separated list of group names
- `--server, -s <server>` The registered OCM server

Examples:

```
cec unshare-type BlogType -u user1,user2  
cec unshare-type BlogType -u user1,user2 -g group1,group2  
cec unshare-type BlogType -u user1,user2 -s UAT
```

```
-----  
cec download-type  
-----  
Usage: cec download-type <name>
```

Downloads types from OCM server. The content field editors and forms for the types will also be downloaded. Specify the server with `-s <server>` or use the one specified in `cec.properties` file.

Options:

- `--help, -h` Show Help
- `--server, -s <server>` The registered OCM server

Examples:

```
cec download-type BlogType          Download content type BlogType and  
save to local folder src/types/BlogType  
cec download-type BlogType,BlogAuthor Download content type BlogType and  
BlogAuthor and save to local folder  
cec download-type BlogType -s UAT
```

```
-----  
cec upload-type  
-----  
Usage: cec upload-type <name>
```

Uploads types to OCM server. The content field editors and forms for the types will also be uploaded. Specify the server with `-s <server>` or use the one specified in `cec.properties` file.

Options:

- `--help, -h` Show Help
- `--server, -s <server>` The registered OCM server

Examples:

```
cec upload-type BlogType  
cec upload-type BlogType -s UAT  
cec upload-type BlogAuthor,BlogType Place the referenced types first
```

```
-----
cec update-type
-----
```

Usage: cec update-type <action>

Performs action <action> on a type in a local template or on OCM server. Specify the server with -s <server> or use the one specified in cec.properties file. The valid actions are

```
add-content-form
remove-content-form
```

Options:

```
--help, -h           Show Help
--objectname, -o     the content form [required]
--contenttype, -c    the content type [required]
--template, -t       The template the content type is from
--contenttemplate, -t Flag to indicate the template is a content
template
--server, -s         The registered OCM server
```

Examples:

```
cec update-type add-content-form -o form1 -c BlogPost -t
BlogTemplate           Associate content form form1 with content
type BlogPost from local template at src/templates/BlogTemplate
cec update-type add-content-form -o form1 -c BlogPost -t
BlogTemplateContent -n Associate content form form1 with content
type BlogPost from local template at src/content/BlogTemplateContent
cec update-type add-content-form -o form1 -c BlogPost -s
UAT                   Associate content form form1 with content
type BlogPost on the registered server UAT
cec update-type add-content-form -o form1 -c BlogPost -
s                     Associate content form form1 with content
type BlogPost on the server specified in cec.properties file
cec update-type remove-content-form -o form1 -c BlogPost -t
BlogTemplate           Change not to use form1 when create or edit
items of type BlogPost from local template at src/templates/BlogTemplate
cec update-type remove-content-form -o form1 -c BlogPost -t
BlogTemplateContent -n Change not to use form1 when create or edit
items of type BlogPost from local template at src/content/
BlogTemplateContent
cec update-type remove-content-form -o form1 -c BlogPost -s
UAT                   Change not to use form1 when create or edit
items of type BlogPost on the registered server UAT
```

```
-----
cec download-recommendation
-----
```

Usage: cec download-recommendation <name>

Downloads a recommendation from the Content Management server. Specify the server with -s <server> or use the one specified in cec.properties file. Optionally specify repository with -r <repository>. Optionally

specify `-p` to download the published version.

Options:

```
--help, -h          Show Help
--repository, -r    The repository
--published, -p     The flag to indicate published version
--channel, -c       Channel name, required when <published> is set
--server, -s        <server> The registered OCM server
```

Examples:

```
cec download-recommendation Recommendation1 Downloads
Recommendation1
cec download-recommendation Recommendation1 -p -c Channel1 Downloads
Recommendation1 published to channel Channel1
cec download-recommendation Recommendation1 -s UAT
cec download-recommendation Recommendation1 -r Repol
```

```
-----
cec upload-recommendation
-----
```

Usage: `cec upload-recommendation <name>`

Uploads a recommendation to repository `<repository>` on OCM server. Specify the server with `-s <server>` or use the one specified in `cec.properties` file.

Options:

```
--help, -h          Show Help
--repository, -r    The repository [required]
--server, -s        <server> The registered OCM server
```

Examples:

```
cec upload-recommendation Recommendation1 -r Repol
cec upload-recommendation Recommendation1 -r Repol -s UAT
```

```
-----
cec create-contentlayout
-----
```

Usage: `cec create-contentlayout <name>`

Creates a content layout based on a content type from a local template or from OCM server. By default, an "overview" content layout is created. Optionally specify `-s <style>` to create in a different style.

Valid values for `<style>` are:

```
detail
overview
```

Options:

```
--help, -h          Show Help
--contenttype, -c   <contenttype> Content layout is based on
[required]
--template, -t      <template> Content type is from
--server, -r        The registered OCM server
--style, -s         <style> Content layout style: detail | overview
```

```
--addcustomsettings, -a Add support for custom settings when used in Sites
```

Examples:

```
cec create-contentlayout Blog-Post-Overview-Layout -c Blog-Post -t
BlogTemplate
cec create-contentlayout Blog-Post-Detail-Layout -c Blog-Post -t
BlogTemplate -s detail
cec create-contentlayout Blog-Post-Overview-Layout -c Blog-Post -t
BlogTemplate -a Add custom settings when used in Sites
cec create-contentlayout Blog-Post-Overview-Layout -c Blog-Post -
r Use content type Blog-Post from the server
specified in cec.properties file
cec create-contentlayout Blog-Post-Overview-Layout -c Blog-Post -r
UAT -s detail Use content type Blog-Post from the registered
server UAT
```

```
-----
cec add-contentlayout-mapping
-----
```

```
Usage: cec add-contentlayout-mapping <contentlayout>
```

Creates content type and content layout mapping. By default, the mapping is set for "Default". Optionally specify `-s <layoutstyle>` to name the mapping. By default, the mapping is set for desktop. Optionally specify `-m` to set the mapping for mobile.

Options:

```
--help, -h Show Help
--contenttype, -c <contenttype> Content layout is based on
[required]
--template, -t <template> The mapping is for
--layoutstyle, -s <style> Content layout style
--mobile, -m mobile mapping
--server, -r <server> The registered OCM server
```

Examples:

```
cec add-contentlayout-mapping Blog-Post-Detail-Layout -c Blog-Post -t
BlogTemplate
cec add-contentlayout-mapping Blog-Post-Detail-Layout -c Blog-Post -t
BlogTemplate -m
cec add-contentlayout-mapping Blog-Post-Detail-Layout -c Blog-Post -t
BlogTemplate -s Details
cec add-contentlayout-mapping Blog-Post-Overview-Layout -c Blog-Post -
t BlogTemplate -s "Content List Default"
cec add-contentlayout-mapping Blog-Post-Overview-Layout -c Blog-Post -
t BlogTemplate -s Overview
cec add-contentlayout-mapping Blog-Post-Overview-Layout -c Blog-Post -
r UAT Set "Content Item Default" to
Blog-Post-Overview-Layout for content type Blog-Post on server UAT
```

```
-----
cec remove-contentlayout-mapping
-----
```

```
Usage: cec remove-contentlayout-mapping <contentlayout>
```



Removes a content layout mapping. By default, all mappings for the content layout are removed. Optionally specify `-s <layoutstyle>` to name the mapping and `-m` to indicate the mobile mapping.

## Options:

```
--help, -h          Show Help
--contenttype, -c   Content type, required when <server> is specified
--template, -t      <template> The mapping is from
--layoutstyle, -s   <style> Content layout style
--mobile, -m        mobile mapping
--server, -r        <server> The registered OCM server
```

## Examples:

```
cec remove-contentlayout-mapping Blog-Post-Detail-Layout -t BlogTemplate
cec remove-contentlayout-mapping Blog-Post-Detail-Layout -t BlogTemplate -m
cec remove-contentlayout-mapping Blog-Post-Detail-Layout -c Blog-Post -r
UAT
cec remove-contentlayout-mapping Blog-Post-Detail-Layout -c Blog-Post -s
Details -r UAT
```

```
-----
cec add-field-editor
-----
```

Usage: cec add-field-editor <name>

Adds a field editor to a field in a content type.

## Options:

```
--help, -h          Show Help
--template, -t      The template the content type is from [required]
--contenttype, -c   The content type [required]
--field, -f         The field the field editor is for [required]
--contenttemplate, -n Flag to indicate the template is a content template
```

## Examples:

```
cec add-field-editor editor1 -t BlogTemplate -c BlogPost -f
summary Use editor1 as the appearance for field summary in
content type BlogPost from local template at src/templates/BlogTemplate
cec add-field-editor editor1 -t BlogTemplateContent -n -c BlogPost -f
summary Use editor1 as the appearance for field summary in content type
BlogPost from local template at src/content/BlogTemplateContent
```

```
-----
cec remove-field-editor
-----
```

Usage: cec remove-field-editor <name>

Removes a field editor from a field in a content type.

## Options:

```
--help, -h          Show Help
--template, -t      The template the content type is from [required]
--contenttype, -c   The content type [required]
--field, -f         The field the field editor is for [required]
```

```
--contenttemplate, -n Flag to indicate the template is a content
template
```

Examples:

```
cec remove-field-editor editor1 -t BlogTemplate -c BlogPost -f
summary          Remove editor1 as the appearance for field summary
in content type BlogPost from local template at src/templates/
BlogTemplate
cec remove-field-editor editor1 -t BlogTemplateContent -n -c BlogPost
-f summary       Remove editor1 as the appearance for field summary in
content type BlogPost from local template at src/content/
BlogTemplateContent
```

```
-----
cec migrate-content
-----
```

Usage: cec migrate-content <name>

Migrates content from OCI IC server to EC server. Specify the IC server with -s <server> and the EC server with -d <destination>.

Options:

```
--help, -h          Show Help
--server, -s        The registered IC server the content is from
[required]
--destination, -d   The registered EC server to upload the content
[required]
--repository, -r    The repository for the types and items [required]
--channel, -c       The channel to add the content
--collection, -l    The collection to add the content
```

Examples:

```
cec migrate-content collection1 -s ICServer -d ECServer -r
Repol          Migrates content from collection
collection1 on ICServer to repository Repol on ECServer
cec migrate-content collection1 -s ICServer -d ECServer -r Repol -l
newCollection  Migrates content from collection collection1
on ICServer to repository Repol and collection newCollection on ECServer
cec migrate-content collection1 -s ICServer -d ECServer -r Repol -l
newCollection -c channel1 Migrates content from collection collection1
on ICServer to repository Repol, collection newCollection and channel
channel1 on ECServer
```

```
-----
cec compile-content
-----
```

Usage: cec compile-content

Compiles all the content items within the publishing job or list of assets and places the compiled renditions under the "dist" folder. Specify -s <server> to make content queries against this server. Optionally specify -a <assets> comma separated lists of assets. Optionally specify -t <contentType> compile all published assets of this content type. Optionally specify -i <repositoryId> Id of the repository for content

type queries.  
Optionally specify `-d <debug>` to start the compilation with `--inspect-brk` flag.  
Optionally specify `-v <verbose>` to display all warning messages during compilation.

Options:

<code>--help, -h</code>	Show Help
<code>--server, -s</code>	The registered OCM server
<code>--assets, -a</code>	The comma separated list of asset GUIDS
<code>--contenttype, -t</code>	Compile all the published assets of this content type.
<code>--repositoryId, -i</code>	Id of the repository for content type queries.
<code>--renditionJobId, -r</code>	Server invoked rendition job id for a publishing job
<code>--debug, -d</code>	Start the compiler with <code>"--inspect-brk"</code> option to debug compilation
<code>--verbose, -v</code>	Run in verbose mode to display all warning messages during compilation.

Examples:

```
cec compile-content publishingJobId -s UAT      Compiles the content items
in the specified publishing job retrieving content from the server.
cec compile-content publishingJobId -s UAT -d   Waits for the debugger to
be attached. Once attached, compiles the content in the specified
publishing job.
cec compile-content -a GUID1,GUID2 -s UAT      Compiles the assets by
retrieving content from the specified server.
cec compile-content -t Blog -i REPOGUID -s UAT Compiles the published
assets of this content type from the specified server.
Missing required parameters: <publishingJobId> or <assets> and <server>
```

```
-----
cec upload-compiled-content
-----
```

Usage: `cec upload-compiled-content <path>`

Uploads the compiled content to OCM server. Specify the site `<site>` on the server. Specify the server with `-r <server>` or use the one specified in `cec.properties` file.

Options:

<code>--help, -h</code>	Show Help
<code>--server, -s</code>	The registered OCM server

Examples:

```
cec upload-compiled-content dist/items.zip
cec upload-compiled-content dist/items.zip -s UAT
```

```
-----
cec download-taxonomy
-----
```

Usage: `cec download-taxonomy <name>`

Downloads a taxonomy from OCM server. Optionally specify the taxonomy id

with `-i <id>` if another taxonomy has the same name. Specify the server with `-s <server>` or use the one specified in `cec.properties` file. Specify the status of the taxonomy with `-t` and the valid values are

```
promoted
published
```

Options:

```
--help, -h      Show Help
--status, -t    The taxonomy status [promoted | published] [required]
--id, -i       Taxonomy Id
--server, -s    The registered OCM server
```

Examples:

```
cec download-taxonomy Taxonomy1 -t promoted
cec download-taxonomy Taxonomy1 -i 6A6DC736572C468B90F2A1C17B7CE5E4 -
t promoted
cec download-taxonomy Taxonomy1 -t published -s UAT
```

```
-----
cec upload-taxonomy
-----
```

Usage: `cec upload-taxonomy <taxonomy>`

Uploads a taxonomy to OCM server. Specify `-c <createnew>` to create new taxonomy when one already exists. Specify the server with `-s <server>` or use the one specified in `cec.properties` file.

Options:

```
--help, -h      Show Help
--createnew, -c To create new a taxonomy
--name, -n      The name of the new taxonomy
--abbreviation, -a The abbreviation of the new taxonomy
--description, -d The description of the new taxonomy
--file, -f      Flag to indicate the taxonomy is from file
--server, -s    The registered OCM server
```

Examples:

```
cec upload-taxonomy
Taxonomy1                                Create a new
taxonomy or a draft of existing taxonomy on upload
cec upload-taxonomy Taxonomy1 -s
UAT                                       Create a new taxonomy or
a draft of existing taxonomy on upload on the registered server UAT
cec upload-taxonomy Taxonomy1 -
c                                       Create a new taxonomy on
upload
cec upload-taxonomy Taxonomy1 -c -n Taxonomy1_2 -a t12 -d "Taxonomy1
copy" Create a new taxonomy on upload with given name, abbreviation
and description
cec upload-taxonomy ~/Documents/6A6DC736572C468B90F2A1C17B7CE5E4.json
-f Create a new taxonomy or a draft of existing taxonomy on upload
the JSON file
```

```
-----  
cec control-taxonomy  
-----
```

Usage: cec control-taxonomy <action>

Perform <action> on taxonomy on OCM server. Specify the taxonomy with -n <name> or -i <id>. Specify the server with -s <server> or use the one specified in cec.properties file. The valid actions are

```
promote  
publish  
unpublish
```

Options:

```
--help, -h          Show Help  
--name, -n          Taxonomy name  
--id, -i            Taxonomy Id  
--publishable, -p  Allow publishing of this taxonomy, defaults to true  
--channels, -c      List of channels to publish or unpublish, required when  
<action> is publish or unpublish  
--server, -s        The registered OCM server
```

Examples:

```
cec control-taxonomy promote -n Taxonomy1  
Promote taxonomy Taxonomy1 and allow publishing  
cec control-taxonomy promote -i 6A6DC736572C468B90F2A1C17B7CE5E4 -p false  
Promote the taxonomy and not allow publishing  
cec control-taxonomy publish -n Taxonomy1 -c Channel1,Channel2  
cec control-taxonomy unpublish -n Taxonomy1 -c Channel1  
cec control-taxonomy publish -n Taxonomy1 -c Channel1 -s UAT
```

```
-----  
cec list-translation-jobs  
-----
```

Usage: cec list-translation-jobs

Lists translation jobs from local or from OCM server.

Options:

```
--help, -h          Show Help  
--server, -s        The registered OCM server
```

Examples:

```
cec list-translation-jobs          Lists local translation jobs  
cec list-translation-jobs -s       Lists translation jobs on the server  
specified in cec.properties file  
cec list-translation-jobs -s UAT   Lists translation jobs on the registered  
server UAT
```

```
-----  
cec create-translation-job  
-----
```

Usage: cec create-translation-job <name>

Creates a translation job <name> for a site on OCM server. Specify the server with -r <server> or use the one specified in cec.properties file. Specify -l <languages> to set the target languages, use "all" to select all languages from the translation policy. Optionally specify -c <connector> to set the translation connector. Optionally specify -t <type> to set the content type. The valid values for <type> are:

```
siteAll
siteItems
siteAssets
```

Options:

```
--help, -h          Show Help
--site, -s           <site> Site [required]
--languages, -l     <languages> The comma separated list of languages
used to create the translation job [required]
--connector, -c     The translation connector
--type, -t          The type of translation job contents
--server, -r        The registered OCM server
```

Examples:

```
cec create-translation-job job1 -s Site1 -l all
cec create-translation-job job1 -s Site1 -l all -r UAT
cec create-translation-job job1 -s Site1 -l de-DE,it-IT
cec create-translation-job job1 -s Site1 -l de-DE,it-IT, -t siteItems
cec create-translation-job job1 -s Site1 -l de-DE,it-IT -c Lingotek
```

```
-----
cec download-translation-job
-----
```

Usage: cec download-translation-job <name>

Downloads translation job <name> from OCM server. Specify the server with -s <server> or use the one specified in cec.properties file.

Options:

```
--help, -h          Show Help
--server, -s        The registered OCM server
```

Examples:

```
cec download-translation-job Site1Job
cec download-translation-job Site1Job -s UAT
```

```
-----
cec submit-translation-job
-----
```

Usage: cec submit-translation-job <name>

Submits translation job <name> to translation connection <connection>.

Options:

```
--help, -h          Show Help
--connection, -c    <connection> Connection [required]
```

Examples:

```
cec submit-translation-job Site1Job1 -c connector1-auto
```

```
-----  
cec refresh-translation-job  
-----
```

Usage: cec refresh-translation-job <name>

Refreshes translation job <name> from translation connection.

Options:

```
--help, -h    Show Help  
--server, -s  The registered OCM server
```

Examples:

```
cec refresh-translation-job Site1Job1  
cec refresh-translation-job Site1Job1 -s UAT Refresh translation job  
Site1Job1 on the registered server UAT
```

```
-----  
cec ingest-translation-job  
-----
```

Usage: cec ingest-translation-job <name>

Gets translated job <name> from translation connection and ingest.

Options:

```
--help, -h    Show Help  
--server, -s  The registered OCM server
```

Examples:

```
cec ingest-translation-job Site1Job1          Ingest local translation job  
cec ingest-translation-job Site1Job1 -s DEV  Ingest translation job  
Site1Job1 on the registered server DEV
```

```
-----  
cec upload-translation-job  
-----
```

Usage: cec upload-translation-job <name>

Uploads translation <name> to OCM server, validate and then ingest the translations. Optionally specify -v to validate only. Optionally specify -f <folder> to set the folder to upload the translation zip file. Specify the server with -s <server> or use the one specified in cec.properties file.

Options:

```
--help, -h          Show Help  
--folder, -f        <folder> Folder to upload the translation zip file  
--validateonly, -v  Validate translation job without import.  
--server, -s        The registered OCM server
```

Examples:

```
cec upload-translation-job Site1Job1          File will  
be uploaded to the Home folder.  
cec upload-translation-job Site1Job1 -s UAT  File will
```

be uploaded to the Home folder on registered server UAT  
cec upload-translation-job Site1Job -f Import/TranslationJobs File  
will be uploaded to folder Import/TranslationJobs.  
cec upload-translation-job Site1Job -v  
Validate the translation job without import.

-----  
cec create-translation-connector  
-----

Usage: cec create-translation-connector <name>

Creates the translation connector <name>. By default, it creates a mockTranslationConnector. Optionally specify -f <source> to create from a different source.

Valid values for <source> are:

mockTranslationConnector

Options:

--help, -h Show Help  
--from, -f <source> to create from

Examples:

cec create-translation-connector connector1

-----  
cec start-translation-connector  
-----

Usage: cec start-translation-connector <name>

Starts translation connector <name>. Optionally specify -p <port> to set the port, default port is 8084.

Options:

--help, -h Show Help  
--port, -p Set <port>. Defaults to 8084.  
--debug, -d Start the translation connector server with "--inspect" option

Examples:

cec start-translation-connector connector1  
cec start-translation-connector connector1 -p 7777  
cec start-translation-connector connector1 -d Start the translation connector server with "--inspect" option to allow debugger to be attached.

-----  
cec register-translation-connector  
-----

Usage: cec register-translation-connector <name>

Registers a translation connector. Specify -c <connector> for the connector. Specify -s <server> for the connector server URL. Specify -u <user> and -p <password> for connecting to the server. Specify -f



<fields> for custom fields.

Options:

```
--help, -h      Show Help
--connector, -c <connector> Connector name [required]
--server, -s    <server> Server URL [required]
--user, -u     <user> User name [required]
--password, -p <password> password [required]
--fields, -f   <fields> translation connector custom fields
```

Examples:

```
cec register-translation-connector connector1-auto -c connector1 -s http://
localhost:8084/connector/rest/api -u admin -p SamplePass1 -f
"BearerToken:Bearer token1,WorkflowId:machine-workflow-id,AdditionalData:{}"
```

```
-----
cec create-group
-----
```

Usage: cec create-group <name>

Creates an OCM group on OCM server. Specify the server with -s <server>. Set the group type with -t <type>. The valid group types are

```
PUBLIC_OPEN
PUBLIC_CLOSED
PRIVATE_CLOSED
```

Options:

```
--help, -h      Show Help
--type, -t     The group type [PUBLIC_OPEN | PUBLIC_CLOSED | PRIVATE_CLOSED]
--server, -s   <server> The registered OCM server
```

Examples:

```
cec create-group Group1          Create group Group1, people can
add themselves to the group and share content with the group
cec create-group Group1 -t PUBLIC_CLOSED  Create group Group1, only group
managers can add members but people can share content with the group
cec create-group Group1 -t PRIVATE_CLOSED Create group Group1, only group
managers can add members and only members can share content with the group
cec create-group Group1 -s DEV
```

```
-----
cec delete-group
-----
```

Usage: cec delete-group <name>

Deletes an OCM group on OCM server. Specify the server with -s <server>.

Options:

```
--help, -h      Show Help
--server, -s   <server> The registered OCM server
```

Examples:

```
cec delete-group Group1
```

```
cec delete-group Group1 -s DEV
```

```
-----  
cec add-member-to-group  
-----
```

Usage: cec add-member-to-group <name>

Adds users and groups to an OCM group and assign a role on OCM server. Specify the server with -s <server>. The valid roles are

```
MANAGER  
MEMBER
```

Options:

```
--help, -h      Show Help  
--users, -u     The comma separated list of user names  
--groups, -g    The comma separated list of group names  
--role, -r      The role [MANAGER | MEMBER] to assign to the users or  
groups [required]  
--server, -s    The registered OCM server
```

Examples:

```
cec add-member-to-group Group1 -u user1,user2 -g Group2,Group3 -r  
MEMBER  
cec add-member-to-group Group1 -u user1,user2 -g Group2,Group3 -r  
MEMBER -s DEV
```

```
-----  
cec remove-member-from-group  
-----
```

Usage: cec remove-member-from-group <name>

Removes users and groups from an OCM group on OCM server. Specify the server with -s <server>.

Options:

```
--help, -h      Show Help  
--members, -m   The comma separated list of user and group names  
[required]  
--server, -s    The registered OCM server
```

Examples:

```
cec remove-member-from-group Group1 -m user1,user2,Group2,Group3  
cec remove-member-from-group Group1 -m user1,user2,Group2,Group3 -s  
DEV
```

```
-----  
cec create-encryption-key  
-----
```

Usage: cec create-encryption-key <file>

Create an encryption key to encrypt/decrypt password for servers and save to <file>. Use NodeJS 10.12.0 or later.

Options:  
 --help, -h Show Help

Examples:  
 cec create-encryption-key ~/.ceckey Create encryption key and save to  
 file ~/.ceckey

-----  
 cec register-server  
 -----

Usage: cec register-server <name>

Registers a OCM server. Specify -e <endpoint> for the server URL. Specify -u  
 <user> and -p <password> for connecting to the server. Optionally specify -k  
 <key> to encrypt the password. Optionally specify -t <type> to set the  
 server type. The valid values for <type> are:

```
pod_ec
pod_ic
dev_ec
dev_pod
dev_osso
```

and the default value is pod\_ec.

For pod\_ec server, optionally specify <idcsurl>, <clientid>, <clientsecret>  
 and <scope> for headless commands.

Options:  
 --help, -h Show Help  
 --endpoint, -e <endpoint> Server endpoint [required]  
 --user, -u <user> User name [required]  
 --password, -p <password> Password [required]  
 --key, -k The key file used to encrypt the password  
 --type, -t <type> Server type  
 --idcsurl, -i <idcsurl> Oracle Identity Cloud Service Instance URL  
 --clientid, -c <clientid> Client ID  
 --clientsecret, -s <clientsecret> Client secret  
 --scope, -o <clientsecret> Scope  
 --timeout, -m Timeout in millisecond when try to login to the  
 server. Defaults to 30000ms.

Examples:  
 cec register-server server1 -e http://server1.com -u user1 -p samplePass1 -  
 i http://idcs1.com -c clientid -s clientsecret -o https://primary-audience-  
 and-scope The server is a tenant on Oracle Public cloud  
 cec register-server server1 -e http://server1.com -u user1 -p  
 samplePass1  
 The server is a tenant on Oracle Public cloud  
 cec register-server server1 -e http://server1.com -u user1 -p samplePass1 -  
 m 60000  
 The server is a tenant on Oracle Public cloud  
 cec register-server server1 -e http://server1.git.oraclecorp.com.com -u  
 user1 -p samplePass1 -t  
 dev\_ec The server

is a standalone development instance

```
cec register-server server1 -e http://server1.com -u user1 -p
samplePass1 -k
~/.ceckey
```

The password will be encrypted

```
-----
cec set-oauth-token
-----
```

Usage: cec set-oauth-token <token>

Set OAuth token for a registered server.

Options:

```
--help, -h    Show Help
--server, -s  The registered OCM server [required]
```

Examples:

```
cec set-oauth-token token1 -s UAT Set OAuth token for server UAT,
all CLI commands using UAT will be headless
```

```
-----
cec list
-----
```

Usage: cec list

Lists local or server resources such components and templates. Specify the server with -s <server> or use the one specified in cec.properties file. Optionally specify -t <types> to list specific types of resources on the OCM server.

Valid values for <types> on the server are:

```
channels
components
localizationpolicies
recommendations
repositories
sites
templates
taxonomies
translationconnectors
```

Options:

```
--help, -h    Show Help
--types, -t   <types> The comma separated list of resource types
--server, -s  <server> The registered OCM server
```

Examples:

```
cec list                List all local resources
cec list -s            List resources on the server
specified in cec.properties file
cec list -t components,channels -s    List components and channels
on the server specified in cec.properties file
```

```
cec list -t components,channels -s UAT List components and channels on
the registered server UAT
```

```
-----
cec execute-get
-----
```

```
Usage: cec execute-get <endpoint>
```

Makes an HTTP GET request to a REST API endpoint on OCM server. Specify the server with `-s <server>`.

Options:

```
--help, -h    Show Help
--file, -f    The file to save the result [required]
--server, -s  The registered OCM server
```

Examples:

```
cec exeg "/sites/management/api/v1/sites?links=none" -f allsites.json -s
DEV
cec exeg "/content/management/api/v1.1/channels?links=none" -f
allchannels.json -s DEV
cec exeg "/documents/api/1.2/folders/self/items" -f homefolderitems.json -
s DEV
```

```
-----
cec install
-----
```

```
Usage: cec install
```

Creates an initial source tree in the current directory.

With `cec install`, your source can be in a separate directory to the `cec` command install files, and you no longer need your source to be within a `sites-toolkit` directory.

The `cec.properties` file can be used to specify server settings. It will be picked up from the source directory, or can be specified with environment variable `CEC_PROPERTIES`

Use `cec develop` to start a dev/test server for your source. Different ports can be used for the server, to enable multiple source trees to exist.

Options:

```
--help, -h  Show Help
```

Examples:

```
cec install
```

```
-----
cec develop
-----
```

```
Usage: cec develop
```

Starts a test server in the current folder. Specify the server with `-s <server>` or use the one specified in `cec.properties` file. Optionally specify

-p <port> to set the port, default port is 8085.

Options:

```
--help, -h      Show Help
--port, -p      Set <port>. Defaults to 8085.
--server, -s    The registered OCM server
--debug, -d     Start the server with "--inspect"
```

Examples:

```
cec develop
cec develop -p 7878
cec develop -p 7878 -s UAT
```

```
-----
cec sync-server
-----
```

Usage: cec sync-server

Starts a sync server in the current folder to sync changes notified by web hook from <server> to <destination> server. Specify the source server with -s <server> and the destination server with -d <destination>. Optionally specify -p <port> to set the port, default port is 8086. To run the sync server over HTTPS, specify the key file with -k <key> and the certificate file with -c <certificate>. Set authorization option with -a and the valid values are

```
none
basic
header
```

Options:

```
--help, -h      Show Help
--server, -s    The registered OCM server for sync source
[required]
--destination, -d  The registered OCM server for sync destination
[required]
--authorization, -a  The authorization method [none | basic | header]
for the web hook event, defaults to basic
--username, -u    The username used to authenticate the web hook
event when <authorization> is basic
--password, -w    The password used to authenticate the web hook
event when <authorization> is basic
--values, -v      The comma separated list of name-value pairs
used to authenticate the web hook event when <authorization> is header
--port, -p      Set port. Defaults to 8086.
--key, -k       The key file for HTTPS
--certificate, -c  The certificate file for HTTPS
```

Examples:

```
cec sync-server -s DEV -d UAT -u admin -w
samplePass1          Use Basic authorization
cec sync-server -s DEV -d UAT -u admin -w samplePass1 -p
7878                Use Basic authorization and port set to 7878
cec sync-server -s DEV -d UAT
```

```

Use Basic authorization and the username and password will be prompted to
enter
  cec sync-server -s DEV -d UAT -u admin                               Use
Basic authorization and the password will be prompted to enter
  cec sync-server -s DEV -d UAT -a header -v key1:value1,key2:value2  Use
Header authorization
  cec sync-server -s DEV -d UAT -a none                               No
authorization
  cec sync-server -s DEV -d UAT -k ~/keys/key.pem -c ~/keys/cert.pem  The
sync server will start over HTTPS
Missing required arguments: server, destination

```

```

-----
cec webhook-server
-----
Usage: cec webhook-server

```

Starts a server in the current folder to handle events notified by web hook from <server>. Optionally specify -p <port> to set the port, default port is 8087. The supported event types are

```

seo - refresh Detailed page in the Prerender cache

```

```

Options:
  --help, -h           Show Help
  --type, -t           The webhook server type [seo] [required]
  --contenttype, -c   The content type [required]
  --detailpage, -d   The full url of the site detail page for this type
[required]
  --server, -s        The registered OCM server [required]
  --port, -p          Set port. Defaults to 8087.

```

```

Examples:
  cec webhook-server -t seo -s DEV -c Blog -d "/site/blogsite/detailpage"
  cec webhook-server -t seo -s DEV -c Blog,Author -d "/site/blogsite/
blogdetail,/site/blogsite/authordetail"
  cec webhook-server -t seo -s DEV -c Blog -d "/site/blogsite/detailpage" -p
7878
Missing required arguments: type, contenttype, detailpage, server

```

## Testowanie przy użyciu lokalnego zestawu testów

Przed zaimportowaniem składników niestandardowych, szablonów i układów zawartości do usługi Oracle Content Management należy uruchomić je przy użyciu lokalnego zestawu testów.

Aby uruchomić lokalny zestaw testów, należy:

1. W oknie terminala wpisać polecenie `cd cec`.
2. Wpisać polecenie `cec develop &` lub `cec develop --server <nazwa_serwera> &`
3. Aby wyświetlić składniki, szablony i układy zawartości, działające w lokalnym zestawie testów, otworzyć przeglądarkę, a następnie stronę `http://localhost:8085`.

4. Składniki, szablony, motywy itp. można znaleźć w następujących katalogach:
  - `cec/src/main/components`
  - `cec/src/main/templates`
  - `cec/src/main/themes`

## Uaktualnienie do jQuery 3.5.x

Za pomocą zestawu OCE Toolkit można identyfikować niepoprawnie zamknięte znaczniki html.

Aby wybrać poprawkę w zakresie bezpieczeństwa w analizatorze składni (parserze) HTML, należy uaktualnić bibliotekę do wersji do jQuery 3.5.x.

<https://blog.jquery.com/2020/04/10/jquery-3-5-0-released>

Analiza składniowa niektórych napisów HTML przy użyciu biblioteki jQuery 3.5.x daje inne wyniki niż przy użyciu wersji 3.4.x. W napisach tych są uwzględniane znaczniki samozamykające się, niezgodnie ze standardem HTML; na przykład: "`<div />`". Takie napisy, zwłaszcza gdy stanowią część dłuższej sekwencji znaczników, mogą być analizowane w wersji 3.5.x inaczej niż w wersji 3.4.x.

Aby w plikach JSON oraz plikach HTML i JS składników odszukać niepoprawnie zamknięte znaczniki html, należy użyć polecenia `cec create-asset-report <serwis>`.

## Tworzenie różnych elementów dla Oracle Content Management za pomocą usługi Developer Cloud Service

Zestaw OCE Toolkit — integracja usługi Developer Cloud Service — pomaga tworzyć szablony, motywy, składniki niestandardowe i układy zawartości serwisów dla Oracle Content Management.

Używając zestawu OCE Toolkit, można korzystać z repozytoriów zasobów, z plików oraz z folderów dostępnych w Oracle Content Management. OCE Toolkit zawiera narzędzia pozwalające tworzyć i rozwijać niestandardowe składniki i szablony serwisów, w tym motywy i układy zawartości. Zawiera także repozytorium Git oraz oprogramowanie do wykonywania testów lokalnych, ułatwiające szybkie tworzenie iteracyjne, a także przykładowe testy jednostkowe, od których można zacząć.

OCE Toolkit pomaga przy wykonywaniu następujących zadań:

- Konfigurowanie lokalnego środowiska programistycznego pod kątem używania instancji Oracle Content Management do lokalnego tworzenia i testowania składników, szablonów, motywów i układów zawartości
- Tworzenie składników, szablonów serwisów i układów zawartości z przykładów, uruchamianie w zestawie testów tych elementów, eksplorowanie ich oraz opracowywanie składników, szablonów, motywów i układów zawartości w środowisku Developer Cloud Service



- Importowanie — z Oracle Content Management do środowiska Developer Cloud Service — szablonów lub składników, które zostały utworzone na potrzeby zarządzania źródłami i przyszłych potrzeb programistycznych
- Eksportowanie składników, szablonów lub układów zawartości z środowiska Developer Cloud Service do usługi Oracle Content Management, w której będą używane
- Kopiowanie już istniejących składników, szablonów lub układów zawartości
- Tworzenie testów jednostkowych
- Optymalizowanie składników
- Wprowadzanie składników i szablonów do środowiska wykonawczego Oracle Content Management

W poniższych tematach opisano, jak skonfigurować środowisko Developer Cloud Service pod kątem opracowywania składników niestandardowych, szablonów i motywów serwisów oraz układów zawartości:

1. [Korzystanie z Developer Cloud Service — informacje podstawowe](#)
2. [Logowanie do konsoli Developer Cloud Service dla Oracle Content Management](#)
3. [Tworzenie projektu w usłudze Developer Cloud Service](#)
4. [Dodawanie zestawu OCE Toolkit do kodu projektu w nowym repozytorium Git](#)
5. [Testowanie niestandardowych składników, szablonów i układów zawartości przy użyciu lokalnego zestawu testów](#)
6. [Scalanie zmian](#)

Następujące tematy zawierają więcej informacji o korzystaniu z zestawu OCE Toolkit usługi Oracle Content Management :

- [Korzystanie z narzędzia cec opartego na wierszu polecenia](#)
- Tworzenie składników niestandardowych za pomocą usługi Developer Cloud Service
- Tworzenie szablonów za pomocą usługi Developer Cloud Service
- Tworzenie układów zawartości

## Korzystanie z Developer Cloud Service — informacje podstawowe

Oracle Developer Cloud Service jest to usługa PaaS (Platform as a Service) oparta na chmurze, służąca do tworzenia oprogramowania. To hostowane środowisko dla infrastruktury tworzenia aplikacji. Udostępnia oparte na standardach mechanizmy integracji typu "open-source" do tworzenia aplikacji, współpracy nad nimi i ich wdrażania w Oracle Cloud

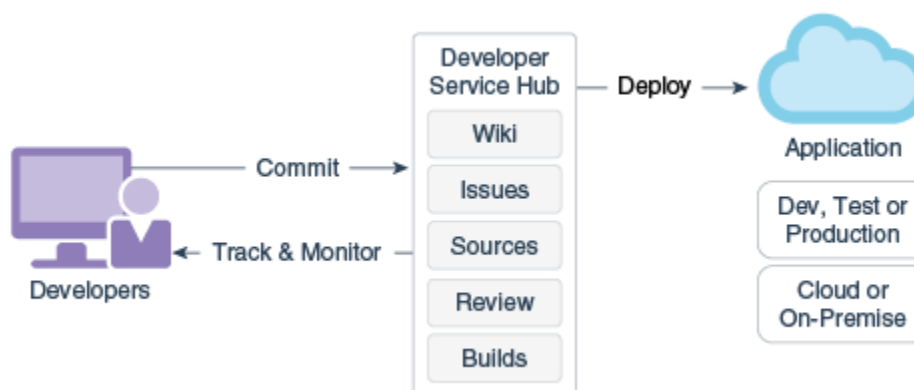
Usługa Developer Cloud Service jest kolekcją oprogramowania i usług hostowanych w Oracle Cloud, przydatnych do efektywnego zarządzania cyklem życia tworzenia aplikacji poprzez integrację systemów Git i Maven, a także opisów problemów i stron wiki. Korzystając z usługi Oracle Developer Cloud Service, można zatwierdzać kod źródłowy aplikacji w repozytorium Git w Oracle Cloud, śledzić przydzielone problemy i wady, udostępniać informacje za pomocą stron wiki, wspólnie sprawdzać kod źródłowy oraz monitorować kompilaty projektów. Po pomyślnym przetestowaniu można wprowadzić projekt do środowiska wykonawczego Oracle Content Management.

## Logowanie do konsoli Developer Cloud Service dla Oracle Content Management

Korzystając z konsoli Developer Cloud Service, można rozpocząć tworzenie składników niestandardowych dla usługi Oracle Content Management.

Będąc administratorem usług Oracle Cloud, można — za pomocą interfejsu administracyjnego aplikacji "Moje usługi" — tworzyć usługi Cloud i zarządzać nimi. Będąc administratorem instancji usługi Oracle Content Management i administratorem usługi Standard Developer Service, można te usługi skonfigurować, a następnie ich używać:

1. Zalogować się do Oracle Cloud, używając informacji udostępnionych dla konta.
2. Aby utworzyć instancje usług Oracle Content Management i Standard Developer Service oraz nimi zarządzać, zalogować się do interfejsu administracyjnego aplikacji "Moje usługi".



3. Sprawdzić e-mail powiązany z usługą Oracle Developer Cloud Service.
4. Skonfigurować instancję Oracle Content Management, używając szczegółów subskrypcji usługi, po czym przejść pod adres URL instancji usługi Oracle Content Management.
5. Przejść pod adres URL usługi Standard Developer Service.
6. Zalogować się do konta Oracle Developer Cloud Service.

Uzyskać dostęp do adresu URL usługi Developer Cloud Service, po czym zalogować się do konsoli.

## Tworzenie projektu w usłudze Developer Cloud Service

Używając szablonu projektu "Content Experience Cloud", można utworzyć projekt w usłudze Developer Cloud Service. Można także utworzyć projekt z pustym repozytorium Git, a następnie zaimportować zestaw OCE Toolkit z instancji Oracle Content Management.

- [Tworzenie projektów Developer Cloud Service za pomocą szablonu usługi Oracle Content Management](#)

- Tworzenie projektu w usłudze Developer Cloud Service za pomocą zestawu OCE Toolkit pobranego z Oracle Content Management
- Dodawanie zestawu OCE Toolkit do kodu projektu w nowym repozytorium Git

## Tworzenie projektów Developer Cloud Service za pomocą szablonu usługi Oracle Content Management

W usłudze Developer Cloud Service można utworzyć projekt dot. tworzenia niestandardowych składników, szablonów, motywów i układów zawartości.

Aby utworzyć projekt, należy:

1. Po zalogowaniu się do konsoli Developer Cloud Service, nacisnąć przycisk **Nowy projekt**.
2. Z listy szablonów wybrać **Content Management**, po czym nacisnąć przycisk **Dalej**.
3. W obszarze "Właściwości projektu" wybrać w polu **Kod znacznikowy wiki** właściwość **CONFLUENCE**.

## Tworzenie projektu w usłudze Developer Cloud Service za pomocą zestawu OCE Toolkit pobranego z Oracle Content Management

W usłudze Developer Cloud Service można utworzyć projekt obejmujący tworzenie składników niestandardowych, szablonów, motywów i układów zawartości.

Aby utworzyć projekt, należy:

1. Po zalogowaniu się do konsoli Developer Cloud Service, nacisnąć przycisk **Nowy projekt**.
2. Nadać nazwę projektowi, podać lub wybrać inne szczegóły projektu, po czym nacisnąć przycisk **Dalej**.
3. Z listy szablonów wybrać **repozytorium początkowe**, po czym nacisnąć przycisk **Dalej**.
4. W obszarze "Właściwości projektu" wybrać właściwość `Puste repozytorium dla repozytorium początkowego`. Nacisnąć przycisk **Zakończ**.

## Dodawanie zestawu OCE Toolkit do kodu projektu w nowym repozytorium Git

Można dodać zestaw OCE Toolkit do nowego, pustego repozytorium Git projektu.

1. W nowym projekcie, w obszarze **REPOSITORIES**, skopiować adres URL HTTP repozytorium Git projektu.
2. Otworzyć okno terminala, po czym wprowadzić następujące polecenie: `git clone <projekt>.git`
  - a. Gdy pojawi się wezwanie, podać hasło do usługi Developer Cloud Service.
  - b. Jeśli zostanie wyświetlony błąd "git is not a command" (git nie jest poleceniem), zainstalować oprogramowanie Git ze strony <https://git-scm.com/downloads>, po czym ponownie wprowadzić polecenie `git clone`.
3. `git clone git@github.com:oracle/content-and-experience-toolkit.git`

Można także pobrać stąd: <https://github.com/oracle/content-and-experience-toolkit/archive/master.zip>

4. `cp -R content-and-experience-toolkit/sites/cec-components <projekt>`
5. `cd <projekt>`
6. `git add cec-components`
7. `git commit -a -m "<komentarze>"`
8. `git push`

## Testowanie niestandardowych składników, szablonów i układów zawartości przy użyciu lokalnego zestawu testów

Przed zaimportowaniem składników niestandardowych, szablonów i układów zawartości do usługi Oracle Content Management należy uruchomić je przy użyciu lokalnego zestawu testów.

Aby uruchomić lokalny zestaw testów, należy:

1. W oknie terminala wpisać polecenie `cd cec-components`.  
Wprowadzić polecenie `npm start &`
2. Aby wyświetlić składniki, szablony i układy zawartości, działające w lokalnym zestawie testów, otworzyć przeglądarkę, a następnie stronę `http://localhost:8085`.

Testując składniki na lokalnym serwerze, można wybrać opcję używania zawartości z lokalnego szablonu lub z serwera Oracle Content Management.

## Scalanie zmian

Po utworzeniu składnika, szablonu lub układu zawartości bądź edycji kodu źródłowego na komputerze, należy w repozytorium Git projektu scalić nowe oraz zmienione składniki i szablony.

Aby scalić zmiany w repozytorium Git, należy w oknie terminala wprowadzić w podanej kolejności poniższe polecenia.

```
cd cec-components git pull
git add .
git status
git commit -a -m "komentarze" git pull
git push
```

## Propagacja zmian ze środowiska testowego do produkcyjnego za pomocą zestawu OCE Toolkit

Po opracowaniu szablonu serwisu można, korzystając z narzędzi CLI (Command-Line Interface) z zestawu OCE Toolkit, przekazywać szablon ze środowiska rozwojowego do testowego, a następnie do produkcyjnego na swoich serwerach Oracle Content Management.

Propagację zmian można przeprowadzić za pomocą poleceń z zestawu Toolkit, używając ich do tworzenia serwisów i zarządzania cyklem ich życia na serwerach rozwojowych, testowych i produkcyjnych. Zmian w serwisach można dokonać na serwerze produkcyjnym, a następnie przekazać te zmiany do środowisk testowego i produkcyjnego. Ten zestaw narzędzi opartych na wierszu polecenia można także wprowadzić do swoich środowisk skryptowych w celu zarządzania swoimi wdrożeniami. Korzystając z narzędzi CLI, można tworzyć nowe elementy, takie jak zasoby i składniki czy aktualizacje istniejącej zawartości.

Poniższa procedura pokazuje, jak za pomocą narzędzia CLI z zestawu OCE Toolkit przeprowadzać propagację zmian ze środowiska rozwojowego do testowego i produkcyjnego:

1. Skonfigurować serwery rozwojowy, testowy i produkcyjny z użyciem tego samego repozytorium i tego samego założenia systemowego dot. lokalizacji.

Aby umożliwić propagację zmian z serwera rozwojowego do serwera testowego, a następnie do serwera produkcyjnego, trzeba na każdym z nich skonfigurować repozytorium o tej samej nazwie i z tym samym założeniem systemowym dot. lokalizacji. Domyślnym założeniem systemowym dot. lokalizacji jest en-US, lecz można użyć innego pod warunkiem, że będzie ono identyczne na każdym z tym trzech serwerów.

Zob. Konfigurowanie repozytoriów zasobów.

2. Zarejestrować serwery rozwojowy, testowy i produkcyjny za pomocą usługi Oracle Content Management.

Przed przystąpieniem do propagacji zmian serwisu trzeba zarejestrować każdy z tych serwerów. Serwer można zarejestrować, używając polecenia `cec register-server` z zestawu OCE Toolkit:

```
cec register-server <name>
```

Określić następujące opcje polecenia:

- `-e <punkt_końcowy>` dla adresu URL serwera.
- `-u <użytkownik>` i `-p <hasło>` dla połączenia z serwerem.
- `-t <typ>` (opcjonalne) w celu określenia typu serwera. Wartością domyślną jest `pod_ec`.

Łącząc się z dzierżawą Oracle Content Management w publicznej chmurze Oracle, należy używać jedynie typu `pod_ec`.

Na przykład poniższe polecenie służy do zarejestrowania serwera będącego dzierżawą w publicznej chmurze Oracle:

```
cec register-server DEV -e https://DEV.example.com -u user1 -p <password>
```

Poniższe polecenie służy do zarejestrowania autonomicznej rozwojowej instancji Oracle Content Management:

```
cec register-server DEV -e https://DEV.git.oraclecorp.example.com -u user1 -p <password>
```

Po zarejestrowaniu serwera Oracle Content Management można wyszczególnić jego zawartość, używając polecenia `cec list` z zestawu Toolkit.

Poniższe polecenie służy do wyszczególnienia zawartości serwera rozwojowego:

```
cec-components> cec list -s DEV
- Logged in to remote server: <host:port>
Channels:
  Name                               Token
  StarterSite                         <site-id>

Components:
  Name                               Type
Published
  FooterBar                           Component group
  StarterComponent                     Local component
  StarterFooter                       Component group
  StarterNavMenu                      Local component

Localization policies:
  Name                               Required languages
Optional Languages
  en-US                               en-US

Repositories:
  Name
  r

Sites:
  Type          Published  Online  Theme
  Name                               StarterSiteTheme
Enterprise

Templates:
  Name                               Theme                               Type
  StarterTemplate                   StarterTheme                       Standard
```

3. Wysłać szablon serwisu do serwera rozwojowego, po czym utworzyć serwis, używając tego szablonu.

Szablon serwisu można utworzyć za pomocą polecenia `cec create-template`, po czym można wysłać uzyskany szablon do serwera rozwojowego. Można wówczas utworzyć serwis z szablonu, używając polecenia `cec create-site`. Poniższe polecenia służą do utworzenia i wysłania szablonu:

```
- cec create-template blog -f BlogTemplate
- cec upload-template blog -s DEV
```

Poniższe polecenie służy do utworzenia — z wysłanego szablonu — serwisu o nazwie `blog`:

```
cec-components> cec create-site blog -t blog -r r -l "en-US" -d "en-US" --server DEV
- Logged in to remote server: <https:<host:<port>
- establish user session
- get template
- get repository
```

```

- get localization policy
- creating enterprise site . . .
  name             blog
  template         blog
  site prefix      blog
  repository       r
  localization policy en-US
  default language en-US
- submit create site site
- create site in process: percentage 95
- create site in process: percentage 95
- create site in process: percentage 95
- create site in process: percentage 95
- create site in process: percentage 95
- site created

```

#### 4. Opublikować serwis i przełączyć go do trybu online na serwerze rozwojowym.

Po utworzeniu serwisu można go opublikować i przełączyć do trybu online, używając polecenia `cec control-site`:

```

cec-components> cec control-site
Usage: cec contrl-site <action>
Perform <action> on site in CEC server. Specify the site with -s <site>
Specify the server with -r <server>.

```

```

publish
unpublish
bring-online
take-offline

```

Options:

```

--site, -s    <site> Site
--server, -r  <server> The registered CEC server
--help, -h    Show help

```

Examples:

```

cec control-site publish -s Site1           Publish site Site1 on
the server
cec control-site publish -s Site1 -r UAT     Publish site Site1 on
the registered server UAT
cec control-site unpublish -s Site1 -r UAT   Inpublish site Site1 on
the registered server UAT
cec control-site bring-online -s Site1 -r UAT Bring site Site1 online
on the registered server UAT
cec control-site take-offline -s Site1 -r UAT Take site Site1 offline
on the registered server UAT

```

Not enough non-option arguments: got 0, need at least 1

```

cec-components> cec control-site publish --site blog --server DEV
- Logged in to the remote server: https://<host>:<port>
- establish user session
- get site: runtimeStatus: offline publishStatus: unpublished
- submit publish site
- publish in process: percentage 20

```

- publish in process: percentage 40
- publish in process: percentage 49
- publish in process: percentage 49
- publish in process: percentage 50
- publish in process: percentage 50

5. Aby przenieść serwis ze środowiska rozwojowego (DEV) do środowiska testów akceptacyjnych (UAT), trzeba ten serwis spakować. Modelem pakowania, umożliwiającym przenoszenie serwisów między serwerami, jest szablon. Utworzyć nowy szablon z serwisu utworzonego na serwerze rozwojowym, po czym ten szablon pobrać.

Za pomocą polecenia `cec create-template-from-site`, użytego w poniższym przykładzie, jest z serwisu `blog` tworzony szablon o nazwie `blog2`.

```
cec create-template-from-site blog2 -s blog
```

Pobrać z serwera rozwojowego utworzony szablon, używając polecenia `cec download-template`:

```
cec-components: cec download-template blog2 --server DEV
- Logged in to remote server: https://<host>:<port>
- establish user session
- export template
- template download to /Users/<user-name>/devenv/git/webclient/
developer/sites-toolkit/cec-components/dist/blog2.zip
- the template will be at /Users/<user-name>/devenv/git/webclient/
developer/sites-toolkit/cec-components/src/main/templates/blog2
- the theme for the template will be at /Users/<user-name>/
devenv/git/webclient/developer/sites-toolkit/cec-components/src/
main/themes/blogTheme
- create link _scs_theme_root_
- create link _scs_design_name_
- override component /Users/<user-name>devenv/git/webclient/
developer/sites-toolkit/cec-components/src/main/components/Starter-
Blog-Author-Summary
- override component /Users/<user-name>devenv/git/webclient/
developer/sites-toolkit/cec-components/src/main/components/Starter-
Blog-Post-Content
- override component /Users/<user-name>devenv/git/webclient/
developer/sites-toolkit/cec-components/src/main/components/Starter-
Blog-Post-Header
- override component /Users/<user-name>devenv/git/webclient/
developer/sites-toolkit/cec-components/src/main/components/Starter-
Blog-Post-Search-Result
- override component /Users/<user-name>devenv/git/webclient/
developer/sites-toolkit/cec-components/src/main/components/Starter-
Blog-Post-Post-Sidebar
- override component /Users/<user-name>devenv/git/webclient/
developer/sites-toolkit/cec-components/src/main/components/Starter-
Blog-Post-Summary
- set themeName to blogTheme in siteinfo.json
- unzip tmlate content file
*** template is ready to test: https://localhost:8085/templates/
```



```
blog2
cec upload-template blog2 --server UAT
```

6. Wysłać szablon w celu utworzenia typów zawartości i map układów zawartości.

```
cec upload-template blog2 --server UAT
```

7. Wysłać szablon, wykluczając z niego elementy zawartości (szablon zawartości).

```
cec upload-template blog2 --server UAT -x
```

Takie postępowanie ma na celu utworzenie serwisu z zawartością mającą identyfikatory GUID identyczne z występującymi w oryginalnym serwisie. Jeśli serwis zostałyby utworzony z szablonu z zawartością, to cała zawartość w nowym serwisie miałaby nowe identyfikatory GUID. Ponieważ chcemy zezwolić na aktualizację zawartości (a nie tworzenie nowej), trzeba wykluczyć zawartość z szablonu.

8. Utworzyć serwis z szablonu.

```
cec create-site blog -t blog2 -r r -l "en-US" -d "end-US" --server UAT
```

9. Wysłać szablon zawartości do kanału i kolekcji serwisu. Trzeba to zrobić, ponieważ szablon zawartości został wykluczony (w punkcie 7) z szablonu serwisu.

```
cec upload-content blog2 -t -r r -c blog -l "blog site" --server UAT
```

10. Opublikować serwis i przełączyć go do trybu online na serwerze testowym.

Opublikować serwis i przełączyć go do trybu online, używając polecenia `cec control-site`:

```
cec-components> cec control-site publish --site blog --server UAT
- Logged in to the remote server: https://<host>:<port>
- establish user session
- get site: runtimeStatus: offline publishStatus: unpublished
- submit publish site
- publish in process: percentage 20
- publish in process: percentage 40
- publish in process: percentage 49
- publish in process: percentage 49
- publish in process: percentage 50
- publish in process: percentage 50
- publish blob finished
```

11. Jeśli na serwerze DEV zostaną dokonać jakieś zmiany w serwisie blog, można przeprowadzić ich propagację do serwisu utworzonego na serwerze UAT.
12. Utworzyć kolejny szablon z serwisu (aby szablon zawierał dokonane zmiany).

```
cec create-template-from-site blog3 -s blog --server DEV
```

**13. Pobrać szablon.**

```
cec download-template blog3 -s DEV
```

**14. Wysłać szablon, po czym — aby przeprowadzić propagację zmian do środowiska testowego — utworzyć z tego szablonu serwis.**

```
cec upload-template blog3 -s UAT
```

To polecenie tworzy lub aktualizuje wszelkie składniki i motywy, które zostały zmienione, lecz wyklucza zawartość.

**15. Użyć polecenia `update-site`, aby wydzielić zawartość i zaktualizować strony.**

```
cec update-site blog -t blog3 - UAT
```

Na przykład:

```
cec-components> cec update-site blog -t blog3 --server UAT
Updating site: blog3
- Logged in to remote server: https://<host>:<port>
- pages          : updating file# 6   of 6   files
- content        : updating file# 3   of 3   files
- System Files   : updating file# 5   of 5   files
- controller     : no files in update, removing files on
server
- favicons       : no files in update, removing files on
server
- misc           : no files in update, removing files on
server
- seo            : no files in update, removing files on
server
- system         : no files in update, removing files on
server
- created content file /Users/<user-name>/devenv/git/webclient/
developer/sites-toolkit/cec-components/dist/blog3_export.zip
- upload content file
- get CSRF token
- submit import job, updating content
- import job in progress. . .
- import job in progress. . .
- import job in progress. . .
- content imported:
Update Site Results:
- Site Pages      : completed with 0 errors.
- Embedded Content : completed with 0 errors.
- System Files    : completed with 0 errors.
- Settings Files  : completed with 0 errors.
- Content Update  : completed with 0 errors.
```

**16. Sprawdzić serwis, aby upewnić się, że zmiany zostały wprowadzone.****17. Przeprowadzić propagację z serwera UAT do serwera PROD, postępując jak przy przenoszeniu serwisu z serwera DEV do UAT.**

18. Utworzyć serwis na serwerze produkcyjnym, przełączyć serwis do trybu online, po czym sprawdzić zmiany.

Można, używając polecenia `cec list` — wyszczególnić zawartość serwisu produkcyjnego i upewnić się, że zawiera zmiany, które zostały dokonane w środowisku rozwojowym. Można także sprawdzić serwis, aby potwierdzić propagację zmian do wersji produkcyjnej.

## Szyfrowanie hasła

Po zarejestrowaniu serwera za pomocą zestawu OCE Toolkit, trzeba zaszyfrować hasło, aby udostępnić serwer do lokalnego użycia.

1. Zarejestrować serwer Oracle Content Management, używając polecenia `cec register-server` zawierającego hasło podane tekstem jawnym.
2. Zaszyfrować hasło, używając polecenia `cec create-encryption-key`.

```
cec create-encryption-key <file> [alias: cek]
Create an encryption key to encrypt/decrypt password for servers.
```

3. Zarejestrować serwer z kluczem szyfrowania, dzięki czemu serwer stanie się dostępny dla lokalnego tworzenia i testowania.

Zaszyfrowane hasła są przechowywane w pliku połączenia z serwerem. Podczas łączenia się z zarejestrowanym serwerem hasło jest deszyfrowane.

## Rejestrowanie serwera

Serwer można zarejestrować do użycia z zestawem OCE Toolkit.

Używając polecenia `cec register-server` z kluczem szyfrowania, można zarejestrować serwer Oracle Content Management dla lokalnego tworzenia i testowania.

Gdy serwer zostanie zarejestrowany, zaszyfrowanie hasła uczyni serwer dostępnym do użycia z zestawem OCE Toolkit. Zob. [Szyfrowanie hasła](#).

## Tworzenie raportu dot. użytkownika i uprawnień w zakresie serwisu

Można utworzyć raport pozwalający zweryfikować i poprawić uprawnienia elementów serwera docelowego przy propagacji serwisu od testów do produkcji.

Należy użyć polecenia `cec create-asset-report`, po czym sprawdzić:

1. Sprawdzić członkostwo i przydział kanałów dla wszystkich artefaktów serwisu:
  - Motyw
  - Szablon
  - Składniki
  - Typ zawartości
2. Oznakować problemy, które można samodzielnie rozwiązać.

Na przykład:

```
cec create-asset-report blog1 -s <registered-server> -o
```

```
cec create-asset-report trbcent -s <registered-server> -o
```

Ten raport jest generowany w postaci pliku JSON, który można sprawdzić pod kątem problemów z użytkowaniem i uprawnieniami. W celu poprawy uprawnień można używać następujących poleceń:

- `cec share-type`: Udostępnianie (użytkownikom) typów na serwerze Oracle Content Management.
- `cec unshare-type`: Cofanie (określonym użytkownikom) dostępu do typów na serwerze Oracle Content Management.
- `cec share-repository`: Udostępnianie repozytorium (i używanym przez niego typów) na serwerze Oracle Content Management.
- `cec unshare-repository`: Cofanie (użytkownikom) dostępu do repozytorium na serwerze Oracle Content Management.
- Na przykład:

```
cec share-repository Repol -u <user-name1>,<user-name2> -r manager -t -s <registered-server>
```

## Pobieranie oraz wysyłanie dokumentów i folderów

Dokumenty i foldery można pobierać oraz wysyłać do/z serwera Oracle Content Management.

Są dostępne następujące polecenia pobierania oraz wysyłania dokumentów i folderów:

```
cec download-folder <path>           Downloads folder from CEC
server.                               [alias: dlfd]
cec upload-folder <path>           Uploads folder to CEC
server.                               [alias: ulfd]
cec download-file <file>           Downloads file <file> from CEC
server.                               [alias: dlf]
cec upload-file <file>           Uploads file <file> to CEC
server.                               [alias: ulf]
cec-share-folder <name>
cec-unshare-folder <name>
```

Za pomocą polecenia `cec-share-folder <nazwa>` można udostępnić użytkownikom folder na serwerze Oracle Content Management oraz przypisać im rolę. Należy określić serwer za pomocą opcji `"-s <serwer>"` lub użyć serwera określonego w pliku `cec.properties`. Poprawne role:

- manager
- contributor
- downloader

- `viewer`

W przypadku pobierania można określić hierarchię folderów.

## Tworzenie serwisu z szablonu i zachowywanie tych samych identyfikatorów GUID zawartości

Będąc programistą, można — za pomocą polecenia z zestawu OCE Toolkit — utworzyć serwis Oracle Content Management z szablonu przy jednoczesnym zachowaniu tych samych identyfikatorów GUID zawartości.

Należy użyć następującego polecenia z zestawu OCE Toolkit:

```
update create-site-from-template --reuse-content
```

Polecenie to tworzy serwis na serwerze Oracle Content Management oraz zachowuje — podczas tworzenia serwisu — identyfikatory (ID) zawartości. Zachowanie identyfikatorów zawartości jest wymagane dla wielokrotnych uruchomień "od testu do produkcji" w celu zapobieżenia duplikacji elementów zawartości w serwerze docelowym.

## Tworzenie szablonu firmowego z serwisu standardowego

Będąc programistą, można — za pomocą polecenia CLI z zestawu OCE Toolkit — utworzyć szablon firmowy z serwisu standardowego.

Domyślnie polecenie `create-template` tworzy szablon standardowy (jeśli serwis jest serwisem standardowym) lub szablon firmowy (jeśli serwis jest serwisem firmowym). Można także utworzyć szablon firmowy z serwisu standardowego.

Należy uruchomić polecenie `cec create-template` z nową opcją `-enterprise`:

```
cec create-template EnterpriseTempl -s StandardSite1 -e
```

## Importowanie i eksportowanie taksonomii

Za pomocą poleceń z zestawu OCE Toolkit można zaimportować taksonomie z komputera lokalnego do serwera Oracle Content Management lub wyeksportować taksonomie z serwera do komputera lokalnego.

Polecenie `cec download-taxonomy <nazwa>` eksportuje taksonomię z usługi Oracle Content Management. Taksonomia jest pobierana z serwera Oracle Content Management.

Dla tego polecenia można użyć następujących opcji:

- `--status, -t [promoted | published] [required]`: Określa status taksonomii.
- `--id, -i`: Określa ID taksonomii, jeśli inna taksonomia ma identyczną nazwę.
- `--server, -s`: Określa zarejestrowany serwer Oracle Content Management lub używa innego określonego w pliku `cec.properties`.
- `--help, -h`: Wyświetla pomoc odnoszącą się do polecenia.

Przykłady użycia polecenia `download-taxonomy`:

```
cec download-taxonomy Taxonomy1 -t promoted
```

```
cec download-taxonomy Taxonomy1 -i 6A6DC736572C468B90F2A1C17B7CE5E4 -  
t promoted
```

```
cec download-taxonomy Taxonomy1 -t published -s UAT
```

Polecenie `cec upload-taxonomy <taksonomia>` importuje taksonomię do usługi Oracle Content Management . Taksonomia jest wysyłana do serwera Oracle Content Management.

Dla tego polecenia można użyć następujących opcji:

- `--createnew, -c`: Tworzy nową taksonomię.
- `--name, -n`: Określa nazwę dla nowej taksonomii.
- `--abbreviation, -a`: Określa skrót dla nowej taksonomii.
- `--description, -d`: Określa opis nowej taksonomii.
- `--file, -f`: Sygnalizuje, czy taksonomia pochodzi z pliku.
- `--server, -s`: Określa zarejestrowany serwer Oracle Content Management lub używa innego określonego w pliku `cec.properties`.
- `--T2P`: sprawdza, czy istnieje zasób lub taksonomia o tym samym ID w docelowej instancji Oracle Content Management. Jeśli ma wartość `true`, zasób jest dodawany jako nowa wersja, a taksonomia jako wersja robocza (przesłaniająca istniejącą). W przeciwnym razie nowy zasób lub nowa taksonomia są tworzone z tymi samymi identyfikatorami.
- `--New`: Zawsze tworzy nowy zasób lub nową taksonomię w docelowej instancji Oracle Content Management.
- `--help, -h`: Wyświetla pomoc odnoszącą się do polecenia.

Poniżej jest przedstawionych kilka przykładów polecenia **upload-taxonomy**:

```
cec upload-taxonomy Taxonomy1
```

Create a new taxonomy or a draft of an existing taxonomy on upload

```
cec upload-taxonomy Taxonomy1 -s UAT
```

Create a new taxonomy or a draft of an existing taxonomy on upload on the registered server UAT

```
cec upload-taxonomy Taxonomy1 -c
```

Create a new taxonomy on upload

```
cec upload-taxonomy Taxonomy1 -c -n Taxonomy1_2 -a t12 -d
```

Create a new taxonomy on upload with the given name, abbreviation

```
"Taxonomy1 copy" and description cec upload-taxonomy
```

Create a new taxonomy or a draft of an existing taxonomy in <file-name>.json -f and upload the JSON file

Używając narzędzi CLI "od testów do produkcji" z zestawu OCE Toolkit, można zautomatyzować import i eksport zasobów (wraz z modelem zawartości) ze źródłowego serwera Oracle Content Management do docelowego serwera Oracle Content Management.

Menedżer lub administrator zawartości może — za pomocą poleceń z zestawu OCE Toolkit — importować i eksportować taksonomię w środowisku "od testów do produkcji". Mając uprawnienia menedżera, można dodać wersję roboczą taksonomii. Mając uprawnienia administratora zawartości, można utworzyć nową taksonomię.

Dla środowiska "od testów do produkcji" są dostępne operacje związane z cyklem życia taksonomii, takie jak promocja taksonomii, przypisanie taksonomii do repozytorium i jej opublikowanie. Zasoby można importować, wraz z informacjami o ich klasyfikacji oraz taksonomiami, z pliku importu, a eksportować — do pliku eksportu.

## Importowanie i eksportowanie rekomendacji

Za pomocą poleceń z zestawu OCE Toolkit można zaimportować rekomendacje z serwera Oracle Content Management do komputera lokalnego lub wyeksportować rekomendacje z komputera lokalnego do serwera.

Polecenie `cec download-recommendation <nazwa>` eksportuje rekomendację z usługi Oracle Content Management. Rekomendacja jest pobierana z serwera Oracle Content Management.

Dla tego polecenia można użyć następujących opcji:

- `--status, -t [promoted | published] [required]`: Określa status rekomendacji.
- `--id, -i`: Określa ID rekomendacji, jeśli inna rekomendacja ma identyczną nazwę.
- `--server, -s`: Określa zarejestrowany serwer Oracle Content Management lub używa innego określonego w pliku `cec.properties`.
- `--help, -h`: Wyświetla pomoc odnoszącą się do polecenia.

Przykłady użycia polecenia `download-recommendation`:

```
cec download-recommendation Recommendation1 -t promoted
```

```
cec download-recommendation Recommendation1 -i  
6A6DE836572C468B90F2A1C17B7CE5E4 -t promoted
```

```
cec download-recommendation recommendation -t published -s UAT
```

Polecenie `cec upload-recommendation <nazwa>` importuje rekomendację do usługi Oracle Content Management. Rekomendacja jest wysyłana do serwera Oracle Content Management.

Dla tego polecenia można użyć następujących opcji:

- `--createnew, -c`: Tworzy nową rekomendację.
- `--name, -n`: Określa nazwę dla nowej rekomendacji.
- `--abbreviation, -a`: Określa skrót dla nowej rekomendacji.
- `--description, -d`: Określa opis nowej rekomendacji.
- `--file, -f`: Sygnalizuje, czy rekomendacja pochodzi z pliku.

- `--server, -s`: Określa zarejestrowany serwer Oracle Content Management lub używa innego określonego w pliku `cec.properties`.
- `--help, -h`: Wyświetla pomoc odnoszącą się do polecenia.

Poniżej jest przedstawionych kilka przykładów polecenia **upload-recommendation**:

```
cec upload-recommendation Recommendation1
```

Create a new recommendation or a draft of an existing recommendation on upload

```
cec upload-recommendation Recommendation1 -s UAT
```

Create a new recommendation or a draft of an existing recommendation on upload on the registered server UAT

```
cec upload-recommendation Recommendation1 -c
```

Create a new recommendation on upload

```
cec upload-recommendation Recommendation1 -c -n Recommendation 1_2 -a t12 -d
```

Create a new recommendation on upload with the given name, abbreviation

```
"Recommendation1 copy" and description cec upload-recommendation
```

Create a new recommendation or a draft of an existing recommendation in `<file-name>.json -f` and upload the JSON file

## Dodawanie lub usuwanie zawartości kolekcji

Będąc programistą, można — za pomocą poleceń z zestawu OCE Toolkit — dodać zawartość do kolekcji lub usunąć zawartość z kolekcji.

Poniższe polecenie `cec-control-content` dodaje wszystkie elementy, znajdujące się w repozytorium **Repo1**, do kolekcji **Collection1**, znajdujące się na zarejestrowanym serwerze **UAT**:

```
cec control-content add -l Collection1 -r Repo1 -s UAT
```

Poniższe polecenie `cec-control-content` usuwa wszystkie elementy z kolekcji **Collection1**, znajdujące się na zarejestrowanym serwerze **UAT**:

```
cec control-content remove -l Collection -s UAT
```

## Tworzenie i rozwijanie niestandardowych edytorów pól za pomocą zestawu OCE Toolkit

OCE Toolkit umożliwia tworzenie składników typu "wygląd pola" (Field Appearance). Programiści mogą tworzyć edytory pól niestandardowych i tymi edytorami zarządzać.

W przypadku składnika typu "wygląd pola" można wykonywać następujące zadania:



- Otwieranie, kopiowanie lub usuwanie składnika
- Publikowanie składnika lub anulowanie jego publikacji
- Eksportowanie lub importowanie składnika
- Dodawanie lub usuwanie współużytkowników składnika
- Wyświetlanie właściwości
- Wybór loga składnika

Listę składników można filtrować na podstawie typu "wygląd pola".

Do tworzenia niestandardowych edytorów pól są dostępne następujące polecenia z zestawu OCE Toolkit:

```
cec add-field-editor <name>                Adds a field editor to
a field in a content type.                [alias: afe]
    cec remove-field-editor <name>        Removes a field editor
from a field in a content type.          [alias: rfe]
```

```
-----
cec add-field-editor
-----
Usage: cec add-field-editor <name>
```

Adds a field editor to a field in a content type.

Options:

```
--template, -t      The template the content type is from [required]
--contenttype, -c   The content type [required]
--field, -f         The field the field editor is for [required]
--contenttemplate, -n Flag to indicate the template is a content template
--help, -h         Show help [boolean]
```

Examples:

```
cec add-field-editor editor1 -t BlogTemplate -c BlogPost -f
summary          Use editor1 as the appearance for field summary in
content type BlogPost from local template at src/templates/BlogTemplate
    cec add-field-editor editor1 -t BlogTemplateContent -n -c BlogPost -f
summary          Use editor1 as the appearance for field summary in content type
BlogPost from local template at src/content/BlogTemplateContent
```

```
-----
cec remove-field-editor
-----
Usage: cec remove-field-editor <name>
```

Removes a field editor from a field in a content type.

Options:

```
--template, -t      The template the content type is from [required]
--contenttype, -c   The content type [required]
--field, -f         The field the field editor is for [required]
--contenttemplate, -n Flag to indicate the template is a content template
--help, -h         Show help [boolean]
```

Examples:

```
cec remove-field-editor editor1 -t BlogTemplate -c BlogPost -f
summary          Remove editor1 as the appearance for field summary
in content type BlogPost from local template at src/templates/
BlogTemplate
cec remove-field-editor editor1 -t BlogTemplateContent -n -c BlogPost
-f summary       Remove editor1 as the appearance for field summary in
content type BlogPost from local template at src/content/
BlogTemplateContent
```

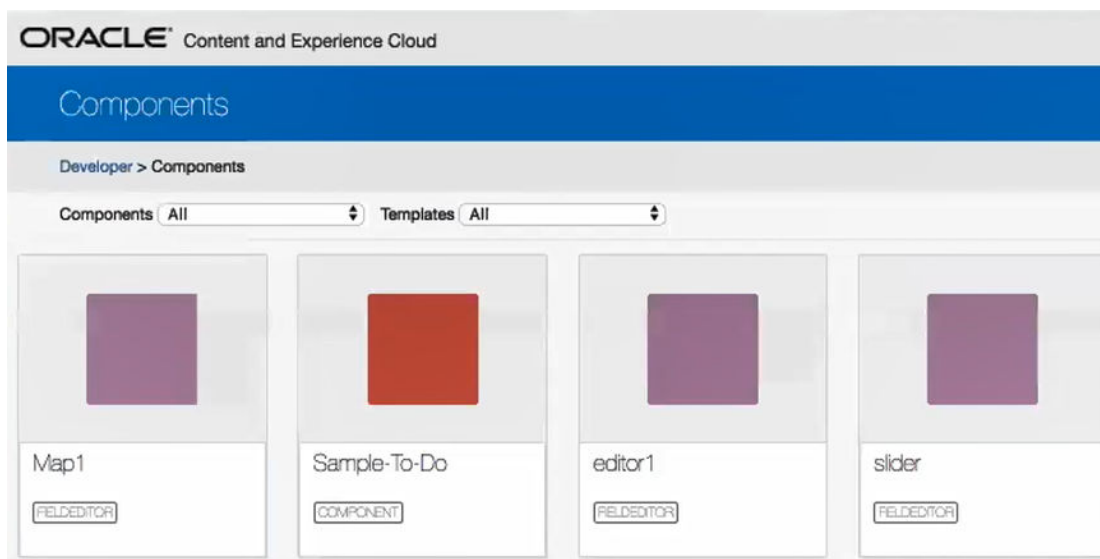
Do zestawu OCE Toolkit są dołączone następujące składniki typu "wygląd pola":

- TextFieldEditor
- SliderFieldEditor
- MapFieldEditor

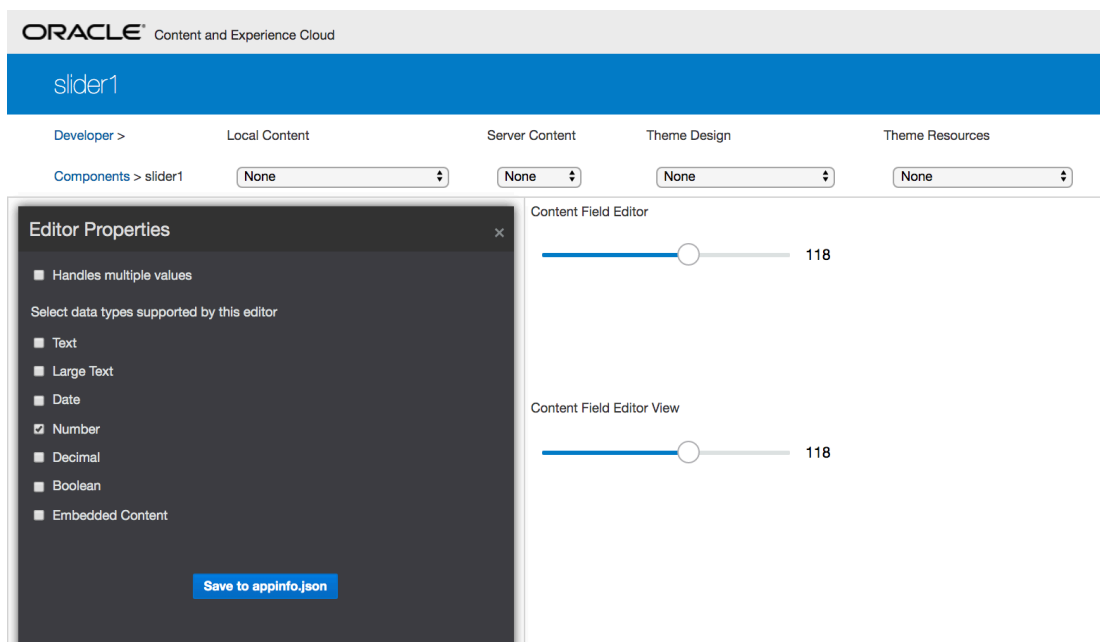
Na poniższym rysunku są pokazywane polecenia z zestawu OCE Toolkit, których można użyć do utworzenia przykładowych składników "wygląd pola".

```
2
3 # create
4 cec cc editor1 -f TextFieldEditor
5
6 cec cc slider -f SliderFieldEditor
7
8 cec cc Map1 -f MapFieldEditor
9
10 #local testing
11
12 # add to content type field
13 cec add-field-editor editor1 -t SimpleContent -n -c SimpleType -f title
14
15 cec add-field-editor slider -t SimpleContent -n -c SimpleType -f value
16
17
18 # upload editors
19 cec ulcp editor1,slider -p -s
20
21 # upload content
22
23 cec cr Repo5 -s
24
25 cec upload-content SimpleContent -r Repo5 -s
26
```

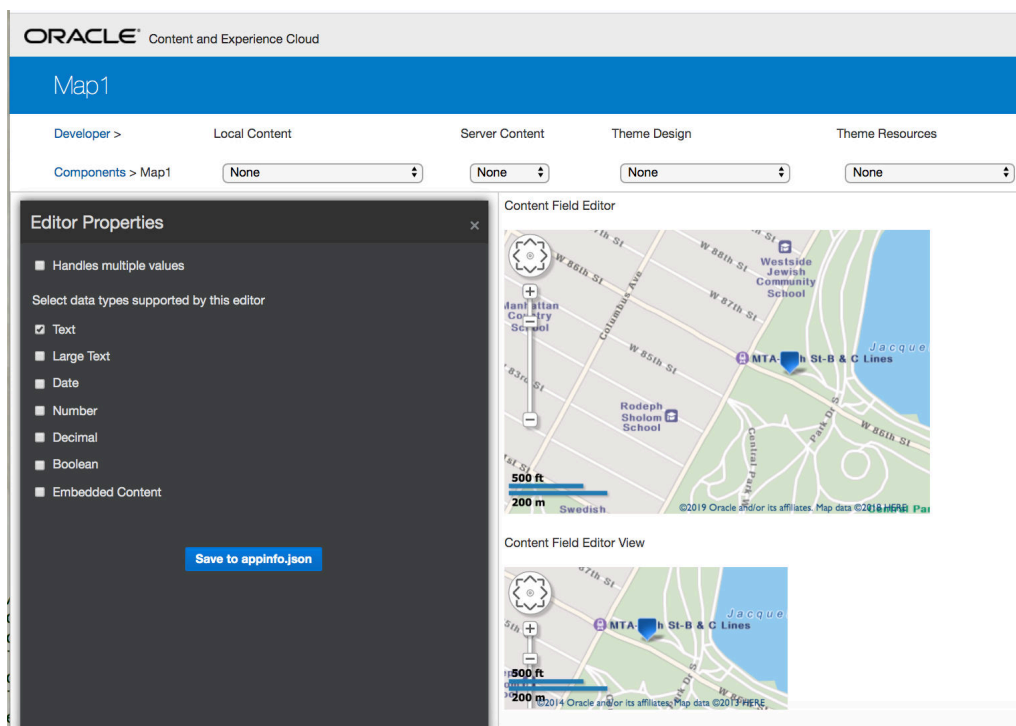
Składniki "wygląd pola" można utworzyć na swoim serwerze lokalnym, przetestować je, a następnie wysłać do instancji Oracle Content Management. Na poniższym rysunku są pokazywane te składniki z localhost:8085.



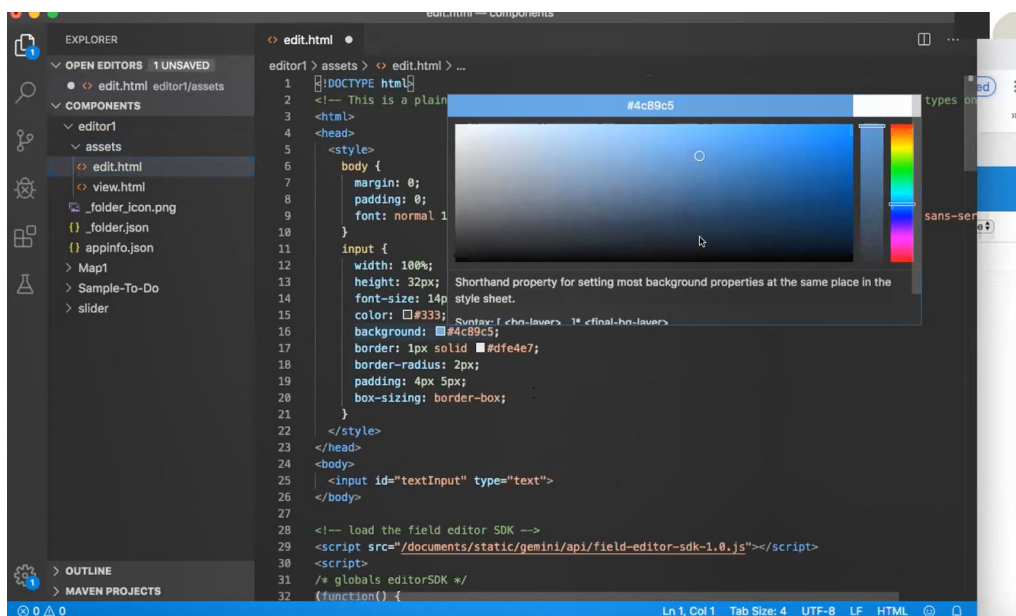
Każdy ze składników, na przykład suwak, można przetestować na serwerze lokalnym. Można wówczas wybrać właściwości dla składnika, po czym go zapisać.



W przypadku składnika "mapa" można kliknąć na mapie, aby podać lokalizację jako wartość w edytorze.



Plik HTML dla składnika można edytować w celu zmiany jego ustawień, takich jak kolor tła.



Za pomocą polecenia z zestawu OCE Toolkit można powiązać lokalnie edytor pola z polem o danym typie zawartości:

```
cec add-field-editor editor1 -t SimpleContent -n -c SimpleType -f title
```

```
cec add-field-editor slider -t SimpleContent -n -c SimpleType -f value
```

Po zakończeniu konfigurowania i testowania niestandardowych edytorów pól można te edytory wysłać do instancji Oracle Content Management, używając poleceń z zestawu OCE Toolkit:

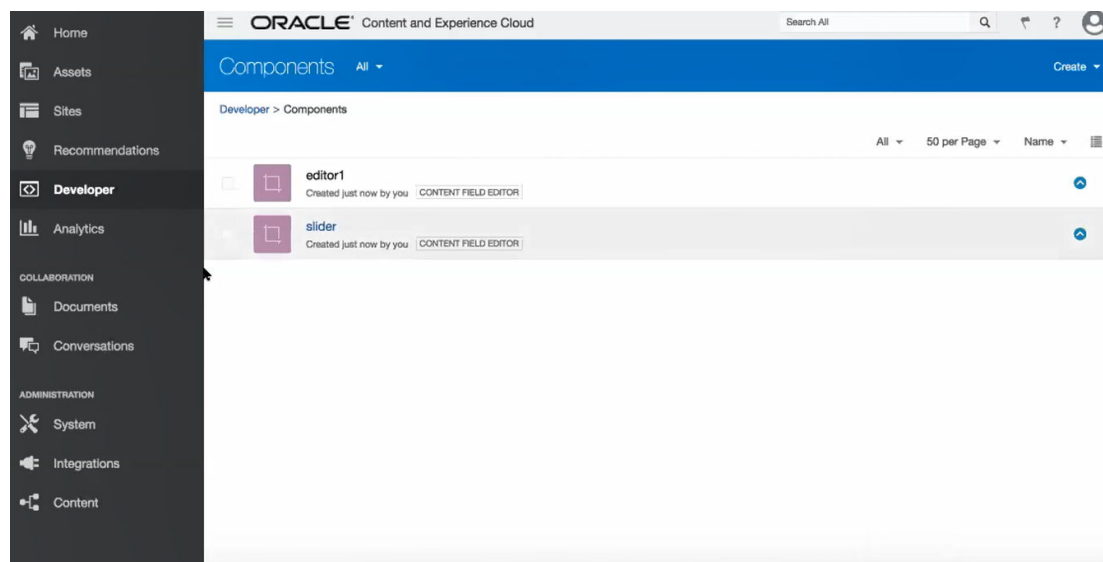
```
# upload editors
cec ulcp editor1,slider -p -s Latest

# upload content

cec cr Repo5 -s

cec upload-content SimpleContent -r Repo5 -s
```

Gdy niestandardowe edytory pól zostaną wysłane, nastąpi import składników do instancji Oracle Content M



## Przenoszenie lub aktualizowanie serwisu z jednego serwera do innego

Będąc programistą, można za pomocą polecenia z zestawu OCE Toolkit utworzyć lub zaktualizować serwis oraz jego zawartość z serwera A do serwera B.

Domyślnie wszystkie zasoby są transferowane. Chcąc przetransferować tylko opublikowane zasoby, należy użyć opcji "-p". Serwer źródłowy należy określić za pomocą opcji -s <serwer>, a serwer docelowy — za pomocą opcji -d <cel>. Jeśli serwis zawiera zasoby z innych repozytoriów, można opcjonalnie podać odwzorowanie repozytorium; w przeciwnym razie zasoby te nie będą transferowane.

Można zaktualizować serwis lub przenieść go z środowiska testowego do produkcyjnego, używając następującego polecenia:

```
cec transfer-site site --from server --to server --repository r
--localization-policy l
```

Informacje o dodatkowych opcjach są dostępne pod hasłem [Korzystanie z narzędzia cec opartego na wierszu polecenia](#).

## Transfer serwisu bez elementów zawartości

Będąc programistą, można użyć zestawu OCE Toolkit do przeniesienia — z serwera Oracle Content Management do innego serwera — serwisu bez elementów zawartości.

Aby przenieść serwis z większą liczbą zasobów zawartości, trzeba oddzielić serwis od jego zawartości. Można dodać opcję `--excludecontent` (skrót: `-x`) do polecenia `transfer-site`.

Po ustawieniu tej opcji zostanie przeniesiony tylko serwis.

Na przykład:

```
cec transfer-site Site1 -s DEV -d UAT -r Repository1 -l
LocalizationPolicy1 -x
```

## Grupowe pobieranie lub wysyłanie elementów zawartości serwisu

Będąc programistą, można — transferując serwis z jednego serwera Oracle Content Management na drugi — pobierać lub wysyłać elementy zawartości w grupach.

Aby przenieść serwis z większą liczbą zasobów zawartości, trzeba oddzielić serwis od jego zawartości. Po przetransferowaniu serwisu bez zawartości (`cec transfer-site --excludecontent`) można przetransferować jego zawartość, używając polecenia `cec transfer-site-content`.

Na przykład:

```
cec transfer-site-content <name>
```

W wyniku tego polecenia są tworzone skrypty transferu zawartości serwisu firmowego z jednego serwera Oracle Content Management do innego. To polecenie służy do transferowania dużej liczby elementów zawartości przy użyciu partii. Domyślnie polecenie to nie uruchamia skryptów i są wówczas transferowane wszystkie zasoby. Chcąc przetransferować tylko opublikowane zasoby, należy użyć opcji `-p`.

Serwer źródłowy należy określić za pomocą opcji `-s <server>`, a serwer docelowy — za pomocą opcji `-d <cel>`.

## Opcje

```
--destination, -d The registered CEC server to transfer the content
[required]
--repository, -r Repository [required]
--publishedassets, -p The flag to indicate published assets only
--number, -n The number of items in each batch, defaults to 500
--execute, -e Execute the scripts
--help, -h Show help [boolean]
```

Jeśli nie została ustawiona opcja `--execute`, to — gdy zostanie wykonane polecenie `cec transfer-site-content` — należy uruchomić wygenerowany skrypt `<nazwa_serwisu>_downloadcontent`, aby pobrać zawartość z serwera źródłowego, oraz polecenie `<nazwa_serwisu>_uploadcontent`, aby wysłać pobraną zawartość do serwera docelowego.

## Przykłady

```
cec transfer-site-content Sitel -s DEV -d UAT -r Repository1 Generate script
Sitel_downloadcontent and Sitel_uploadcontent
cec transfer-site-content Sitel -s DEV -d UAT -r Repository1 -e Generate
script Sitel_downloadcontent and Sitel_uploadcontent and execute them
cec transfer-site-content Sitel -s DEV -d UAT -r Repository1 -n 200
cec transfer-site-content Sitel -s DEV -d UAT -r Repository1 -p
```

# Indeksowanie stron serwisu za pomocą zestawu OCE Toolkit

Za pomocą zestawu OCE Toolkit można utworzyć elementy zawartości dla tekstu na stronach serwisu oraz włączyć wyszukiwanie stron dla serwisu.

W następujących tematach opisano, jak indeksować strony serwisu za pomocą zestawu OCE Toolkit:

1. [Tworzenie typu zawartości dla tekstu na stronie serwisu](#)
2. [Tworzenie elementów zawartości indeksu strony za pomocą zestawu OCE Toolkit](#)
3. [Dodawanie wyszukiwania zawartości do serwisu w Oracle Content Management](#)

## Tworzenie typu zawartości dla tekstu na stronie serwisu

Dla typu zawartości należy określić nazwę, wartości wymaganych pól oraz domyślny układ zawartości dla typu zawartości.

- Nazwa typu  
Należy określić dowolną poprawną nazwę typu zawartości.
- Pola  
Poniższe pola są wymagane.

Nazwa pola	Typ pola	Liczba wartości pola danych	Opis
site	Tekstowe	Jedna wartość	Nazwa serwisu
pageid	Tekstowe	Jedna wartość	ID strony

Nazwa pola	Typ pola	Liczba wartości pola danych	Opis
pagename	Tekstowe	Jedna wartość	Nazwa strony
pageurl	Tekstowe	Jedna wartość	URL strony
pagedescription	Tekstowe	Jedna wartość	Opis strony
keywords	Tekstowe	Więcej niż jedna (brak maksimum)	Cały tekst na stronie oraz wartości ze wszystkich pól tekstowych elementów zawartości strony, uzyskane za pomocą polecenia index-site z zestawu OCE Toolkit

```

{{#fields}}
<div class="indextype"></div>
<div>
  <a href="{{pageFullURL}}" title="{{pagename}}">{{pagename}}</a>
</div>
{{/fields}}

```

```

content.fields.pageFullURL = SCSRenderAPI.getSitePrefix() +
content.fields.pageurl;

```

- Utworzyć układ zawartości dla typu.

Układ zawartości powinien wyświetlać nazwę serwisu oraz adres URL prowadzący do strony. Na przykład w pliku `layout.html`:

```

{{#fields}}
<div class="indextype"></div>
<div>
  <a href="{{pageFullURL}}"title="
{{pagename}}">{{pagename}}</a>
</div>
{{/fields}}

```

- W pliku `render.js` wygenerować pełny adres URL strony:

```

content.fields.pageFullURL =
SCSRenderAPI.getSitePrefix() +
content.fields.pageurl;

```

- Ustawić układ zawartości jako domyślny układ zawartości dla typu.

```

content.fields.pageFullURL =
SCSRenderAPI.getSitePrefix() + content.fields.pageurl;

```



## Tworzenie elementów zawartości indeksu strony za pomocą zestawu OCE Toolkit

Za pomocą polecenia z zestawu OCE Toolkit można utworzyć elementy zawartości indeksu strony.

Wymagania wstępne:

- OCE Toolkit został zainstalowany i skonfigurowany w komputerze lokalnym.
- Serwis w usłudze Oracle Content Management został opublikowany.
- Elementy zawartości strony serwisu zostały opublikowane w kanale serwisu.

Za pomocą interfejsu opartego na wierszu polecenia należy wprowadzić następujące polecenie z zestawu OCE Toolkit:

```
cec index-site site name -c content type name -p
```

W poleceniu tym *site name* jest nazwą serwisu, *content type name* jest typem zawartości tworzoną dla tekstu strony, a opcja `-p` zleca opublikowanie elementów zawartości indeksu stron po utworzeniu.

Usage: `cec index-site <site>`

Create content item for each page with all text on the page. If the page index content item already exists for a page, updated it with latest text on the page. Specify `-c <contenttype>` to set the page index content type. Optionally specify `-p` to publish the page index items after creation or update.

Options:

```
--contenttype, -c <contenttype> page index content type
--publish, -p      publish page index items
--help, -h        Show
```

help

[boolean]

Examples:

```
cec index-site Site1 -c PageIndex
cec index-site Site1 -c PageIndex -p
```

Chcąc zobaczyć składnię wywołania, można użyć polecenia `cec index-site -h`

## Dodawanie wyszukiwania zawartości do serwisu w Oracle Content Management

Funkcję wyszukiwania zawartości można dodać do serwisu Oracle Content Management poprzez stronę wyszukiwania i pole wyszukiwania.

Aby dodać do serwisu wyszukiwanie zawartości, należy:

1. [Dodawanie do serwisu strony wyszukiwania](#)

## 2. Dodawanie do motywu pola wyszukiwania

### Dodawanie do serwisu strony wyszukiwania

Można dodać stronę wyszukiwania do serwisu oraz składnik "lista zawartości" do strony wyszukiwania.

Aby dodać stronę wyszukiwania, należy:

1. Dodać stronę do serwisu, po czym ustawić ją jako stronę wyszukiwania.
2. Dodać do strony składnik "lista zawartości".
3. Ustawić **typ zawartości** dla wcześniej utworzonego typu zawartości indeksu strony.

### Dodawanie do motywu pola wyszukiwania

Aby pole wyszukiwania było wyświetlane na każdej stronie serwisu, można dodać pole wyszukiwania do strony HTML układu motywu.

Na przykład:

```
<div align="center">
<input id="searchonpage" type="text" size="30" placeholder="Search on
page. . ."/>
</div>
```

1. Dodać pole danych wejściowych:

```
<script>
  // Get the search field element
  const node = document.getElementById('searchonpage');
  // Get the search string from the url if it exists
  var params = (new URL(document.location)).searchParams;
  var defaultStr = params && params.get('default');
  if (defaultStr) {
    if (defaultStr.lastIndexOf('*') === defaultStr.length - 1) {
      defaultStr = defaultStr.substring(0, defaultStr.length
- 1);
    }
    // Display the search string in the search field
    node.value = defaultStr;
  }
  // When enter from the search field, go to the site search page
  with the search string
  node.addEventListener('keydown', function onEvent(event) {
    if (event.key === "Enter") {
      var inputElem = event.srcElement || event.target;
      var siteSearchPageUrl = 'search.html';
      var searchUrl = SCSRenderAPI.getSitePrefix() +
        siteSearchPageUrl +
        '?contentType=indextype&default=' + inputElem.value
+ '*';
      window.location = searchUrl;
    }
  });
}
```

```
});  
</script>
```

2. Dodać kod JavaScript na końcu treści HTML.

## Indeksowanie serwisu wielojęzycznego za pomocą zestawu OCE Toolkit

Za pomocą zestawu OCE Toolkit można indeksować wielojęzyczne (MLS) serwisy pod kątem tłumaczeń oraz wyszukiwania stron i elementów zawartości.

Przed opublikowaniem serwisu można utworzyć i przetestować indeks serwisu wielojęzycznego. Do poindeksowania serwisu wielojęzycznego służy polecenie `cec index-site` z zestawu OCE Toolkit. Aby wyświetlić informacje Pomocy na temat tego polecenia, należy przejść do katalogu `cec-components`, po czym je uruchomić bez żadnych opcji:

```
cec-components> cec index-site  
Usage: cec index-site <site>
```

Create content item for each page with all text on the page. If the page index content item already exists for a page, updated it with latest text on the page. Specify `-c <contenttype>` to set the page index content type. Optionally specify `-p` to publish the page index items after creation or update.

Options:

```
--contenttype, -c <contenttype> page index content type  
--publish, -p      publish page index items  
--help, -h        Show
```

help

[boolean]

Examples:

```
cec index-site Site1 -c PageIndex  
cec index-site Site1 -c PageIndex -p
```

Elementy indeksu stron istnieją z podziałem na strony i języki. Elementy indeksu stron dla poszczególnych języków są tworzone jako tłumaczenia elementów indeksu stron w języku domyślnym. Jeśli w działającym serwisie uruchomimy zapytanie, funkcje wyszukiwania i listy zawartości wybiorą język z adresu URL strony. W ten sposób wyszukiwanie jest automatycznie filtrowane.

Aby można było opublikować serwis wielojęzyczny, trzeba przedtem go poindeksować i przetłumaczyć — w tym celu jest potrzebne zlecenie tłumaczenia. Zob. [Tworzenie na serwerze Oracle Content Management nowego serwisu lub nowego zlecenia tłumaczenia zasobu](#).

Wymagany jest język domyślny (angielski). Dla każdego obsługiwanego języka (wymaganego i opcjonalnego) należy uruchomić tworzenie indeksu i utworzyć tłumaczenie jego elementów. Jeśli polecenie indeksowania zostanie uruchomione dwa razy, nastąpi aktualizacja indeksu.

Aby poindeksować, przetłumaczyć i opublikować serwisu wielojęzyczny za pomocą poleceń z zestawu OCE Toolkit, należy:

1. Utworzyć typ zawartości dla serwisu, po czym uczynić ten typ dostępnym w repozytorium. Zob. [Tworzenie typu zawartości dla tekstu na stronie serwisu](#).
2. Wybrać założenie systemowe dot. weryfikacji.
  - a. Z lewej nawigacji bocznej wybrać opcję **Zasoby**.
  - b. Z menu **Zasoby** wybrać opcję **Założenia systemowe dot. lokalizacji**.
  - c. Wybrać założenie systemowe dot. lokalizacji.
  - d. Zmodyfikować założenie systemowe dot. lokalizacji (jeśli trzeba), tak aby były uwzględniane języki, które mają być używane do indeksowania i tłumaczenia serwisu. Na przykład, jeśli w założeniu systemowym jest uwzględniany tylko język angielski, można dodać język francuski i hiszpański.
3. Pobrać zlecenie tłumaczenia. Można tłumaczyć tylko zasoby używane w serwisie.
4. [Przetłumaczyć](#) serwis. Serwis można tłumaczyć ręcznie lub poprzez zintegrowany łącznik.
5. Po zakończeniu tłumaczenia wysłać zlecenie tłumaczenia.
6. Aby można było poindeksować serwis, użyć polecenia `cec index-site`. Aby ustawić typ zawartości indeksowanej strony, określić opcję `-c <typ_zawartości>`.

Aby opublikować serwis, można także określić opcję `-p`. Następnie, przed opublikowaniem zmian w aktywnym serwisie, można sprawdzić indeksowanie i tłumaczenie.

Na przykład w wyniku poniższego polecenia `cec index-site` jest tworzony indeks serwisu, w którym są używane języki angielski, francuski i hiszpański. Obsługiwane przez serwis języki (w tym język domyślny) pochodzą z przypisanego założenia systemowego L10n.

```
cec index-site Demo2 -c search_content_type -p
- Logged in to remote server: server-URL
- establish user session
- get CSRF token
- site: Demo2, default language: en-US, channel token: channel-token
- site localization policy: search_localization_policy
- query site repository
- query content type search_content_type
- query site structure
- content types used in the site: search_blog
- query page data
- query content on the pages
- will create 11 page index items
- will update 0 page index items
- will remove 1 page index items
- create page index item for Blog
- create page index item for Privacy Policy
- create page index item for Search
- create page index item for Components
- create page index item for Navigtion
- create page index item for Detail Page
- create page index item for Pages
- create page index item for Page Content
```

- create page index item for Developing Templates
- create page index item for Themes
- add page index items to site channel
- remove page index items for page Search from site channel
- will create/update translate for fr-FR,es-ES
- query site structure with locale fr-FR
- query page data (fr-FR)
- query content on the pages (fr-FR)
- will create 11 page index items (fr-FR)
- will update 0 page index items (fr-FR)
- will remove 1 page index items (fr-FR)
- create page index item for Themes (fr-FR)
- create page index item for Navigation (fr-FR)
- create page index item for Pages (fr-FR)
- create page index item for Detail Page (fr-FR)
- create page index item for Search (fr-FR)
- create page index item for Page Content (fr-FR)
- create page index item for Components (fr-FR)
- create page index item for Developing Templates (fr-FR)
- create page index item for Blog (fr-FR)
- create page index item for Home (fr-FR)
- create page index item for Privacy Policy (fr-FR)
- add page index items to site channel
- set page index items in fr-FR as translated
- remove page index items for page Search from site channel
- query site structure with locale es-ES
- query page data (es-ES)
- query content on the pages (es-ES)
- will create 11 page index items (es-ES)
- will create 0 page index items (es-ES)
- create page index item for Pages (en-ES)
- create page index item for Home (en-ES)
- create page index item for Themes (en-ES)
- create page index item for Components (en-ES)
- create page index item for Privacy Policy (en-ES)
- create page index item for Detail Page (en-ES)
- create page index item for Page Content (en-ES)
- create page index item for Navigation (en-ES)
- create page index item for Developing Templates (en-ES)
- create page index item for Search (en-ES)
- create page index item for Blog (en-ES)
- add page index items to site channel
- set page index items in es-ED as translated
- publish job submitted
- publish in proogress
- publish in progress
- publish page index items finished

7. Opublikować serwis, aby dołączyć do niego tłumaczenia.

## Tworzenie uproszczonego składnika, łatwiejszego do dalszego rozwijania

Za pomocą zestawu OCE Toolkit można utworzyć uproszczony składnik, łatwiejszy do dalszego rozwijania.

Uproszczony składnik `SimpleHTML`, ułatwiający rozpoczęcie pracy nad składnikami niestandardowymi, jest dostępny w zestawie OCE Toolkit:

```
cec create-component -f SimpleHTML
```

Jest także dostępny przykład składnika `JET`, od którego można rozpocząć pracę:

```
cec create-component MyComp -f JET-CCA-Demo-Card
```

## Konfigurowanie usługi kompilacji serwisu

W celu skonfigurowania usługi kompilacji serwisu (w Oracle Content Management) można utworzyć obraz Docker serwera kompilacji. Usługa ta udostępnia opcję definiowania serwera kompilacji, który ma być używany przed opublikowaniem.

Punkt końcowy serwera kompilacji jest określany — jako adres URL — w polu **URL punktu końcowego kompilacji** na stronie **Administrowanie>System>Serwisy i zasoby**. W polu tym można wprowadzić w pełni kwalifikowany URL, który ma zostać zarejestrowany z serwerem, po czym — aby zweryfikować punkt końcowy — można nacisnąć przycisk **Test**. Zob. Ustawianie adresu URL punktu końcowego kompilacji.

Usługa kompilacji serwisu skompiluje strony serwisu — gdy będzie uzyskiwany dostęp do opublikowanego serwisu, będą zwracane strony HTML. Serwer Oracle Content Management, gdy opublikuje serwis, wywoła usługę kompilacji serwisu (jeśli została skonfigurowana) w celu skompilowania stron.

Po określeniu punktu końcowego dla serwisu, można na karcie **Dostarczanie serwisu statycznego** okna dialogowego **Właściwości serwisu** włączyć automatyczną kompilację. Publikowany serwis jest kompilowany przez serwer kompilacji.

Usługa kompilacji serwisu stanowi rozszerzenie poleceń z zestawu Toolkit. Można samodzielnie uruchomić polecenie `cec compilation-server`, lecz obraz Docker umożliwia utworzenie serwera kompilacji w trybie standardowym, a następnie dostrojenie konfiguracji usługi.

Aby utworzyć obraz Docker i opublikować plik Docker, należy:

1. Pobrać z serwisu GitHub informacje o obrazie Docker.  
Pobierając informacje z serwisu GitHub, uzyskuje się trzy dodatkowe pliki, znajdujące się pod serwerem kompilacji. Jednym z obrazów Docker jest serwer kompilacji, zawierający plik Docker i pliki readme.
2. Po utworzeniu obrazu Docker uruchomić polecenie `run.sh`.  
W wyniku tego polecenia jest pobierany i instalowany zestaw OCE Toolkit oraz jest tworzony katalog źródłowy.

3. Można wówczas dostosować środowisko i uruchomić polecenie `run.sh`. Można określić używany port, limity czasu oraz wszelkie inne ustawienia, które mają zostać zmienione, następnie przeprowadzić aktualizację, po czym użyć polecenia `.sh`.
4. Po pobraniu informacji o obrazie Docker i zmienieniu katalogu serwera kompilacji można skonstruować plik Docker.  
Jeśli już został pobrany zestaw OCE Toolkit i nie chcemy niczego ponownie pobierać, można usunąć opcję `no-cache`.
5. Wyszczególnić obraz lub obrazy Docker.
6. Jeśli obraz Docker będzie dostępny, zarejestrować go.
7. Sprawdzić, czy działa.  
Przejsć do pliku wyjątków REST API i zobaczyć, które wersje są obsługiwane. Ten test pozwala sprawdzić, że obraz faktycznie działa.
8. Zarejestrować serwer. Jest wymagana pierwsza część nazwy serwera; wszystko inne w niej zawarte zostanie obsłużone.
9. Wrócić do katalogu "System" i zasobów serwisów, skąd można zarejestrować punkt końcowy kompilacji (nacisnąć przycisk **Zapisz**).
10. Przejsć do właściwości serwisu, po czym określić, co ma być publikowane, po czym go opublikować.

## Kompilowanie serwisu w celu poprawy wydajności stron serwisu w trybie wykonawczym

Kompilacja serwisu w Oracle Content Management może poprawić jego wydajność w trybie wykonawczym oraz przyspieszyć działanie stron serwisu. W wyniku kompilacji jest tworzony statyczny plik HTML dla każdej strony w serwisie, który działa dokładnie jak strona oryginalna.

### Kompilowanie serwisu — omówienie

Składające się na stronę serwisu pliki metadanych są łączone podczas kompilacji; w ten sposób unika się żądań do serwera, które normalnie są wymagane w trybie wykonawczym. Koszt renderowania stron widoku zmienia się w jednorazowy koszt kompilacji.

Przez zmniejszenie liczby żądań w trybie wykonawczym uzyskuje się poprawę wydajności. Między innymi można uniknąć wszystkich zapytań dotyczących zawartości, ponieważ wyniki tych zapytań można skompilować bezpośrednio w stronie.

Uzyskuje się poprawę funkcjonowania, ponieważ strona jest od razu renderowana. Można w ten sposób uniknąć różnych problemów, takich jak chwilowe wyświetlenie zawartości bez zastosowanych stylów (FOUC — flash of unstyled content) lub pojawianie się składników na stronie w przypadkowej kolejności (na przykład najpierw pojawia się stopka, która następnie — w miarę renderowania składników — przesuwa się w dół strony).

Aby skompilować strony serwisu Oracle Content Management, trzeba wyeksportować serwis jako szablon, a następnie — używając narzędzi z zestawu OCE Toolkit — skompilować szablon i wysłać uzyskane strony statyczne do pierwotnego serwisu. W kolejnych podrozdziałach jest omówiony proces kompilacji oraz sposób użycia tego modelu do kompilowania swojego serwisu.

## Interakcja z prerenderem

Zadaniem prerendera jest zwrócenie wyników nadających się dla wyszukiwarek. Od stron wytworzonych przez prerenderer nie oczekuje się, aby działały jak strony oryginalne. Nie jest także wykonywany żaden kod JavaScript. Zwracany jest jedynie wstępnie renderowany kod HTML dla przeglądarek, obsługujący funkcje SEO wyszukiwania tekstów.

Oczekuje się, że strony statyczne, utworzone wskutek kompilacji serwisu, będą działały i zachowywały się w przeglądarce tak samo jak oryginalne strony dynamiczne serwisu. Obejmuje to także dynamiczne uruchamianie nieskompilowanych składników oraz wykonywanie zapytań w trybie wykonawczym, kiedy to użytkownik chce otrzymywać w wynikach dane aktualne, a nie ustalone podczas kompilacji.

Jeśli prerenderer zostanie włączony przez administratora dzierżawy, strony statyczne także mogą być dostarczane przez serwer prerendera. Dzięki temu indeksery i procesy przeszukujące mogą uwzględniać części stron statycznych, które nadal mogą być renderowane dynamicznie (na przykład listy zawartości).

Zapytania dotyczące zawartości zwracają elementy zawierające wartości pól z formatowaniem, a zatem — gdy dla list zawartości są używane układy zawartości — nie trzeba pobierać osobno wartości.

## Renderowanie stron serwisu z użyciem kontrolera

Polecenie `cec compile-template` umożliwia renderowanie stron serwisu bezpośrednio w przeglądarce, bez przechodzenia przez kontroler stron serwisu udostępniany przez Oracle Content Management.

Jeśli nie został użyty kompilator do utworzenia stron statycznych HTML, serwis Oracle Content Management do renderowania stron używa modelu kontrolera. Wiąże się to — zanim strona zostanie wyświetlona w przeglądarce — z pewną liczbą zapytań mających na celu uzyskanie informacji o serwisie, o stronie w serwisie oraz o szablonach używanych do renderowania strony.

### Uwaga:

Dotatkowo są kierowane inne żądania, takie jak zlecające uzyskanie pliku `controller.js`.

## Renderowanie skompilowanych stron serwisu

Korzystając z polecenia `cec compile-template` można skompilować wszystkie te etapy w jeden statyczny plik HTML, co poprawia wydajność serwisu w trybie wykonawczym przez ograniczenie lub wyeliminowanie tych żądań kierowanych do serwera. Ponadto można dalej zredukować żądania dotyczące zasobów, które nie zmieniają się w trybie wykonawczym, kompilując je w strony statyczne.

Domyślny proces kompilacji wykonuje pracę istniejącego kompilatora w trybie wykonawczym, tworząc fizyczną stronę HTML dla każdego z plików `page.json` w serwisie. Stronę taką można następnie wdrożyć w serwisie — zostanie opublikowana i użyta, gdy zostanie wywołany odpowiadający jej adres URL.



## Kompilowanie szablonu

Aby skompilować serwis, wymagany jest dostęp do motywów i składników serwisu. Szablon serwisu to model pakietu dla serwisu. Szablon zawiera wszystkie zasoby wymagane dla migracji i uruchamiania serwisu.

Za pomocą szablonu serwisu można wyeksportować serwis do środowiska OCE Toolkit, w którym można skompilować serwis z szablonu. W poniższych tematach opisano kompilowanie serwisu z szablonu **BlogTemplate**, dostarczanego w zestawie OCE Toolkit.

## Konfigurowanie

Polecenie `cec compile-template` jest dostępne z zestawu OCE Toolkit. Polecenie to jest instalowane wraz z instalacją zestawu. Aby zainstalować zestaw OCE Toolkit, należy postępować zgodnie ze standardowymi instrukcjami.

Po zainstalowaniu zestawu OCE Toolkit można — aby utworzyć szablon — uruchomić polecenie `cec`.

1. Zainstalować katalog programistyczny zestawu OCE Toolkit:

```
mkdir cec-src
cd ./cec-src
cec install

# Start up the CEC Toolkit Development server for testing
cec develop &
```

2. Utworzyć szablon.

Standardowy szablon `BlogTemplate` został zaktualizowany o niestandardowe kompilatory. Utworzyć szablon z tego źródła:

```
# create a new template (must be in the cec-src folder if not already
there)
cd ./cec-src
cec create-template BlogTemplate -f BlogTemplate
```

Standardowy szablon **BlogTemplate** rozszerza poprzedni szablon o niestandardowe kompilatory:

- Kompilatory układów zawartości

```
cec-src/src/components/Starter-Blog-Post-Summary/assets/compile.js
cec-src/src/components/Starter-Blog-Author-Summary/assets/compile.js
cec-src/src/components/Starter-Blog-Post-Content/assets/compile.js
cec-src/src/components/Starter-Blog-Post-Header/assets/compile.js
cec-src/src/components/Starter-Blog-Post-Sidebar/assets/compile.js
```

- Kompilatory układów strony

```
cec-src/src/themes/BlogStarterTheme/layouts/post-detail-compile.js
cec-src/src/themes/BlogStarterTheme/layouts/home-compile.js
cec-src/src/themes/BlogStarterTheme/layouts/authors-compile.js
cec-src/src/themes/BlogStarterTheme/layouts/common-compile.js
```

```
cec-src/src/themes/BlogStarterTheme/layouts/about-compile.js
cec-src/src/themes/BlogStarterTheme/layouts/index-compile.js
```

## Kompilowanie szablonu

W procesie kompilacji szablonu jest dla każdej ze stron zawartych w pliku `structure.json` szablonu serwisu tworzona statyczna strona HTML.

Podczas kompilacji metadane strony (na przykład `100.json`) są łączone z układem strony (na przykład `about.html`), tak że uzyskuje się stronę HTML, która zostanie od razu wyświetlona bez konieczności użycia kontrolera w trybie wykonawczym. Kompilatory niestandardowe pozwalają w jeszcze większym stopniu ograniczyć wykonywanie kodu JavaScript w trybie wykonawczym, kompilując ze stroną elementy uprzednio dynamiczne, takie jak nawigacja czy składniki.

Polecenie `cec compile-template` wykonuje w odniesieniu do każdej ze stron serwisu następujące czynności:

- Wczytanie metadanych serwisu i stron
- Wczytanie układu stron i zastosowanie kompilatora układu stron
- Dla każdego gniazda na stronie:
  - Rozwinięcie gniazda przy użyciu tabeli zdefiniowanej w pliku `page.json`
  - Dla każdego składnika w gnieździe:
    - \* Zastosowanie kompilatora składnika
    - \* Wstawienie wygenerowanego znacznikowego kodu składnika w odpowiednim miejscu w gnieździe
- Rozwinięcie wszelkich makr w skompilowanym kodzie znacznikowym i wstawienie obiektu SCS JavaScript, używanego przez renderer Oracle Content Management w trybie wykonawczym
- Zapisanie skompilowanego kodu znacznikowego strony w folderze `src/templates/<szablon>/static`

Aby skompilować swój serwis, należy uruchomić:

```
> cec compile-template BlogTemplate
Compile Template: compiling template BlogTemplate
Oracle Content Management Site Compiler

createPage: Processing pageId 100. Preview URL:
http://localhost:8085/templates/BlogTemplate/index.html
createPage: Processing pageId 105. Preview URL:
http://localhost:8085/templates/BlogTemplate/post-detail.html
createPage: Processing pageId 401. Preview URL:
http://localhost:8085/templates/BlogTemplate/about.html
createPage: Processing pageId 402. Preview URL:
http://localhost:8085/templates/BlogTemplate/search.html
createPage: Processing pageId 403. Preview URL:
http://localhost:8085/templates/BlogTemplate/authors.html
All page creation calls complete.
```

```
Creating detail pages:
```

```
createPage: Processing detail pageId 105. Preview URL:
http://localhost:8085/templates/BlogTemplate/post-detail/1481786063051-developing-content-layout- for-content-and-experience-cloud
createPage: Processing detail pageId 105. Preview URL:
http://localhost:8085/templates/BlogTemplate/post-detail/1481786063052-dynamic-dom- manipulation-in-content-layout
createPage: Processing detail pageId 105. Preview URL:
http://localhost:8085/templates/BlogTemplate/post-detail/1481786063045-getting-media-url-in- the-content-layout
createPage: Processing detail pageId 105. Preview URL:
http://localhost:8085/templates/BlogTemplate/post-detail/1481786063053-getting-reference- items-in-content-layout
createPage: Processing detail pageId 105. Preview URL:
http://localhost:8085/templates/BlogTemplate/post-detail/1481786063048-navigating-to-a- search-page-with-search-query
createPage: Processing detail pageId 105. Preview URL:
http://localhost:8085/templates/BlogTemplate/post-detail/1481786063050-alex-read
createPage: Processing detail pageId 105. Preview URL:
http://localhost:8085/templates/BlogTemplate/post-detail/1481786063047-jerrold-summers
createPage: Processing detail pageId 105. Preview URL:
http://localhost:8085/templates/BlogTemplate/post-detail/1481786063049-kelly-emerson
createPage: Processing detail pageId 105. Preview URL:
http://localhost:8085/templates/BlogTemplate/post-detail/1481786063043-samantha-howard
createPage: Processing detail pageId 105. Preview URL:
http://localhost:8085/templates/BlogTemplate/post-detail/1481786063046-raising-triggers-from- content-layout
createPage: Processing detail pageId 105. Preview URL:
http://localhost:8085/templates/BlogTemplate/post-detail/1481786063044-rendering-the-content- layout-using-mustache-template
All detail page creation calls complete.
```

## Błędy kompilacji

Podczas kompilacji mogą pojawiać się trzy typy komunikatów: "Informacja", "Ostrzeżenie" i "Błąd".

1. Komunikaty informacyjne dotyczą problemów, o których należy wiedzieć, lecz zazwyczaj komunikaty są oczekiwane jako część procesu kompilacji i obejmują:
  - Elementy zawartości typu wypełniacz, które będą renderowane w trybie wykonawczym.
  - Brakujące kompilatory układów stron. Układy stron mogą nie zawierać żadnego elementu dynamicznego i dlatego ewentualny brak kompilatora nie jest uznawany za problem.
  - Elementy oznaczone jako renderowane przy dostępie, czyli takie, które w trybie wykonawczym mają być renderowane dynamicznie, mimo że strona została skompilowana.
2. Komunikaty ostrzegawcze dotyczą problemów, które prawdopodobnie będą mieć wpływ na wydajność serwisu i które — jeśli to możliwe — należy wyeliminować. Elementy z tej kategorii obejmują:
  - Brak kompilatorów układów stron lub kompilatorów składników niestandardowych. Bez nich składniki będą renderowane dynamicznie na stronie, tak jak poprzednio.

- Brakujące odwzorowania układów zawartości. Jest mało prawdopodobne, że do renderowania elementów zawartości powinien być używany systemowy, domyślny układ zawartości.
3. Komunikaty o błędach dotyczą problemów i sygnalizują błędy kompilacji. Jeśli będzie to możliwe, kompilacja stron będzie kontynuowana, lecz ogólna kompilacja zakończy się błędem.
- Najczęściej przyczyną są błędy kodu JavaScript w kompilatorach niestandardowych. Wszystkie błędy trzeba poprawić.

**Uwaga:**

W celu uniknięcia powtarzania się komunikatów dany komunikat pojawi się tylko raz podczas kompilacji bez względu na to, czy błąd wystąpił na jednej czy wielu stronach.

Dla poprzedniego przykładu został zwrócony następujący status ukończenia kompilacji:

```
Compilation completed with 0 errors and 3 warnings.
to display warnings, run with --verbose (-v) option.
```

Jeśli etap kompilacji zostanie ponownie uruchomiony z opcją "-v", uzyska się następujący wynik:

```
> cec compile-template BlogTemplate --verbose
Compile Template: compiling template BlogTemplate
Oracle Content Management Site Compiler

createPage: Processing pageId 100. Preview URL:
http://localhost:8085/templates/BlogTemplate/index.html
createPage: Processing pageId 105. Preview URL:
http://localhost:8085/templates/BlogTemplate/post-detail.html
Info: no content item specified for placeholder: a890a65c-
c0fc-451b-966b-e606ca18a1f4
component will render at runtime.
Info: no content item specified for placeholder: c90bbc10-
c9d8-4a54-8dd4-7a8251e8efbb
component will render at runtime.
Info: no content item specified for placeholder: f12691e1-79ab-4d1f-
a8b9-3af8c638dd26
component will render at runtime.
createPage: Processing pageId 401. Preview URL:
http://localhost:8085/templates/BlogTemplate/about.html
createPage: Processing pageId 402. Preview URL:
http://localhost:8085/templates/BlogTemplate/search.html
Info: Component: "fdfd0392-e901-48f6-8044-36803c836aa1" of type "scs-
contentlist" marked as
"render on access", will not be compiled.
Info: Component: "ba9f3711-4367-444e-ae38-71289fc10e73" of type "scs-
contentlist" marked as
```

```
"render on access", will not be compiled.
createPage: Processing pageId 403. Preview URL:
http://localhost:8085/templates/BlogTemplate/authors.html
All page creation calls complete.

Creating detail pages:
createPage: Processing detail pageId 105. Preview URL:
http://localhost:8085/templates/BlogTemplate/post-detail/1481786063051-developing-content- layout-for-content-and-experience-cloud
createPage: Processing detail pageId 105. Preview URL:
http://localhost:8085/templates/BlogTemplate/post-detail/1481786063052-dynamic-dom- manipulation-in-content-layout
createPage: Processing detail pageId 105. Preview URL:
http://localhost:8085/templates/BlogTemplate/post-detail/1481786063045-getting-media-url-in- the-content-layout
createPage: Processing detail pageId 105. Preview URL:
http://localhost:8085/templates/BlogTemplate/post-detail/1481786063053-getting-reference- items-in-content-layout
createPage: Processing detail pageId 105. Preview URL:
http://localhost:8085/templates/BlogTemplate/post-detail/1481786063048-navigating-to-a- search-page-with-search-query
createPage: Processing detail pageId 105. Preview URL:
http://localhost:8085/templates/BlogTemplate/post-detail/1481786063050-alex-read
Warning: failed to find content layout map entry for: Starter-Blog-
Author:header. Will compile
using the system default layout.
Warning: failed to find content layout map entry for: Starter-Blog-
Author:content. Will compile
using the system default layout.
Warning: failed to find content layout map entry for: Starter-Blog-
Author:sidebar. Will compile
using the system default layout.
createPage: Processing detail pageId 105. Preview URL:
http://localhost:8085/templates/BlogTemplate/post-detail/1481786063047-jerrold-summers
createPage: Processing detail pageId 105. Preview URL:
http://localhost:8085/templates/BlogTemplate/post-detail/1481786063049-kelly-emerson
createPage: Processing detail pageId 105. Preview URL:
http://localhost:8085/templates/BlogTemplate/post-detail/1481786063043-samantha-howard
createPage: Processing detail pageId 105.
Preview URL: http://localhost:8085/templates/BlogTemplate/post-detail/1481786063046-raising-triggers-from- content-layout
createPage: Processing detail pageId 105. Preview URL:
http://localhost:8085/templates/BlogTemplate/post-detail/1481786063044-rendering-the-content- layout-using-mustache-template
All detail page creation calls complete.

Compilation completed with 0 errors and 3 warnings.

*** compiled template is ready to test
*** to render non-compiled pages, remove compiled files from under: /
private/tmp/cec-
src/src/templates/BlogTemplate/static
```

## Kompilowanie strony szczegółów

Podczas kompilowania strony szczegółów są scalane wszystkie elementy zawartości napotkane w trakcie kompilacji. Następnie jest rekompilowana strona szczegółów dla każdego znalezionej zawartości, przy czym do zdefiniowania adresu URL nowej strony szczegółów jest używana wartość opisowa (slug).

Generowanie wyników kompilacji odbywa się w dwóch etapach:

1. Kompilowanie strony
2. Kompilowanie strony szczegółów

W poprzednim przykładzie są wyświetlane ostrzeżenia o braku wpisów odwzorowań układu zawartości dla strony **Starter-Blog-Author**. My jednak nie chcemy uzyskać stron szczegółów dla strony **Starter-Blog-Author**. Strona szczegółów ma zostać utworzona tylko dla elementów zawartości **Starter-Blog-Post**. Aby usunąć te błędy, można wykluczyć elementy zawartości z kompilacji strony szczegółów, jeśli te elementy w swoich ustawieniach nie zawierają jawnego odwołania do strony szczegółów; służy do tego następująca opcja:

```
--noDefaultDetailPageLink, -o Do not generate compiled detail page for
items/content lists
    that use the default detail page.
```

Po ponownym uruchomieniu kompilacji z wykluczeniem strony szczegółów, przy którym jest używana domyślna strona szczegółów, uzyskuje się następujący wynik:

```
> cec compile-template BlogTemplate --noDefaultDetailPageLink
Compile Template: compiling template BlogTemplate
Oracle Content Management Site Compiler

createPage: Processing pageId 100. Preview URL:
http://localhost:8085/templates/BlogTemplate/index.html
createPage: Processing pageId 105. Preview URL:
http://localhost:8085/templates/BlogTemplate/post-detail.html
createPage: Processing pageId 401. Preview URL:
http://localhost:8085/templates/BlogTemplate/about.html
createPage: Processing pageId 402. Preview URL:
http://localhost:8085/templates/BlogTemplate/search.html
createPage: Processing pageId 403. Preview URL:
http://localhost:8085/templates/BlogTemplate/authors.html
All page creation calls complete.

Creating detail pages:
createPage: Processing detail pageId 105. Preview URL:
http://localhost:8085/templates/BlogTemplate/post-detail/1481786063051-developing-content-layout-for-content-and-experience-cloud
createPage: Processing detail pageId 105. Preview URL:
http://localhost:8085/templates/BlogTemplate/post-detail/1481786063052-dynamic-dom-manipulation-in-content-layout
createPage: Processing detail pageId 105. Preview URL:
http://localhost:8085/templates/BlogTemplate/post-detail/1481786063045-getting-media-url-in-the-content-layout
```

```

createPage: Processing detail pageId 105. Preview URL:
http://localhost:8085/templates/BlogTemplate/post-detail/1481786063053-getting-reference-items-in-content-layout
createPage: Processing detail pageId 105. Preview URL:
http://localhost:8085/templates/BlogTemplate/post-detail/1481786063048-navigating-to-a-search-page-with-search-query
All detail page creation calls complete.

Compilation completed with no errors.

*** compiled template is ready to test
*** to render non-compiled pages, remove compiled files from under: /
private/tmp/cec-
src/src/templates/BlogTemplate/static

```

## Kompilowanie określonych stron

W szablonie nie zawsze trzeba kompilować wszystkie strony. Można wybrać, które strony mają zostać skompilowane. Jest to przydatne, jeśli trwają prace nad określonymi stronami, dla których mają być wykrywane błędy lub które mają być aktualizowane bez konieczności ponawiania kompilacji całego serwisu.

Aby skompilować określone strony, należy użyć opcji `--pages (-p)`, po której następuje lista stron do skompilowania.

### Uwaga:

Uwaga: Jeśli elementy zawartości znajdują się na stronach szczegółów z odwołaniami do listy, to strony szczegółów także zostaną skompilowane, nawet gdy nie są jawnie dołączane.

```

cec compile-template BlogTemplate --pages 401,402
Compile Template: compiling template BlogTemplate
Oracle Content Management Site Compiler

createPage: Processing pageId 401. Preview URL:
http://localhost:8085/templates/BlogTemplate/about.html
createPage: Processing pageId 402. Preview URL:
http://localhost:8085/templates/BlogTemplate/search.html
All page creation calls complete.

Compilation completed with no errors.
*** compiled template is ready to test
*** to render non-compiled pages, remove compiled files from under: /
private/tmp/cec-
src/src/templates/BlogTemplate/static

```

## Kompilowanie serwisu

Na poprzednich etapach pokazano, jak utworzyć i skompilować szablon lokalny. Zazwyczaj będziemy kompilować faktyczne serwisy Oracle Content Management.

Aby skompilować serwis Oracle Content Management, trzeba wyeksportować go do szablonu, a następnie skompilować szablon. W pakiecie szablonu będzie zawarty serwis oraz wszelki kod własny, wymagany do skompilowania serwisu z użyciem motywów i składników eksportowanych z szablonem.

## Kompilowanie serwisu — wymagania wstępne

W poniższych etapach przyjęto założenie, że na serwerze Oracle Content Management został utworzony serwis o nazwie **BlogSite**. Można wysłać poprzedni **BlogTemplate**, a następnie utworzyć ten serwis z tego szablonu.

Na przykład:

```
> cec upload-template BlogTemplate --server UAT

> cec create-site BlogSite --template BlogTemplate --repository
<yourRepository> --
localizationPolicy <yourLocalizationPolicy> --defaultLanguage en-US --
server UAT
```

Po wybraniu lub utworzeniu serwisu można go skompilować serwis, wykonując instrukcje z poniższych sekcji.

## Kompilowanie serwisu

Używając polecenia `cec create-template-from-site` z zestawu OCE Toolkit, można utworzyć szablon serwisu, a następnie skompilować serwis.

```
cec create-template-from-site
```

Options:

```
--site, -s <site> Site to create from [required]
--includeunpublishedassets, -i flag to indicate to include unpublished
content items and digital assets in your template
--server, -r <server> The registered CEC server
--help, -h Show help [boolean]
```

Aby skompilować serwis, należy:

1. Utworzyć szablon z serwisu:

```
> cec create-template-from-site BlogTemplate --site BlogSite --
includeunpublishedassets -
server UAT
```

2. Pobrać szablon:

```
> cec download-template BlogTemplate --server UAT
```



### 3. Skompilować szablon:

```
> cec compile-template BlogTemplate --noDefaultDetailPageLink --verbose --
server UAT
--channelToken e1bb88cdc1e025c8dd278f6b676877a3
```

#### Uwaga:

Trzeba będzie uzyskać token kanału publikowania (opcja `--channelToken (-c)`) dla swojego serwisu na serwerze. Token ten będzie używany dla wszystkich zapytań w obrębie serwisu z szablonu.

### 4. Wysłać skompilowane strony serwisu. Skopiować skompilowane pliki statyczne do folderu statycznego serwisu:

```
> cec upload-static-site-files src/templates/BlogTemplate/static --site
BlogSite --server UAT
```

#### Przywracanie funkcjonowania serwisu bez kompilacji

Aby przywrócić funkcjonowanie serwisu bez kompilacji, trzeba z niego usunąć pliki statyczne, które zostały do niego wysłane:

```
> cec delete-static-site-files BlogSite --server UAT
```

Jeśli serwis został wcześniej opublikowany, trzeba go ponownie opublikować. Zostanie wówczas usunięty z serwisu folder "static" — sam serwis nie zostanie usunięty.

## Kompilatory niestandardowe

Proces kompilacji można usprawnić za pomocą kompilatorów niestandardowych. Kompilatory niestandardowe można wywoływać w celu skompilowania układu strony, układu sekcji, składnika niestandardowego lub układu zawartości ze stroną i uniknięcia dynamicznego dodawania składnika w trybie wykonawczym.

Jeśli dla składnika nie istnieje kompilator niestandardowy lub kompilator składnika nie zwraca kodu znacznikowego, to składnik będzie renderowany w trybie wykonawczym tak, jakby nigdy nie został skompilowany.

Kompilatory niestandardowe są obsługiwane dla następujących składników:

- Układy strony
- Układy sekcji
- Składniki niestandardowe
- Układy zawartości

W podanych dalej przykładach są pokazane kompilatory układu strony i kompilatory układu zawartości. Kompilatory układu sekcji i składników niestandardowych opierają się na tym samym modelu co kompilator układu zawartości.

Jeśli nie istnieje żaden kompilator niestandardowy, składnik jest renderowany z użyciem pliku `render.js`.

Jeśli składnik niestandardowy istnieje, to jest wywoływany, po czym uzyskany kod HTML jest wstawiany do strony. Kompilator niestandardowy może także zasignalizować, że w trybie wykonawczym składnik będzie potrzebował wstawienia (hydratacji) kodu JavaScript. W takim przypadku z pliku `render.js` jest wywoływana funkcja `hydrate()` składnika. Jeśli składnik nie wymaga hydratacji, plik `render.js` składnika nigdy nie jest ładowany.

 **Uwaga:**

Kompilacja jest funkcją tylko dla trybu wykonawczego (opublikowanego serwisu). Jeśli serwis jest wyświetlany w trybie edycji, nawigacji lub podglądu, strony działają jak zwykle i wszystkie składniki są zawsze dodawane do strony w sposób dynamiczny.

## Ograniczenia

Operacja `cec compile-template` jest operacją aplikacji NodeJS wykonywaną poza przeglądarką. Ponieważ strona nie jest renderowana w przeglądarce, nie istnieje żaden obiekt DOM ani okno, a biblioteki JavaScript funkcjonujące po stronie klienta, takie jak JQuery, VueJS lub KnockoutJS, nie będą działać.

Wprawdzie jest możliwe utworzenie obiektu DOM za pomocą biblioteki takiej jak JSDOM i uruchomienie jej po stronie klienta, to jednak nie ma z tego większych korzyści. Do prostej analizy składniowej HTML, można użyć analizatora składni (parser) HTML NodeJS, takiego jak cheerio.

HTML zwracany z kodu znacznikowego musi być poprawnym kodem HTML. Będzie przechodził przez analizator składni i tylko pomyślnie przeanalizowany HTML zostanie dodany do strony. Ma to na celu potwierdzenie, że skompilowany HTML nie będzie zawierał niezgodnych znaczników, które mogłyby się przyczynić do uszkodzenia gniazd.

## Wykrywanie błędów kompilatorów niestandardowych

Podczas tworzenia kompilatorów niestandardowych trzeba uruchomić wykrywanie błędów kodu.

Polecenie `cec compile-template` jest związane z opcją `--debug (-d)`, która uruchamia kompilator z ustawioną flagą `--inspect-brk`, dzięki czemu do procesu może zostać dołączony debugger. Następnie można wykonać standardowe wykrywanie błędów w celu sprawdzenia kodu.

```
cec compile-template BlogTemplate --noDefaultDetailPageLink --debug
```

```
Debugger listening on ws://127.0.0.1:9229/8a8eba83-42d2-476b-adc1-  
b29ab4e92642  
For help see https://nodejs.org/en/docs/inspector
```

Dodatkowo za pomocą opcji `--pages (-p)` można ograniczyć liczbę stron kompilowanych do określonej strony.

## Kompilatory układów strony

Kompilatorem układów stron jest moduł JavaScript NodeJS (CommonJS), który kompiluje odpowiednie układy stron.

Kompilator układów stron dla określonego układu strony jest definiowany poprzez powiązanie nazwy z rozszerzeniem `-compile.js`:

- `src`
  - `themes`
    - \* `<yourTheme>`
      - \* `layouts`
        - \* `<yourPageLayout>.html`
        - \* `<yourPageLayout>-compile.js`

Jeśli rozszerzenie `-compile.js` nie istnieje dla układu strony, nie zostanie zastosowana żadna kompilacja niestandardowa.

Kompilator układów stron musi zaimplementować interfejs `compile()`, który zwraca obietnicę; na przykład `about-compile.js`:

```
var mustache = require('mustache');

var PageCompiler = function () {};

PageCompiler.prototype.compile = function (args) {
  var self = this,
      layoutMarkup = args.layoutMarkup;

  self.SCSCompileAPI = args.SCSCompileAPI;

  return new Promise function (resolve, reject) {
    var compiledPage = layoutMarkup,
        id = self.SCSCompileAPI.navigationRoot;

    // page is compiled so there is no FOUC, can remove the opacity
    workaround
    compiledPage = compiledPage.replace('opacity: 0;', 'opacity: 1;');

    // remove the dynamic menu creation, we'll be compiling it here
    compiledPage = compiledPage.replace('<script
src="_scs_theme_root_/assets/js/topnav.js"></script>', '');

    // add link to Home page. . .
    var homePageURL = (self.SCSCompileAPI.getPageLinkData(id) ||
{}).href;
    if (homePageURL) {
      compiledPage = compiledPage.replace('class="navbar-brand"
href="#"', 'class="navbar-brand" href="' + homePageURL + '"');
    }

    // build the menu and add it to the page
```

```

        var navMenu = self.createNavMenu();
        compiledPage = compiledPage.replace('<!-- navigation menu goes
in
here -->', navMenu);

        // return the compiled page
        resolve(compiledPage);
    });
};

// Create the navigation menu that was previously dynamically generated
on
each page
PageCompiler.prototype.createNavMenu = function () {
    . . .
}

module.exports = new PageCompiler();

```

## Kompilatory składników

Wszystkie niestandardowe kompilatory składników opierają się na tym samym modelu co kompilatory stron i mogą być tworzone dla układów sekcji, składników niestandardowych oraz układów zawartości.

W trakcie kompilacji polecenie `cec compile-template` wyszukuje plik `compile.js` w tej samej lokalizacji, w `render.js` dla składnika:

- `src`
  - `components`
    - \* `<składnik>`
      - \* `assets`
        - \* `render.js`
        - \* **`compile.js`**

Jeśli ten plik nie istnieje, składnik nie jest kompilowany i będzie renderowany w trybie wykonawczym.

Jeśli ten plik istnieje, musi implementować interfejs `compile()` zwracający obietnicę. Na przykład poniższy kompilator `Starter-Blog-Author-Summary` jest niestandardowym kompilatorem układu zawartości:

```

var fs = require('fs'),
    path = require('path'),
    mustache = require('mustache');

var ContentLayout = function (params) {
    this.contentClient = params.contentClient;
    this.contentItemData = params.contentItemData || {};
    this.scsData = params.scsData;
};

ContentLayout.prototype = {

```

```

contentVersion: '>=1.0.0 <2.0.0',

compile: function () {
  var compiledContent = '',
      content = JSON.parse(JSON.stringify(this.contentItemData)),
      contentClient = this.contentClient;

  // Store the id
  content.fields.author_id = content.id;

  if (this.scsData) {
    content.scsData = this.scsData;
    contentType = content.scsData.showPublishedContent === true ?
'published' : 'draft';
    secureContent = content.scsData.secureContent;
  }

  // calculate the hydrate data
  content.hydrateData = JSON.stringify({
    contentId: content.id,
    authorName: content.fields['starter-blog-author_name']
  });

  try {
    // add in style - possible to add to <head> but inline for
simplicity
    var templateStyle = fs.readFileSync(path.join(__dirname,
'design.css'), 'utf8');
    content.style = '<style>' + templateStyle + '</style>';

    var templateHtml = fs.readFileSync(path.join(__dirname,
'layout.html'), 'utf8');
    compiledContent = mustache.render(templateHtml, content);
  } catch (e) {
    console.error(e.stack);
  }

  return Promise.resolve({
    content: compiledContent,
    hydrate: true // note that we want to hydrate this component
using the render.js hydrate() function. This is required for when the user
clicks on the author
  });
}
};

module.exports = ContentLayout;

```

## Kompilatory układów zawartości

Kompilatory układów zawartości są używane do generowania kodu HTML układu zawartości, gdy jest publikowany element zawartości o określonym typie zawartości.

W trakcie kompilacji polecenie `cec compile-content` wyszukuje plik `compile.js` w tej samej lokalizacji, w której znajduje się plik `render.js` dla składnika:

- src
  - components
    - \* <składnik>
      - \* assets
        - \* render.js
        - \* **compile.js**

Jeśli ten plik nie istnieje, układ nie jest kompilowany i będzie renderowany w trybie wykonawczym.

Jeśli ten plik istnieje, musi implementować interfejs `compile()` zwracający obietnicę. Poniżej są pokazane treść kompilatora układu i uzyskany wynikowy plik `layout.html`:

```
var fs = require('fs'),
    path = require('path'),
    mustache = require('mustache');

var ContentLayout = function (params) {
  this.contentClient = params.contentClient;
  this.contentItemData = params.contentItemData || {};
  this.scsData = params.scsData;
};

ContentLayout.prototype = {
  contentVersion: '>=1.0.0 <2.0.0',

  compile: function () {
    var compiledContent = '',
        content = JSON.parse(JSON.stringify(this.contentItemData)),
        contentClient = this.contentClient;

    // Store the id
    content.fields.author_id = content.id;

    if (this.scsData) {
      content.scsData = this.scsData;
    }

    try {
      // add in style - possibly add to head but inline for
      // simplicity
      var templateStyle = fs.readFileSync(path.join(__dirname,
        'designn.css'), 'utf8');
      content.style = '<style>' + templateStyle + '</style>';

      var templateHtml = fs.readFileSync(path.join(__dirname,
        'layout.html'), 'utf8');
      compiledContent = mustache.render(templateHtml, content);
    } catch (e) {
      console.error(e.stack);
    }

    return Promise.resolve({
```

```

        content: compiledContent,
        hydrate: true
    });
}
};

```

```
module.exports = ContentLayout;
```

Uzyskany plik `layout.html` można wstawić do aplikacji internetowej lub w dowolnym innym miejscu, w którym jest potrzebny statyczny kod HTML układu zawartości. Plik `layout.html` jest generowany w tym samym katalogu, w którym znajduje się plik `compile.js`.

```

{{{style}}}
{#{fields}}
<div class="author-container">
    <span class="author-name" onclick='{{{scsData.contentTriggerFunction}}}
("starter-blog-post_author eq \"{{author_id}}\")'>{{{starter-blog-
author_name}}}</span>
{/{fields}}

```

## SCSCompileAPI

Podobnie jak `SCSRenderAPI`, istnieje `SCSCompileAPI`, który jest przekazywany do każdej funkcji kompilacji.

Zawiera poniższe właściwości i funkcje.

- Właściwości:
  - **navigationRoot**: ID węzła będącego głównym węzłem serwisu.
  - **navigationCurr**: ID węzła bieżącej strony.
  - **structureMap**: Wszystkie węzły hierarchicznej struktury serwisu; dostęp do nich uzyskuje się za pomocą ID.
  - **siteInfo**: Wszystkie właściwości serwisu.
- Funkcje:
  - **getContentClient**: Uzyskuje instancję `contentClient`, która będzie używana dla wywołań Content API.
  - **getCustomSiteProperty**: Uzyskuje właściwości niestandardowe w trakcie kompilowania serwisu. Dane i właściwości niestandardowe, służące do dostosowania serwisu, można przechowywać w pliku `siteinfo.json` na poziomie poziomu serwisu. Na przykład można zmienić teksturę tła strony, łączy stopki lub nazwę serwisu.

## Hydratacja składników

Terminem "hydratacja" (ang. hydration) jest określany proces dodawania kodu JavaScript do skompilowanego kodu HTML na stronie, gdy HTML jest renderowany w przeglądarce.

Na przykład, jeśli na stronie znajdują się dwa składniki, które mają być renderowane jako nadrzędny i podrzędny, to kliknięcie na elemencie w nadrzędnym wymaga aktualizacji składnika podrzędnego. Wszystko to jest obsługiwane przez kod JavaScript wykonywany na stronie. Aby to działało, trzeba po wyrenderowaniu obu tych składników *hydratyzować* ich

kod HTML, dodając do elementów w składniku nadrzędnym procedurę obsługi zdarzenia `on click` oraz dodając nasłuch do składnika podrzędnego, aby był on ponownie renderowany (na podstawie ładunku przekazanego w zdarzeniu), gdy wystąpi zdarzenie `on click`.

Kod HTML jest wstawiany do strony przez kompilatory składników. Jeśli składnik wymaga dodatkowego kodu JavaScript, który ma zostać wykonany w trybie wykonawczym w celu dodania np. procedur obsługi zdarzeń, jest dostępnych kilka możliwości, w tym użycie wstawkowego kodu JavaScript lub funkcji hydratacji. Wybór rozwiązania zależy od wymagań użytkownika.

## Wprowadzanie kodu JavaScript jako wstawkowego

Znacznik `<script>` można wstawić bezpośrednio w zwróconym, skompilowanym kodzie znacznikowym. Skrypt zostanie wykonany przy uruchomieniu strony.

Na przykład:

```
<script src="/_sitesclouddelivery/renderer/libs/scs-core/jssor-
slider/js/jssor.slider.min.js" type="text/javascript"></script>
<div id="slider_container_c46b122d-978a-429d-aa25-9b5698428f6f"
style="position: relative; top: 0px; left: 0px; height: 400px; width:
600px;
background-color: rgb(68, 68, 68); visibility: visible;" data-jssor-
slider="1">
. . .
</div>
<script>
    (function () {
        // get the required options
        var options =
        {"$FillMode":2,"$AutoPlay":false,"$AutoPlayInterval":3000,"$SlideDuratio
n":50
0,"$ArrowKeyNavigation":true,"$HWA":false,"$BulletNavigatorOptions":
{"$Chance
ToShow":1,"$AutoCenter":1,"$SpacingX":5},"$ArrowNavigatorOptions":
{"$ChanceTo
Show":1,"$AutoCenter":2,"$Steps":1},"$ThumbnailNavigatorOptions":
{"$ChanceToS
how":0,"$DisplayPieces":7,"$SpacingX":8,"$ParkingPosition":240}};

        // select the JSSOR value options
        options.$BulletNavigatorOptions.$Class
= $JssorBulletNavigator$;
        options.$ArrowNavigatorOptions.$Class
= $JssorArrowNavigator$;
        options.$ThumbnailNavigatorOptions.$Class =
$JssorThumbnailNavigator$;

        // create the slider
        var slider = new $JssorSlider$
("slider_container_c46b122d-978a-
429d-aa25-9b5698428f6f", options);

        // resize, maintaining aspect ratio
```



```

    var container = slider.$Elmt.parentElement;
    if (container) {
        slider.$ScaleWidth(container.getBoundingClientRect().width);
    }
  })();
</script>

```

## Funkcja hydratacji

Zamiast używać kodu JavaScript w postaci wstawkowej, można w swoim pliku `render.js` zawrzeć funkcję *hydratacji* w trybie wykonawczym, gdy jest zwracany skompilowany kod znacznikowy. W ten sposób unika się powtarzania znaczników `<script>` i umożliwia wykorzystanie istniejącego kodu JavaScript do zarządzania zdarzeniami.

Nawet jeśli plik `render.js` zostanie załadowany, to jednak funkcja `render()` *nie* jest wywoływana podczas hydratacji. Wywoływana jest tylko funkcja `hydrate()`.

### Uwaga:

Jeśli skompilowany składnik nie informuje, że wymaga hydratacji, plik `render.js` skł adnika nigdy nie jest ładowany.

Na przykład niestandardowy kompilator układu zawartości zwraca - `{ hydrate: true }`.

```

return Promise.resolve({
  content: compiledContent,

  hydrate: true // note that we want to hydrate this component using the
render.js hydrate() function. This is required for when the user clicks on
the author
});

```

Ponadto, jeśli jest to wymagane, kompilator niestandardowy może dodać właściwości *hydrate*, które będzie wyszukiwał w trybie wykonawczym. Na przykład:

```

Compiler:
  // calculate the hydrate data
  content.hydrateData = JSON.stringify({
    contentId: content.id,
    authorName: content.fields['starter-blog-author_name']
  });
  . . .

```

```

Template:
<div class="author-container" data-hydrate="{{hydrateData}}">

```

Na koniec, jeśli składnik poinformuje, że wymaga hydratacji, to w trybie wykonawczym zostanie załadowany plik `render.js` składnika i zostanie wywołana funkcja `hydrate()`, przekazując pojemnik `<div>` zawierający skompilowany kod znacznikowy.

Na przykład `render.js` z funkcją `hydrate()`:

```
function selectAuthor(parentObj, contentId, authorName) {
    var $parentObj = $(parentObj);
    $parentObj.find(".author-name").click($.proxy(function () {
        $(".author-name").removeClass('author-selected');
        $(event.target).addClass('author-selected');
    }, self));
    if (window.location.href.indexOf("default=" + contentId) >= 0) {
        $(".author-name").each(function () {
            if (this.innerText === authorName) {
                $(this).addClass('author-selected');
            }
        });
    }
}

...

hydrate: function (parentObj) {
    var $parentObj = $(parentObj),
        hydrateData = $parentObj.find('.author-
container').attr('data-
hydrate');
    if (hydrateData) {
        var data = JSON.parse(hydrateData);
        selectAuthor(parentObj, data.contentId, data.authorName);
    }
},
render: function (parentObj) {
    ...
    try {
        // Mustache
        template = Mustache.render(templateHtml, content);

        if (template) {
            $(parentObj).append(template);
        }
        selectAuthor(parentObj, this.contentItemData.id,
content.fields['starter-blog-author_name']);
    } catch (e) {
        console.error(e.stack);
    }
}
}
```

## Publikowanie

Po skompilowaniu stron statycznych i wysłaniu ich do folderu "static" serwisu trzeba — aby je uaktywnić — opublikować lub ponownie opublikować serwis. Analogicznie, aby przywrócić funkcjonowanie serwisu z nieskompilowanymi stronami, trzeba z niego usunąć pliki statyczne, po czym go opublikować lub ponownie opublikować.

Podczas publikowania następuje przygotowanie stron statycznych (które zostały wysłane) do dostarczenia. Ponieważ pliki te są kopiowane w trakcie procesu

publikowania, wydajność operacji publikowania może się pogorszyć proporcjonalnie do liczby plików.

Operacja publikowania przyjmuje bieżący zestaw plików statycznych i udostępnia je do dostarczania. Pliki te mogą (lecz nie muszą) być zsynchronizowane ze zmianami, które nastąpiły w serwisie dynamicznym, i mogą (lecz nie muszą) odzwierciedlać serwis dynamiczny. Aktualizacja kolekcji plików statycznych w odpowiednim czasie jest zadaniem twórcy serwisu.

## Dostarczanie serwisu statycznego — kolejność

Jeśli z serwisem są powiązane pliki statyczne, są one dostarczane dla odpowiednich adresów URL przychodzących do serwera. Jeśli przychodzący URL nie odpowiada plikowi statycznemu, to dla żądania jest zwracany plik `controller.html` serwisu. Jest to zgodne z istniejącym dynamicznym modelem dostarczania serwisu.

Przez serwisy Oracle Content Management mogą być także — za pomocą powiązanego pliku JSON — definiowane przekierowania 301 i 302. Jeśli przekierowania zostaną zdefiniowane, to będą mieć pierwszeństwo przed plikami statycznymi. Jeśli URL jest zgodny jednocześnie z regułą przekierowania i plikiem statycznym, to z serwera zostanie dostarczone przekierowanie.

Ocena adresu URL przy dostarczaniu serwisu jest dokonywana w następujący sposób:

1. Czy URL jest zgodny ze skonfigurowanym przekierowaniem?  
Jeśli tak, jest wydawana odpowiedź przekierowania.
2. Czy URL odpowiada plikowi statycznemu?  
Jeśli dla serwisu został skonfigurowana lista mobilnych agentów statycznych użytkownika i żądanie przychodzi z przeglądarki uwzględnionej na tej liście, jest dostarczany mobilny plik statyczny.
3. W przeciwnym razie jest dostarczany plik `controller.html` serwisu dynamicznego.

### Uwaga:

Jeśli mobilne pliki statyczne są powiązane z serwisem i klient do dostarczania używa sieci CDN, to CDN (zazwyczaj Akamai) trzeba skonfigurować tak, aby żądania z przeglądarek mobilnych były przechowywane w pamięci podręcznej osobno od standardowych żądań z przeglądarek typu Desktop.

Jeśli CDN nie zostanie skonfigurowana z osobnym buforowaniem żądań mobilnych i standardowych, to przeglądarki mobilne mogą otrzymywać odpowiedzi standardowe, a przeglądarki typu Desktop — odpowiedzi przeznaczone dla przeglądarek mobilnych.

## Buforowanie nagłówków

Nagłówki HTTP, zawarte w odpowiedziach z serwerów internetowych, pomagają ustalić sposób przechowywania stron w pamięci w podręcznej. Strony statyczne są także dostarczane z odpowiednimi nagłówkami, ułatwiającymi buforowanie przez przeglądarki.

W przypadku zabezpieczonych serwisów w odpowiedziach będą wysyłane następujące nagłówki:

- Cache-Control: no-store
- Pragma: no-cache

W przypadku standardowych, niezabezpieczonych serwisów będą wysyłane następujące nagłówki:

- Cache-Control: max-age=300
- Edge-Control: !no-store,max-age=2592000,downstream-ttl=1800  
Nagłówek Edge-Control pomaga określić buforowanie w sieci CDN.

Jeśli w jednym z tych dwóch obszarów nagłówki zostały dostosowane, odpowiedź będzie zawierać nie standardowe nagłówki (tu wymienione), ale nagłówki dostosowane.

Odpowiedzi te można kontrolować na poziomie dzierżawy lub serwisu.

## Strony szczegółów

Strony szczegółów, dostępne w serwisach Oracle Content Management, umożliwiają pokazywanie informacji dotyczących elementów zawartości.

Ta sama strona szczegółów może być używana do obsługi wielu adresów URL. Każdy z tych adresów URL będzie wyświetlał tę samą strukturę strony, lecz będzie pokazywać zawartość związaną z elementami zawartości, których wartości opisowe to odpowiednio `item1.html`, `item2.html` i `item3.html`. W takiej sytuacji kompilator szablonu cec mógłby utworzyć cztery pliki:

- `/detail/item1.html`
- `/detail/item2.html`
- `/detail/item3.html`
- `/detail.html`

Finalny plik umożliwia wyświetlanie (w serwisie internetowym) nowo opublikowanego materiału bez konieczności ponownej kompilacji i ponownego publikowania serwisu. W tym przykładzie element zawartości z wartością opisową `item4.html` jest publikowany po przełączeniu serwisu do trybu online. Statyczna strona `/detail.html` umożliwia dynamiczne wyświetlanie nowego elementu w serwisie. URL `/detail/item4.html` dostarczy stronę `detail.html`, lecz będzie na niej pokazywana zawartość związana z elementem zawartości `item4.html`.

Strona `detail.html` do wyświetlania zawartości jest generowana przez kompilator cec. Z tego powodu względne adresy URL, zawarte w skompilowanej stronie `detail.html`, będą mieć dodatkowe segmenty nadrzędne (`./`). A zatem strona `detail.html`, jeśli wystąpi do niej bezpośrednie odwołanie, nie zostanie wyświetlona poprawnie. Dlatego nie należy odwoływać się do strony `detail.html` ani dodawać jej do nawigacji po stronach.

## Dodawanie elementów zawartości do kanału

Używając polecenia `control-content` z zestawu OCE Toolkit, można dodawać elementy zawartości do kanału w serwerze Oracle Content Management.

Polecenie `control-content <czynność>` zawiera czynność `add` dodającą elementy zawartości do kanału Oracle Content Management:

```
cec control-content add -c Channel1 -r Repo1 -s UAT
```

Polecenie to dodaje wszystkie elementy z repozytorium `Repo1` do kanału `Channel1` w zarejestrowanym serwerze "UAT".

Można określić serwer za pomocą opcji `-s <serwer>` lub użyć serwera określonego w pliku `cec.properties`.

W poleceniu `content-usage` można użyć następujących czynności:

- `publish`
- `unpublish`
- `add`
- `remove`

Dostępne są następujące polecenia `content-usage`:

- `--channel, -c kanał [wymagane]`
- `--repository, -r repozytorium [wymagane, gdy <czynność> to add]`
- `--server, -s zarejestrowany serwer Oracle Content Management`
- `--help, -h wyświetlenie pomocy [wartość logiczna]`

Przykłady polecenia `control-content`:

```
cec control-content publish -c Channel1
```

Publish all items in channel `Channel1` on the server specified in the `cec.properties` file

```
cec control-content publish -c Channel1 -s UAT
```

Publish all items in channel `Channel1` on the registered server `UAT`

```
cec control-content unpublish -c Channel1 -s UAT
```

Unpublish all items in channel `Channel1` on the registered server `UAT`

```
cec control-content add -c Channel1 -r Repo1 -s UAT
```

Add all items in repository `Repo1` to channel `Channel1` on the registered server `UAT`.

```
cec control-content remove -c Channel1 -s UAT
```

Remove all items in channel `Channel1` on the registered server `UAT`

## Kompilowanie serwisu dla urządzeń mobilnych

Korzystając z zestawu OCE Toolkit, można skompilować układ mobilny internetowej strony serwisu. Układ mobilny może dla tej samej zawartości różnić się od układu strony typu Desktop. Układy te mogą także być identyczne.

W edytorze serwisów można wybrać dla urządzeń mobilnych ten sam układ strony, co dla urządzeń typu Desktop, bądź określić inny. Za pomocą zestawu OCE Toolkit można osobno skompilować układ statyczny dla urządzeń mobilnych.

Strona serwisu może być wyświetlana inaczej w urządzeniu mobilnym. Przykładowo może nie mieć banera wyświetlanego w układzie typu Desktop.

Wyświetlana w zestawie OCE Toolkit strona Pomocy dla polecenia `cec compile-template` pokazuje opcję `targetDevice` pozwalającą wskazać konkretne urządzenie przy kompilowaniu szablonu serwisu:

```
C:\git\webclient\developer\test\sites-compiler\cec-install>cec compile-template --help
Usage: cec compile-template <source>

Compiles all the pages within the site of the template and places the compiled pages under the sites assets folder.
Optionally specify -s <server> to make content queries against this server (requires channelToken).
Optionally specify -c <channelToken> to use this channelToken when generating any content URLs.
Optionally specify -t <contentType> [draft | published] content to retrieve from the server type, defaults to published.
Optionally specify -p <pages> the set of pages to compile. --inspect-brk flag.
Optionally specify -d <debug> to start the compilation with --inspect-brk flag.
Optionally specify -r <recurse> recurse through all child pages of specified pages.
Optionally specify -l <includeLocale> include default locale when creating pages.
Optionally specify -a <targetDevice> [desktop | mobile] target device type when using adaptive layouts.
Optionally specify -v <verbose> to display all warning messages during compilation.

Options:
--server, -s           The registered CEC server
--channelToken, -c    The channel access token to use for content URLs
--type, -t            The type of content to retrieve from the server [published | draft]
--pages, -p          The list of pages to compile
--recurse, -r        Compile all child pages of those specified in the page list
--debug, -d          Start the compiler with "--inspect-brk" option to debug compilation
--noDetailPages, -e  Do not generate compiled detail pages
--noDefaultDetailPageLink, -o Do not generate compiled detail page for items/content lists that use the default detail page
--targetDevice, -a   The target device type when using adaptive layouts [desktop | mobile]
--includeLocale, -l Include default locale when creating pages
--verbose, -v        Run in verbose mode to display all warning messages during compilation.
--help, -h          Show help

Examples:
cec compile-template Temp1                               Compiles the site in template Temp1 using content stored in the template.
cec compile-template Temp1 -c channelToken              Compiles the site in template Temp1 using the given channelToken for any content URLs.
cec compile-template Temp1 -c channelToken -s UAT -t draft Compiles the site in template Temp1 retrieving draft content from the specified server.
cec compile-template Temp1 -p 104,112,183 -r            Compiles the specified pages in the site in template Temp1 including all child pages.
cec compile-template Temp1 -d                          Waits for the debugger to be attached. Once attached, compiles the site in template Temp1.

C:\git\webclient\developer\test\sites-compiler\cec-install>
```

Kompilując serwis, można określić, czy ma być on kompilowany dla urządzeń typu Desktop czy urządzeń mobilnych. Pliki typu Desktop są umieszczane w folderze `static/_files`. Pliki mobilne są umieszczane w folderze `static/_mobilefiles`.

```

C:\git\webclient\developer\test\sites-compiler\cec-install>cec compile-template Corporate-Site-Template --targetDevice desktop
Compile Template: compiling template Corporate-Site-Template
Oracle Content and Experience Site Compiler

Compiling: desktop pages
-----
createPage: Processing pageId 10. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/index.html
createPage: Processing pageId 100. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/developing-templates.html
createPage: Processing pageId 110. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/developing-templates/themes.html
createPage: Processing pageId 120. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/developing-templates/pages.html
createPage: Processing pageId 130. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/developing-templates/navigation.html
createPage: Processing pageId 140. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/developing-templates/page-content.html
createPage: Processing pageId 150. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/developing-templates/components.html
createPage: Processing pageId 160. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/developing-templates/sign-in.html
createPage: Processing pageId 200. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/privacy-policy.html
All page creation calls complete.

Compilation completed with 0 errors and 5 warnings.
to display warnings, run with --verbose (-v) option.

*** compiled template is ready to test
*** to render non-compiled pages, remove compiled files from under: C:\git\webclient\developer\test\sites-compiler\cec-install\src\templates\Corporate-Site-Template\static

C:\git\webclient\developer\test\sites-compiler\cec-install>cec compile-template Corporate-Site-Template --targetDevice mobile
Compile Template: compiling template Corporate-Site-Template
Oracle Content and Experience Site Compiler

Compiling: mobile pages
-----
createPage: Processing pageId 10. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/index.html
createPage: Processing pageId 100. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/developing-templates.html
createPage: Processing pageId 110. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/developing-templates/themes.html
createPage: Processing pageId 120. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/developing-templates/pages.html
createPage: Processing pageId 130. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/developing-templates/navigation.html
createPage: Processing pageId 140. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/developing-templates/page-content.html
createPage: Processing pageId 150. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/developing-templates/components.html
createPage: Processing pageId 160. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/developing-templates/sign-in.html
createPage: Processing pageId 200. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/privacy-policy.html
All page creation calls complete.

Compilation completed with 0 errors and 6 warnings.
to display warnings, run with --verbose (-v) option.

*** compiled template is ready to test
*** to render non-compiled pages, remove compiled files from under: C:\git\webclient\developer\test\sites-compiler\cec-install\src\templates\Corporate-Site-Template\static

```

Gdy szablon dla urządzeń mobilnych zostanie skompilowany, polecenie `upload-static-site-files` z zestawu OCE Toolkit będzie obsługiwać pliki mobilne.

## Cykl życia serwisu oraz skompilowane strony

Gdy ze strony jest tworzony szablon, nie są w nim uwzględniane skompilowane strony. Ma to na celu uniknięcie problemu, kiedy to są dostarczane strony statyczne, podczas gdy programista serwisu oczekuje na dostarczenie stron dynamicznych.

Jeśli następnie zostanie utworzony serwis z szablonu, trzeba skompilować strony serwisu, a następnie wysłać je do nowego serwisu.

## Tworzenie na serwerze Oracle Content Management nowego serwisu lub nowego zlecenia tłumaczenia zasobu

Za pomocą zestawu OCE Toolkit można utworzyć zlecenie tłumaczenia serwisu lub zasobu w Oracle Content Management.

Zanim będzie można poindeksować serwis wielojęzyczny, trzeba utworzyć zlecenie tłumaczenia. Aby utworzyć zlecenie tłumaczenia, należy:

1. Z górnego menu na stronie "Serwisy" wybrać opcję **Tłumacz**.
2. W oknie dialogowym **Tworzenie zlecenia tłumaczenia** podać nazwę zlecenia, po czym wybrać domyślny język źródłowy, języki docelowe oraz zawartość zlecenia tłumaczenia.

Można wybrać, czy pakiet tłumaczeń ma zawierać całą zawartość serwisu i wszystkie zasoby docelowe, tylko zawartość serwisu czy tylko zasoby docelowe dla kanału publikowania serwisu.

Z tłumaczenia można wykluczyć dowolne elementy zawartości, które zostały skonfigurowane przy użyciu ustawienia **Nie tłumacz**. Na przykład nazwy produktów są zazwyczaj nie tłumaczone.

3. Aby utworzyć zlecenie tłumaczenia, nacisnąć przycisk **Utwórz**.
4. Aby wyświetlić listę dostępnych zleceń, użyć polecenia z zestawu OCE Toolkit:

```
cec components> cec list-translation-jobs
Asset translation jobs:
Name                               Status
Source Language Target Languages    Pending
Languages

Site translation jobs:
Name                               Status
Source Language Target Languages    Pending
Languages

demo1                               READY      en-
US          fr-FR,es-ES           fr-FR,es-ES

searchdemo1                         TRANSLATED en-
US          fr-FR,es-ES
```

5. Pobrać zlecenie tłumaczenia:

```
cec components> cec download-translation-job demo1
- translation job downloaded to /Users/<user-name>/Dev/webclient/
  developers/sites-toolkit/cec-components/demo.zip
- update the translation job status to INPROGRESS.
cec components> cec translate dmo1.zip -l all -t demo1-xlate.zip
- target languages: fr-FR,ex-ES
- translation finished: /Users/<user-name>/Dev/webclient/
  developers/sites-toolkit/cec-components/demo1-xlate.zip
```

6. Otworzyć pakiet tłumaczeń, po czym utworzyć foldery zasobów dla języków, dla których jest wykonywane tłumaczenie:

```
Unzip demo1-xlate.zip
ARchive: emol-xlate.zip
replace assets/job.json? [n]o, [A]ll, [N]one, [r]ename: A
  inflating assets/job.json
  inflating site/job.json
  inflating assets/es-ES/CORE47653001483240C1AAF180C435F189AB-
  search_siteSearch202.json
  inflating assets/es-ES/COREA570227E12194356BAA16A80A78A2670-
  entry1.json
  inflating assets/es-ES/CORED977BC199A3B494596F0D467CAADF7FA-
  entry2.json
  inflating assets/fr-FR/CORE47653001483240C1AAF18DC435F1B9A8-
  search_siteSearch202.json
  inflating assets/fr-FR/COREA570227E12194356BAA16A80A78A2670-
  entry1.json
  inflating assets/fr-FR/CORED977BC199A3B494596F0D467CA4DF7FA-
  entry2.json
  inflating assets/root/CORE476530014B3240C1AAF18DC435F1B948-
  search_siteSearch202.json
  inflating assets/root/COREA570227E12194356BAA16A80A7842870-
```



```

entry1.json
  inflating assets/root/CORED977BC199A38494596F0D467CA4DF7FA-entry2.json
  inflating site/es-ES/10.json
  inflating site/es-ES/100.json
  inflating site/es-ES/110.json
  inflating site/es-ES/120.json
  inflating site/es-ES/130.json
  inflating site/es-ES/140.json
  inflating site/es-ES/150.json
  inflating site/es-ES/200.json
  inflating site/es-ES/201.json
  inflating site/es-ES/202.json
  inflating site/es-ES/203.json
  inflating site/es-ES/siteinfo.json
  inflating site/es-ES/structure.json
  inflating site/fr-FR/10.json
  inflating site/fr-FR/100.json
  inflating site/fr-FR/110.json
  inflating site/fr-FR/120.json
  inflating site/fr-FR/130.json
  inflating site/fr-FR/140.json
  inflating site/fr-FR/150.json
  inflating site/fr-FR/200.json
  inflating site/fr-FR/201.json
  inflating site/fr-FR/202.json
  inflating site/fr-FR/203.json
  inflating site/fr-FR/siteinfo.json
  inflating site/fr-FR/structure.json
  inflating site/root/10.json
  inflating site/root/100.json
  inflating site/root/110.json
  inflating site/root/120.json
  inflating site/root/130.json
  inflating site/root/140.json
  inflating site/root/150.json
  inflating site/root/200.json
  inflating site/root/201.json
  inflating site/root/202.json
  inflating site/root/203.json
  inflating site/root/siteinfo.json
  inflating site/root/structure.json
  inflating
  inflating
  inflating
  inflating
  inflating
  inflating
  inflating
  inflating

```

## 7. Zaimportować zlecenie tłumaczenia:

```

cec-components> cec import-translation-job demo1-xlate.zip
- Logged in to remote server: <server url>
- file demo1-xlate.zip uploaded to home folder, version 1

```

```
- importing: percentage 5
- importing: percentage 60
- import demol finished
```

## Tłumaczenie serwisu przez dostawcę usług tłumaczeniowych (LSP)

Tłumaczeniami serwisu na wiele języków można zarządzać za pomocą interfejsu CLI z zestawu OCE Toolkit oraz dostawcy usług tłumaczeniowych (LSP).

Założenie systemowe dotyczące lokalizacji serwisu określa język domyślny, np. angielski amerykański (en-US), oraz jeden lub większą liczbę języków alternatywnych, np. niemiecki i francuski. Używane w serwisie napisy tekstowe mogą zostać przetłumaczone na określone języki alternatywne. Jeśli przed tłumaczeniem nastąpi zmiana języka serwisu, napisy tekstowe nadal będą wyświetlane w języku domyślnym.

OCE Toolkit udostępnia w interfejsie CLI następujące opcje tłumaczeniowe:

Translation		
cec list-translation-jobs	Lists translation jobs.	[alias: ltj]
cec create-translation-job <name>	Creates a translation job <name> for a site on CEC server.	[alias: ctj]
cec download-translation-job <name>	Downloads translation job <name> from CEC server.	[alias: dtj]
cec submit-translation-job <name>	Submits translation job <name> to translation connection <connection>.	[alias: stj]
cec ingest-translation-job <name>	Gets translated job <name> from translation connection and ingest.	[alias: itj]
cec upload-translation-job <name>	Uploads translation job <name> to CEC server.	[alias: utj]
cec create-translation-connector <name>	Creates translation connector <name>.	[alias: ctc]
cec start-translation-connector <name>	Starts translation connector <name>.	[alias: stc]
cec register-translation-connector <name>	Registers a translation connector.	[alias: rtc]

Używając polecenia `cec list-translation-jobs`, można wyświetlić zlecenia tłumaczeń, które już są na serwerze. Na przykład:

```
cec ltj -s
Server: <server-name>
Asset translation jobs:
Name                               Status      Source Language
Target Languages                    Pending Languages
testHash                            INPROGRESS en-US
fr-FR,de-DE                         fr-FR,de-DE
```

Site translation jobs:

Name	Status	Source Language	Target
Languages	Pending Languages		
demoTest	TRANSLATED	en-US	de-
DE, fr-FR			

Wprowadzenie dowolnego polecenia `cec` bez parametrów lub z opcją `-h` wyświetla pomoc dot. polecenia. Zob. [Korzystanie z narzędzia `cec` opartego na wierszu polecenia](#).

Następujące sekcje zawierają więcej informacji o tłumaczeniu serwisu przez dostawcę usług tłumaczeniowych (LSP):

1. [Tworzenie zlecenia tłumaczenia za pomocą zestawu OCE Toolkit](#)
2. [Wyświetlanie zleceń tłumaczeń](#)
3. [Tworzenie łącznika z tłumaczeniem](#)
4. [Generowanie mapy serwisu dla serwisów wielojęzycznych](#)
5. [Przesyłanie zlecenia tłumaczenia do dostawcy usług tłumaczeniowych \(LSP\)](#)
6. [Wysyłanie zlecenia tłumaczenia do serwera](#)

## Tworzenie zlecenia tłumaczenia za pomocą zestawu OCE Toolkit

Za pomocą polecenia z zestawu OCE Toolkit można utworzyć w lokalnym systemie zlecenie tłumaczenia serwisu.

Aby utworzyć nowe zlecenie tłumaczenia serwisu, należy użyć polecenia `cec create-translation-job`. Polecenie to wyszukuje wszystkie zasoby serwisu, a następnie tworzy plik zip zawierający wszystkie elementy z tego serwisu, które trzeba przetłumaczyć.

```
cec create-translation-job FridayDemo -s Take2 -l all
- Logged in to remote server: <server-name>
- establish user session
- site: Take2, default language: en-US
- query channel
- site localization policy: MyLP
- target languages: de-DE, fr-FR
- create translation job submitted
- creating: percentage 50
- translation job FridyDemo created
```

Informacje o opcjach tłumaczeń są dostępne pod hasłem [Tworzenie na serwerze Oracle Content Management nowego serwisu lub nowego zlecenia tłumaczenia zasobu](#).

## Wyświetlanie zleceń tłumaczeń

Można wyświetlić zlecenia tłumaczeń w serwerze w celu zweryfikowania, czy zlecenie zostało utworzone i czy można nad nim pracować.

```
cec list-translation-jobs -s
Server: <server-name>
Asset translation jobs:
Name                               Status           Source Language Target
```

Languages	Pending Languages	
testHash	INPROGRESS	en-US
fr-FR,de-DE	fr-FR,de-DE	
Site translation jobs:		
Name	Status	Source Language
Target Languages	Pending Languages	
demoTest	TRANSLATED	en-US
de-DE,fr-FR		
FridayDemo	READY	en-US
de-DE,fr-FR	de-DE,fr-FR	

Należy zwrócić uwagę, że stanem zlecenia `FridayDemo` jest `READY` (Gotowe).

## Tworzenie łącznika z tłumaczeniem

Przy tłumaczeniu serwisu może pomóc dostawca usług tłumaczeniowych (LSP). Mając łącznik z tłumaczeń powiązany z LSP, można przysyłać i przyjmować zlecenia tłumaczenia.

Przed przesłaniem zlecenia tłumaczenia trzeba utworzyć łącznik z tłumaczeniem. Chcąc tłumaczyć serwis bez udziału LSP, można utworzyć łącznik imitujący. Do utworzenia łącznika z tłumaczeniem, a następnie do uruchomienia łącznika, należy użyć kolejno poleceń `cec create-translation-connector` i `cec start-translation-connector`:

```
cec create-translation-connector connector1
- translation connector connector1 created at <sites-toolkit folder>/
cec-components/src/main/connectors/connector1
- install connector
. . .
Start the connector: cec start-translation-connector connector1 [-p
<port>]
cec start-translation-connector connector1 -p 7777
NodeJS running. . .:
Site page: http://localhost:7777
```

Do przetestowania łącznika z tłumaczeniem należy uruchomić łącznik dla przewidywanych API, użyć zestawu OCE Toolkit:

1. Zarejestrować łącznik z zestawem OCE Toolkit.

```
>cec register-translation-connector
```

2. Otworzyć zestaw, po czym przejść na stronę "Połączenia z tłumaczeniami".

```
>http://localhost:8085/public/translationconnections.html
```

3. Przejść przez etapy ze strony weryfikacji łącznika z tłumaczeniem. Jest w nich — do weryfikacji łącznika — używany plik `translationBundle.zip` z folderu `/data` w środowisku łącznika.

Łącznik z tłumaczeniem, przeznaczony do użycia z Oracle Content Management, można utworzyć za pomocą zestawu Translation Connector SDK. Jest to przykładowa implementacja NodeJS interfejsu API łącznika z tłumaczeniem. Przyjmuje ona plik zip

zlecenia tłumaczenia z Oracle Content Management, tłumaczy wszystkie zasoby zawarte w pliku, po czym zwraca nowy plik zip zawierający wszystkie tłumaczenia.

SDK wymaga, aby użytkownik miał dostęp do dostawcy usług tłumaczeniowych (LSP) wykonującego tłumaczenie napisów. W zestawie SDK jest zawarty imitacyjny serwer LSP, pozorujący odpowiedzi od LSP przez dołączenie docelowych ustawień narodowych do napisów.

Translation Connector SDK składa się z trzech głównych modułów.

- **Łącznik (Connector):** Łącznik z tłumaczeniem, implementujący wymagany interfejs Oracle Content Management Translation Connector API.
- **Menedżer zleceń (Job Manager):** Przykładowy menedżer zleceń bazujący na systemie plików, określający stan zleceń tłumaczenia realizowanych przez dostawcę usług tłumaczeniowych.
- **Dostawca (Provider):** Implementacja określonego zestawu API, wymaganych przez LSP do dostarczania dokumentów do tłumaczenia i pobierania przetłumaczonych.

Można skopiować imitacyjny obiekt Provider JS i zaimplementować wszystkie zawarte w nim metody.

## Generowanie mapy serwisu dla serwisów wielojęzycznych

Za pomocą zestawu OCE Toolkit można wygenerować mapę serwisu dla serwisów wielojęzycznych oraz opublikować ją w serwisie.

Używając polecenia `cec create-site map <site>`, można utworzyć mapę serwisu dla serwisów wielojęzycznych w serwerze Oracle Content Management. Na przykład:

```
cec create-site-map Site1 -u http://www.example.com/site1
```

W wyniku tego polecenia następuje przejście przez strukturę serwisu, zostaje utworzona hierarchia mapy serwisu zgodna z hierarchią stron serwisu oraz zostaje utworzona mapa serwisu pod podanym adresem URL na serwerze Oracle Content Management.

Opcje poleceń są następujące:

```
--url, -u          <url> Site
URL                                                         [required]
--changefreq, -c  How frequently the page is likely to change
--file, -f        Name of the generated site map file
--publish, -p     Upload the site map to CEC server after creation
--help, -h       Show
help                                                     [boolean]
```

Poprawnymi wartościami dla opcji `<changefreq>` są:

- always
- hourly
- daily
- weekly
- monthly

- yearly
- never
- auto

Przykłady polecenia `cec create-site-map`:

```
cec create-site-map Site1 -u http://www.example.com/site1
cec create-site-map Site1 -u http://www.example.com/site1 -f sitemap.xml
cec create-site-map Site1 -u http://www.example.com/site1 -p
cec create-site-map Site1 -u http://www.example.com/site1 -c weekly -p
```

W celu opublikowania serwisu tworzy się aktualizację serwisu, aktualizuje mapę serwisu, po czym zatwierdza aktualizację.

## Przesyłanie zlecenia tłumaczenia do dostawcy usług tłumaczeniowych (LSP)

OCE Toolkit udostępnia plik zip, który można wysłać do dostawcy usług tłumaczeniowych (LSP) w celu rozpoczęcia pracy nad zleceniem tłumaczenia.

Zlecenie tłumaczenia można przesłać do LSP, używając z łącznika z tłumaczeniem. Przesyłanie może zająć trochę czasu, ponieważ łącznik musi rozpakować plik i przesłać wszystkie poszczególne pliki do LSP. Następnie LSP może utworzyć projekt dla zlecenia tłumaczenia. Gdy pliki zostaną zaimportowane do projektu, można zacząć wybierać pliki do tłumaczenia. Następnie LSP rozpocznie monitorowanie statusu tłumaczeń.

Aby sprawdzić status, należy wyszczególnić lokalnie zlecenia tłumaczeń, używając polecenia `cec list-translation-jobs` bez opcji. Gdy statusem zlecenia będzie `READY TO INGEST` (Gotowe do przyjęcia), można pobrać plik zip od LSP w celu przyjęcia zlecenia tłumaczenia. Łącznik z tłumaczeniami przesłał plik zip do LSP, LSP przetłumaczył pliki, łącznik pobrał z powrotem pliki od LSP, zawarte w pliku zip, który można pobrać i przyjąć.

```
cec list-translation-jobs
Local translation jobs:
Name                               Status           Source
Language Target Languages
FridayDemo                          READY TO INGEST en-
US                                   de-DE,fr-FR
demoTest                             READY TO INGEST en-
US                                   de-DE,fr-FR
```

W procesie przyjmowania pliku zip następuje ściągnięcie (z łącznika) zlecenia tłumaczenia do zestawu OCE Toolkit.

```
cec ingest-translation-job FridayDemo
- use connection <lsp name>
- query translation connection to get job status
```

```

- get translation
- translation saved to <sites-toolkit folder>/cec-components/dist/
FridayDemo-translated.zip
- validate translation file
- translation job ingested to <sites-toolkit folder>/cec-components/src/
main/translationJobs/FridayDemo

```

Po przyjęciu pliku zip, statusem zlecenia tłumaczenia — wyświetlanym, gdy zlecenia tłumaczenia zostaną wyszczególnione lokalnie — jest TRANSLATED (Przetłumaczone).

```

cec list-translation-jobs
Local translation jobs:
Name                               Status           Source Language
Target Languages
FridayDemo                         TRANSLATED      en-US
de-DE,fr-FR
demoTest                           READY TO INGEST en-US
de-DE,fr-FR

```

Przetłumaczone zlecenie można wysłać do serwera Oracle Content Management. Zazwyczaj zlecenie jest poddawane początkowemu szybkiemu tłumaczeniu, które jest odsyłane w celu sprawdzenia. Tłumaczenie serwisu może zająć kilka tygodni, w tym przyjmowanie tłumaczenia od LSP, korekty tłumaczeń i ponowne ich przesyłanie.

## Wysyłanie zlecenia tłumaczenia do serwera

Po wykonaniu zlecenia tłumaczenia można je wysłać do serwera Oracle Content Management i sprawdzić w swoim serwisie.

Plik zip z tłumaczeniem można wysłać do serwera, używając polecenia `cec upload-translation-job`.

```

cec upload-translation-job FridayDemo
- created translation job zip file <sites-toolkit folder>cec-components/
dist/FridayDemo.zip
- Logged in to remote server: <server-name>
- file FridayDemo.zip uploaded to home folder, version 1
- importing: percentage 5
- importing: percentage 60
- importing: percentage 60
- import FridayDemo finished

```

Gdy zlecenie tłumaczenia zostanie wysłane, status zlecenia zmienia się na INPROGRESS (w toku):

```

cec list-translation-jobs -s
Server: <server-name>
Asset translation jobs:
Name                               Status           Source Language Target
Languages                          Pending Languages
testHash                           INPROGRESS      en-US           fr-
FR,de-DE                          fr-FR,de-DE
Site translation jobs:

```

Name	Status	Source Language
Target Languages	Pending Languages	
demoTest de-DE, fr-FR	TRANSLATED	en-US
FridayDemo de-DE, fr-FR	INPROGRESS	en-US

Chcąc zweryfikować tłumaczenie, można sprawdzić napisy w zasobach w tłumaczonym serwisie.



# Część VI

## Dodatki

Dostępne są następujące dodatki:

- [Samouczek: Tworzenie składników za pomocą technologii Knockout](#)
- [Sites SDK — opis](#)
- [Rozwiązywanie problemów](#)

# Samouczek: Tworzenie składników za pomocą technologii Knockout

Niniejszy samouczek prowadzi przez etapy pracy z obiektami JavaScript, w których są wykorzystywane standardowe funkcje modelu widoku (ViewModel) i szablonów (Template) Knockout, mające na celu utworzenie składnika przechowywanego w Katalogu składników Oracle Content

- [Tworzenie składników za pomocą technologii Knockout — wprowadzenie i wymagania wstępne](#)
- [Etap 1: Tworzenie składnika](#)
- [Etap 2: Sprawdzanie struktury renderowania składnika lokalnego](#)
- [Etap 3: Sprawdzanie struktury ustawień składnika lokalnego](#)
- [Etap 4: Wyświetlanie nowej właściwości w składniku](#)
- [Etap 5: Rejestrowanie wyzwalaczy](#)
- [Etap 6: Uaktywnianie wyzwalaczy](#)
- [Etap 7: Rejestrowanie czynności](#)
- [Etap 8: Wykonywanie czynności](#)
- [Etap 9: Tworzenie osobnego tytułu dla każdego wystąpienia składnika](#)
- [Etap 10: Używanie zagnieżdżonych składników z funkcjami edycji miejscowej](#)
- [Etap 11: Obsługa różnych układów](#)
- [Etap 12: Definiowanie stylów niestandardowych](#)
- [Etap 13: Renderowanie składnika w ramce wstawkowej](#)
- [Etap 14: Używanie stylów niestandardowych, gdy składnik jest renderowany w ramce wstawkowej](#)
- [Etap 15: Integracja z funkcjami "Cofnij" i "Ponów" strony](#)
- [Etap 16: Zarządzanie zasobami](#)
- [Samouczek — przegląd](#)

## Tworzenie składników za pomocą technologii Knockout — wprowadzenie i wymagania wstępne

Niniejszy samouczek zawiera instrukcje i procedury weryfikacji umożliwiające utworzenie przykładowego składnika z użyciem obiektów JavaScript, korzystających ze standardowych funkcji modelu widoku i szablonów Knockout JS.

Kod (zawarty w plikach wypełnianych treścią podczas tworzenia składnika), do którego występują odwołania w tych przykładach, można wykorzystać, aktualizując jedynie szablon `.html` i `viewModel` JavaScript przy użyciu własnego kodu.

 **Uwaga:**

Podczas gdy Oracle Content Management nie narzuca technologii JavaScript używanej przy tworzeniu składników, to jednak zazwyczaj fabryczna funkcja JavaScript jest identyczna dla każdej implementacji składnika, w którym została wybrana struktura JavaScript.

### Wymagania wstępne

Ten samouczek koncentruje się jedynie na implementacji składnika. Więcej informacji ogólnych dotyczących składników jest dostępnych pod hasłem Tworzenie składników.

Aby można było wykonać instrukcje z tego samouczka, trzeba spełnić następujące wymagania:

- Mieć dostęp do instancji Oracle Content Management z uprawnieniami do tworzenia serwisów i składników.
- Zsynchronizować serwer instancji Oracle Content Management z komputerem lokalnym, na którym jest używana aplikacja Oracle Content Management typu Desktop lub jest używany składnik niestandardowy. Zob. Tworzenie składników niestandardowych za pomocą usługi Developer Cloud Service.

Ponadto należy znać następujące koncepcje i środowiska JavaScript:

- Wykrywanie błędów JavaScript przy użyciu przeglądarki
- Domknięcia (closures) JavaScript
- Programowanie z użyciem specyfikacji JavaScript AMD (Asynchronous Module Definition)
- Struktury RequireJS i KnockoutJS

Kontynuacja: [Etap 1: Tworzenie składnika](#).

## Etap 1: Tworzenie składnika

Wyjaśniono tu, jak utworzyć składnik niestandardowy w Oracle Content Management.

Utworzony składnik niestandardowy, aby mógł być używany przez Oracle Content Management, musi zostać zarejestrowany. Oracle Content Management można poinformować o składniku, rejestrując go za pomocą strony "Składniki" z konstruktora serwisów.

Istnieją dwa typy rejestrowanych składników.

- **Składnik lokalny:**
  - Jest to składnik, którego pliki są przechowywane na serwerze instancji Oracle Content Management.
  - Główną zaletą jest to, że nie trzeba się zajmować problemami związanymi z używaniem różnych domen i protokołów, ponieważ wszystkie pliki znajdują się w lokalizacji serwisu.
  - Wadą jest to, że nie można wykonać żadnego algorytmu warstwy pośredniej serwera Oracle Content Management i dlatego do komunikowania się

z odległymi serwerami obsługującymi technologię CORS trzeba używać REST API.

- Ten typ składnika można osadzić bezpośrednio na stronie albo wybrać renderowanie go w ramce wstawkowej.
- **Składnik odległy:**
  - Składnik, którego pliki są przechowywane na serwerze odległym; rejestrowane są jedynie adresy URL dla renderowania i panelu "Ustawienia" składnika.
  - Zaletą składnika odległego jest to, że jeśli istnieją algorytmy działające po stronie serwera, to w trakcie tworzenia zawartości składnika muszą zostać one wykonane.
  - Wadą jest to, że — aby uzyskać dostęp do tych adresów URL — trzeba rozwiązać wszelkie potencjalne problemy związane z różnymi domenami i protokołami.
  - Składniki odległe są zawsze renderowane w ramce wstawkowej na stronie.



#### Aby utworzyć i zarejestrować składnik lokalny, należy:

1. Na stronie startowej Oracle Content Management wybrać opcję **Programista**.  
Zostanie wyświetlona strona **Programista**.
2. Nacisnąć przycisk **Wyświetl wszystkie składniki**.
3. Z menu wybrać opcję **Utwórz składnik lokalny**.
4. Podać nazwę składnika; na przykład **A\_Local\_Component**.
5. Podać opcjonalny opis.
6. Nacisnąć przycisk **Utwórz**.

Gdy zostanie to wykonane, na liście składników pojawi się składnik o nazwie `A_Local_Component`.

#### Sprawdzanie wyników dla etapu 1

Pomyślnie utworzony składnik jest widoczny na palecie składników dla każdego tworzonego serwisu. Aby zweryfikować utworzenie składnika, należy:

1. Utworzyć serwis o nazwie `localComponentTest`.
2. Wybrać serwis i nacisnąć przycisk **Otwórz**.
3. Nacisnąć przycisk **Edytuj**.
4. Utworzyć aktualizację serwisu, a następnie podać jej nazwę i opcjonalny opis.
5. Wybrać stronę serwisu.
6. Aby wyświetlić listę składników niestandardowych, kliknąć na ikonie  znajdującej się w bocznej palecie, po czym wybrać opcję **Niestandardowe**.
7. Wybrać z listy składników niestandardowych składnik `A_Local_Component` i przeciągnąć go na stronę.  
Zostaje wyświetlone domyślne renderowanie utworzonego składnika lokalnego.
8. Na banerze kliknąć na ikonie  dla składnika, który został przeciągnięty na stronę.
9. Wybrać opcję **Ustawienia**.
10. Zmienić opcję wyrównania, a następnie ustawić styl dla składnika.
11. Zamknąć panel "Ustawienia".

Kolejne etapy wyjaśniają, w jaki sposób składnik niestandardowy jest konstruowany i jak go modyfikować do własnych celów. Kontynuacja: [Etap 2: Sprawdzanie struktury składnika lokalnego](#).

## Etap 2: Sprawdzanie struktury renderowania składnika lokalnego

Sprawdzimy tu strukturę plików domyślnych, utworzonych dla składnika lokalnego.

Dla prostego przykładu Hello World cztery obiekty JavaScript i znaczna liczba linii kodu mogą wydawać się zbyt wielkim nakładem, lecz mają one jedynie służyć zarówno jako podstawa konstruowania bardziej złożonych składników, jak i prezentacja interakcji z cyklem życia stron Oracle Cloud Sites Service.

Aby sprawdzić strukturę składnika lokalnego, należy:

1. Na stronie startowej Oracle Content Management wybrać opcję **Programista**.  
Zostanie wyświetlona strona **Programista**.
2. Nacisnąć przycisk **Wyświetl wszystkie składniki**.
3. Z menu wybrać opcję **Utwórz składnik lokalny**.
4. Podać nazwę składnika; na przykład **A\_Local\_Component**.
5. Podać opcjonalny opis.
6. Nacisnąć przycisk **Utwórz**.

Gdy zostanie to wykonane, na liście składników pojawi się składnik o nazwie A\_Local\_Component .

1. Używając klienta synchronizującego typu Desktop Oracle Content Management, zlokalizować składnik, po czym zsynchronizować go z systemem plików.

Nie mając klienta typu Desktop, można na karcie "Składniki" interfejsu usługi Oracle Content Management wyświetlić wszystkie składniki, po czym wybrać składnik, a następnie — aby wyświetlić pliki — drążyć w dół.

2. Pod składnikiem można zobaczyć następujące pliki:

```
assets
  render.js
  settings.html
appinfo.json
_folder_icon.jpg
```

3. Z katalogu `/assets` wybrać plik `render.js`, a następnie go otworzyć.

Najważniejsze uwagi dotyczące pliku `render.js` to:

- Jest strukturyzowany jako moduł JavaScript AMD, tak że może być uznany za wymagany na stronie.
- Zawiera odwołania do KnockoutJS i JQuery, które już zostały załadowane jako część strony Oracle Content Management.

Przyjrzyjmy się strukturze pliku `render.js`.

W pliku `render.js` są zawarte dwa obiekty JavaScript, implementujące wymagane API składników Oracle Content Management: `sampleComponentFactory` i `SampleComponentImpl`. Obiekty te stanowią przykład implementacji mającej na celu tworzenie dowolnych składników opartych na KnockoutJS. Implementacja tych obiektów zmienia się w zależności od używanej technologii.

- `sampleComponentFactory`
  - Ten obiekt jest zwracany przez moduł AMD `render.js`.
  - Jest to bardzo prosty obiekt "Factory" implementujący jeden interfejs `createComponent()`.
  - Bardziej złożone implementacje mogą używać wartości `args` przekazywanej w celu zwrócenia różnych implementacji składnika na podstawie parametru `viewModel`. Dzięki temu można uzyskać implementację składnika w trybie wykonawczym, mniej obciążającą niż uzyskana za pomocą konstruktora serwisów.
- `SampleComponentImpl`
  - Główną funkcją w tym obiekcie jest funkcja `render` służąca do renderowania składnika na stronie.

W celu wyrenderowania składnika Knockout na stronie funkcja `render` dynamicznie dodaje szablon do strony, po czym stosuje do niego wiązania `viewModel`.
  - Pozostała część implementacji obsługuje inicjalizację parametru `viewModel` i szablonu oraz komunikację między stroną a składnikiem.

Ostatnie dwa obiekty z pliku `render.js` — `sampleComponentTemplate` i `SampleComponentViewModel` — zapewniają niestandardową implementację składnika. Implementacja zmienia się w zależności od wymagań.

- `sampleComponentTemplate`
  - Ten obiekt zapewnia tworzenie szablonu KnockoutJS. Przed podjęciem próby wyświetlenia czegokolwiek czeka, aż składnik będzie miał zainicjalizowane wszystkie dane.
- `SampleComponentViewModel`
  - `viewModel` pobiera w imieniu składnika informacje składowane w Oracle Content Management, po czym — bazując na tych danych — wybiera odpowiedni układ składnika.
  - Ogólne obiekty Knockout "observable", używane przez szablon do obsługi dostępu w imieniu składnika do składowanych metadanych:

```
self.imageWidth = ko.observable('200px');
self.alignImage = ko.observable();
self.layout = ko.observable();
self.showTopLayout = ko.observable();
self.showStoryLayout = ko.observable();
```
  - Integracja wyzwalaczy i czynności:

**Wyzwalacz:** Funkcja uaktywniająca wyzwalacz Oracle Content Management ze składnika, która może zostać powiązana z czynnościami związanymi z innymi składnikami na stronie.

```
self.imageClicked = function (data, event) {
  self.raiseTrigger("imageClicked"); // matches appinfo.json
};
```

**Czynność:** Funkcja obsługująca wywołanie zwrotne, gdy składnikowi jest zlecane wykonanie czynności z użyciem danego ładunku.

```
self.executeActionsListener = function (args) {
  // get action and payload
  var payload = args.payload,
      action = args.action;

  // handle 'setImageWidth' actions
  if (action && action.actionName === 'setImageWidth') {
    $.each(payload, function(index, data) {
      if (data.name === 'imageWidth') {
        self.imageWidth(data.value);
      }
    });
  }
};
```

**Wywołanie zwrotne** mające na celu wykonanie — na żądanie — jakichkolwiek zarejestrowanych czynności.

```
SitesSDK.subscribe(SitesSDK.MESSAGE_TYPES.EXECUTE_ACTION,
$.proxy(self.executeActionsListener, self));
```

– Subskrypcje związane z cyklem życia składnika:

- \* Inicjalizacja składnika: Upewnienie się, że składnik nie będzie renderowany, dopóki nie zostaną pobrane wszystkie dane. Jest obsługiwane przez obiekty Knockout "observable".

```
self.componentLayoutInitialized = ko.observable(false);
self.customSettingsDataInitialized = ko.observable(false);
```

Uzyskanie początkowych wartości wszelkich wymaganych właściwości. Jest obsługiwane przez wywołania zwrotne pobierania danych.

```
SitesSDK.getProperty('componentLayout',
self.updateComponentLayout);
SitesSDK.getProperty('customSettingsData',
self.updateCustomSettingsData);
```

- \* Aktualizacje metadanych: wywołanie zwrotne, gdy metadane składnika składowane w imieniu składnika ulegną zmianie; na przykład, gdy użytkownik wywoła panel "Ustawienia" i zmieni dane.

```
SitesSDK.subscribe(SitesSDK.MESSAGE_TYPES.SETTINGS_UPDATED,
$.proxy(self.updateSettings, self));
```

#### Uwaga:

Ponieważ serwer Oracle Content Management zawsze ustawia dla plików `.html` typ MIME (mime-type), nie można wysłać pliku `.html` i w celu załadowania go użyć wymaganej wtyczki `"text!"`. Z tego powodu trzeba — w celu załadowania szablonu za pomocą wtyczki `"text!"` — użyć innego rozszerzenia albo wprowadzić szablon bezpośrednio w kodzie JavaScript, jak pokazano w udostępnionych danych.

### Sprawdzanie wyników dla etapu 2

Wiemy już, jak jest tworzona struktura renderowania niestandardowego składnika. Aby sprawdzić, czy działa, należy:

1. Zaktualizować w pliku `render.js` obiekt `sampleComponentTemplate`, zmieniając poniższą linię. Zmienić następujący kod:

```
'<!-- ko if: initialized -->'+
```

Zamiast tego użyć następującego kodu:

```
'<!-- ko if: initialized -->'+
'<div data-bind="text:\'image width is: \' + imageWidth()"></div>' +
```

2. Zsynchronizować (lub wysłać) składnik z serwerem instancji Oracle Content Management.
3. Edytować stronę w obszarze serwisu, a następnie przeciągnąć niestandardowy składnik `A_Local_Component` na stronę.

W tym momencie powinniśmy w składniku widzieć informację `image width is: 260px`.

4. Wyświetlić panel "Ustawienia", po czym nacisnąć przycisk **Ustawienia niestandardowe**.
5. Zmienić wartość pola **Szerokość obrazu** na `300px`.
6. W tym momencie w składniku nastąpią dwie zmiany:
  - a. Rozmiar domyślnego obrazu zostanie zmieniony z 260 pikseli na 300 pikseli.
  - b. Dodany tekst zostanie zmieniony na `image width is 300px`.

Kontynuacja: [Etap 3: Sprawdzanie struktury ustawień składnika lokalnego](#).

## Etap 3: Sprawdzanie struktury ustawień składnika lokalnego

Sprawdzimy tu strukturę ustawień, określonych dla składnika lokalnego.



W katalogu `/assets`, oprócz pliku `render.js`, istnieje wstępnie utworzony plik `settings.html`. Plik `settings.html` określa renderowanie danych dla ustawień niestandardowych składnika. W implementacji domyślnej, w danych ustawień niestandardowych występuje jedna właściwość `imageWidth`.

Aby sprawdzić strukturę składnika lokalnego, należy:

1. Używając klienta synchronizującego typu Desktop Oracle Content Management, zlokalizować składnik, po czym zsynchronizować go z systemem plików.

Nie mając klienta synchronizującego typu Desktop, można na karcie **Składniki** internetowego interfejsu Oracle Content Management wybrać składnik, a następnie — aby wyświetlić pliki — drążyć w dół.

2. Pod składnikiem można zobaczyć następujące pliki:

```
assets
  render.js
  settings.html
  appinfo.json
  _folder_icon.jpg
```

Z katalogu `/assets` wybrać plik `settings.html`, a następnie go otworzyć. W przeciwieństwie do pliku `render.js`, plik `settings.html` używa ramki wstawkowej w panelu "Ustawienia" w konstruktorze serwisów i dlatego — aby poprawnie renderować zawartość w ramce wstawkowej — musi mieć dostęp do plików wspomagających. Konstruktor serwisów jest wymagany do zarządzania serwisem, tak aby wszelkie błędy w kodzie JavaScript mogły być izolowane od konstruktora serwisów. Dlatego właśnie plik `settings.html` używa ramki wstawkowej.

W pliku `settings.html` istnieją następujące główne obszary:

- Szablon Knockout do renderowania panelu "Ustawienia".

```
<!-- ko if: initialized() -->
<div class="scs-component-settings">
  <div>
    <!-- Width -->
    <label id="widthLabel" for="width" class="settings-heading"
data-bind="text: 'Image Width'"></label>
    <input id="width" data-bind="value: width"
placeholder="example: 200px or 33%" class="settings-text-box">
  </div>
</div>
<div data-bind="setSettingsHeight: true"></div>
<!-- /ko -->
```

- Niestandardowa procedura obsługi mająca na celu dostosowanie wysokości ramki wstawkowej po wyrenderowaniu panelu "Ustawienia".

```
ko.bindingHandlers.scsCompComponentImpl
```

- Knockout ViewModel do zastosowania do szablonu Knockout.

```
SettingsViewModel
```

Podstawowymi elementami `SettingsViewModel` są:

- Subskrypcje cyklu życia elementu.
- Inicjalizacja składnika:
  - Upewnienie się, że składnik nie będzie renderowany, dopóki nie zostaną pobrane wszystkie dane. Jest obsługiwane przez obiekty Knockout "observable".

```
self.initialized = ko.observable(false);
```

- Upewnienie się, że nie zostanie podjęta próba aktualizacji danych, dopóki nie będziemy gotowi.

```
self.saveData = false;
```

- Uzyskanie początkowych wartości wszelkich wymaganych właściwości. Jest obsługiwane przez wywołania zwrotne pobierania danych.

```
SitesSDK.getProperty('customSettingsData', function (data) {
  //update observable
  self.width(data.width);

  // note that viewModel is initialized and can start saving data
  self.initialized(true);
  self.saveData = true;
});
```

- Zapisanie wszelkich zmian właściwości (danych dla ustawień niestandardowych).

```
self.save = ko.computed(function () {
  var saveconfig = {
    'width': isNaN(self.width()) ? self.width() : self.width() + 'px'
  };

  // save data in page
  if (self.saveData) {
    SitesSDK.setProperty('customSettingsData', saveconfig);
  }
}, self);
```

Aby dodać kolejną właściwość, która ma zostać zarejestrowana, należy:

1. Zaktualizować interfejs użytkownika, tak aby nowa wartość była wyświetlana.
2. Zainicjalizować wartość przy użyciu bieżącej wartości przechowywanej dla składnika.
3. Zapisać wszelkie zmiany wartości z powrotem do składnika.

Aby dodać kolejną właściwość do składnika niestandardowego, należy wprowadzić w pliku `settings.html` następujące zmiany:

1. Dodać kolejny obiekt "observable" obsługujący nową właściwość. Zmienić następujący kod:

```
self.width = ko.observable();
```

Zamiast tego użyć następującego kodu:

```
self.width = ko.observable();
self.imageBannerText = ko.observable();
```

2. Uzyskać dowolną bieżącą wartość dla nowej właściwości, gdy panel "Ustawienia" zostanie po raz pierwszy wyświetlony. Zmienić następujący kod:

```
self.width(data.width);
```

Zamiast tego użyć następującego kodu:

```
self.width(data.width);
self.imageBannerText(data.imageBannerText);
```

3. Zapisać wszelkie zmiany wprowadzone do nowej właściwości. Zmienić następujący kod:

```
'width': isNaN(self.width()) ? self.width() : self.width() +
'px'
```

Zamiast tego użyć następującego kodu:

```
'width': isNaN(self.width()) ? self.width() : self.width() + 'px',
'imageBannerText': self.imageBannerText()
```

4. Aby wyświetlić nowe pole, dodać interfejs użytkownika. Zmienić następujący kod:

```
<label id="widthLabel" for="width" class="settings-heading" data-
bind="text: 'Image Width'"></label>
<input id="width" data-bind="value: width" placeholder="example:
200px or 33%" class="settings-text-box">
```

Zamiast tego użyć następującego kodu:

```
<label id="widthLabel" for="width" class="settings-heading" data-
bind="text: 'Image Width'"></label>
<input id="width" data-bind="value: width" placeholder="example:
200px or 33%" class="settings-text-box">

<label id="imageBannerTextLabel" for="imageBannerText"
class="settings-heading" data-bind="text: 'Image Banner'"></label>
<input id="imageBannerText" data-bind="value: imageBannerText"
placeholder="Text to display above an image" class="settings-text-
box">
```

5. Zsynchronizować (lub wysłać) plik `settings.html`.

Gdybyśmy teraz uruchomili, pole zostałoby wyświetlone. Jednak rozmiar panelu "Ustawienia" nie zmienia się automatycznie. Ponieważ zwiększyliśmy rozmiar panelu, musimy także zaktualizować wpis rejestracyjny `components.json` o nowy rozmiar.

1. Pobrać plik `appinfo.json` (znajduje się na tym samym poziomie co katalog `assets/` składnika), po czym zaktualizować rozmiar panelu "Ustawienia". Zmienić następujący kod:

```
"settingsHeight": 90,
```

Zamiast tego użyć następującego kodu:

```
"settingsHeight": 160,
```

2. Zsynchronizować (lub wysłać) plik `appinfo.json`.

### Sprawdzanie wyników dla etapu 3

Teraz powinniśmy być w stanie zobaczyć i wprowadzić nową właściwość, którą dodaliśmy do panelu "Ustawienia".

1. Aby konstruktor serwisów mógł pobrać zmiany do składnika, odświeżyć stronę serwisu.
2. Przełączyć stronę do trybu edycji.
3. Przeciągnąć składnik na stronę.
4. Wyświetlić panel "Ustawienia" składnika.
5. Nacisnąć przycisk **Ustawienia niestandardowe**.

Będą wyświetlane dwa pola dla właściwości zawartych w pliku `settings.html`.

Kontynuacja: [Etap 4: Wyświetlanie nowej właściwości w składniku](#).

## Etap 4: Wyświetlanie nowej właściwości w składniku

Pod koniec tego etapu będziemy potrafili wprowadzić wartość nowej właściwości w panelu "Ustawienia" oraz zobaczyć zmianę w składniku niestandardowym, odzwierciedlającą tę nową właściwość. Aktualizacje nowej właściwości będą automatycznie zapisywane wraz ze stroną.

W pliku `render.js` trzeba zaktualizować w składniku następujące dwa obiekty JavaScript:

- `SampleComponentViewModel`
- `sampleComponentTemplate`

Otworzyć do edycji plik `render.js` i zaktualizować `SampleComponentViewModel`, tak aby zawierał nową właściwość. Zmienić tę właściwość:

```
self.showStoryLayout = ko.observable();
```

Zamiast niej użyć:

```
self.showStoryLayout = ko.observable();  
self.imageBannerText = ko.observable();
```

Zaktualizować `SampleComponentViewModel`, tak aby uzyskiwać wszelkie zmiany wartości. Zmienić tę właściwość:

```
self.imageWidth(customData && customData.width);
```

Zamiast niej użyć:

```
self.imageWidth(customData && customData.width);  
self.imageBannerText(customData && customData.imageBannerText);
```

Zmienić `sampleComponentTemplate`, tak aby nowa właściwość była wyświetlana. Zmienić tę właściwość:

```
'<div data-bind="text: \'image width is: \' + imageWidth()"'></div>' +
```

Zamiast niej użyć:

```
'<div data-bind="text: imageBannerText"'></div>' +
```

Zsynchronizować składnik z serwerem Oracle Content Management lub wysłać składnik na ten serwer.

Składnik został zmodyfikowany, tak aby wyświetlał nową właściwość. Składnik — w przeciwieństwie do panelu "Ustawienia", który jest osadzony w ramce wstawkowej na stronie — jest wstawiany bezpośrednio na stronie i dlatego, gdy jego rozmiar się zwiększa, automatycznie zwiększa się także dostępny dla niego obszar.

#### Sprawdzanie wyników dla etapu 4

Aby zobaczyć wyświetlaną nową właściwość:

1. Aby konstruktor serwisów mógł pobrać zmiany do składnika, odświeżyć stronę serwisu.
2. Przełączyć stronę do trybu edycji.
3. Przeciągnąć składnik na stronę.
4. Wyświetlić panel "Ustawienia" składnika.
5. Nacisnąć przycisk **Ustawienia niestandardowe**.
6. Zmienić `Image Banner` na `Workspace`.

Dokonana aktualizacja składnika zostanie odzwierciedlona na stronie — nad obrazem pojawi się baner `Workspace`.

Kontynuacja: [Etap 5: Rejestrowanie wyzwalaczy](#).

## Etap 5: Rejestrowanie wyzwalaczy

W tej części zobaczymy, w jaki sposób można zarejestrować wyzwalacz Oracle Content Management, wybierany za pomocą opcji "Czynności wyzwalacza" z karty "Łącze" panelu "Ustawienia" składnika.

Wyzwalacze stanowią część mechanizmu komunikacji między składnikami Oracle Content Management. Każdy składnik może uaktywniać dowolną liczbę wyzwalaczy. Składnik może dostarczać do wyzwalacza ładunek, który jest przekazywany do czynności uruchamianej w chwili uaktywnienia wyzwalacza. Użytkownicy mogą wybrać czynności, które powinny być uruchamiane dla poszczególnych wyzwalaczy. Na koniec składniki, które są projektowane z myślą o ich współpracy, mogą automatycznie uaktywniać wyzwalacze w celu uruchamiania czynności w innym składniku bez potrzeby definiowania interakcji między składnikami.

W przypadku dodawanych składników wyzwalacze są rejestrowane w ramach danych rejestracyjnych. Aby dodać wyzwalacz, należy zaktualizować tablicę właściwości "triggers", dodając do niej obsługiwany wyzwalacz. Trzeba także określić ładunek obsługiwany przez wyzwalacz, aby mógł zostać utworzony interfejs użytkownika umożliwiający użytkownikowi odwzorowywanie wartości z ładunku na właściwości obsługiwane przez czynność.

Należy otworzyć plik `appinfo.json` do edycji i sprawdzić wpis `"triggers":[]`.

```
"triggers": [{
  "triggerName": "imageClicked",
  "triggerDescription": "Image clicked",
  "triggerPayload": [{
    "name": "payloadData",
    "displayName": "Trigger Payload Data"
  }]
}],
```

W tym wpisie można zobaczyć:

- Nazwę "imageClicked" wyzwalacza (triggerName), która powinna być unikatowa i zazwyczaj jest przypisywana do przestrzeni nazw na podstawie ID składnika niestandardowego.
- Opis "Image clicked" wyzwalacza (triggerDescription), używany w oknie dialogowym, w interfejsie użytkownika, do wyświetlenia wyzwalacza.
- Wartość "payloadData" ładunku wyzwalacza (triggerPayload). Użytkownicy będą mogli wybierać wpisy z tego ładunku i odwzorowywać je na pola w danej czynności.

#### Sprawdzanie wyników dla etapu 5

Wyzwalacz można zobaczyć i wybrać na karcie **Łącze** w panelu "Ustawienia" składnika:

1. Aby konstruktor serwisów mógł pobrać zmiany do składnika, odświeżyć stronę serwisu.
2. Przełączyć stronę do trybu edycji.
3. Przeciągnąć składnik na stronę.
4. Wyświetlić panel "Ustawienia" składnika.
5. Wybrać kartę **Łącze** znajdującą się na górze panelu "Ustawienia".
6. Jako typ łącza wybrać opcję **Czynności wyzwalacza**.
7. Kliknąć na zarejestrowanym wyzwalaczu **Image clicked**.
8. W oknie dialogowym przeciągnąć czynność **Pokaż alarm** z sekcji "Czynności związane ze stroną". ("Czynności związane ze stroną" to wbudowane czynności dostarczane przez Oracle Content Management.)

9. W polu **Komunikat** wybrać wartość **Trigger Payload Data**, czyli nazwę wpisu w ładunku, która była wyświetlana podczas rejestrowania wyzwalacza. Można ją zmienić na dowolną nazwę.

Teraz można zarejestrować wyzwalacz i odwzorować go na wbudowaną czynność przekazywaną przez wartość. W następnym etapie pokażemy, w jaki sposób wyzwalacz jest uaktywniany w celu wykonania czynności.

Kontynuacja: [Etap 6: Uaktywnianie wyzwalaczy](#).

## Etap 6: Uaktywnianie wyzwalaczy

Pokażemy tutaj, jak uaktywniać zarejestrowany wyzwalacz.

Wyzwalacze mogą być uaktywniane w dowolnej chwili przez składnik. Zazwyczaj wyzwalacz jest uaktywniany wskutek interakcji użytkownika, takiej jak naciśnięcie przycisku lub wybranie wiersza w tabeli. Składnik może jednak uaktywnić wyzwalacz na podstawie dowolnych kryteriów — na przykład, gdy dane ulegną zmianie spowodowanej wywołaniem REST.

W tym przykładzie kliknięcie na obrazie uaktywni wyzwalacz przekazujący bieżącą wartość właściwości `whoAreYou`.

Otworzyć plik `render.js`, po czym przyjrzeć się obiektowi `SampleComponentViewModel`.

Aby uaktywnić wyzwalacz, należy:

1. Przyjrzeć się funkcji w obiekcie `SampleComponentViewModel`, która wywołuje pakiet Sites SDK w celu uaktywnienia wyzwalacza.

```
self.raiseTrigger = function (triggerName) {
  SitesSDK.publish(SitesSDK.MESSAGE_TYPES.TRIGGER_ACTIONS, {
    'triggerName': triggerName,
    'triggerPayload': {
      'payloadData': 'some data here'
    }
  });
};
```

2. Teraz potrzebujemy czegoś w interfejsie użytkownika, co wywołałoby funkcję uaktywniającą wyzwalacz. W pliku `render.js` zaktualizować obiekt `sampleComponentTemplate`, tak aby zawierał ten wpis:

```
'<div data-bind="attr: {style: imageStyle, \'data-layout\':
alignImage()}, click: imageClicked">' +
```

W obiekcie `SampleComponentViewModel` jest widoczna funkcja JavaScript, która zostanie wywołana po kliknięciu na obrazie. Funkcja ta wywołuje Sites SDK i zleca mu uaktywnienie wszystkich czynności zdefiniowanych dla wyzwalacza "imageClicked", tj. wartości przekazanej z wiązania `click` (w punkcie 2). Przekazywane są także `triggerPayload` zawierający jedno pole `payloadData` oraz wartość statyczna 'some data here'. Te wartości `imageClicked` i `whoAreYou` są zgodne z zawartymi w pliku `appinfo.json`, za pomocą którego wyzwalacz został zarejestrowany (poprzedni etap).

W przykładowym kodzie wyzwalacz jest uaktywniany przez powiązanie danych z wiązania `click` i przekazuje nazwę `imageClicked` wyzwalacza. Obecnie są dostępne trzy renderowania składnika `<scs-image>`, bazujące na stylu wybranym przez użytkownika. Aby zapewnić uaktywnianie wyzwalacza dla każdego z tych układów, należy dokonać w pliku `render.js` następujących zmian:

- Uaktywnianie wyzwalaczy z różnych układów. Odszukać dwa wystąpienia tego kodu:

```
'<div data-bind="attr: {style: imageStyle, \'data-layout\': alignImage()}">' +
```

Zmienić ten kod na następujący:

```
'<div data-bind="attr: {style: imageStyle, \'data-layout\': alignImage()}, click: imageClicked">' +
```

- Określić ładunek przekazywany do wyzwalaczy. Zmienić następujący kod:

```
self.raiseTrigger = function (triggerName) {  
    SitesSDK.publish(SitesSDK.MESSAGE_TYPES.TRIGGER_ACTIONS, {  
        'triggerName': triggerName,  
        'triggerPayload': {  
            'payloadData': 'some data here'  
        }  
    });  
};
```

Zamiast tego użyć następującego kodu:

```
self.raiseTrigger = function (triggerName) {  
    SitesSDK.publish(SitesSDK.MESSAGE_TYPES.TRIGGER_ACTIONS, {  
        'triggerName': triggerName,  
        'triggerPayload': {  
            'payloadData': self.imageBannerText() // pass banner text as  
payload  
        }  
    });  
};
```

- Zsynchronizować (lub wysłać) plik `render.js` z serwerem instancji Oracle Content Management.

Teraz po sprawdzeniu wymaganego kodu można podpiąć wyzwalacz, tak aby był uaktywniany przez składnik niestandardowy, gdy użytkownik kliknie na przycisku.

### Sprawdzanie wyników dla etapu 6

Teraz powinniśmy być w stanie zarejestrować czynność uaktywniającą wyzwalacz oraz uruchamiać czynność, która ma zostać wykonana po uaktywnieniu wyzwalacza:

1. Aby konstruktor serwisów mógł pobrać zmiany do składnika, odświeżyć stronę serwisu.
2. Przełączyć stronę do trybu edycji.
3. Przeciągnąć składnik na stronę.



4. Wyświetlić panel "Ustawienia" składnika.
5. Wybrać kartę **Łącze** znajdującą się na górze panelu "Ustawienia".
6. Jako typ łącza wybrać opcję **Czynności wyzwalacza**.
7. Kliknąć na zarejestrowanym wyzwalaczu **imageClicked**.
8. W oknie dialogowym przeciągnąć czynność **Pokaż alarm** z sekcji **Czynności związane ze stroną**.
9. W polu **Komunikat** wybrać wartość **payloadData**, czyli ładunek wprowadzony podczas rejestrowania wyzwalacza.
10. Zamknąć panel "Ustawienia", po czym przełączyć konstruktor serwisów do trybu podglądu.
11. Kliknąć na obrazie w składniku.

Pojawi się alarm `no message defined` (nie zdefiniowano komunikatu), ponieważ nie została określona wartość `imageBannerText`.
12. Przełączyć panel "Ustawienia" do trybu edycji, po czym wyświetlić ponownie panel "Ustawienia" składnika.
13. Nacisnąć przycisk **Ustawienia niestandardowe**, po czym wpisać `Workplace`.
14. Zamknąć panel "Ustawienia", po czym przełączyć stronę do trybu podglądu.
15. Kliknąć na obrazie w składniku.

Teraz powinien być wyświetlany baner `Workplace`, wywołany wskutek zmiany dokonanej w wiązaniu `click`.

Gdy wyzwalacz zostanie uaktywniony, może być wykonywana dowolna liczba czynności.

 **Uwaga:**

Nie ma określonej kolejności wykonywania czynności. Mimo że czynności są wywoływane w kolejności, w jakiej są wyszczególnione, to jednak nie ma okresu oczekiwania na ukończenie poprzedniej czynności — następna czynność jest wywoływana od razu. Jeśli czynność wykonuje wywołanie asynchroniczne, to jej działanie może nie zostać ukończony, zanim zostanie wywołana następna czynność.

Kontynuacja: [Etap 7: Rejestrowanie czynności](#).

## Etap 7: Rejestrowanie czynności

Czynności Oracle Content Management są wywoływane dla składników, gdy zostaną uaktywnione odpowiednie wyzwalacze.

Składnik może mieć zarejestrowaną dowolną liczbę czynności, przy czym można zdefiniować ładunek obsługiwany przez daną czynność. Użytkownik, wybierając czynność, może wypełnić danymi przekazywany do niej ładunek.

Podobnie jak w przypadku rejestrowania wyzwalaczy, można — używając danych rejestracyjnych w pliku `appinfo.json` — zarejestrować czynności obsługiwane

przez składnik. Aby przejrzeć rejestrację przykładowej czynności w składniku, należy otworzyć plik `appinfo.json` i odszukać kod "actions".

```
"actions": [{
  "actionName": "setImageWidth",
  "actionDescription": "Update the image width",
  "actionPayload": [{
    "name": "imageWidth",
    "description": "Image Width in pixels",
    "type": {
      "ojComponent": {
        "component": "ojInputText"
      }
    },
    "value": ""
  }]
}]
```

Zarejestrowana czynność będzie od razu widoczna w oknie dialogowym czynności, które jest wywoływane, gdy użytkownik kliknie na wyzwalaczu na karcie **Łącze** w panelu ustawień składnika.

#### Sprawdzanie wyników dla etapu 7

1. Aby konstruktor serwisów mógł pobrać zmiany do składnika, odświeżyć stronę serwisu.
2. Przełączyć stronę do trybu edycji.
3. Przeciągnąć składnik na stronę.
4. Przeciągnąć na stronę składnik "przycisk".
5. Wyświetlić panel "Ustawienia" składnika "przycisk".
6. Na karcie "Ogólne" zmienić etykietę przycisku na **Naciśnij mnie..**
7. Z panelu "Ustawienia" wybrać kartę **Łącze**.
8. Jako **Typ łącza** wybrać opcję **Czynności wyzwalacza**.
9. Dla składnika "przycisk" wybrać wyzwalacz **Naciśnięcie przycisku**.
10. W oknie dialogowym rozwinąć w lewej palecie składnik `A_Local_Component`.
11. Przeciągnąć na stronę czynność **Update the image width** ze składnika `A_Local_Component`.
12. W polu **Szerokość obrazu w pikselach** wprowadzić wartość **300px**.

Zobaczyliśmy, jak zarejestrować czynność i jak będzie ona pokazywana w interfejsie użytkownika. Następnie nauczymy się, jak obsługiwać czynność ze składnika, gdy zostanie ona wywołana.

Kontynuacja: [Etap 8: Wykonywanie czynności](#).

## Etap 8: Wykonywanie czynności

Pod koniec tego etapu będziemy potrafili przeciągać na stronę składniki wykonujące czynności w danym składniku. Wykorzystywana jest przy tym rejestracja czynności utworzona w poprzednim etapie.

Składnik, aby móc wykonać czynność, musi nasłuchiwać komunikatu `EXECUTE_ACTION`. W komunikacie tym jest także zawarty ładunek przekazany do czynności, z którego trzeba wyodrębnić oczekiwane wartości.

Aby był prowadzony nasłuch komunikatu `EXECUTE_ACTION`, należy otworzyć plik `render.js` do edycji i zaktualizować w nim obiekt `SampleComponentViewModel`, używając następującego kodu:

```
SitesSDK.subscribe('EXECUTE_ACTION', $.proxy(self.executeActionsListener, self));
```

Po otrzymaniu komunikatu `EXECUTE_ACTION` zostanie wykonana powiązana funkcja wywołania zwrotnego:

```
self.executeActionsListener = function (args) {  
  // get action and payload  
  var payload = args.payload,  
      action = args.action;  
  
  // handle 'setImageWidth' actions  
  if (action && action.actionName === 'setImageWidth') {  
    $.each(payload, function(index, data) {  
      if (data.name === 'imageWidth') {  
        self.imageWidth(data.value);  
      }  
    });  
  }  
}
```

Kod ten tworzy funkcję JavaScript wykonywania czynności, a następnie używa biblioteki Sites SDK do wywołania tej funkcji, gdy tylko zostanie zgłoszony komunikat `EXECUTE_ACTION`.

Składnik ten będzie wywoływany przy każdym zgłoszeniu komunikatu `EXECUTE_ACTION`, a o tym, czy czynność ta zostanie wykonana, decyduje sam składnik. Dlatego, aby mieć pewność, że jest to składnik obsługiwany, trzeba sprawdzić jego nazwę.

Ładunek dla czynności ma postać tablicy wartości. Zazwyczaj trzeba będzie odnaleźć w tablicy odpowiednie wartości ładunku.

 **Uwaga:**

Ponieważ nasłuch funkcji działa w trybie wywołania zwrotnego, należy — aby mieć pewność, że w chwili uruchomienia funkcji jest możliwy dostęp do obiektu `viewModel` — użyć domknięcia (closure) JavaScript albo odpowiednio powiązać funkcję.

#### Sprawdzanie wyników dla etapu 8

1. Aby konstruktor serwisów mógł pobrać zmiany do składnika, odświeżyć stronę serwisu.
2. Przełączyć stronę do trybu edycji.

3. Przeciągnąć składnik na stronę.
4. Przeciągnąć składnik "przycisk" na stronę.
5. Wyświetlić panel "Ustawienia" składnika "przycisk".
6. Na karcie "Ogólne" zmienić etykietę przycisku na `Click me!`
7. Wybrać kartę **Łącze** znajdującą się na górze panelu "Ustawienia".
8. Jako typ łącza wybrać opcję **Czynności wyzwalacza**.
9. Dla składnika "przycisk" wybrać wyzwalacz **Naciśnięcie przycisku**.
10. W oknie dialogowym rozwinąć składnik `A_Local_Component` (po lewej stronie).
11. Przeciągnąć czynność **Aktualizacja szerokości obrazu** ze składnika `A_Local_Component` do obszaru prawego panelu.
12. W polu **Szerokość obrazu w pikselach** wprowadzić wartość `300px`.
13. Przełączyć stronę do trybu podglądu.
14. Nacisnąć przycisk **Click me!**

W tym miejscu rozmiar obrazu zostanie zwiększony do `300px`.



#### Uwaga:

Wyzwalacze i czynności są projektowane z myślą o obsłudze komunikacji między składnikami. Nie są projektowane tak, aby można było za ich pomocą tworzyć stany lub nimi zarządzać. Po odświeżeniu strony powróci ona do swojego pierwotnego stanu, ponieważ nie są uruchamiane żadne wyzwalacze ani nie są wykonywane żadne czynności.

Kontynuacja: [Etap 9: Tworzenie osobnego tytułu dla każdego wystąpienia składnika](#).

## Etap 9: Tworzenie osobnego tytułu dla każdego wystąpienia składnika

Wyjaśniono tu, jak można tworzyć osobne tytuły dla różnych wystąpień składnika

Gdy składnik zostanie przeciągnięty na stronę, jego baner będzie zawierał napis `A_Local_Component`. Jest to akceptowalne, jeśli użytkownik przeciąga na stronę tylko jedno wystąpienie składnika; jeśli jednak jest ich przeciąganych kilka wystąpień składnika, to użytkownik może — aby je rozróżnić — nadawać im różne tytuły.

Tytuł składnika można zaktualizować za pomocą pakietu Sites SDK. Tutaj zaktualizujemy tytuł, używając właściwości `imageBannerText`.

Aby zaktualizować tytuł, należy otworzyć plik `render.js` do edycji, po czym dodać do obiektu `SampleComponentViewModel` następujący kod:

```
self.updateDescription = ko.computed(function () {
  SitesSDK.setProperty('description', self.imageBannerText());
});
```

To obliczenie Knockout zaktualizuje opis składnika, gdy obiekt "observable" `imageBannerText` ulegnie zmianie.

#### Sprawdzanie wyników dla etapu 9

1. Aby konstruktor serwisów mógł pobrać zmiany do składnika, odświeżyć stronę serwisu.
2. Przełączyć stronę do trybu edycji.
3. Przeciągnąć składnik na stronę.
4. Wyświetlić panel "Ustawienia" składnika.
5. Nacisnąć przycisk **Ustawienia niestandardowe**.
6. Zmienić **Baner obrazu** na **Workplace**.
7. Zamknąć panel "Ustawienia", po czym umieścić wskaźnik myszy na składniku, aby wyświetlić baner.

Powinien być wyświetlany baner z napisem `A_Local_Component Workplace`.

Kontynuacja: [Etap 10: Używanie zagnieżdżonych składników z funkcjami edycji miejscowej](#).

## Etap 10: Używanie zagnieżdżonych składników z funkcjami edycji miejscowej

Składniki Oracle Content Management są implementowane przy użyciu architektury składników KnockoutJS. Znaczy to, że — jeśli do implementacji składników używa się biblioteki KnockoutJS — wbudowane składniki Oracle Content Management można zawrzeć bezpośrednio w szablonie.

#### Uwaga:

Ponieważ wbudowane składniki Oracle Content Management mogą działać tylko na stronie Oracle Content Management, nie można używać zagnieżdżonych składników, jeśli składnik jest renderowany w ramce wstawkowej.

Aby móc używać zagnieżdżonych składników, należy:

1. Zaimplementować składnik, używając KnockoutJS.
2. Zawrzeć składnik, używając RequireJS, i użyć tej samej zmiennej instancyjnej Knockout "`ko`", która została utworzona przez Oracle Content Management.

Jest to wymagane, ponieważ Oracle Content Management rozszerza Knockout o składniki i składniki te nie będą dostępne, jeśli użytkownik będzie używał własnej instancji KnockoutJS.

W tej części zobaczymy, w jaki sposób są w składniku niestandardowym renderowane składniki "obraz" (image), "akapit" (paragraph) i "tytuł" (title) pochodzące z Oracle Content Management. Użytkownik będzie mógł edytować je bezpośrednio na stronie oraz wywoływać panel "Ustawienia" dla zagnieżdżonego składnika.

Chcąc zobaczyć, jak te składniki są zawierane w szablonie, można otworzyć plik `render.js` i przyjrzeć się `sampleComponentTemplate`. Poniżej jest pokazana renderowana sekcja domyślna:

```
'<!-- ko if: alignImage() !== \'right\' -->' +
'<div style="display:flex;">' +
'<div data-bind="attr: {style: imageStyle, \'data-layout\': alignImage()},
click: imageClicked">' +
'<scs-image params="{ scsComponent: { \'renderMode\': mode, \'parentId\':
id, \'id\': \'imageId\'}, \'data\': imageData } }"></scs-image>' +
'</div>' +
'<div data-bind="attr: {style: paragraphStyle}">' +
'<scs-title params="{ scsComponent: { \'renderMode\': mode, \'parentId\':
id, \'id\': \'titleId\'}, \'data\': titleData } }"></scs-title>' +
'<scs-paragraph params="{ scsComponent: { \'renderMode\': mode,
\'parentId\': id, \'id\': \'paragraphId\'}, \'data\': paragraphData } }"></
scs-paragraph>' +
'</div>' +
'</div>' +
'<!-- /ko -->' +
```

Przyglądając się zagnieżdżonemu składnikowi `<scs-image>`, można zobaczyć następujący wpis:

```
'<scs-image params="{ scsComponent: { \'renderMode\': mode, \'parentId\':
id, \'id\': \'imageId\'}, \'data\': imageData } }"></scs-image>' +
```

Dane `scsComponent`, przekazywane do wiązania `params` szablonu, zawierają:

- `renderMode`: Tryb, w którym działa konstruktor szablonów. Za pomocą tego parametru można włączać i wyłączać funkcje. Na przykład, gdy jest używany przez składnik `<scs-title>`, dodaje w trybie edycji edytor tekstów sformatowanych.
- `parentId`: Parametr wymagany, aby składnik Oracle Content Management wiedział, że jest renderowany jako zagnieżdżony. Wszystkie zmiany, dokonane w zagnieżdżonym składniku, będą zapisywane w danych dla składnika niestandardowego.
- `id`: Unikatowy ID zagnieżdżonego składnika. Następnie będzie powiązany z przestrzenią nazw na podstawie ID składnika niestandardowego.
- `data`: Początkowe dane zagnieżdżonego składnika. Jeśli składnik nie zostanie zmodyfikowany, to będzie renderowany z użyciem tych początkowych danych.

Wartości `id` i `mode` są przekazywane w obiekcie `SampleComponentViewModel` do składnika niestandardowego, a zatem — aby uzyskać te wartości — nie trzeba tego obiektu modyfikować:

```
// Store the args
self.mode = args.viewMode;
self.id = args.id;
```

Składnia wszystkich pozostałych obsługiwanych zagnieżdżonych składników (na przykład `<scs-image>`, `<scs-title>`) ma ten sam wzorec, co składnia składnika `<scs-paragraph>`.

### Sprawdzanie wyników dla etapu 10

1. Aby konstruktor serwisów mógł pobrać zmiany do składnika, odświeżyć stronę serwisu.
2. Przełączyć stronę do trybu edycji.
3. Przeciągnąć składnik na stronę.
4. W składniku kliknąć na tekście *Będąc autorem strony, można edytować...*, a następnie zaktualizować opis za pomocą edytora tekstów z formatowaniem.
5. Aby wyświetlić aktualizację, przełączyć do trybu podglądu.
6. Przełączyć z powrotem do trybu edycji.
7. Wyświetlić panel "Ustawienia" składnika.
8. Kliknąć na łączu **Składniki**, które jest wyświetlane po wykryciu zagnieżdżonego składnika.
9. Kliknąć na składniku **akapit**, będącym wykrytym składnikiem zagnieżdżonym.

Można teraz zaktualizować właściwości składnika "akapit" użytego w składniku.

#### Uwaga:

Dopóki nie zostanie utworzona instancja składnika, Oracle Content Management nie dowie się o zagnieżdżonych składnikach, które mogą istnieć w szablonie. Chcąc poinformować Oracle Content Management o ukrytych zagnieżdżonych składnikach, można użyć `SitesSDK.setProperty('visibleNestedComponents', [])`; API. Aby ukryte zagnieżdżone składniki domyślnie były ujawniane, trzeba w rejestracji składnika zaktualizować tablicę `"nestedComponents": []`.

Kontynuacja: [Etap 11: Obsługa różnych układów](#).

## Etap 11: Obsługa różnych układów

Zajmiemy się tu układami pozwalającymi użytkownikowi zmieniać sposób wyświetlania składnika.

Składnik niestandardowy może obsługiwać dowolną liczbę układów, które chcemy udostępnić użytkownikowi do wyboru. Każdy z tych układów zmienia sposób wyświetlania składnika niestandardowego. Układy stanowią rozszerzenie danych rejestracji.

Aby zobaczyć trzy układy obsługiwane w przykładowym kodzie, należy przyjrzeć się wpisowi `"componentLayouts"` w pliku `appinfo.json`:

```
"componentLayouts": [  
  {  
    "name": "default",  
    "displayName": "IMAGE_LEFT_LAYOUT"  
  },  
]
```

```

    {
      "name": "right",
      "displayName": "IMAGE_RIGHT_LAYOUT"
    },
    {
      "name": "top",
      "displayName": "IMAGE_TOP_LAYOUT"
    }
  ],

```

Jeśli wyświetlimy panel "Ustawienia" składnika niestandardowego, zobaczymy opcję przełączania się między stylami. Aby umożliwić reagowanie składnika na zmianę wyboru, plik `render.js` zawiera kod pobierający obecnie wybraną wartość i nasłuchujący jej zmiany.

Otworzyć plik `render.js` do edycji, po czym przyjrzeć się obiektowi `SampleComponentViewModel`.

- Istnieje obiekt "observable" `layout`, do którego występuje odwołanie w szablonie:

```
self.layout = ko.observable();
```

- Istnieje funkcja aktualizacji (`update`) przeprowadzanej, gdy ta wartość się zmieni:

```

self.updateComponentLayout = $.proxy(function (componentLayout) {
  var layout = componentLayout ? componentLayout : 'default';
  self.layout(layout);
  self.alignImage(layout === 'right' ? 'right' : 'left');
  self.showTopLayout(layout === 'top');
  self.showStoryLayout(layout === 'default' || layout === 'right');

  self.componentLayoutInitialized(true);
}, self);

```

- Kod inicjalizacyjny uzyskuje pierwotną wartość układu i wywołuje funkcję aktualizacji:

```
SitesSDK.getProperty('componentLayout', self.updateComponentLayout);
```

Nasłuch zmian właściwości nasłuchuje wszelkich zmian tej właściwości i wywołuje funkcję aktualizacji:

```

self.updateSettings = function (settings) {
  if (settings.property === 'componentLayout') {
    self.updateComponentLayout(settings.value);
  } else if (settings.property === 'customSettingsData') {
    self.updateCustomSettingsData(settings.value);
  }
};

```

```
SitesSDK.subscribe(SitesSDK.MESSAGE_TYPES.SETTINGS_UPDATED, $.proxy(self.updateSettings, self));
```



Na koniec obiekt szablonu `sampleComponentTemplate` zawiera kod odzwierciedlający zmiany w tej wartości:

```
'<!-- ko if: alignImage() === \'right\' -->' +
```

Łącznie zmiany te umożliwiają wybranie układu w panelu "Ustawienia" i zaktualizowanie składnika.

#### Sprawdzanie wyników dla etapu 11

1. Aby konstruktor serwisów mógł pobrać zmiany do składnika, odświeżyć stronę serwisu.
2. Przełączyć stronę do trybu edycji.
3. Przeciągnąć składnik na stronę.
4. Wyświetlić panel "Ustawienia" składnika.
5. Wybrać z właściwości "Layout" opcję **Image Right**.

W tym miejscu składnik zostanie zaktualizowany i będzie wyświetlać składnik "`<scs-image>`".

Kontynuacja: [Etap 12: Definiowanie stylów niestandardowych](#).

## Etap 12: Definiowanie stylów niestandardowych

Tworzone składniki są traktowane jak wszelkie inne w plikach `design.json` i `design.css`, zawartych w motywie używanym dla serwisu.

Aby dodać swój własny styl dla składnika niestandardowego, należy potwierdzić wartość `id`, użytą podczas rejestrowania składnika. W pliku `appinfo.json` było to `"id": "hello-world"`.

Używając tej wartości, otworzyć plik `design.json` do edycji, po czym dodać nowe style, które mają być obsługiwane dla tej wartości `id`. Na przykład otworzyć do edycji plik `/designs/default/design.json` ze swojego motywu, po czym dodać następujący kod:

```
"hello-world": {  
  "styles": [{  
    "name": "Plain",  
    "class": "hello-world-default-style"  
  },  
  {  
    "name": "Gothic",  
    "class": "hello-world-gothic-style"  
  }]  
},
```

Jeśli zostanie wywołany panel "Ustawienia" składnika, to na karcie "Styl" powinny być widoczne dwie opcje: **Plain (default)** i **Gothic**. Przełączanie się między tymi opcjami niczego jednak nie zmieni, dopóki nie zostaną zdefiniowane klasy wyszczególnione w pliku `design.css`.

Otworzyć plik `design.css` do edycji, po czym dodać klasy CSS swojego stylu. Na przykład otworzyć do edycji plik `/designs/default/design.css` ze swojego motywu, po czym dodać następujący kod:

```
.hello-world-default-style .scs-component-content {
  font-family: "Helvetica Neue", "Helvetica", "Arial", sans-serif;
  font-size: 24px;
  font-weight: normal; }

.hello-world-gothic-style .scs-component-content {
  font-family: "Century Gothic", "CenturyGothic", "AppleGothic", sans-serif;
  font-size: 32px;
  font-weight: bold; }
```

Zapisać, po czym zsynchronizować pliki z serwerem instancji Oracle Content Management.

#### Sprawdzanie wyników dla etapu 12

1. Aby konstruktor serwisów mógł pobrać zmiany do składnika, odświeżyć stronę serwisu.
2. Przełączyć stronę do trybu edycji.
3. Przeciągnąć składnik na stronę.
4. Wyświetlić panel "Ustawienia" składnika.
5. Przejść do karty "Styl".
6. Przełączać się między stylami **Gothic** i **Plain**, zdefiniowanymi w pliku `design.json`.

Można zaobserwować zmianę rozmiaru czcionki w składniku, spowodowaną przełączeniem stosowanej klasy CSS.

Kontynuacja: [Etap 13: Renderowanie składnika w ramce wstawkowej](#).

## Etap 13: Renderowanie składnika w ramce wstawkowej

Do tej pory w przykładzie był pokazywany składnik lokalny renderowany wstawkowo na stronie. Można także wybrać renderowanie składnika w ramce wstawkowej.

Na przykład można wybrać renderowanie składnika w ramce wstawkowej, jeśli składnik dokonuje takiej aktualizacji strony, przy której zmiana właściwości pociąga za sobą konieczność ponownego utworzenia całej strony. Ponadto składniki odległe są zawsze renderowane w ramce wstawkowej.

Zawarte w tej części przykłady pochodzą z plików tworzonych, gdy podczas tworzenia składnika lokalnego została wybrana opcja **Utwórz składnik, który będzie renderowany w ramce iFrame**. Można także wziąć ten zbiór plików i wprowadzić go na serwerze odległym, tak aby były stosowane tak samo do składników odległych.

Podobieństwa między składnikami zawartymi i niezawartymi w ramach wstawkowych

#### Panel "Ustawienia"

Ponieważ panel "Ustawienia" jest zawsze na stronie umieszczany w ramce wstawkowej, kod panelu "Ustawienia" nie zmienia się bez względu na to, czy składnik używa czy nie używa ramki wstawkowej. Dla obu przypadków jest tworzony ten sam kod panelu "Ustawienia".

#### Sites SDK API

SDK API jest identyczny dla obu przypadków. Ten sam kod jest używany do uaktywniania wyzwalaczy, nasłuchiwanie czynności oraz uzyskiwania i ustawiania wartości właściwości. Mimo że niektóre właściwości mogą nie mieć zastosowania w obu przypadkach (na przykład nie można ustawić właściwości "height" dla składnika, który nie używa ramki wstawkowej), to jednak API nie ulega zmianie. Z tego powodu można kopiować kod między tymi oboma typami składników — przykładowy kod, omawiany w tym samouczku, będzie działał w obu przypadkach.

**Różnice między składnikami zawartymi i niezawartymi w ramach wstawkowych**

### Struktura plików i zależności

Jeśli podczas tworzenia składnika zostanie wybrana opcja **Utwórz składnik, który będzie renderowany w ramce iFrame**, zostaną utworzone następujące pliki:

```
<component name>
  assets
    css
      app-styles.css
    js
      jquery.mn.js
      knockout.mn.js
      sites.min.js
    render.html
    settings.html
  appinfo.json
  _folder_icon.jpg
```

Pliki te pozwalają od razu uruchomić składnik w ramce wstawkowej na stronie. Główne różnice między tą strukturą a strukturą standardowego składnika lokalnego są następujące:

- Zależności JavaScript:
  - Uzyskuje się kompletną kopię tych plików, tak że składnik będzie działał. Pliki te są wymagane do uruchomienia przykładowego składnika w ramce wstawkowej. W zależności od własnych wymagań można dodawać i usuwać pliki do/z tego katalogu.
  - Ponieważ wszystkie pliki z katalogu `assets` składnika są wypychane do serwisu publicznego, gdy składnik jest publikowany, wszystko zawarte w katalogu `js` będzie dostępne zarówno w konstruktorze serwisów, jak i w środowisku wykonawczym.
  - Uwaga: Pliki te są tworzone w celu ułatwienia użytkownika. Zamiast tworzyć ich osobne dla każdego ze składników w ramach wstawkowych, powinno się te pliki konsolidować w motywie lub w innej lokalizacji publicznej.
- `render.html`:
  - Jest to pełny dokument HTML w przeciwieństwie do pliku `render.js` dla składników standardowych, będącego modulem AMD.

### Zarządzanie wysokością przez składnik

Jednym z problemów związanych z używaniem ramki wstawkowej jest zarządzanie jej wysokością. Jeśli nie zostanie to poprawnie zrobione, dla składnika na stronie pojawią się paski przewijania, co może być zjawiskiem niepożądanym.

W celu zarządzania wysokością ramki wstawkowej składnik musi poinformować stronę, jaka ma być wysokość ramki. W przypadku składników odległych mogą pojawić się problemy z obsługą różnych domen; trzeba wówczas — do zlecenia ustawienia wysokości ramki wstawkowej na wymaganą po wyrenderowaniu składnika na stronie — użyć funkcji komunikowania się z pakietu Sites SDK. Służy do tego API `SitesSDK.setProperty('height', {wartość})`. (zob. Oracle Content and Experience — SDK.)

Na przykład można utworzyć funkcję `setHeight` oraz niestandardową procedurę obsługi wiązania wywoływaną, gdy składnik zostanie wyrenderowany na stronie.

- Funkcja aktualizacji wysokości:

```
// set the height of the iFrame for this App
self.setHeight = function () {
  // use the default calculation or supply your own height value as
  a second parameter
  SitesSDK.setProperty('height');
};
```

- Niestandardowa procedura obsługi wiązania Knockout, wywołująca funkcję `setHeight`, gdy składnik zostanie wyrenderowany na stronie lub zmienią się właściwości:

```
ko.bindingHandlers.sampleAppSetAppHeight = {
  update: function (element, valueAccessor, allBindings, viewModel,
bindingContext) {
    // create dependencies on any observables so this handler is called
    whenever it changes
    var imageWidth = viewModel.imageWidth(),
        imageUrl = viewModel.imageUrl(),
        titleText = viewModel.titleText(),
        userText = viewModel.userText();

    // re-size the iFrame in the Sites page now the template has rendered
    // Note: If you still see scrollbars in the iframe after this, it is
    likely that CSS styling in your app is the issue
    viewModel.setHeight();
  }
};
```

- Aktualizacja szablonu do wywoływania procedury obsługi wiązania:

```
<div data-bind="sampleAppSetAppHeight: true"></div>
```

### Rejestracja wyzwalaczy i czynności

Podczas gdy rejestracja wyzwalaczy i czynności dla składników niebędących w ramach wstawkowych jest zawarta w pliku `appinfo.json`, to w przypadku składników w ramach wstawkowych sam składnik jest odpowiedzialny za dostarczenie tych informacji. Służą do tego następujące dwa API:

```
SitesSDK.subscribe('GET_ACTIONS', self.getAppActions);
SitesSDK.subscribe('GET_TRIGGERS', self.getAppTriggers);
```

Oto przykład użycia tych API:

```
// Register TRIGGERS meta-data
SampleAppViewModel.prototype.getAppTriggers = function (args) {
  var triggers = [{
    "triggerName": "imageClicked",
    "triggerDescription": "Image clicked",
    "triggerPayload": [{
      "name": "payloadData",
      "displayName": "Trigger Payload Data"
    }]
  }]

  return triggers;
};

// Register ACTIONS meta-data
SampleAppViewModel.prototype.getAppActions = function (args) {
  var actions = [{
    "actionName": "setImageWidth",
    "actionDescription": "Update the image width",
    "actionPayload": [{
      "name": "imageWidth",
      "description": "Image Width in pixels",
      "type": {
        "ojComponent": {
          "component": "ojInputText"
        }
      },
      "value": ""
    }]
  }]

  return actions;
};
```

### Dostęp do stylów w motywie

Składnik, ponieważ jest renderowany w ramce wstawkowej, nie ma dostępu do stylów zawartych w motywie. Sites SDK udostępnia interfejs API umożliwiający pobranie tych stylów, tak aby mogły one zostać zastosowane do elementów w ramce wstawkowej.

Zagadnienie to jest omówione dokładniej pod hasłem [Etap 14: Używanie stylów niestandardowych, gdy składnik jest renderowany w ramce wstawkowej](#).

### Mieszane użycie protokołów HTTPS i HTTP

Ponieważ Oracle Content Management używa protokołu HTTPS, wszystkie zasoby, do których występuje odwołanie na stronie, także muszą używać protokołu HTTPS. Zasoby zawierają bazowy plik `.html`, który jest renderowany w ramce wstawkowej wraz ze wszystkimi plikami, do których się on odwołuje.

Ten wymóg dotyczy głównie zasobów odległych, ale trzeba jednak pamiętać o tym ograniczeniu. Zasoby dla składników lokalnych używających ramek wstawkowych są udostępniane przez serwer Oracle Content Management, wskutek czego składniki te korzystają z właściwego protokołu.

Kontynuacja: [Etap 14: Używanie stylów niestandardowych, gdy składnik jest renderowany w ramce wstawkowej](#).

## Etap 14: Używanie stylów niestandardowych, gdy składnik jest renderowany w ramce wstawkowej

Składniki, gdy są renderowane w ramce wstawkowej, nie mają bezpośredniego dostępu do pliku `design.css`. Zamiast tego jest wykonywany dodatkowy krok, mający na celu uzyskanie adresu URL dla pliku `design.css` składnika i dodanie tego adresu do strony. Następnie trzeba zaktualizować składnik, tak aby odzwierciedlał wybrany przez użytkownika styl.

Aby dołączyć plik `design.css` i używać go w składniku, trzeba dokonać zmian w pliku `render.html`:

1. Odszukać i dołączyć adres URL do pliku `design.css`.
2. Uzyskać wartość wybranej klasy stylu, gdy się ona zmieni.
3. Zaktualizować szablon, tak aby odzwierciedlał wybraną klasę `styleClass`.
4. Odzwierciedlić w składniku zmiany dokonane w wybranej klasie stylu.
5. Zapewnić zmianę rozmiaru ramki wstawkowej, gdy styl zostanie zmieniony.

Poniżej są podane szczegółowe instrukcje edycji pliku `render.html`:

1. Odszukać i dołączyć adres URL do pliku `design.css`.

Dodać dynamicznie plik `design.css` do sekcji `<head>` strony. Po załadowaniu pliku, ustawić wysokość ramki wstawkowej, ponieważ mogła zostać zmieniona przez stosowanie stylów.

Dodać do obiektu `viewModel` następujący kod:

```
// Dynamically add any theme design URL to the <head> of the page
self.loadStyleSheet = function (url) {
    var $style,
        styleSheetDeferred = new $.Deferred(),
        attempts = 100,
        numAttempts = 0,
        interval = 50,
        pollFunction = function () {
            // try to locate the style sheet
            for (var i = 0; i < document.styleSheets.length; i++) {
                try {
                    // locate the @import sheet that has an href based on
our expected URL
                    var sheet = document.styleSheets[i],
                        rules = sheet && sheet.cssRules,
                        rule = rules && rules[0];
                    // check whether style sheet has been loaded
                    if (rule && (rule.href === url)) {
                        styleSheetDeferred.resolve();
                        return;
                    }
                }
            }
        } catch (e) {}
    };
};
```

```

    }
    if (numAttempts < attempts) {
        numAttempts++;
        setTimeout(pollFunction, interval);
    } else {
        // didn't find style sheet so complete anyway
        styleSheetDeferred.resolve();
    }
};

// add the themeDesign stylesheet to <head>
// use @import to avoid cross domain security issues when
determining when the stylesheet is loaded
$style = $('<style type="text/css">@import url("' + url + '"</
style>');
$style.appendTo('head');

// kickoff the polling
pollFunction();

// return the promise
return styleSheetDeferred.promise();
};

// update with the design.css from the Sites Page
SitesSDK.getSiteProperty('themeDesign', function (data) {
    if (data && data.themeDesign && typeof data.themeDesign ===
'string') {
        // load the style sheet and then set the height
        self.loadStyleSheet(data.themeDesign).done(self.setHeight);
    }
});

```

## 2. Uzyskać wartość wybranej klasy stylu, gdy się ona zmieni.

Utworzyć obiekt "observable" do śledzenia zmian wartości właściwości styleClass :

```
self.selectedStyleClass = ko.observable();
```

Należy pamiętać, że — dopóki nie ma klasy stylu — nie można renderować. Zmienić następujący kod:

```
self.customSettingsDataInitialized = ko.observable(false);
self.initialized = ko.computed(function () {
    return self.customSettingsDataInitialized();
}, self);
```

Zamiast tego użyć następującego kodu:

```
self.customSettingsDataInitialized = ko.observable(false);
self.styleClassInitialized = ko.observable(false);
self.initialized = ko.computed(function () {
    return self.customSettingsDataInitialized() &&
```

```
self.styleClassInitialized();
}, self);
```

Uzyskać początkową wartość wybranej klasy stylu, dodając:

```
self.updateStyleClass = function (styleClass) {
    self.selectedStyleClass((typeof styleClass === 'string') ?
styleClass : 'hello-world-default-style'); // note that this 'hello-
world' prefix is based on the app name
    self.styleClassInitialized(true);
};
SitesSDK.getProperty('styleClass', self.updateStyleClass);
```

3. Zaktualizować szablon, tak aby odzwierciedlał klasę styleClass. Zmienić następujący kod:

```
<p data-bind="attr: {id: 'titleId'}, text: titleText"></p>
```

Zamiast tego użyć następującego kodu:

```
<p data-bind="attr: {id: 'titleId'}, text: titleText, css:
selectedStyleClass"></p>
```

4. Odzwierciedlić w składniku zmiany dokonane w wybranej klasie stylu. Zmienić następujący kod:

```
if (settings.property === 'customSettingsData') {
    self.updateCustomSettingsData(settings.value);
}
```

Zamiast tego użyć następującego kodu:

```
if (settings.property === 'customSettingsData') {
    self.updateCustomSettingsData(settings.value);
}
if (settings.property === 'styleClass') {
    self.updateStyleClass(settings.value);
}
```

5. Zapewnić zmianę rozmiaru ramki wstawkowej, gdy styl zostanie zmieniony. Zmienić następujący kod:

```
// create dependencies on any observables so this handler is called
whenever it changes
var imageWidth = viewModel.imageWidth(),
    imageUrl = viewModel.imageUrl(),
    titleText = viewModel.titleText(),
    userText = viewModel.userText();
```



Zamiast tego użyć następującego kodu:

```
// create dependencies on any observables so this handler is called
whenever it changes
var imageWidth = viewModel.imageWidth(),
    imageUrl = viewModel.imageUrl(),
    titleText = viewModel.titleText(),
    userText = viewModel.userText(),
    selectedStyleClass = viewModel.selectedStyleClass();
```

6. Zapisać, po czym zsynchronizować pliki z serwerem instancji Oracle Content Management.

#### Sprawdzanie wyników dla etapu 14

1. Aby konstruktor serwisów mógł pobrać zmiany do składnika, odświeżyć stronę serwisu.
2. Przełączyć stronę do trybu edycji.
3. Przeciągnąć składnik na stronę.
4. Wyświetlić panel "Ustawienia" składnika.
5. Przejść do karty "Styl".
6. Przełączać się między stylami **Gothic** i **Plain**, zdefiniowanymi w pliku `design.json`.

Można zaobserwować zmianę rozmiaru czcionki w składniku, spowodowaną przełączaniem stosowanej klasy CSS.

Kontynuacja: [Etap 15: Integracja z funkcjami "Cofnij" i "Ponów" strony](#).

## Etap 15: Integracja z funkcjami "Cofnij" i "Ponów" strony

Ponieważ Oracle Content Management przechowuje właściwości na rzecz składnika niestandardowego, zmiany w nich stają się automatycznie elementem działania funkcji **Cofnij** i **Ponów** strony.

Aby mieć pewność, że jest wiadome, co się stanie, gdy użytkownik naciśnie przycisk **Cofnij** lub **Ponów**, te zdarzenia cofania powinny mieć miejsce tylko wtedy, gdy użytkownik faktycznie coś zrobi na stronie. Na przykład wywołanie panelu "Ustawienia" składnika niestandardowego nie powinno spowodować aktualizacji właściwości na stronie, dopóki użytkownik nie zmieni którejś z nich. Samo zainicjalizowanie właściwości w panelu "Ustawienia" nie powinno być przyczyną zdarzenia aktualizacji.

Jeśli takie funkcjonowanie nie zostanie zapewnione, to można mieć do czynienia z nieoczekiwanym działaniem. Strona będzie nadal działała, ale wrażenia użytkownika ulegną pogorszeniu. Na przykład mogą mieć miejsce następujące sytuacje:

- Wywołanie panelu "Ustawienia" spowoduje uaktywnienie przycisku **Zapisz**.
- Użytkownik musi nacisnąć kilka razy przycisk **Cofnij**, aby zaobserwować jakikolwiek skutek tej czynności.
- Stos funkcji `Ponów` zostaje usunięty, ponieważ składnik zapisał nieoczekiwaną zmianę i zaktualizował stos funkcji `Ponów` o nową wartość.

Udostępniony w tym samouczku przykładowy kod dla panelu "Ustawienia" pokazuje, jak zapewnić zapis wsteczny tylko przy wywołaniu metody `saveData`, a nie przy inicjalizacji. Podobne środki ostrożności należy podjąć w odniesieniu do samego składnika, tak aby nie następowała aktualizacja danych `customSettingsData`, dopóki nie nastąpi interakcja użytkownika (choć zazwyczaj jest to mniej istotne).

Kontynuacja: [Etap 16: Zarządzanie zasobami](#).

## Etap 16: Zarządzanie zasobami

Opisano i objaśniono tu, jak zarządzać zasobami używanymi przez składnik.

Do zasobów należą składniki (w tym niestandardowe), o których Oracle Content Management musi wiedzieć, aby móc zarządzać cyklem ich życia.

### Folder zawartości Oracle Content Management

Każdy tworzony w Oracle Content Management serwis jest wyposażony we własny folder `content` zawartości. Jest to folder ukryty, który zazwyczaj nie jest wyświetlany. Gdy serwis jest publikowany, w systemie plików są publikowane wszystkie pliki zawarte w folderze `content`.

Na przykład, jeśli przy użyciu składnika "obraz" zostanie wybrany obraz, Oracle Content Management sporządzi kopię tego wybranego obrazu i umieści ją w folderze zawartości. Używany URL zawsze wskazuje skopiowaną wersję obrazu, tak że jeśli obraz oryginalny zostanie usunięty, serwis nadal będzie poprawnie działał. Dotyczy to także innych składników udostępnianych przez Oracle Content Management, takich jak "galeria", "siatka galerii", "dokument", " pasek społecznościowy", "pobieranie plików", a także obrazów tła dla gniazd i grup składników.

Składnik niestandardowy, aby móc uczestniczyć w cyklu życia zasobu, musi poinformować Oracle Content Management o wszelkich zasobach, którymi usługa ma zarządzać w jego imieniu. Ponieważ wiąże się to ze sporządzeniem kopii zasobu, składnik niestandardowy musi także wybrać zasób — używając zestawów API z Oracle Content Management API — tak aby było wiadomo, jak nim zarządzać.

### Zarządzanie adresami URL

URL zasobu zmienia się w zależności od pewnych kryteriów.

- URL składnika, używany w trybie wykonawczym, różni się od adresu URL, używanego dla składnika przez konstruktor serwisów
- Jeśli zostanie skopiowana strona, Oracle Content Management również sporządzi kopię wszystkich zasobów z folderu zawartości, do których występują odwołania, tak że nigdy nie ma dwóch składników wskazujących ten sam zasób w folderze zawartości
- Jeśli na stronę zostanie przeciągnięta grupa składników (składnik `componentGroup`), zostaną sporządzone kopie wszystkich zasobów, do których odwołuje się którykolwiek zasób zawarty w grupie składników

Ponadto, mimo że względny URL może być odpowiedni dla składnika lokalnego, dla składników odległych jest wymagany w pełni kwalifikowany URL każdego zasobu, którym Oracle Content Management ma zarządzać, dzięki czemu zawartość ramki `iFrame` zasobu może być renderowana z użyciem pełnego adresu URL.

Ponieważ nie można zakładać, że adres URL będzie statyczny, trzeba odwołania do ID utrzymywać w swoim kodzie i pobierać URL zasobu tylko wtedy, gdy zasób ten będzie renderowany.

### Zarządzanie zasobami

Dostępne są następujące Sites SDK API do zarządzania zasobami:

```
SitesSDK.getProperty('componentAssets', callback);
```

- Uzyskuje tablicę bieżących zasobów
- Każdy wpis zasobu składa się z następujących elementów:
  - **id**: Unikatowy ID zasobu.
  - **title**: Metadane tytułu Oracle Content Management.
  - **description**: Metadane opisu Oracle Content Management.
  - **fileName**: Oryginalna nazwa wybranego pliku. Przydatne do wyświetlenia w panelu "Ustawienia" dla składnika niestandardowego (dzięki temu użytkownicy będą wiedzieć, który plik wybrali). Nie jest to nazwa pliku skopiowanego do folderu zawartości.
  - **source**: URL (z obsługą makr) zasobu. Wartość ta będzie się zmieniać i dlatego składnik nie powinien się do niej odwoływać, ale musi zostać zapisana jako część zasoby.
  - **url**: W pełni kwalifikowany URL zasobu, oparty na kontekście, w którym wywołano `getProperty()`.

```
SitesSDK.setProperty('componentAssets', [assets]);
```

- Należy wywołać, aby zapisać wszystkie zasoby, którymi Oracle Content Management ma zarządzać w imieniu bieżącego użytkownika.
- Jeśli ta metoda nie zostanie wywołana, to nie zostanie zapisany żaden zasób.
- Wszelkie zasoby, których nie ma w tej tablicy, zostaną usunięte podczas publikowania serwisu.
- Parametr `assets` ma postać tablicy zasobów w tym samym formacie, w jakim uzyskano je za pomocą metody `getProperty` — tablica ta jest także zwracana przez selektor plików (`filePicker`).

#### Uwaga:

Nie jest składowana żadna wartość `url`. Jest ona dynamicznie tworzona, gdy wystąpi pytanie o zasoby.

```
SitesSDK.filePicker(options, callback);
```

- API do wywołania selektora plików umożliwiające wybranie listy zasobów.
- Przy pomyślnym wyborze zasobów jest uaktywniane wywołanie zwrotne przekazujące ich tablicę.
- W tym momencie nic nie jest zapisywane; to składnik decyduje o wywołaniu metody `setProperty('componentAssets', [assets]);` w celu zapisania elementów z tego wyboru w połączeniu z innymi zasobami, które mają zostać zapisane.

### Przykład wyboru zasobu

Pokazano tu, jak wybrać zasób, składować jego ID oraz ponownie pobrać faktyczne wartości ze składowanych zasobów.

1. Edytować plik `settings.html`.
2. Zmienić obiekt "template", tak aby zawierał blok Image selection.

```
<div>
  <!-- Image selection -->
  <label id="imageLabel" for="imageAsset" class="settings-heading" data-
bind="text: 'Image'"></label>
  <input id="imageAsset" data-bind="value: imageName" readonly
class="settings-text-box">
  <button id="imageSelect" type="button" class="selectbutton" data-
bind="click: showFilePicker">Select Image</button>
</div>
```

3. Zmienić `viewModel`, dodając obiekt "observable" służący do przechowywania ID wybranego zasobu.

```
self.imageID = ko.observable();
```

4. Zmienić `viewModel`, tak aby wybór zasobu odbywał się przez wywołanie selektora plików i była wyświetlana nazwa wybranego zasobu.

```
//
// handle component assets
//
self.assets = []

// bring up a file picker to select the assets
self.showFilePicker = function () {
  // select an image
  SitesSDK.filePicker({
    'multiSelect': false,
    'supportedFileExtensions': ['jpg', 'png']
  }, function (result) {
    if (result.length === 1) {
      // update the array of assets
      self.assets = result;

      // update the image in customSettingsData
      self.imageID(result[0].id);
    }
  });
};

// update the display name based on the assets
self.imageName = ko.computed(function () {
  var imageName = '',
      imageID = self.imageID();
  for (var i = 0; i < self.assets.length; i++) {
    if (self.assets[i].id === imageID) {
```

```

                imageName = self.assets[i].fileName;
                break;
            }
        }

        return imageName
    }, self);

```

5. Zaktualizować viewModel, tak aby przed uzyskaniem customSettingsData były najpierw uzyskiwane zasoby. Kod ten wywoła także self.imageName, gdy obiekt "observable" self.ImageID() ulegnie zmianie.

```

SitesSDK.getProperty('componentAssets', function (assets) {
    self.assets = assets;
    SitesSDK.getProperty('customSettingsData', function (data) {
        //update observable
        self.imageWidth(data.imageWidth);
        self.imageID(data.imageID);
        self.titleText(data.titleText);
        self.userText(data.userText);

        // note that viewModel is initialized and can start saving
data
        self.initialized(true);
        self.saveData = true;
    });
});

```

6. Na koniec zmodyfikować funkcję save, tak aby został zapisany imageID, oraz zaktualizować componentAssets o listę zasobów, do których występują odwołania.

```

self.save = ko.computed(function () {
    var saveconfig = {
        'imageWidth': isNaN(self.imageWidth()) ?
self.imageWidth() : self.imageWidth() + 'px',
        'imageID': self.imageID(),
        'titleText': self.titleText(),
        'userText': self.userText()
    };

    // store the selected asset and update custom settings
    if (self.saveData) {
        SitesSDK.setProperty('componentAssets', self.assets);
        SitesSDK.setProperty('customSettingsData', saveconfig);
    }
}, self);

```

#### Sprawdzanie wyników dla wyboru zasobu

1. Aby konstruktor serwisów mógł pobrać zmiany do składnika, odświeżyć stronę serwisu.
2. Przełączyć stronę do trybu edycji.
3. Przeciągnąć składnik na stronę.

4. Wyświetlić panel "Ustawienia".
5. Nacisnąć przycisk **Wybierz obraz**.
6. Wyszukać (lub wysłać) obraz, a następnie go wybrać.  
Warto zwrócić uwagę, że zostaje składowana nazwa wybranego obrazu.
7. Zamknąć panel "Ustawienia".
8. Wyświetlić ponownie panel "Ustawienia".  
Nazwa obrazu jest ponownie odzwierciedlana.

#### Przykład renderowania zasobu

Pokazano tu, jak pobierać zasoby i renderować je w składniku, oraz dynamicznie aktualizować składnik, gdy zostaną zmienione wartości w panelu "Ustawienia".

#### Uwaga:

Mimo że jest to przykład odnoszący się do składnika lokalnego, zawartego w ramce wstawkowej na stronie, to jednak podobny kod będzie działał dla składników renderowanych miejscowo na stronie.

1. Edytować plik `render.html`.
2. Zmienić obiekt "template", tak aby zawierał zasób.  

```
<!-- ko if: imageURL -->
<div style="flex-shrink:0;">
  <img data-bind="attr: {style: imageStyle, id: 'imageId', src:
imageURL, alt: '', title: ''}, click: imageClicked" />
</div>
<!-- /ko -->
```
3. W modelu widoku (viewModel) utworzyć dwa obiekty "observable" w celu uzyskania wartości `imageID` z `customSettingsData` oraz przechowywania adresu `imageURL` uzyskanego ze składowanej listy zasobów.

```
self.imageID = ko.observable();
self.imageURL = ko.observable();
```

4. Zaktualizować viewModel, tak aby — gdy zmieni się `imageID` — pobierał URL odpowiadającego mu zasobu "obraz".

```
self.imageID.subscribe(function (imageID) {
  // whenever the image changes get the updated referenced asset
  SitesSDK.getProperty('componentAssets', function (assets) {
    for (var i = 0; i < assets.length; i++) {
      if (assets[i].id === imageID) {
        self.imageURL(assets[i].url);
        break;
      }
    }
  })
})
```

```
    });  
  });
```

5. Zaktualizować `viewModel`, tak aby uzyskiwał ID z `customSettingsData`.

#### Sprawdzanie wyników dla renderowania zasobu

1. Aby konstruktor serwisów mógł pobrać zmiany do składnika, odświeżyć stronę serwisu.
2. Przełączyć stronę do trybu edycji.
3. Przeciągnąć składnik na stronę.
4. Wyświetlić panel "Ustawienia".
5. Nacisnąć przycisk **Wybierz obraz**.
6. Wyszukać (lub wysłać) obraz, a następnie go wybrać.  
Warto zwrócić uwagę, że zostaje składowana nazwa wybranego obrazu.
7. Zamknąć panel "Ustawienia".

W tym momencie powinniśmy w składniku widzieć wyrenderowany wybrany obraz.

Kontynuacja: [Samouczek — przegląd](#).

## Samouczek — przegląd

Ten samouczek pokazuje, jak utworzyć dostosowany składnik z użyciem Knockout Component Factory.

Głównym celem tego samouczka jest pokazanie, że — korzystając z tego wzorca — można utworzyć dowolny składnik niestandardowy, aktualizując jedynie obiekty `SampleComponentViewModel` i `sampleComponentTemplate`

JavaScript. Obiekty `sampleComponentFactory` i `SampleComponentImpl` nie zmieniły się, gdy przechodziliśmy przez ten samouczek. Byliśmy w stanie zaimplementować te zmiany bez konieczności zajmowania się komunikacją ze stroną oraz byliśmy w stanie wykonać następujące zadania:

- Komunikowanie (składnikowi) zmian dokonanych w panelu "Ustawienia" i utrwalanie ich.
- Uruchamianie wyzwalaczy i czynności oraz wchodzenie w interakcję z innymi składnikami na stronie.
- Tworzenie układów i wykorzystywanie zagnieżdżonych składników.
- Definiowanie stylów specyficznych dla składników.

W przykładzie tym, w celu zachowania przejrzystości, składnik niestandardowy został rozdzielony na pewną liczbę plików. W celu optymalizacji operacji należałoby rozważyć — aby uniknąć wielu pobrań — odpowiednie spakowanie swoich plików.

Na koniec należy zauważyć, że samouczek ten jest odpowiedni w zakresie składników opartych na technologii Knockout. Zamierzając tworzyć składniki niestandardowe przy użyciu innej technologii JavaScript, takiej jak AngularJS, trzeba ponownie zaimplementować obiekt `SampleComponentImpl`, tak aby zapewnić odpowiednią komunikację z tym środowiskiem oraz implementację składnika właściwą dla tej technologii. To zadanie wykracza poza zakres tego samouczka.

## Sites SDK — opis

Oracle Content Management JavaScript SDK (Sites SDK) zawiera zestaw funkcji umożliwiających składnikom pełniejszą integrację z Oracle Content Management.

Sites SDK można pobrać z serwera Oracle Content Management:

```
http://{server}/_sitesclouddelivery/renderer/app/sdk/js/sites.min.js
```

Definicje Sites API (Application Programming Interface) obejmują wszystkie zdefiniowane przestrzenie nazw globalnego obiektu `SitesSDK`:

- `SitesSDK.getProperty('componentAssets', callback)`
- `SitesSDK.getProperty(propertyName, callback)`
- `SitesSDK.getSiteProperty(propertyName, callback)`
- `SitesSDK.setProperty('componentAssets', [assets])`
- `SitesSDK.setProperty(propertyName, propertyValue)`
- `SitesSDK.filePicker({options}, callback)`
- `SitesSDK.openDocumentPicker(options)`
- `SitesSDK.publish(messageType, payload)`
- `SitesSDK.subscribe(messageType, callback)`
- `SitesSDK.Utils.addSiteThemeDesign(cssUrl)`
- `SitesSDK.Utils.Logger`

Głównym zadaniem Sites SDK jest umożliwienie użytkownikom konstruowania aplikacji i zarządzania nimi. Oferuje kilka możliwości:

- Komunikacja między ramką wstawkową a obszarem domeny
  - Umożliwia składnikowi komunikowanie się ze stroną bez względu na to, gdzie składnik jest hostowany
  - Uwzględnia uczestnictwo w cyklu życia modelu strony:
    - \* tworzenie składnika
    - \* renderowanie składnika przy użyciu składowanych właściwości
    - \* aktualizowanie składnika i zapisywanie dokonanych zmian
    - \* usuwanie składnika ze strony
  - Przechowuje i pobiera ustawienia niestandardowe, używane do konfigurowania składnika do użycia na stronie
  - Używa komunikacji między składnikami niezbędnej dla wyzwalaczy i czynności
- Podpowiedzi dot. geometrii strony (nie dotyczy składników niestandardowych)



- Umożliwia stronie dynamiczną zmianę rozmiaru ramki wstawkowej, gdy zostanie to zleczone przez składnik
- Daje w wyniku składnik wyglądający jak część strony oraz stronę przesuwającą się w odpowiedzi na zdarzenia zmiany rozmiaru składnika (bez pasków przewijania)
- Rozpoznawanie przez składniki, że są wywoływane przez Oracle Content Management
  - Komunikacja między składnikiem a stroną obejmuje token podpisany kluczem prywatnym.
  - Sites SDK umożliwia składnikowi zdekodowanie tokenu oraz ułatwia weryfikację wywołującego.

Funkcje Sites SDK dołącza się, podając znaną ścieżkę skryptu. Na przykład:

```
<script type="text/javascript"
src="<sdk_install_dir_path_prefix>/sites.min.js">
</script>
```

Sites SDK zawiera następujące obiekty globalne:

- SitesSDK
- SitesSDK.Utils

#### SitesSDK

Globalny obiekt `SitesSDK` może być używany w punktach końcowych renderowania i ustawień.

Polecenie	Zadanie
<code>SitesSDK.getProperty(propertyName, callback)</code>	Uzyskuje wartość nazwanej właściwości z serwisu-hosta.
<code>SitesSDK.setProperty(propertyName, propertyValue)</code>	Ustawia wartość nazwanej właściwości na <i>propertyValue</i> .
<code>SitesSDK.getSiteProperty(propertyName, callback)</code>	Uzyskuje wartość nazwanej właściwości serwisu z serwisu-hosta. Na przykład może pobrać bieżący projekt motywu używany przez serwis-host.
<code>SitesSDK.publish(messageType, payload)</code>	Wysła komunikat do serwera i przyjmuje <i>messageType</i> oraz obiekt JSON jako ładunek ( <i>payload</i> ).
<code>SitesSDK.subscribe(messageType, callback)</code>	Nasłuch komunikatów rozsyłanych z serwisu-hosta. Wywołanie jest asynchroniczne. Są to wywołania zwrotne JavaScript, a zatem funkcja powinna używać domknięcia (closure) JavaScript albo odpowiednio powiązać funkcję, tak aby miała dostęp do odpowiedniego kontekstu.

## SitesSDK.Utils

Przestrzeń nazw `SitesSDK.Utils` zawiera funkcje narzędziowe, które mogą być wywoływane przez wszystkie punkty końcowe składnika Oracle Content Management. Jest dostępna dla panelu "Ustawienia" dla składnika odległego, lecz nie jest dostępna dla wstawkowego składnika lokalnego.

### Ostrzeżenie:

Te funkcje nie są oficjalnie obsługiwane przez Oracle; służą jedynie jako przykładowe implementacje. Używanie tych funkcji odbywa się na wyłączne ryzyko bieżącego użytkownika.

Polecenie	Zadanie
<a href="#">SitesSDK.Utils.Logger</a>	Zwraca obiekt logger używany przez Sites SDK. Poziom rejestrowania w dzienniku można ustawić na jeden z następujących: <ul style="list-style-type: none"> <li>• debug</li> <li>• log</li> <li>• info</li> <li>• error</li> <li>• warn</li> </ul>
<a href="#">SitesSDK.Utils.addSiteThemeDesign(cssUrl)</a>	Dodaje do składnika motyw z bieżącego serwisu.
<code>Utils.getStyle(elem,styleProp)</code>	Metoda obliczająca wysokość ramki wstawkowej.

## SitesSDK.getProperty('componentAssets', callback)

Ta funkcja zwraca listę zasobów obecnie przechowywanych na rzecz składnika w obrębie serwisu.

### Parametry

Nazwa	Typ	Opis
<code>componentAssets</code>	Napis	Wywołuje twórcę listy.
<code>callback</code>	Funkcja	Funkcja wywołania zwrotnego

### Użycie

```
// get/set list of assets
SitesSDK.getProperty('componentAssets', callback);
SitesSDK.setProperty('componentAssets', [assets]);

// invoking list creator
// after calling this, the developer needs to call setProperty against
// 'componentAssets' to persist all currently used assets
```

```
//
// args:
// fileType - list of supported file types
// multiSelect - single or multiple file select
//
// returns a list of selected assets. If user cancels out of the
picker,
// the callback is never called
SitesSDK.filePicker({options}, callback);

//
// example 'componentAssets' JSON returned:
//
'componentAssets': [{
  'name': <nameOfAssetInContentFolder>, // this is used to uniquely
and persistently identify the asset. It is typically prefixed by the
originalName; followed by a 16 digit string
  'originalName': <originalName>, // name of the asset as
selected from DOCS
  'description': <description>, // description/other
attributes that are available from DOCS
  'url': <fully qualified url to the asset>
}]
```

## SitesSDK.getProperty(propertyName, callback)

Ta funkcja uzyskuje wartość nazwanej właściwości z serwisu-hosta.

### Parametry

Nazwa	Typ	Opis
propertyName	Napis	Nazwa właściwości.
callback	Funkcja	Funkcja wywołania zwrótnego.

### Powiązane właściwości

Nazwa właściwości	Opis	Ustawienia	Renderowanie
height	Wysokość ramki	ND	Tak
width	Szerokość ramki	ND	Tak
customSettingsData	Dane ustawień	Tak	Tak
styleClass	Wybrana klasa stylu	Tak	Tak

Każdy inny atrybut stylu, na przykład borderColor, borderRadius, alignment, borderStyle, margin

## Użycie

```
SitesSDK.getProperty('customSettingsData', function (propertyData) {
    // store settings data
    self.configuration(
        {
            'id': propertyData.id || self.id,
            'url': propertyData.url || self.params.url,
            'limit': propertyData.limit || self.params.limit
        }
    );
});
```

### Dodatkowe powiązane właściwości pobierane spod adresu URL renderingu

Oprócz używania funkcji `getProperty()` do pobierania wartości właściwości, inne powiązane właściwości, które mogą być pobierane spod adresu URL renderingu, zostały zawarte w poniższej tabeli.

Nazwa właściwości	Opis
id	ID składnika
instance	Zaszyfrowany token
width	Szerokość ramki iFrame
height	Wysokość ramki iFrame
viewMode	Wartości trybu renderowania
locale	Zbiór języków w serwisie lub przeglądarce
settingsId	ID bieżących ustawień

### Dodatkowe właściwości pobierane spod adresu URL ustawień

Oprócz używania funkcji `getProperty()` do pobierania wartości właściwości, inne właściwości, które mogą być pobierane spod adresu URL ustawień, zostały zawarte w poniższej tabeli.

Nazwa właściwości	Opis
currCompId	ID edytowanego składnika
instance	Zaszyfrowany token
width	Szerokość ramki iFrame ustawień
locale	Zbiór języków w serwisie lub przeglądarce
OrigSettingsId	ID pierwotnych ustawień składnika
settingsId	ID nowych ustawień

## SitesSDK.getSiteProperty(propertyName, callback)

Ta funkcja uzyskuje wartość nazwanej właściwości serwisu z serwisu-hosta. Przykładem jest pobieranie bieżącego projektu motywu używanego przez serwis-host.

### Parametry

Nazwa	Typ	Opis
propertyName	napisowy (string)	Nazwa właściwości serwisu
callback	funkcja	Funkcja wywołania zwrotnego, służąca do odczytu wartości właściwości

### Właściwości serwisu

Nazwa właściwości	Opis	Ustawienia	Renderowanie
theme	URL projektu bieżącego motywu	ND	Tak

### Użycie

```
SitesSDK.getSiteProperty('theme',function(data){
    console.log('Theme data ' + JSON.stringify(data));
    // check if we got an url back
    if ( data.url && typeof data.url === 'string' ) {
        if ( data.url !== '' ) {
            // theme is loaded so dynamically inject theme
            SitesSDK.Utils.addSiteThemeDesign(data.url);
        }
    }
});
```

## SitesSDK.setProperty('componentAssets', [assets])

Ta funkcja aktualizuje listę zasobów serwisu przechowywanych na rzecz składnika niestandardowego.

### Parametry

Nazwa	Typ	Opis
componentAssets	Napis	Tworzy listę zasobów składnika.
assets	Obiekt JSON	Zwraca listę wybranych zasobów.

## Argumenty

Nazwa właściwości	Opis
filetypes	Lista obsługiwanych typów plików.
multiSelect	Wybór jednego lub większej liczby plików.

## Użycie

```
// get/set list of assets
SitesSDK.getProperty('componentAssets', callback);
SitesSDK.setProperty('componentAssets', [assets]);

// invoking list creator
// after calling this, the developer needs to call setProperty against
// 'componentAssets' to persist all currently used assets
//
// args:
//   fileTypes - list of supported file types
//   multiSelect - single or multiple file select
//
// returns a list of selected assets. If user cancels out of the picker,
// the callback is never called
SitesSDK.filePicker({options}, callback);

//
// example 'componentAssets' JSON returned:
//
'componentAssets': [{
  'name': <nameOfAssetInContentFolder>, // this is used to uniquely and
  persistently identify the asset. It is typically prefixed by the
  <originalName> followed by a 16 digit string
  'originalName': <originalName>, // name of the asset as selected
  from DOCS
  'description': <description>, // description/other attributes
  that are available from DOCS
  'url': <fully qualified url to the asset>
}]
```

# SitesSDK.setProperty(propertyName, propertyValue)

Ta funkcja ustawia wartość nazwanej właściwości na *propertyValue*.

## Parametry

Nazwa	Typ	Opis
propertyName	Napis	Nazwa właściwości.
propertyValue	Obiekt JSON	Wartość właściwości.

### Powiązane właściwości

W tej tabeli są zawarte powiązane właściwości, które mogą być pobierane przy użyciu funkcji `setProperty()`.

Nazwa właściwości	Opis	Ustawienia	Renderowanie
<code>height</code>	Wysokość ramki	ND	Tak
<code>width</code>	Szerokość ramki	ND	Tak
<code>customSettingsData</code>	Dane ustawień	Tak	Tak
<code>styleClass</code>	Wybrana klasa stylu	Tak	Tak
<code>description</code>	Używana na banerze (na stronie) dla instancji składnika lokalnego		
<code>layout</code>	Zwraca obecnie wybrany układ tylko dla składników lokalnych. Może być aktualizowana z panelu "Ustawienia" na podstawie wartości z pliku <code>appinfo.json</code> .		
<code>renderStatus</code>	Zezwala składnikowi niestandardowemu na zgłoszenie, że renderowanie zostało wykonane.	<code>true</code>	
Każdy inny atrybut stylu, na przykład <code>borderColor</code> , <code>borderRadius</code> , <code>alignment</code> , <code>borderStyle</code> , <code>margin</code>			

### Użycie

```
// configuration that can be saved
var saveconfig = {
    //current settings id
    'id': self.appSettingsProperties['settingsId'],
    'url': self.url(),
    'limit': self.limit()
};
// save property 'customSettingsData'
SitesSDK.setProperty('customSettingsData',saveconfig);
```

## SitesSDK.filePicker({options}, callback)

Ta funkcja zwraca listę wybranych plików.

### Parametry

Nazwa	Typ	Opis
options	Napis	Opcje określające zwracane wybrane pliki.
callback	Funkcja	Funkcja wywołania zwrotnego. Jeśli użytkownik anuluje selektor, wywołanie zwrotne nie zostanie uaktywnione.

### Użycie

```
// get/set list of assets
SitesSDK.getProperty('componentAssets', callback);
SitesSDK.setProperty('componentAssets', [assets]);

// invoking list creator
// after calling this, the developer needs to call setProperty against
// 'componentAssets' to persist all currently used assets
//
// args:
//   fileType - list of supported file types
//   multiSelect - single or multiple file select
//
// returns a list of selected assets. If user cancels out of the picker,
// the callback is never called
SitesSDK.filePicker({options}, callback);

//
// example 'componentAssets' JSON returned:
//
'componentAssets': [{
  'name': <nameOfAssetInContentFolder>, // this is used to uniquely and
  persistently identify the asset. It is typically prefixed by the
  {originalName} followed by a 16 digit string
  'originalName': <originalName>, // name of the asset as selected
  from DOCS
  'description': <description>, // description/other attributes
  that are available from DOCS
  'url': <fully qualified url to the asset>
}]
```



## SitesSDK.openDocumentPicker(options)

Ta funkcja zwraca listę wybranych plików.

Funkcja `SitesSDK.openDocumentPicker()` przyjmuje jeden argument, a następnie zwraca obietnicę, która się przekłada na wybrane dokumenty. Dostępne opcje są zgodne z opcjami składnika `documentsView`, opisanymi pod hasłem *Osadzanie UI widoku dokumentów* (składnik "documentsView") dla API V2 w podręczniku *Integrowanie i rozszerzanie usług*.

### Użycie

```
SitesSDK.openDocumentPicker({
  selectable: "any",
  layout: "grid"
}).then(function (selection) {
  console.log(selection);
});
```

Gdy dokumenty zostaną pobrane, trzeba poinformować usługę Oracle Content Management — za pomocą właściwości `SitesSDK.getProperty('componentAssets', callback)` i `SitesSDK.setProperty('componentAssets', [assets])` — że identyfikatory tych dokumentów mają być przechowywane z serwisem. W przeciwnym razie dokumenty te nie będą publikowane z serwisem i mogą zostać usunięte.

## SitesSDK.openAssetPicker(options)

Ta funkcja zwraca listę wybranych plików.

Funkcja `SitesSDK.openAssetPicker()` przyjmuje jeden argument, a następnie zwraca obietnicę, która się przekłada na wybrane zasoby. Dostępne opcje są zgodne z opcjami składnika `assetsView`, opisanymi pod hasłem *UI osadzania widoku zasobów* (składnik "assetsView") dla API V2 w podręczniku *Integrowanie i rozszerzanie usługi Oracle Content Management*.

### Użycie

```
SitesSDK.openAssetPicker({
  select: "single"
}).then(function (selection) {
  console.log(selection);
});
```

Gdy zasoby zostaną pobrane, trzeba poinformować usługę Oracle Content Management — za pomocą właściwości `SitesSDK.getProperty('componentAssets', callback)` i `SitesSDK.setProperty('componentAssets', [assets])` — że identyfikatory tych zasobów mają być przechowywane z serwisem. W przeciwnym razie zasoby te nie będą publikowane z serwisem i mogą zostać usunięte.

## SitesSDK.publish(messageType, payload)

Ta funkcja wysyła komunikat do serwera. Przyjmuje `messageType` oraz obiekt JSON jako ładunek (payload). Typ komunikatu jest rozpoznawany i obsługiwany na stronie. Jeśli nie ma procedury obsługi, przekazywany typ komunikatu jest ignorowany.

### Parametry

Nazwa	Typ	Opis
<code>messageType</code>	napisowy (string)	Typ komunikatu
<code>payload</code>	Obiekt JSON	Ładunek komunikatu

### Typy komunikatów

W tej tabeli są zawarte typy komunikatów, które są znane składnikom lokalnym (używającym ramek wstawkowych) i odległym oraz które mogą być wysyłane przy użyciu funkcji `SitesSDK.publish()`.

Typ komunikatu	Opis
<code>SETTINGS_UPDATED</code>	Używany w panelu "Ustawienia" do wysyłania zaktualizowanego komunikatu o ustawieniach do końcowego punktu renderingu.
<code>TRIGGER_ACTIONS</code>	Używany w końcowym punkcie renderingu składnika do uaktywniania czynności w serwisie-goście.

### Użycie

```
// raise trigger
SitesSDK.publish(SitesSDK.MESSAGE_TYPES.TRIGGER_ACTIONS, {
    'triggerName': 'scsChangeSettings',
    'triggerPayload': { 'url':
this.configuration()['url'],'feedcount': this.configuration()['limit'] }
});
```

## SitesSDK.subscribe(messageType, callback)

Ta funkcja jest praktycznie nasłuchem komunikatów rozsyłanych z serwisu-hosta. Wywołanie jest asynchroniczne.

Zarejestrowane wywołanie zwrotne (callback) jest wywoływane, gdy SDK otrzyma komunikat określonego typu. Jeśli wywołanie zwrotne zwróci wartość, to następnie zostanie ona zwrócona do strony. Jeśli do tej funkcji nie zostanie przekazane żadne wywołanie zwrotne, to nastąpi usunięcie nasłuchu zarejestrowanego dla tego konkretnego typu komunikatu.

Są to wywołania zwrotne JavaScript, a zatem funkcja powinna używać domknięcia (closure) JavaScript albo odpowiednio powiązać funkcję, tak aby miała dostęp do odpowiedniego kontekstu.

## Parametry

Nazwa	Typ	Opis
messageType	napis	Typ komunikatu: <ul style="list-style-type: none"> <li>• SETTINGS_UPDATED</li> <li>• TRIGGER_ACTIONS</li> <li>• EXECUTE_ACTION</li> <li>• GET_ACTIONS</li> <li>• GET_TRIGGERS</li> <li>• COPY_CUSTOM_DATA</li> <li>• PASTE_CUSTOM_DATA</li> </ul>
callback	funkcja	Funkcja wykonywana, gdy z serwisu-hosta zostanie otrzymany komunikat.

## Typy komunikatów

Typ komunikatu	Opis
SETTINGS_UPDATED	Ten komunikat należy subskrybować, aby były przysyłane powiadomienia, gdy w panelu ustawień zostanie zmieniona jakaś wartość. Zazwyczaj składnik reaguje, aktualizując swoją realizację w interfejsie użytkownika.
TRIGGER_ACTIONS	Publikacja tego komunikatu uaktywnia wyzwalacz i przekazuje ładunek.
EXECUTE_ACTION	Ten komunikat należy subskrybować, aby były obsługiwane czynności niestandardowe. Zazwyczaj nasłuch obsługuje ten komunikat, wykonując podaną czynność.
GET_ACTIONS	Ten komunikat należy subskrybować, aby zadeklarować tablicę czynności, które ten składnik może uruchamiać. Zwraca tablicę czynności.
GET_TRIGGERS	Ten komunikat należy subskrybować, aby zadeklarować tablicę wyzwalaczy, które ten składnik może uaktywniać. Zwraca tablicę wyzwalaczy.
COPY_CUSTOM_DATA	Ten komunikat należy subskrybować, aby było obsługiwane kopiowanie danych (ustawień niestandardowych) do schowka. Zwraca obiekt reprezentujący dane, które mają zostać umieszczone w schowku.
PASTE_CUSTOM_DATA	Ten komunikat należy subskrybować, aby było obsługiwane wklejanie danych (ustawień niestandardowych) ze schowka.

## Użycie

Niektóre nasłuchcy oczekują przekazania danych, a od innych nasłuchów oczekuje się zwracania danych.

### Przykład 1:

Rejestracja nasłuchu komunikatów typu EXECUTE\_ACTION:

```
SitesSDK.subscribe(SitesSDK.MESSAGE_TYPES.EXECUTE_ACTION,  
self.executeAction,self);
```

Zarejestrowana funkcja callback powinna przyjmować argument, jak poniżej:

```
// typical signature of a callback function registered with a message  
// type  
function (args) {  
    var payload = args.detail.message.payload,  
        action = args.detail.message.action,  
        actionName = action && action.actionName;  
  
    // do something here with the payload data  
  
}
```

### Przykład 2:

Nasłuch COPY\_CUSTOM\_DATA jest zazwyczaj implementowany przy użyciu kodu podobnego do następującego:

```
// listen for COPY_CUSTOM_DATA request  
SitesSDK.subscribe(SitesSDK.MESSAGE_TYPES.COPY_CUSTOM_DATA,
```

Metoda copyCustomDataListener() zwraca obiekt reprezentujący dane, które mają zostać skopiowane do schowka, na przykład:

```
// Handle Copy Style (save customSettingsData to the clipboard)  
self.copyCustomDataListener = function() {  
    return {  
        prop1: this.prop1(),  
        prop2: this.prop2()  
    };  
};
```

### Przykład 3:

Żądanie PASTE\_CUSTOM\_DATA przyjmuje dane, ale nic nie musi zwracać:

```
// listen for PASTE_CUSTOM_DATA request  
SitesSDK.subscribe(SitesSDK.MESSAGE_TYPES.PASTE_CUSTOM_DATA,  
pasteCustomDataListener);
```

Wraz z:

```
// Handle Paste Style (apply customSettingsData from the clipboard)  
self.pasteCustomDataListener = function(data) {  
    ...  
};
```

## SitesSDK.Utils.addSiteThemeDesign(cssUrl)

Ta funkcja tworzy element `link` w znaczniku HTML `<head>` bieżącej strony. Źródło jest ustawione na ścieżkę `cssUrl`.

### ▲ Ostrzeżenie:

Funkcje w przestrzeni nazw `Utils` nie są oficjalnie obsługiwane przez Oracle. Służą jako przykładowe implementacje. Używanie tych funkcji odbywa się na wyłączne ryzyko bieżącego użytkownika.

### Parametry

Nazwa	Typ	Opis
<code>cssUrl</code>	napisowy (string)	Ścieżka URL bieżącego projektu motywu. Należy używać tylko ze składnikami odległymi. Składniki lokalne (realizowane w ramce <code>iFrame</code> ) muszą pobierać właściwości ze strony, a nie z adresu URL.

### Użycie

Ta funkcja jest zwykle używana w połączeniu z pobieraniem bieżącego projektu motywu z serwisu hosta, jak pokazano w przykładowym kodzie:

```
// fetch current theme design from host site and then add it to the page

SitesSDK.getSiteProperty('theme',function(data){
    // check if we got an url back
    if ( data.url && typeof data.url === 'string' ) {
        if ( data.url !== '' ) {
            // theme is loaded so dynamically inject theme
            SitesSDK.Utils.addSiteThemeDesign(data.url);
        }
    }
});
```

## SitesSDK.Utils.Logger

Ta funkcja zwraca obiekt logger używany przez narzędzie Sites SDK.

### ▲ Ostrzeżenie:

Funkcje w przestrzeni nazw `Utils` nie są oficjalnie obsługiwane przez Oracle. Służą one jako przykładowe implementacje. Używanie funkcji `Utils` odbywa się na wyłączne ryzyko bieżącego użytkownika.

Poziom rejestrowania w dzienniku można ustawić na jeden z następujących: debug, log, info, error, warn.

## Rozwiązywanie problemów

Podczas tworzenia serwisu mogą pojawić się pewne problemy. Można je łatwo rozwiązać.

- Podczas próby utworzenia serwisu nie widać żadnych szablonów
- Nie można usunąć serwisu
- Nie można otworzyć drzewa serwisu ani edytować strony
- Został dodany składnik, lecz nie pojawia się on na stronie
- Składniki "foldery", "pliki" i "konwersacje" nie działają
- Została wysłana nowa wersja obrazu, lecz nie pokazuje się on na stronie
- Po zmianie układu strony znikła część jej zawartości
- Dodano składnik oparty na innej usłudze, lecz ten składnik nie działa
- W moim firmowym serwisie jest wyświetlane ostrzeżenie

### Podczas próby utworzenia serwisu nie widać żadnych szablonów

Szablony — aby można było utworzyć z nich serwis — muszą zostać zainstalowane i udostępnione.

Szablony oferowane przez Oracle Content Management muszą — aby można było z nich korzystać — zostać zainstalowane i udostępnione przez administratora. Proszę się skontaktować z administratorem.

Analogicznie, po utworzeniu szablonu (poprzez importowanie, kopiowanie lub utworzenie z serwisu), nikt nie może z niego korzystać, dopóki szablon ten nie zostanie jawnie udostępniony.

Zob. Konfigurowanie ustawień serwisów w podręczniku *Administering Oracle Content Management*.

### Nie można usunąć serwisu

Poniżej przedstawiono kilka powodów, dlaczego nie można usunąć serwisu:

- Serwis działa w trybie online.  
Najpierw trzeba przełączyć serwis do trybu offline, a dopiero potem usunąć. Tylko osoba mająca przypisaną, w odniesieniu do serwisu, rolę menedżera może przełączyć serwis do trybu offline.
- Brak wymaganych uprawnień.  
Serwis może zostać usunięty przez jego twórcę (właściciela serwisu) lub osobę, której ten serwis został udostępniony i której została przypisana rola współtwórcy lub menedżera.

Szczegółowe informacje są dostępne pod hasłem [Zarządzanie serwisem i ustawienia serwisów](#).

## Nie można otworzyć drzewa serwisu ani edytować strony

Konstruktor serwisów domyślnie otwiera się w trybie podglądu, w którym można zobaczyć serwis, lecz nie można go edytować. Serwis można edytować tylko wtedy, gdy edytor działa w trybie edycji.

Upewnić się, że przełącznik  jest ustawiony w pozycji **Edytuj**.

## Został dodany składnik, lecz nie pojawia się on na stronie

Istnieje kilka możliwych przyczyn.

- Sprawdzić poprawność adresu URL powiązanego ze składnikiem odległym (aplikacją). Zob. [Rejestrowanie składników odległych](#).
- Edytor ujmuje składniki odległe w ramkę wstawkową HTML (znacznik <iframe>). Jednak nie wszystkie składniki odległe mogą być umieszczane we wstawkowej ramce.

O tym, czy jest to możliwe, trzeba się dowiedzieć od dostawcy składnika.

## Składniki "foldery", "pliki" i "konwersacje" nie działają

Składniki, które komunikują się z Oracle Content Management, wymagają określonych zasobów i ustawień.

Następujące składniki wymagają dostępu do zasobów w Oracle Content Management:

- Lista folderów
- Lista plików
- Menedżer dokumentów
- Konwersacja

Składniki "lista folderów" i "lista plików" wymagają dostępu do REST API.

Należy sprawdzić, czy są spełnione następujące wymagania:

- Składnik, aby mógł korzystać z operacji REST dotyczących plików i folderów, musi mieć dostęp do punktu końcowego API.

Domyślnie punkt końcowy Oracle Content Management REST API jest dostępny, jeśli jest używany standardowy URL podany dla serwisu. Jeśli jest używany własny URL, może okazać się konieczne jawne udzielenie dostępu do punktu końcowego "/documents" dla REST API.

Zob. [Jak odwzorować URL serwisu?](#)

## Po zmianie układu strony znikła część jej zawartości

Jeśli zostanie wybrany układ z mniejszą liczbą gniazd lub z gniazdami mającymi inne nazwy, istniejąca w innych gniazdach zawartość nie zostanie wyświetlona w nowym układzie.



Zawartość nie zostanie usunięta; nie będzie jednak można jej wyświetlić, jeśli wybrany układ nie będzie miał gniazda o tej samej nazwie.

Należy się upewnić, że został wybrany układ z tą samą liczbą gniazd. Jeśli wybrany układ ma tę samą liczbę gniazd, lecz część zawartości nadal się nie wyświetla, to prawdopodobnie gniazda mają inne nazwy. Należy się zwrócić do projektanta motywu o wyeliminowanie tej rozbieżności.

## Została wysłana nowa wersja obrazu, lecz nie pokazuje się on na stronie

Gdy zostanie wybrany obraz, który ma być używany wraz ze składnikiem, w serwisie jest tworzona i składowana unikatowa kopia obrazu.

Wysyłane pliki są składowane w repozytorium serwisu. Jeśli zostanie wysłany plik, którego nazwa jest identyczna z nazwą pliku już istniejącego, jest tworzona nowa wersja pliku.

Widziany na stronie plik obrazu jest kopią pliku wybranego z repozytorium, przy czym plik ten nie jest automatycznie aktualizowany. Ma to na celu zapobieganie nieodwracalnym zmianom w istniejących stronach, gdy plik zostanie zaktualizowany w repozytorium.

Aby nastąpiła aktualizacja obrazu używanego ze składnikiem, trzeba jawnie wybrać obraz na stronie właściwości składnika. Szczegółowe informacje są dostępne pod hasłem [Obrazy](#).

## Dodano składnik oparty na innej usłudze, lecz ten składnik nie działa

Serwis można integrować z różnymi procesami i usługami.

Jeśli składnik został utworzony w oparciu o integrację z usługą Process Cloud Service, Oracle Intelligent Advisor (dawniej Oracle Policy Automation) i/lub innymi usługami, obie usługi muszą używać tej samej domeny tożsamości. Należy dowiedzieć się od osoby, która skonfigurowała integrację, czy te dwie usługi używają tej samej domeny tożsamości.

## W moim firmowym serwisie jest wyświetlane ostrzeżenie

Jeśli firmowy serwis zawiera układy zawartości w wersji 1.0, to w konstruktorze serwisów oraz podczas wyświetlania serwisu będzie widoczne ostrzeżenie, ponieważ tych układów nie będzie można wyrenderować. Trzeba je zaktualizować do układów zawartości w wersji 1.1.

Taka sytuacja może wystąpić w dwóch przypadkach:

- Jeśli istniejący serwis zostanie uczyniony tłumaczalnym, to nastąpi jego automatyczne przekształcenie w serwis firmowy poprzez przypisanie do niego języka domyślnego oraz założenia systemowego dot. lokalizacji. Jeśli szablon serwisu zawiera układy zawartości w wersji 1.0, to tych układów nie będzie można wyrenderować.
- Jeśli firmowy serwis zostanie utworzony ze starszego szablonu, w którym są używane układy zawartości w wersji 1.0, to tych układów nie będzie można wyrenderować.

Problem ten można rozwiązać, aktualizując układy zawartości do wersji 1.1. Aby zobaczyć różnicę, można utworzyć nowy układ zawartości w Katalogu składników. Trzeba będzie dodać linię rejestracji wersji układu zawartości oraz zmienić `data.fieldname` na `fields.fieldname`.