

Oracle® Cloud

Построение сайтов с помощью Oracle Content Management



F26959-20
June 2021



Oracle Cloud Построение сайтов с помощью Oracle Content Management,

F26959-20

Copyright © 2018, 2021, Корпорация Oracle и/или аффилированные с ней лица.

Основной автор: Bruce Silver

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

Участники: 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.

Содержание

Предисловие

Аудитория	xvi
Доступность документации для лиц с ограниченными возможностями	xvi
Многообразие и инклюзивность	xvi
Связанные ресурсы	xvii
Условные обозначения	xvii

Часть I Введение

1 Обзор Oracle Content Management

Доступ к Oracle Content Management	1-1
Общие сведения о ролях	1-2
Управление активами	1-2
Управление содержимым	1-3
Совместная работа над контентом	1-3
Создание сайтов	1-3
Интеграция и расширение Oracle Content Management	1-4
Начало работы	1-4
Миграция в Oracle Cloud Infrastructure	1-5

2 Начало работы с построением сайтов

Начало работы с сайтами	2-1
Общие сведения о процессе создания сайта	2-2
Подробнее о разработке сайта	2-4
Общие сведения об управлении сайтами	2-6

3 Создание своего первого веб-сайта

Перед началом работы	3-1
Шаг 1. Настройка среды	3-2

Импорт пользовательского компонента Minimal-NavMenu	3-2
Публикация пользовательского компонента Minimal-NavMenu	3-3
Добавление образцов изображений	3-4
Шаг 2. Настройка веб-сайта	3-6
Создание веб-сайта	3-6
Редактирование веб-сайта	3-7
Шаг 3. Публикация веб-сайта	3-27
Дополнительные действия	3-30

Часть II Создание и редактирование сайтов

4 Создание сайтов

Создание сайтов	4-1
Копирование сайтов	4-4
Управление запросами сайта	4-5
Изменение сведений о запросе сайта	4-7
Просмотр политик запросов сайта	4-7

5 Редактирование сайтов

Вводные сведения о странице "Site Builder"	5-1
Общие сведения об обновлениях сайтов	5-4
Использование обновления	5-6
Редактирование рекомендаций и советов	5-7
Использование стилей и форматирование	5-11
Настройки сайта	5-12
Добавление пользовательских свойств сайта	5-14
Работа с таблицами	5-15
Загрузка файлов сайта	5-17

6 Использование шаблонов и тем для сайтов

Общие сведения о шаблонах	6-1
Создание шаблона на основе сайта	6-5
Изменение сведений о шаблоне	6-6
Изменение политик шаблона	6-7
Изменение аудитории или статуса шаблона	6-9
Управление шаблонами	6-9
Экспорт и импорт шаблонов	6-13
Общие сведения о темах	6-15

Управление темами	6-16
Публикация тем	6-18

7 Управление пользовательскими компонентами и макетами

Общие сведения о пользовательских компонентах	7-1
Общие сведения о макетах	7-3
Использование пользовательских компонентов и макетов	7-5
Регистрация удаленных компонентов	7-8
Создание локальных компонентов, макетов, редакторов полей контента или форм контента	7-9
Экспорт и импорт компонентов или макетов	7-11

8 Работа со страницами сайта

Переход к странице	8-1
Просмотр страниц	8-1
Добавление страниц	8-3
Перемещение страниц	8-6
Удаление страниц	8-7
Изменение настроек страницы	8-8
Изменение макета страницы	8-11
Изменение фона или темы	8-12

9 Организация содержимого страницы

Добавление компонентов и макетов разделов	9-1
Работа с активными элементами содержимого	9-4
Использование триггеров и действий	9-4
Использование горизонтальных макетов разделов	9-6
Использование макетов с двумя и тремя столбцами	9-7
Использование вертикальных макетов разделов	9-9
Использование макетов разделов с вкладками	9-10
Использование макетов разделов со слайдами	9-11

10 Использование встроенных компонентов

Основные компоненты	10-2
Заголовки	10-2
Разделы	10-4
Обычный текст	10-6
Кнопки	10-7

Структурные компоненты	10-9
Разделители	10-9
Разделители	10-10
Медиа-компоненты	10-10
Изображения	10-11
Галереи	10-13
Сетка галереи	10-17
Видео YouTube	10-21
Видео	10-22
Компоненты документов	10-24
Документы	10-24
Список папок	10-25
Списки файлов	10-26
Диспетчер документов	10-28
Библиотека проекта	10-30
Социальные компоненты	10-32
Социальная панель	10-33
Кнопки "Нравится" и "Рекомендую" на Facebook	10-34
Как поделиться и подписаться в Twitter	10-34
Компонент диалога	10-35
Список диалогов	10-38
Компоненты процесса	10-39
Форма запуска процесса	10-39
Список задач процесса	10-41
Форма сведений о задаче	10-44
Элементы содержимого	10-46
Компонент элемента содержимого	10-46
Заполнитель содержимого	10-48
Список содержимого	10-49
Поиск в содержимом	10-56
Рекомендация	10-58
Другие компоненты	10-59
Карты	10-60
Заголовки	10-61
Статьи	10-63
Изображения с текстом	10-65
Группы компонентов	10-67
Использование компонента "Совместный просмотр" на странице	10-68
Oracle Intelligent Advisor	10-69
Oracle Visual Builder	10-70

11 Управление сайтами

Вводные сведения о странице "Сайты"	11-1
Управление сайтами и настройки сайтов	11-3
Изменение описания сайта, логотипа или внедренных свойств	11-8
Настройка свойств поисковых систем	11-9
Включение сервиса предварительного рендеринга для поисковой оптимизации	11-11
Задание параметров доставки статического сайта	11-13
Включение автоматической компиляции при публикации	11-13
Переопределение заголовков управления кэшем по умолчанию для скомпилированных сайтов	11-14
Указание мобильных агентов пользователей для поддержки скомпилированных адаптивных макетов	11-14
Указание и настройка персонализированных URL-адресов	11-15
Включение интеграции с Cobrowse	11-18
Добавление отслеживания аналитики	11-19

12 Публикация сайтов

Перевод сайта в онлайн-режим или автономный режим	12-1
Публикация изменений сайта	12-2

13 Безопасность сайтов

Общие сведения о безопасности сайта	13-1
Изменение настроек безопасности сайта	13-4

14 Работа с многоязычными сайтами

Обзор многоязычных сайтов	14-1
Перевод сайта	14-2
Управление заданиями перевода сайтов	14-4
Языковые стандарты для перевода	14-5
Пользовательские региональные параметры для перевода	14-5
Настройка псевдонима региональных параметров для перенаправления URL-адреса	14-6

15	Использование перенаправление сайтов или сопоставления URL-адресов	
	Планирование перенаправлений	15-1
	Простое сопоставление строк	15-1
	Упрощенное применение символов шаблона	15-2
	Добавление перенаправлений сайта	15-2
	Задание правил перенаправлений в файле JSON	15-3
	Загрузка файла правил перенаправления на сайт	15-9
	Сопоставление URL-адреса сайта	15-10
16	Повышение производительности сайта	
	Использование кэширования для повышения производительности	16-1
	Кэширование времени выполнения	16-2
	Кэширование в Site Builder	16-3
	Визуализация без прокрутки (ATF)	16-3

Часть IV Разработка для сайтов

17	Настройка проектов и стилей	
	О проектах	17-1
	Проектные файлы	17-1
	Дизайн адаптивной таблицы	17-4
	Настройка стилей списка диалогов	17-8
	Настройка стилей списков папок и файлов	17-10
	Настройка значков социальной панели	17-11
	Настройка новых расширений стилей интервью для Oracle Intelligent Advisor	17-12
18	Общие сведения об использовании фона	
	Общие сведения о фоне и темах	18-1
	Порядок реализации фона	18-1
	Где хранятся настройки	18-2
19	Задание триггеров и действий	
	Использование триггеров и действий	19-1
	Задание триггеров	19-2

Установка действий	19-3
--------------------	------

20 Разработка шаблонов

О шаблонах	20-1
Структура базового шаблона	20-2
Создание шаблона	20-4
Экспорт шаблона	20-5
Импорт шаблона	20-6
Работа с начальным шаблоном	20-7
Создание шаблона сайта на основе шаблона Bootstrap или шаблона разработки веб-сайтов	20-11
Разработка шаблонов в Developer Cloud Service	20-17
Вход в консоль Developer Cloud Service для Oracle Content Management	20-17
Создание шаблона в Developer Cloud Service	20-18
Создание шаблонов в Developer Cloud Service	20-18
Копирование шаблона в Developer Cloud Service	20-19
Импорт шаблона в Developer Cloud Service	20-20
Объединение изменений	20-20
Экспорт шаблона из Developer Cloud Service	20-20

21 Разработка тем

О темах	21-1
Базовая структура темы	21-3
Навигация на сайте	21-5
Создание темы	21-7
Скрытие компонентов и макетов разделов для темы	21-12
Скрытие параметров компонента выравнивания, ширины или интервала для темы	21-14
Связывание компонентов с темами	21-16
API-интерфейс рендеринга сайтов	21-19

22 Разработка макетов

О макетах	22-1
Поисковая оптимизация (SEO)	22-3
Общие сведения о файле components.json и формате	22-5
Настройка групп панелей инструментов в Site Builder	22-9
Ограничение компонентов в ячейках	22-12
Как сделать контент макета редактируемым	22-15
Создание макета раздела	22-17
Создание макета раздела с поддержкой медленной загрузки (Lazy Load)	22-18

Разработка пользовательских макетов разделов с помощью API	22-19
Разработка макетов контента	22-25
Создание макетов контента с помощью Oracle Content Management	22-27
Передача представления макета макету контента	22-33
Создание URL-адреса страницы сведений о сайте с помощью API	22-34
Локальная разработка макетов контента с помощью Developer Cloud Service	22-35
Создание макета контента с помощью Developer Cloud Service	22-36
Определение модуля RequireJS	22-37
Настройка параметра функции Constructor	22-38
Рендеринг макета контента	22-39
Редактирование макета контента в шаблоне Mustache	22-39
Добавление динамического преобразования DOM	22-40
Определение стилей в файле design.css	22-40
Получение ссылочных позиций	22-40
Получение URL-адреса медиаданных	22-41
Вызов триггеров	22-41
Переход на страницу поиска с помощью запроса поиска	22-41
Развертывание макрокоманд и рендеринг RTF	22-42
Ссылка на страницу сведений	22-43
Разворачивание макросов в запросах списков содержимого	22-43
Разработка надежных макетов контента	22-47
Визуализация элементов контента	22-47
Стандартизация структуры данных для макета контента	22-48
Создание примера шаблона блога	22-52
Добавление сопоставлений макетов контента к шаблонам	22-53
Тестирование макетов контента с помощью локальной тестовой оснастки	22-53
Тестирование с помощью локальной тестовой оснастки	22-53
Импорт шаблонов с макетами контента в Oracle Content Management	22-54

23 Разработка компонентов

О компонентах	23-1
О разработке компонентов	23-3
Создание компонента	23-6
Разработка пользовательских компонентов с помощью Developer Cloud Service	23-7
Разработка пользовательских компонентов для Oracle Content Management	23-7
Разработка пользовательского компонента	23-8
Запись и выполнение модульных тестов	23-10
Оптимизация компонентов (уменьшение) для повышения производительности	23-11
Выполнение заданий непрерывной интеграции	23-12
Разработка переводимых компонентов для многоязычных сайтов	23-12

Построение компонента H1 с помощью панели настроек	23-13
Создание нового локального компонента	23-13
Построение базового компонента H1	23-14
Добавление CSS для компонента	23-17
Добавление панели настроек для изменения текста заголовка	23-18
Обновление темы для выбора стиля компонента H1 другими пользователями	23-20
Создание простого компонента с помощью HTML-шаблонов	23-21
Создание HTML-компонента	23-22
Добавление в HTML-компонент данных, предоставленных пользователем	23-25
Сравнение локальных компонентов с удаленными	23-30
Настройки рендеринга компонентов	23-31
Реализация локального компонента	23-32
Классы стилей для компонентов	23-33
Порядок создания стилей встроенных компонентов	23-35
Основы создания стилей компонентов	23-36
Стили для компонентов	23-37
Задание свойств компонентов	23-44
Компоненты с рендерингом в задаваемых в строках фреймах	23-45
Об идентификаторе экземпляра и структуре компонентов с рендерингом в задаваемых в строках фреймах	23-47
Безопасность удаленных компонентов	23-48
Регистрация удаленного компонента	23-50
Удаление компонента	23-51
Sites SDK	23-52

24 Настройка файла контроллера

О файле контроллера	24-1
Файл контроллера по умолчанию	24-1
Изменение файла по умолчанию controller.html	24-2
Об объекте SCS	24-3
SCS.sitePrefix	24-3
SCS.preInitRendering	24-4
SCS.getDeviceInfo	24-4
Разделы файла контроллера, настраивать которые не нужно	24-4
Использование маркеров для обеспечения переносимости файлов пользовательского контроллера	24-5
Примеры файлов пользовательского контроллера	24-6
Изменение префикса сайта	24-7
Настройка графика ожидания	24-8
Настройка значков сайтов	24-9
Настройка тегов <noscript> и <meta> для обработчиков без JavaScript	24-10

Предварительная выборка файлов JavaScript	24-10
Проверка прав владения сайтом с помощью дополнительной разметки	24-11
Обнаружение дополняющих устройств	24-12
Использование маркеров для улучшения переносимости файла controller.htm	24-13

Часть V Разработка сайтов с помощью инструментария других программ

25 Разработка с помощью инструментария OCE

Настройка инструментария OCE на локальном компьютере	25-1
Установка зависимых компонентов с помощью команды npm	25-2
Использование утилиты командной строки CEC	25-2
Тестирование с помощью локальной тестовой оснастки	25-67
Обновление до jQuery 3.5.x	25-68
Разработка для Oracle Content Management в Developer Cloud Service	25-68
Об использовании Developer Cloud Service	25-69
Вход в консоль Developer Cloud Service для Oracle Content Management	25-70
Создание шаблона в Developer Cloud Service	25-70
Создайте проект Developer Cloud Service с помощью шаблона Oracle Content Management	25-71
Создание проекта в Developer Cloud Service с помощью инструментария OCE из Oracle Content Management	25-71
Добавление инструментария OCE в код проекта в новом репозитории GIT	25-71
Тестирование пользовательских компонентов, шаблонов и макетов контента с использованием локальной тестовой оснастки	25-72
Объединение изменений	25-72
Распространение изменений из среды тестирования в среду производства с помощью инструментария OCE	25-72
Шифрование пароля	25-79
Регистрация сервера	25-79
Создание отчета об использовании и разрешениях для сайта	25-79
Выгрузка и загрузка документов и папок	25-80
Создание сайта на основе шаблона и сохранение одинаковых GUID для контента	25-81
Создание корпоративного шаблона на основе стандартного сайта	25-81
Импорт и экспорт таксономий	25-81
Импорт и экспорт рекомендаций	25-83
Добавление или удаление контента коллекции	25-84
Разработка пользовательских редакторов полей с помощью инструментария OCE	25-85
Перенос сайта с одного сервера на другой или обновление	25-89
Передача сайта без элементов контента	25-90
Выгрузка или загрузка элементов контента для сайта в группах	25-90

Индексирование страниц сайта при помощи инструментария OCE	25-91
Создание типа контента для текста страницы сайта	25-91
Создание элементов контента индекса страниц с помощью инструментария OCE	25-92
Добавление поиска контента на сайт в Oracle Content Management	25-93
Добавление страницы поиска к сайту	25-93
Добавление поля поиска в тему	25-94
Индексирование многоязычного сайта с помощью инструментария OCE	25-95
Создание упрощенного компонента для облегчения разработки компонентов	25-97
Настройка сервиса компиляции сайтов	25-98
Компиляция сайта для повышения производительности во время выполнения для страниц сайта	25-99
Обзор компиляции сайта	25-99
Взаимодействие с предварительным рендерингом	25-99
Рендеринг страниц сайта контроллера	25-100
Рендеринг скомпилированных страниц сайта	25-100
Компиляция шаблонов	25-101
Настройка	25-101
Компиляция своих шаблонов	25-102
Компиляция сайта	25-108
Предварительные условия компиляции сайта	25-108
Компиляция сайта	25-108
Пользовательские компиляторы	25-109
Ограничения	25-110
Отладка настраиваемых компиляторов	25-110
Компиляторы макетов страниц	25-111
Компиляторы компонентов	25-112
Компиляторы макетов контента	25-114
SCSCCompileAPI	25-115
Гидратация компонентов	25-116
Публикация	25-119
Приоритет доставки статического сайта	25-119
Заголовки кэширования	25-120
Страницы сведений	25-120
Добавление элементов контента в канал	25-121
Компиляция сайта для мобильных устройств	25-122
Жизненный цикл сайта и скомпилированные страницы	25-123
Создание новой задачи перевода сайта или актива на сервере Oracle Content Management	25-123
Переводите сайт с помощью поставщика переводческих услуг	25-126
Создание задания перевода с помощью инструментария OCE	25-127
Перечисление заданий перевода	25-128
Создание коннектора перевода	25-128

Создание карты сайта для многоязычного сайта	25-129
Отправка задания перевода поставщику переводческих услуг	25-130
Загрузка задания перевода на сервер	25-131

Часть VI Приложения

26 Учебное пособие. Разработка компонентов с помощью Knockout

Общие сведения и предварительные требования для разработки компонентов с помощью Knockout	26-1
Шаг 1. Создание компонента	26-2
Шаг 2. Проверка структуры рендеринга локального компонента	26-4
Шаг 3. Проверка структура настроек локального компонента	26-8
Шаг 4. Отображение нового свойства в компоненте	26-12
Шаг 5. Регистрация триггеров	26-13
Шаг 6. Срабатывание триггеров	26-14
Шаг 7. Регистрация действий	26-17
Шаг 8. Выполнение действий	26-18
Шаг 9. Создание отдельного заголовка для каждого экземпляра компонента	26-20
Шаг 10. Использование вложенных компонентов с внутрискриптным редактированием	26-21
Шаг 11. Поддержка различных макетов	26-23
Шаг 12. Определение пользовательских стилей	26-25
Шаг 13. Рендеринг компонента во встроенной рамке	26-26
Шаг 14. Использование пользовательских стилей во время рендеринга компонента во встроенной рамке	26-30
Шаг 15. Интеграция с процессом выполнения на странице операций "Отмена" и "Повтор"	26-33
Шаг 16. Управление активами	26-34
Обзор учебного пособия	26-39

27 Ссылка на Sites SDK

SitesSDK.getProperty('componentAssets', callback)	27-4
SitesSDK.getProperty(propertyName, callback)	27-4
SitesSDK.getSiteProperty(propertyName, callback)	27-6
SitesSDK.setProperty('componentAssets', [assets])	27-7
SitesSDK.setProperty(propertyName, propertyValue)	27-8
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
SitesSDK.Utils.Logger	27-14

28 Устранение неполадок

Я пытаюсь создать сайт, но отсутствуют шаблоны	28-1
Не удается удалить сайт	28-1
Я не могу открыть дерево сайта и отредактировать страницу	28-2
Компонент добавлен, но не отображается на странице	28-2
Компоненты папки, файла и диалога не работают	28-2
Часть содержимого исчезла после изменения макета страницы	28-3
Новая версия изображения не отображается на странице после загрузки	28-3
Не работает добавленный компонент, основанный на другом сервисе	28-3
На моем корпоративном сайте отображается предупреждение	28-4

Предисловие

В этом документе описано, как создавать и управлять компонентами взаимодействия, включая сайты, активы и все связанные с ними структуры и политики, с помощью Oracle Content Management.

Аудитория

Данный документ предназначен для пользователей Oracle Cloud, планирующих работать с файлами в веб-браузере и синхронизировать файлы с помощью настольного приложения или мобильного устройства.

Доступность документации для лиц с ограниченными возможностями

Для получения сведений об обязательствах корпорации Oracle в отношении доступности для пользователей с ограниченными возможностями посетите сайт программы доступности Oracle: <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Доступ к службе поддержки Oracle

Клиенты Oracle, которые приобрели пакет услуг по поддержке, имеют доступ к электронной службе технической поддержки My Oracle Support. Для получения сведений посетите страницу <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> или <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> (для пользователей с нарушениями слуха).

Многообразие и инклюзивность

Корпорация Oracle всецело привержена принципам многообразия и инклюзивности. Корпорация Oracle уважает и ценит многообразие сотрудников, которое помогает развивать интеллектуальное лидерство и инновации. В рамках инициативы по формированию более инклюзивной культуры, благотворно воздействующей на сотрудников, клиентов и партнеров, мы удаляем неpolitкорректные термины из интерфейсов наших продуктов и документации. В то же время мы осознаем необходимость поддерживать совместимость с существующими технологиями наших клиентов и обеспечивать их непрерывное обслуживание в процессе развития предложений Oracle и отраслевых стандартов. По причине этих технических ограничений работа по удалению из обращения неpolitкорректных терминов еще не закончена. Чтобы ее завершить, потребуются время и участие третьих сторон.

Связанные ресурсы

За дополнительной информацией обращайтесь к следующим ресурсам 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*

Условные обозначения

В тексте этого документа используются следующие обозначения.

Обозначение	Значение
полужирный	Полужирным шрифтом обозначаются элементы графического интерфейса пользователя, связанные с действиями, и термины, определяемые в тексте или глоссарии.
<i>курсив</i>	Курсивом обозначаются заголовки книг, выделенные сведения или переменные-местозаполнители, которым пользователь присваивает конкретные значения.
monospace	Моноширинным шрифтом указываются команды внутри абзаца, адреса URL, примеры кода, отображаемый на экране или вводимый пользователем текст.

Часть I

Введение

В этом разделе представлена концептуальная информация о создании сайтов с помощью Oracle Content Management, а также простое в использовании руководство по созданию первого сайта. Здесь содержатся указанные ниже главы.

- [Начало работы с построением сайтов](#)
- [Создание своего первого веб-сайта](#)

1

Обзор Oracle Content Management

Oracle Content Management — централизованная платформа контента в облаке для управления многоканальным контентом и ускорения доставки компонентов взаимодействия. Она предоставляет мощные возможности управления потоками операций и совместной работой, позволяющие упростить создание и доставку контента и повысить вовлеченность сотрудников и клиентов.

Oracle Content Management позволяет быстро совместно работать внутри и вне компании на любом устройстве: утверждать контент и создавать контекстуализированные компоненты взаимодействия. Встроенные инструменты, дружественные для бизнеса, существенно облегчают создание новых веб-компонетов взаимодействия с потрясающим контентом. Вы можете управлять цифровым взаимодействием со всеми заинтересованными сторонами, используя одну и ту же платформу контента и одни и те же процессы. Технические и организационные узкие места исчезли, поэтому у вас больше нет препятствий для создания привлекательных возможностей.

Примечание.

Версия Oracle Content Management Starter имеет ограниченный набор функций. Чтобы воспользоваться преимуществами полного набора функций, выполните обновление до версии Premium.

Доступ к Oracle Content Management

После предоставления вам доступа к Oracle Content Management вы получите приветственное письмо с подробными сведениями об URL-адресе экземпляра и имени пользователя. Эта информация потребуется для входа в сервис, поэтому рекомендуется сохранить ее для дальнейшего использования.

Существуют различные способы взаимодействия с Oracle Content Management:

- Веб-интерфейс предоставляет удобный доступ из любимого веб-браузера. Вы можете управлять своим контентом в облаке, обмениваться файлами и папками с другими пользователями, начинать диалоги и участвовать в них, создавать веб-сайты (если разрешено), а также делать многое другое.
- Настольное приложение позволяет синхронизировать файлы и папки между облаком и компьютером. Вы можете синхронизировать собственные файлы и файлы, к которым вам предоставлен общий доступ, чтобы всегда иметь доступ к последним версиям.
- Дополнительный модуль Microsoft Office предоставляет доступ к функциям Oracle Content Management непосредственно из приложений Microsoft Word, Excel, PowerPoint и Outlook.
- Мобильные приложения для Android и iOS обеспечивают удобный доступ к данным на вашем телефоне или других мобильных устройствах. Мобильные приложения

мгновенно становятся знакомыми, так как они выглядят и работают так же, как служба в веб-браузере. Вы можете получать доступ к своему контенту в облаке, искать и сортировать свои файлы и папки, предоставлять общий доступ к контенту и работать с диалогами.

- REST API и SDK предоставляют разработчикам мощные инструменты для программного внедрения функциональных возможностей Oracle Content Management в веб-приложения и мобильные приложения.

Общие сведения о ролях

Доступные функции Oracle Content Management зависят от назначенной роли. В зависимости от роли пользователя видны различные варианты. Стандартные пользователи могут работать с документами, диалогами и сайтами. Корпоративные пользователи также могут получить доступ к активам. Разработчикам доступны возможности создания и настройки элементов веб-сайтов, таких как шаблоны, темы, компоненты и макеты. Администраторам доступны возможности настройки данной службы, интеграции службы с другими бизнес-приложениями и настройки репозитория активов.

В Oracle Content Management существуют роли различных типов:

- **Роли в организации** — ваша роль в организации определяет задачи, которые необходимо выполнять, и способ использования функций.
- **Роли пользователей** — роли пользователей позволяют управлять отображаемыми для пользователя функциями в Oracle Content Management.
- **Роли ресурсов (разрешения)** — возможности, доступные вам при работе с ресурсами, например документами, элементами контента, сайтами или шаблонами, зависят от ролей, назначаемых вам при предоставлении доступа к ресурсу.

Подробнее...

Управление активами

Oracle Content Management предоставляет мощные возможности управления всеми вашими активами для использования в различных каналах, включая веб-сайты, маркетинговые материалы, кампании по электронной почте, интернет-магазины и блоги. Эта платформа предоставляет центр контента для всех ваших активов, где их можно организовать в репозитории и коллекции и создать правила для определения способа и места их использования.

Кроме того, существуют расширенные функции управления и рабочего процесса, которые позволяют управлять активами в процессе их создания и утверждения, а также обеспечить доступность для использования только авторизованных версий.

Можно без труда тегировать и фильтровать активы, чтобы быстро находить необходимые активы и использовать их там, где это разрешено. А функции смарт-контента будут автоматически тегировать и предлагать активы по мере их использования!

Создайте типы активов, чтобы определить, какую информацию необходимо собирать при создании пользователями активов. *Типы цифровых активов* определяют пользовательские атрибуты, необходимые для ваших цифровых

активов (файлов, изображений и видео). *Типы контента* группируют различные части контента в многократно используемые единицы. После этого пользователи могут создавать цифровые активы и элементы контента на основе этих типов активов для согласованного использования и представления между каналами.

[Подробнее...](#)

Управление содержимым

Oracle Content Management позволяет управлять контентом в облаке — все ресурсы находятся в одном месте и доступны из любой точки.

Можно группировать файлы в папках и выполнять общие операции по управлению файлами (копирование, перемещение, удаление и т. д.) во многом таким же образом, как и на локальном компьютере. А поскольку все ваши файлы находятся в облаке, у вас есть доступ к ним, где бы вы ни находились, а также на мобильных устройствах. При установке приложения для ПК весь контент может автоматически синхронизироваться с локальным компьютером, чтобы под рукой всегда были самые последние версии.

[Подробнее...](#)

Совместная работа над контентом

Так как весь контент находится в облаке, легко предоставить общий доступ к вашим файлам и папкам для совместной работы с другими пользователями внутри организации или за ее пределами. Все те, кому вы предоставили общий доступ к своему контенту, имеют доступ к последней информации — где бы они ни находились, когда бы она им не понадобилась. Вы можете предоставить доступ к целым папкам или ссылки на определенные элементы. Все операции доступа к общим элементам регистрируются, чтобы можно было отслеживать способ и время доступа к каждому общему элементу.

Диалоги в Oracle Content Management позволяют сотрудничать с другими людьми, обсуждая темы и публикуя комментарии в реальном времени. Вы можете начать автономный диалог на любую тему, добавляя файлы по мере необходимости. Также можно начать диалог о конкретном файле, папке, активе или сайте для быстрого и простого получения отзывов.

Сохраняются все сообщения, файлы и аннотации, связанные с диалогом, что позволяет легко отслеживать и просматривать обсуждение. Ваши диалоги проводятся в облачной среде, чтобы их можно было просматривать и принимать в них участие в пути с мобильных устройств.

[Подробнее...](#)

Создание сайтов

Oracle Content Management позволяет быстро создавать и публиковать веб-сайты для маркетинга и сообществ — от концепции до запуска — для обеспечения привлекательных возможностей в Интернете. Данный процесс полностью интегрирован: содержание, совместная работа и творческие проекты объединяются в единую среду создания и публикации.

Чтобы быстро начать работу, используйте готовый шаблон, перетаскиваемые компоненты, примеры макетов страниц и темы сайта, чтобы собрать сайт из

предварительно заданных стандартных блоков. Разработчики также могут создавать пользовательские шаблоны, пользовательские темы или пользовательские компоненты, чтобы обеспечить уникальные возможности для работы в Интернете.

Добавляйте видео с YouTube, потоковое видео, изображения, заголовки, абзацы, ссылки на социальные сети и другие объекты сайтов, просто перетаскивая компоненты в соответствующие ячейки на странице. Переключайте темы и делайте ребрендинг сайта одним нажатием кнопки, чтобы обеспечить оптимизированный, единообразный внешний вид и удобство работы в вашей организации.

Вы можете работать над одним или несколькими обновлениями, просматривать обновление на сайте, а затем, когда будете готовы, публиковать обновление одним нажатием кнопки.

Помимо создания и публикации сайтов в Site Builder, платформа Oracle Content Management поддерживает также "автономную" разработку сайтов с использованием REST API, React JS, Node JS и других веб-технологий.

Подробнее...

Интеграция и расширение Oracle Content Management

Поскольку сервис Oracle Content Management реализован как модель Oracle "платформа как услуга" (PaaS), он легко взаимодействует с другими сервисами Oracle Cloud.

Вы можете встроить веб-интерфейс в свои веб-приложения, чтобы пользователи могли взаимодействовать с контентом напрямую. Используйте Application Integration Framework (AIF) для интеграции сторонних сервисов и приложений в интерфейс Oracle Content Management посредством пользовательских действий. Или разработайте коннекторы контента, чтобы добавлять в Oracle Content Management контент, уже созданный в другом месте, централизованно управлять им и использовать его в новых возможностях по нескольким каналам.

Широкий набор REST API и SDK для управления контентом и сайтами, доставки и совместной работы позволяет встроить функциональные возможности Oracle Content Management в веб-приложения.

Создавайте клиентские приложения, которые взаимодействуют с вашими контентом SDK и активами в облаке. Разрабатывайте пользовательские интеграции с объектами совместной работы или извлекайте активы для использования там, где это необходимо. Вы можете использовать и доставлять все материалы и активы, оптимизированные для каждого канала, будь то веб-сайт, сеть доставки контента (CDN) или мобильные приложения.

Подробнее...

Начало работы

Посетите [Справочный центр Oracle](#) с большим количеством ресурсов, включая [документацию](#), [видеоматериалы](#), [пошаговые обзоры](#) и [информацию для разработчиков](#). Это поможет начать работу с Oracle Content Management.

И если потребуется, могут помочь [служба поддержки](#) и [сообщество](#).

Миграция в Oracle Cloud Infrastructure

Если ваша подписка Oracle Content Management еще не запущена в Oracle Cloud Infrastructure (OCI) с помощью консоли инфраструктуры, Oracle рекомендует выполнить миграцию в эту собственную среду OCI. Это позволит вам в будущем воспользоваться преимуществами облачной платформы Oracle.

Миграция не выполняется автоматически. Для запуска процесса необходимо отправить запрос на обслуживание.

[Дополнительные сведения...](#)

2

Начало работы с построением сайтов

Oracle Content Management — централизованная платформа контента в облаке для управления многоканальным контентом и ускорения доставки компонентов взаимодействия. Oracle Content Management позволяет быстро и совместно работать внутри и вне компании на любом устройстве: утверждать контент, управлять цифровыми активами и создавать контекстуализированные компоненты взаимодействия с помощью встроенных инструментов, дружественных для бизнеса.

- [Обзор Oracle Content Management](#)
- [Начало работы с сайтами](#)
- [Общие сведения о процессе создания сайта](#)
- [Подробнее о разработке сайта](#)
- [Общие сведения об управлении сайтами](#)
- [Создание своего первого веб-сайта](#)



Видео

Начало работы с сайтами

Любой пользователь, имеющий соответствующие разрешения, может создать веб-сайт с помощью Oracle Content Management. Для этого не требуются проприетарные инструменты, код и программное обеспечение. Графический пользовательский интерфейс интуитивно понятен и удобен в использовании.

Примечание.

В версии Oracle Content Management Starter можно использовать только один сайт и нельзя выполнять управление сайтом. Для получения полного набора функций выполните обновление до версии Oracle Content Management Premium.

Возможность создания сайта зависит от нескольких факторов:

- Включил ли администратор сервиса функцию создания сайтов.
- Включил ли администратор сервиса управление сайтами. См. [Общие сведения об управлении сайтами](#)
- Разрешено ли создание сайта только администраторам сайта (доступно, когда управление сайтами отключено).
- Предоставил ли администратор вашего сайта шаблоны.

Чтобы создать сайт, необходимо выбрать шаблон. Шаблон содержит все необходимые элементы для создания сайта, в том числе кодовую платформу сайта, стандартный

сайт с примерами страниц и содержимого, тему со стилями, ресурсы (например, изображения), а также пользовательские компоненты. См. [Общие сведения о процессе создания сайта](#).

Если вы создаете корпоративный сайт, он будет связан с репозиторием и должен иметь политику локализации, определяемую языком по умолчанию. Перед созданием сайта необходимо создать репозиторий и политику локализации. Все необходимые для сайта активы и документы хранятся в репозитории, а политики репозитория определяют использование этих активов. Для получения подробных сведений об использовании репозитория см. раздел Общие сведения о репозиториях активов.

При редактировании сайта создается новое обновление или используется существующее. В рамках обновления можно редактировать и добавлять содержимое, изменять настройки стиля, добавлять и удалять страницы, изменять макеты страниц и упорядочивать страницы. Для получения сведений о возможностях работы с редактором см. раздел [Вводные сведения о странице "Site Builder"](#).

Обновления не должны быть готовы все одновременно. Вы и члены рабочей группы можете работать над несколькими обновлениями параллельно и независимо. Например, вы можете работать над обновлением с еженедельными новостями, а другой член рабочей группы добавляет страницы для предстоящей конференции по продажам. Обновления можно редактировать, просматривать и сохранять в любое время. Также можно объединять обновления с базовым сайтом.

По готовности запустите свой сайт Сайт можно опубликовать в сети простым щелчком мыши. Таким образом, можно с легкостью создавать сайты с нуля.

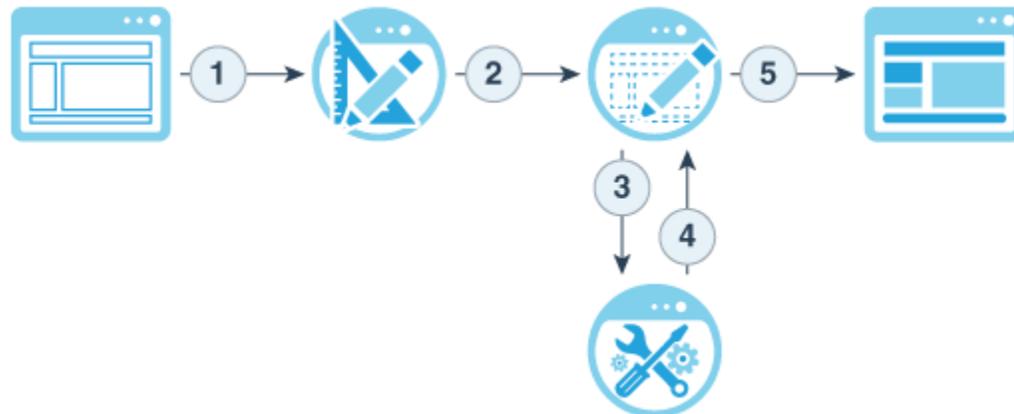
Общие сведения о процессе создания сайта

Ниже кратко описан процесс создания и публикации веб-сайта.

Прежде чем вы начнете работу, ваш администратор сервиса должен включить функцию создания сайтов, указать, кто может создавать сайты, а также установить и предоставить шаблоны, которые вы будете использовать для создания сайтов. Администратор сервиса должен назначить вам как минимум разрешения на выгрузку шаблонов. См. разделы ["Подробнее о ролях пользователей"](#), ["Начало работы с сайтами"](#) и ["Общие сведения об управлении сайтами"](#).

Если вы будете создавать корпоративный сайт, который позволяет использовать активы и многоязычные сайты, он будет связан с репозиторием. Все необходимые для сайта активы и документы хранятся в репозитории, а политики репозитория определяют использование этих активов. Репозиторий должен быть создан до создания корпоративного сайта. См. Общие сведения о репозиториях активов.

Ниже представлен обзор процесса создания сайта со ссылками на подробные сведения.



- 1. Выбор шаблона и имени для сайта:** если управление сайтом включено и ваш сайт требует утверждения, придется подождать, пока он будет утвержден, прежде чем вы сможете выполнить следующий шаг. Сайт остается в автономном режиме до момента публикации.
- 2. Создать сайт.** Создание пустой платформы, которую можно настроить.
- 3. Создать обновление или использовать существующее обновление.** Обновление представляет собой именованную коллекцию изменений для текущего базового сайта. Всякий раз при просмотре или редактировании сайта в редакторе используется обновление. См. [Общие сведения об обновлениях сайтов](#).
- 4. Открыть сайт в редакторе:** Используйте редактор с поддержкой перетаскивания для добавления на сайт страниц и содержимого. См. [Общие сведения об обновлениях сайтов](#).
 - **Добавить страницы на сайт.** Выберите предварительно определенный макет, чтобы быстро определить тип страницы. См. [Добавление страниц](#).
 - **Добавить компоненты на страницу:** Перетащите на страницу текст, изображения, документы и другие элементы. См. [Использование встроенных компонентов](#).
 - **Изменить содержимое или свойства компонента:** Точная настройка интервалов, выравнивания и других свойств компонентов. См. [Использование стилей и форматирование](#).
 - **Переключить макет, используемый для страницы:** Легко и просто измените оформление страницы. См. [Изменение макета страницы](#).
 - **Переключить тему, используемую для страницы:** Легко и просто измените внешний вид и оформление всего сайта. См. [Использование шаблонов и тем для сайтов](#).
 - **Настроить свойства поисковой системы:** Помогите поисковым системам найти свой сайт, чтобы увеличить трафик. См. [Настройка свойств поисковых систем](#).
- 5. Применить обновление сайта:** Обновите сайт с помощью изменений в обновлении. См. [Публикация изменений сайта](#).
 - **Предоставить доступ к сайту другим участникам:** можно предоставить общий доступ к сайту определенным пользователям и назначить им роли, определяющие их полномочия при работе с сайтом. Дополнительные сведения

о предоставлении общего доступа см. в разделе [Общие сведения о безопасности сайта](#).

- **Настроить уровень безопасности сайта.** Выберите, какие пользователи могут получить доступ к вашему опубликованному сайту. См. [Изменение настроек безопасности сайта](#).
- **Опубликовать сайт.** Публикация изменений на сайте. См. [Публикация изменений сайта](#).
- **Перевести сайт в онлайн-режим.** Предоставление доступа к сайту пользователям в соответствии с настройками безопасности сайта. См. [Перевод сайта в онлайн-режим или автономный режим](#).

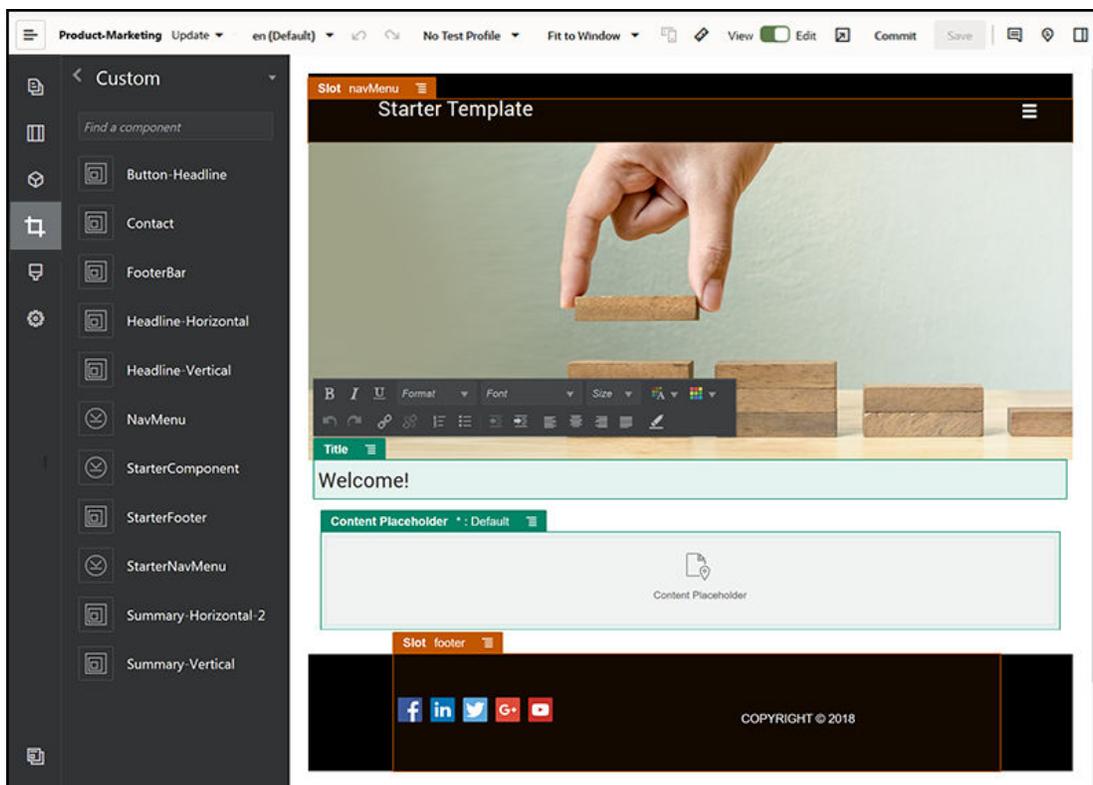
Подробнее о разработке сайта

Как разработчик сайта, вы определяете базовую структуру, которую создатели сайта используют для создания сайтов, например:

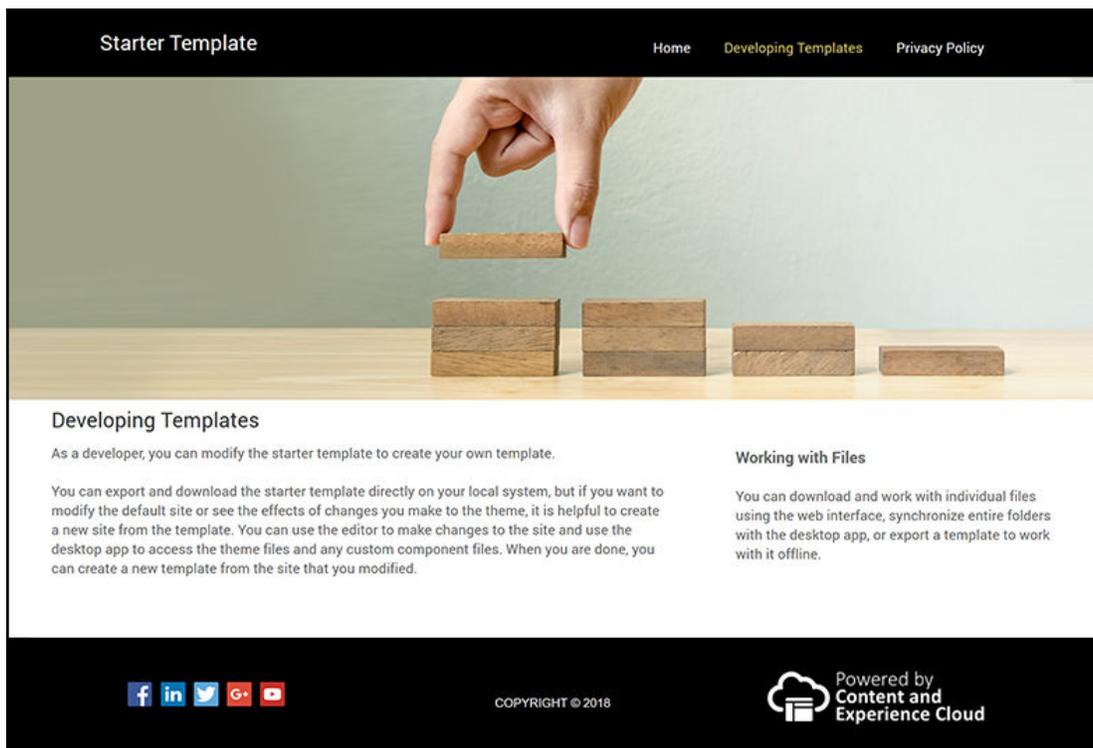
- **стандартные шаблоны**, используются в качестве базовой структуры для сайта, на основе темы, с образцами страниц и контента, пользовательскими компонентами и другими ресурсами, необходимыми для создания сайта
- **темы**, используются для определения общего вида сайта, состоящие из логотипов, таблиц стилей, файлов конфигурации и кода фона, определяющего навигацию по сайту
- **пользовательские компоненты**, используются для добавления определенных типов контента на страницы сайта, что позволяет разрабатывать составные элементы, которые могут быть встроены в страницу сайта с использованием любой технологии страниц по вашему выбору

Разделение представления сайта и контента сайта позволяет создавать сайты в соответствии со стандартами и фирменным стилем вашей организации и дает создателям контента возможность сосредоточиться непосредственно на контенте, что делает разработку и создание сайта более рациональными и эффективными.

Ниже приведен простого шаблона, который может использовать создатель контента при добавлении контента на сайт. Обратите внимание на общие заполнители, которые отображаются на странице:



Ниже приведен пример того, как может выглядеть сайт, созданный с помощью простого шаблона, при предварительном просмотре после добавления контента. Обратите внимание на то, как настроены заполнители для определенных элементов контента и навигации:



Кроме того, разработчики сайтов создают и поддерживают [таблицы стилей](#), [создают сайты](#), используя веб-интерфейс или Инструментарий ОСЕ, и настраивают интеграции между Oracle Content Management и другими сервисами. И, как и любой другой сотрудник, они также совместно работают с другими сотрудниками, открывая общий доступ к контенту, начиная диалоги или участвуя в них, или используя приложения для настольных компьютеров или мобильных устройств.

Для использования Oracle Content Management разработчикам должна быть назначена роль "Стандартный пользователь" или "Корпоративный пользователь". Разработчики с ролью "Стандартный пользователь" могут создавать компоненты, темы и стандартные шаблоны. Разработчики с ролью "Корпоративный пользователь" также могут выполнять создание [макетов](#) и сохранять сайт в качестве стандартного или корпоративного шаблона.

Помимо разработки сайта, Oracle Content Management также можно использовать в автономной среде в качестве мощной и гибкой системы управления контентом (CMS) в облаке.

Общие сведения об управлении сайтами

Управление сайтом позволяет бизнес-пользователям создавать сайты, соответствующие политике компании, и дает администраторам сайтов простой способ отслеживать сайты и управлять ими из централизованного расположения.



Примечание.

В версии Oracle Content Management Starter можно использовать только один сайт и нельзя выполнять управление сайтом. Для получения полного набора функций выполните обновление до версии Oracle Content Management Premium.

Проблема

Сегодня в данной отрасли существует проблема. Компании сталкиваются с бурным ростом интенсивности взаимодействия. Необходимо быстро и централизованно создавать и развертывать компоненты взаимодействия, чтобы соответствовать фирменному стилю и миссии компании. Предприятия часто обнаруживают, что на разработку и доставку компонентов взаимодействия уходит "вечность". Для успешной деятельности вашей компании необходима мощная и адаптируемая технология с поддержкой быстрого масштабирования, позволяющая согласованно создавать множество компонентов взаимодействия и управлять ими — и продвигать инициативы, связанные с товарами или брендами. Вам необходимо быстро и качественно предоставлять доступ к новым компонентам взаимодействия, или вы упустите рыночные возможности и не сможете эффективно привлекать заказчиков и потенциальных клиентов.

Бурный рост интенсивности взаимодействия часто становится результатом многолетней реализации отдельных, узконаправленных инициатив. За это время все подразделения, бренды и команды приобрели системы управления веб-содержимым (WCM) без централизованного надзора. В результате создается сложная среда, которая медленно выходит на рынок, подвержена уязвимости для угроз безопасности и вызывает растущие и непредсказуемые затраты.

Нехватка согласованных процессов создания компонентов взаимодействия наряду с использованием разрозненного сочетания технологий WCM подвергает организацию огромным рискам безопасности. Когда несколько заинтересованных сторон заняты решением лишь небольшой части проблем веб-сайтов, основные обязанности, такие как обеспечение безопасности, становятся фрагментарными, а предприятие остается уязвимым. Все эти компоненты взаимодействия предоставляются различными системами WCM с разными веб-приложениями, и системе надзора за ИТ-инфраструктурой становится сложно, если не невозможно, обеспечить должную защиту всех ресурсов.

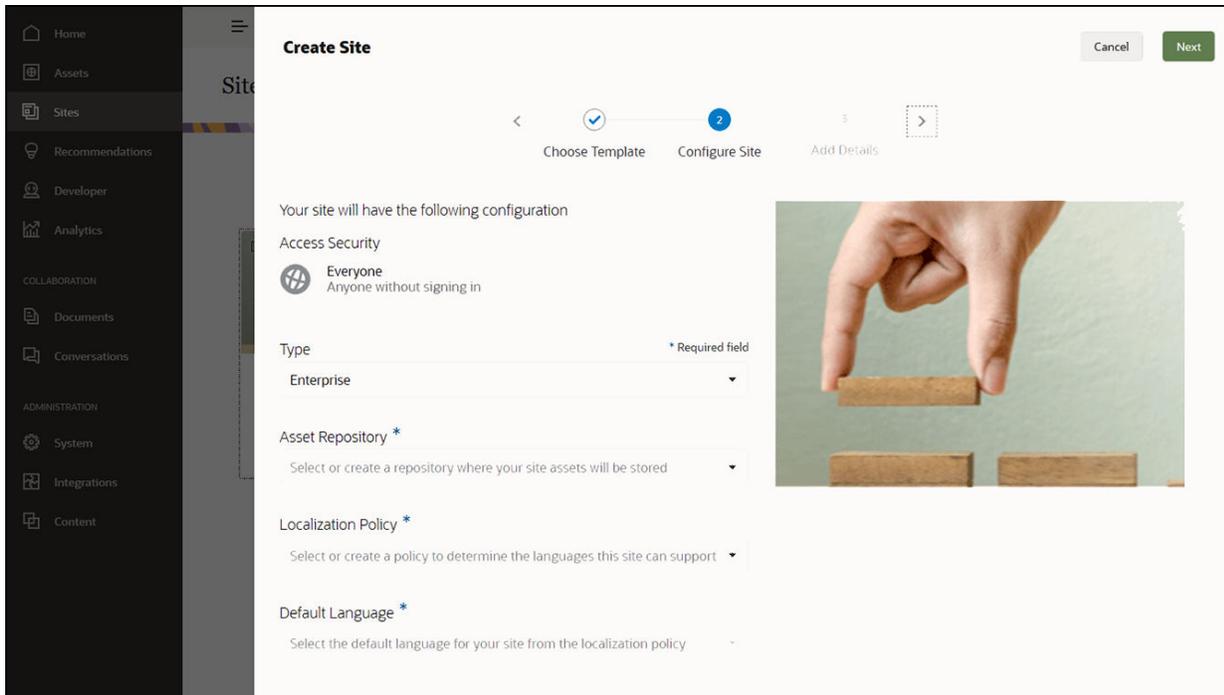
Кроме того, поддержка нескольких дорогостоящих коммерческих систем WCM приводит к дублирующимся затратам. Эта ситуация возникает каждый год с оплатой продления лицензии и постоянными расходами на поддержку внутренних или внешних партнеров.

Это накладывает ограничения на всех.

- Ограничения на бизнес:
 - Нет самообслуживания; зависимость от ИТ-инфраструктуры или дорогих внештатных услуг
 - Невозможность выполнения обновлений без технической помощи
 - Нет дружественных для бизнеса инструментов для управления усилиями
 - Нет централизованного представления всех компонентов взаимодействия и операций
- Ограничения на ИТ-инфраструктуру:
 - К каждому взаимодействию необходимо привлекать ИТ-персонал
 - Недостаточное управление компонентами взаимодействия, созданными на предприятии, включая неконтролируемые внештатные услуги
 - Использование фрагментированных стеков для построения компонентов взаимодействия
 - Необходимость управлять и развертывать изменения содержимого и макетов в сотнях компонентов взаимодействия
- Ограничения на пользователей:
 - Несогласованный обмен сообщениями по различным каналам
 - Устаревшая информация
 - Низкая производительность, ведущая к отказу от канала

Решение

Управление Oracle Content Management упрощает и ускоряет предоставление компонентов взаимодействия бизнес-пользователям, обеспечивая для ИТ-отделов простой способ управления и отслеживания компонентов взаимодействия из централизованного местоположения с возможностью полного управления всем жизненным циклом таких компонентов, снижая затраты на создание и поддержание каждого нового компонента, необходимого компании. Управление встроено в ядро Oracle Content Management; ваш администратор просто должен включить его. См. раздел Конфигурирование настроек сайтов и активов в документе *Administering Oracle Content Management*.



Компоненты

- **Каталог утвержденных шаблонов:** ИТ-разработчики могут заполнить каталог шаблонов набором шаблонов сайтов для различных направлений бизнеса. Они могут ограничить список доступных шаблонов, а также определить список пользователей, для которых эти шаблоны доступны. Они могут применять политики шаблонов, чтобы указать тип безопасности, которому должны соответствовать новые сайты, а также указать, должны ли новые сайты требовать утверждения. Это позволяет быстро и просто запрашивать новые сайты, обеспечивая соблюдение бизнес-пользователями рекомендаций по фирменному стилю и безопасности. См. разделы [Изменение аудитории или статуса шаблона](#) и [Изменение политик шаблона](#).
- **Упрощенные запросы, утверждения и подготовка:** позволяют бизнес-пользователям быстро запрашивать новые сайты с необходимыми утверждениями и автоматической подготовкой. См. разделы [Создание сайтов](#) и [Управление запросами сайта](#).
- **Управление сайтами:** ИТ-отделы могут управлять всеми сайтами из одного места независимо от того, кто создал и развернул сайт. ИТ-пользователи могут отслеживать статус любого развернутого сайта и изменять его. См. [Управление сайтами и настройки сайтов](#).

Преимущества

При использовании управления снижаются затраты на создание и поддержание каждого нового взаимодействия.

- Преимущества для бизнеса:
 - Быстрая подготовка независимо от ИТ-персонала
 - Отделение содержимого от дизайна, что позволяет осуществлять повторное использование

- Глобальное управление компонентами взаимодействия, пользователями и разрешениями из одной консоли
- Использование аналитической информации в операциях взаимодействия с помощью отчетов
- Преимущества для ИТ-инфраструктуры:
 - Нет сложной реализации (облачное решение)
 - Автоматическое масштабирование в соответствии с ростом и сезонными пиками
 - Управление компонентами взаимодействия с обеспечением их безопасности и соответствия корпоративному фирменному стилю и стандартам
- Преимущества для пользователей:
 - Согласованный обмен сообщениями по различным каналам
 - Актуальная информация
 - Оптимальная производительность
 - Безопасное взаимодействие

Процесс

Ниже перечислены шаги, выполняемые при создании сайтов и управлении ими с помощью Управления сайтами:

1. Ваш администратор включает управление. См. раздел Конфигурирование настроек сайтов и активов в документе *Administering Oracle Content Management*.
2. Администратор сайта делает утвержденные шаблоны доступными пользователям для создания сайтов. См. [Изменение аудитории или статуса шаблона](#) и [Изменение политик шаблона](#).
3. Пользователь создает запрос сайта из утвержденного шаблона. См. [Создание сайтов](#).
4. Администратор сайта утверждает запрос сайта, и сайт создается автоматически. Администратор сайта также может отклонить запрос сайта с указанием причины отклонения, а создатель сайта может устранить проблему и отправить запрос повторно. См. [Управление запросами сайта](#).
5. Создатель сайта продолжает процесс создания сайта, редактирует и публикует сайт. См. [Общие сведения о процессе создания сайта](#).
6. Текущие операции по управлению сайтами и обновлениям можно выполнять на странице Сайты. Администраторы сайта могут видеть все сайты на странице Сайты. Другие пользователи могут видеть сайты, которые они создали или участниками которых они являются. См. [Управление сайтами и настройки сайтов](#).

3

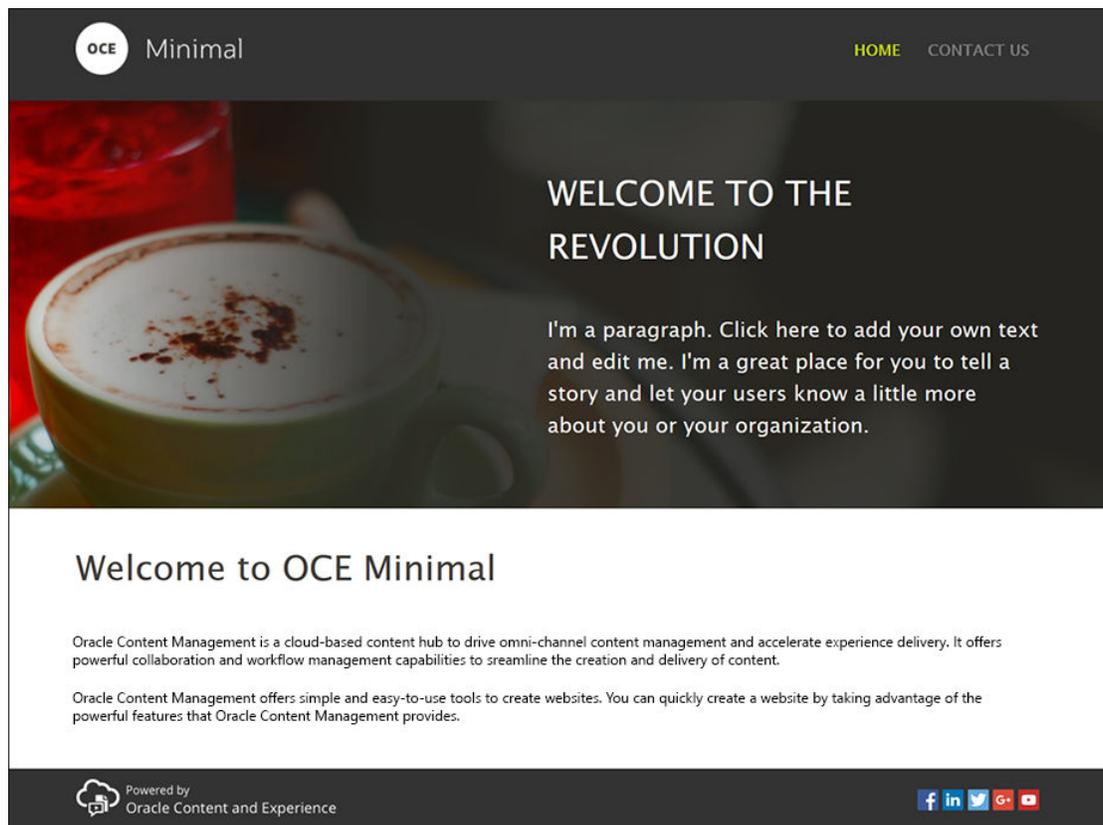
Создание своего первого веб-сайта

Oracle Content Management позволяет легко и быстро создать свой первый веб-сайт.

Из этого учебного пособия вы узнаете, как настроить и опубликовать веб-сайт с помощью готового шаблона **Blank-Template**, доступного в Oracle Content Management. Вы увидите, как легко создать свой первый веб-сайт с помощью Site Builder без необходимости в дополнительном кодировании.

Это учебное пособие состоит из трех основных этапов:

1. [Настройка среды](#)
2. [Настройка веб-сайта](#)
3. [Публикация веб-сайта](#)



Однако прежде чем начнем, давайте сначала [проверим несколько вещей](#).

Перед началом работы

Чтобы начать, пользователь должен иметь доступ к экземпляру Oracle Content Management и следующие [роли](#) приложения:

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

Если у вас нет указанных выше ролей, попросите администратора сервиса назначить их вам.

Получив доступ к экземпляру Oracle Content Management, выполните вход как администратор контента.

[Давайте начнем..](#)

Шаг 1. Настройка среды

Во-первых, необходимо настроить среду, импортировав пользовательский компонент Minimal-NavMenu и загрузив образцы изображений.

Поэтому выгрузите пакет активов **OCESamplesAssetPack.zip**, который доступен на странице <https://www.oracle.com/middleware/technologies/content-experience-downloads.html>. В выгруженном пакете активов найдите файл

OCECreateYourFirstSite_data.zip, который содержит два следующих ZIP-файла:

- **Minimal-NavMenu.zip** с пользовательским компонентом Minimal-NavMenu и
- **Minimal-Images.zip** с образцами изображений, используемыми на веб-сайте.

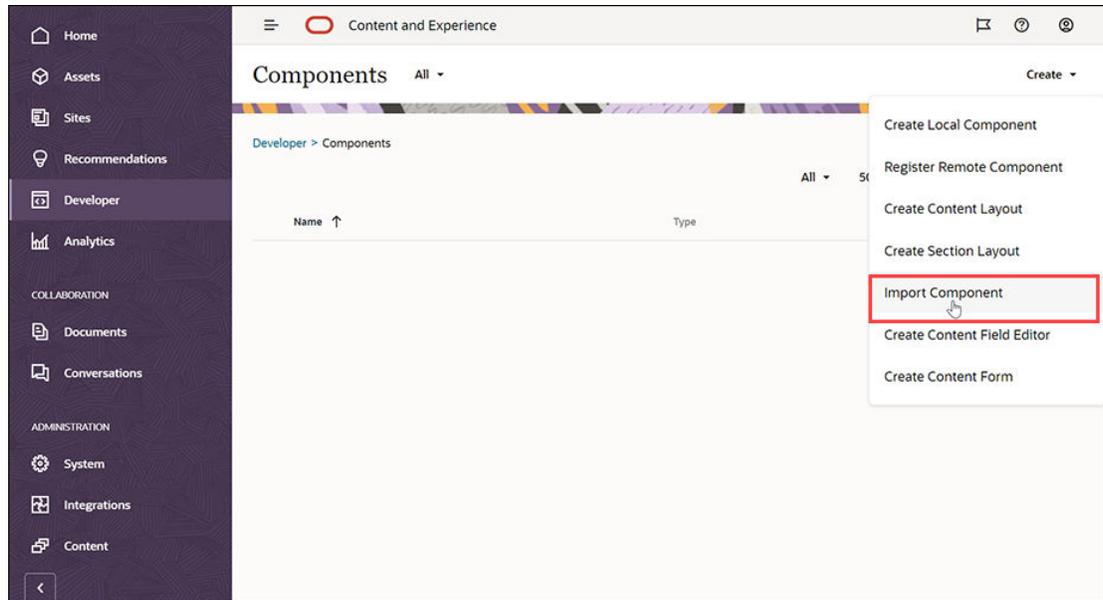
Давайте приступим к настройке среды.

1. [Импорт пользовательского компонента Minimal-NavMenu](#)
2. [Публикация пользовательского компонента Minimal-NavMenu](#)
3. [Добавление образцов изображений](#)

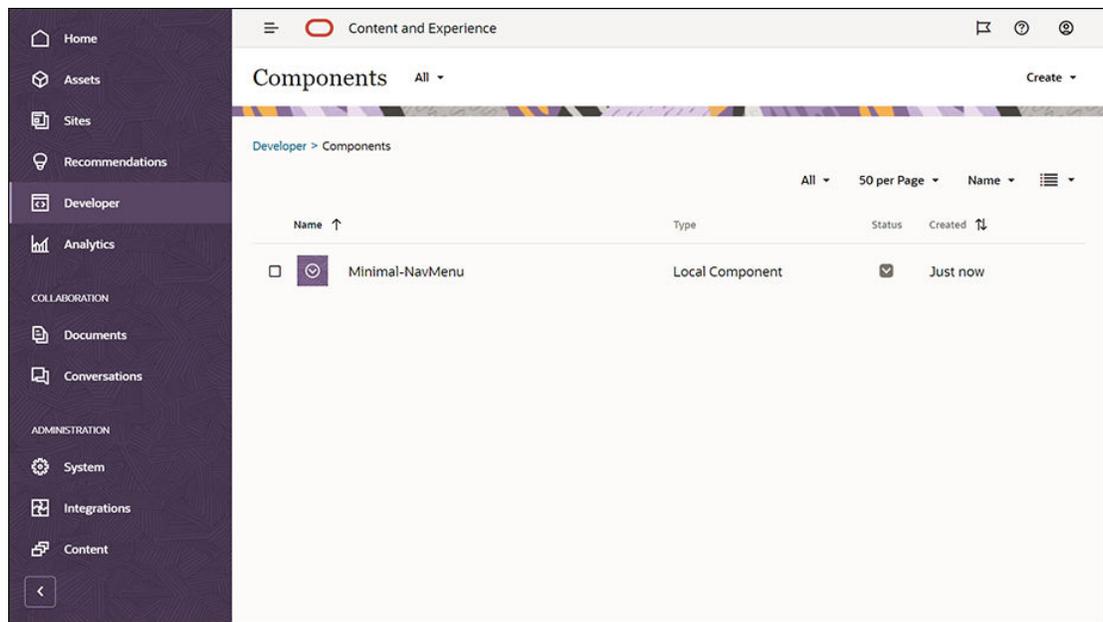
Импорт пользовательского компонента Minimal-NavMenu

После входа в веб-интерфейс Oracle Content Management нажмите **Разработчик** в левом меню навигации, а затем — **Смотреть все компоненты**. Если пункт **Разработчик** не отображается, у вас отсутствуют требуемые [роли](#) пользователя.

На странице "Компоненты" нажмите **Создать** и выберите **Импорт компонента**. Загрузите файл **Minimal-NavMenu.zip** ([доступный в файле OCECreateYourFirstSite_data.zip](#)) и выделите его.



Компонент Minimal-NavMenu сейчас отображается на странице "Компоненты".

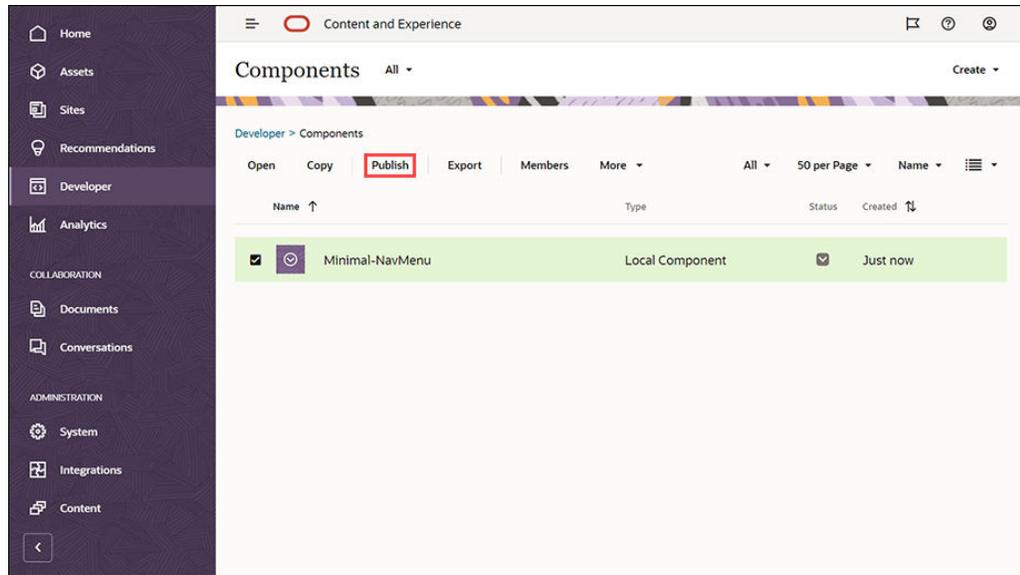


Следующий шаг: публикация пользовательского компонента Minimal-NavMenu

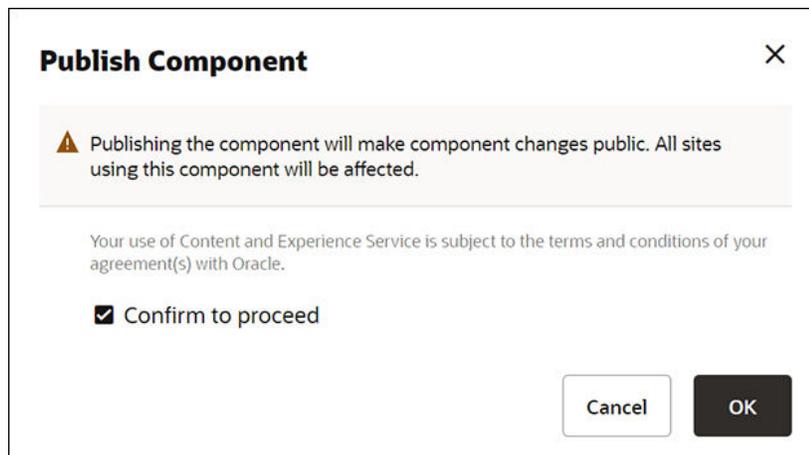
Публикация пользовательского компонента Minimal-NavMenu

Теперь необходимо опубликовать импортированный компонент Minimal-NavMenu.

На странице "Компоненты" выделите компонент Minimal-NavMenu и нажмите **Опубликовать** в строке меню или контекстном меню.



В диалоговом окне "Публикация компонента" выберите **Подтвердить, чтобы продолжить** и нажмите **ОК**.



После публикации компонента Minimal-NavMenu наверху страницы отображается уведомление об этом.

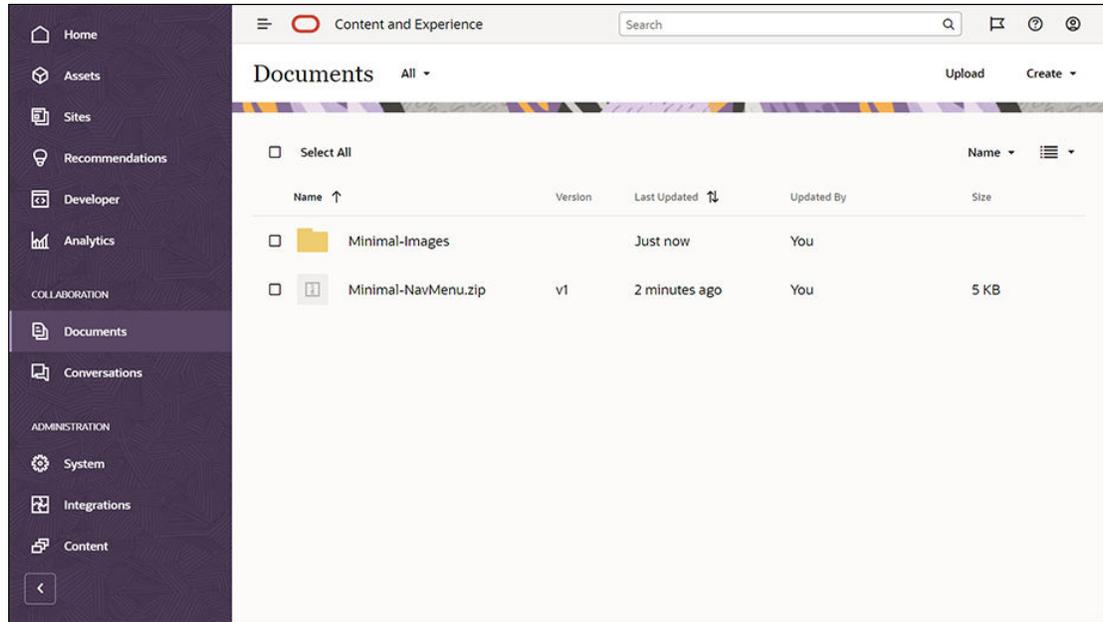
[Следующий шаг: добавление образцов изображений](#)

Добавление образцов изображений

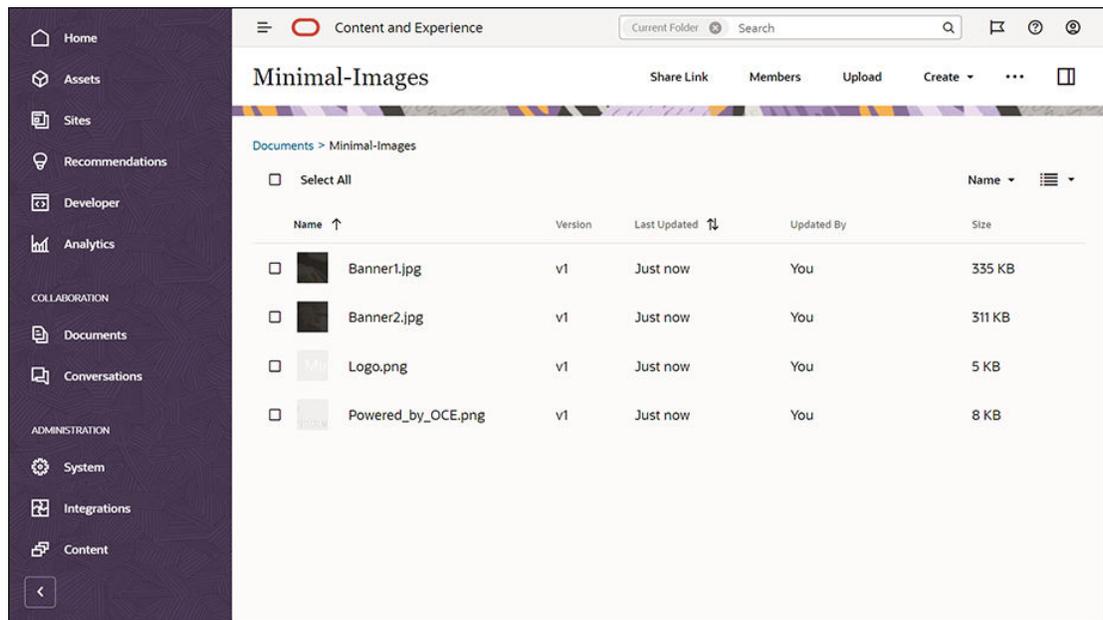
Теперь образцы изображений (которые будут использоваться на созданном веб-сайте) необходимо добавить в раздел **Документы**.

Извлеките содержимое файла **Minimal-Images.zip** ([доступный в файле OCECreateYourFirstSite_data.zip](#)) в папку **Minimal-Images** на своем локальном компьютере.

В левом меню навигации нажмите **Документы**, а затем **Создать**, чтобы создать папку **Minimal-Images**.



Загрузите все изображения из папки **Minimal-Images** локального компьютера в недавно созданную папку **Minimal-Images** в Oracle Content Management.



 **Примечание.**

В шаблоне и на сайте используются изображения Shutterstock. Для повторного использования таких этих изображений они должны быть лицензированы посредством Shutterstock. Эти изображения также можно заменить своими изображениями.

Следующий шаг: настройка веб-сайта

Шаг 2. Настройка веб-сайта

Теперь, когда у вас есть все необходимое для создания веб-сайта, можете приступить к настройке веб-сайта.

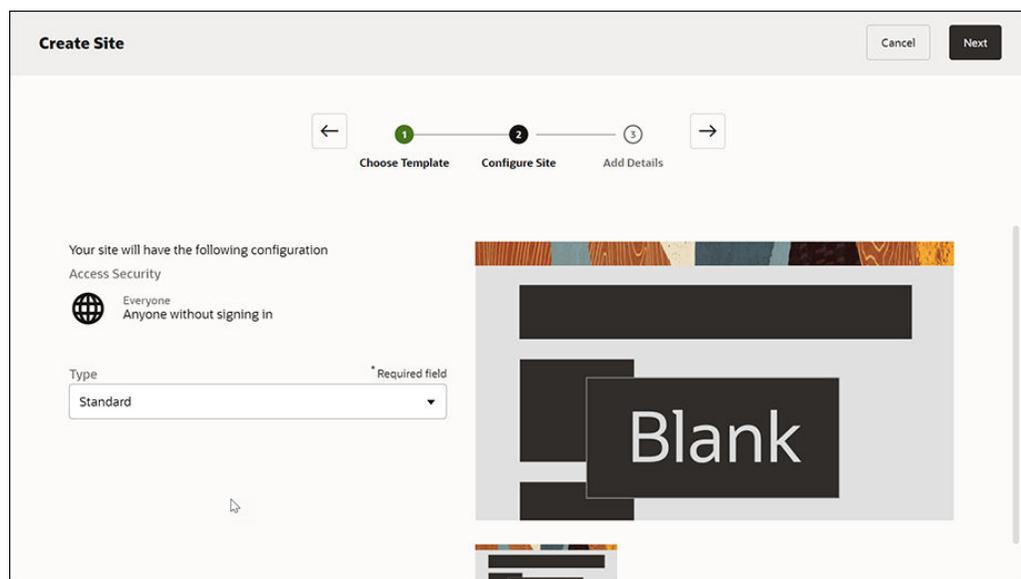
1. [Создание веб-сайта](#)
2. [Редактирование веб-сайта](#)

Создание веб-сайта

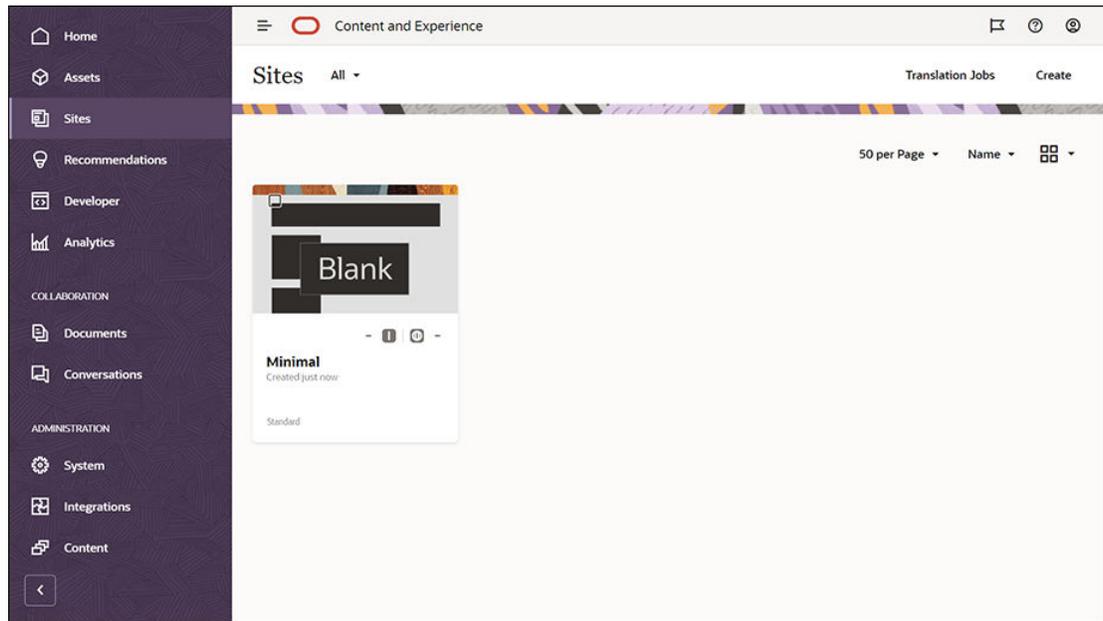
В левом меню навигации нажмите **Сайты**, а затем **Создать**. Выберите **Blank-Template**, а затем нажмите **Далее**.

Если **Blank-Template** не отображается, обратитесь к администратору сервиса. Администратору сервиса необходимо готовые шаблоны сделать доступными для использования. Администратор сервиса обычно [устанавливает готовые шаблоны](#), которые Oracle Content Management предоставляет при инициализации сервиса.

Обязательно выберите для создания стандартный веб-сайт, затем нажмите **Далее**. Введите имя (например, **Minimal**) для своего веб-сайта. Нажмите кнопку **Готово**.



Новый веб-сайт создается и его имя появляется на странице "Сайты".

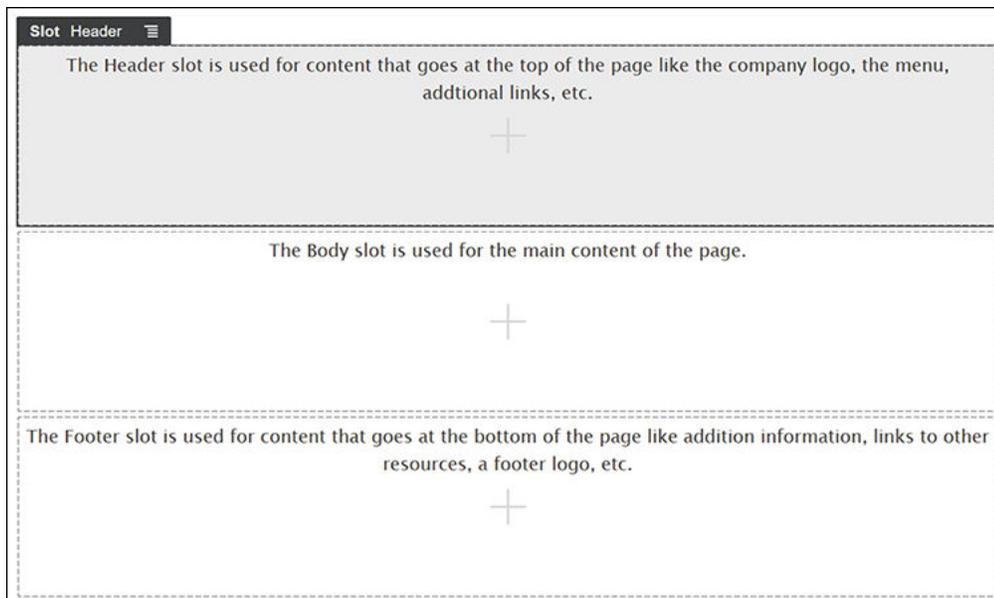


Следующий шаг: редактирование веб-сайта

Редактирование веб-сайта

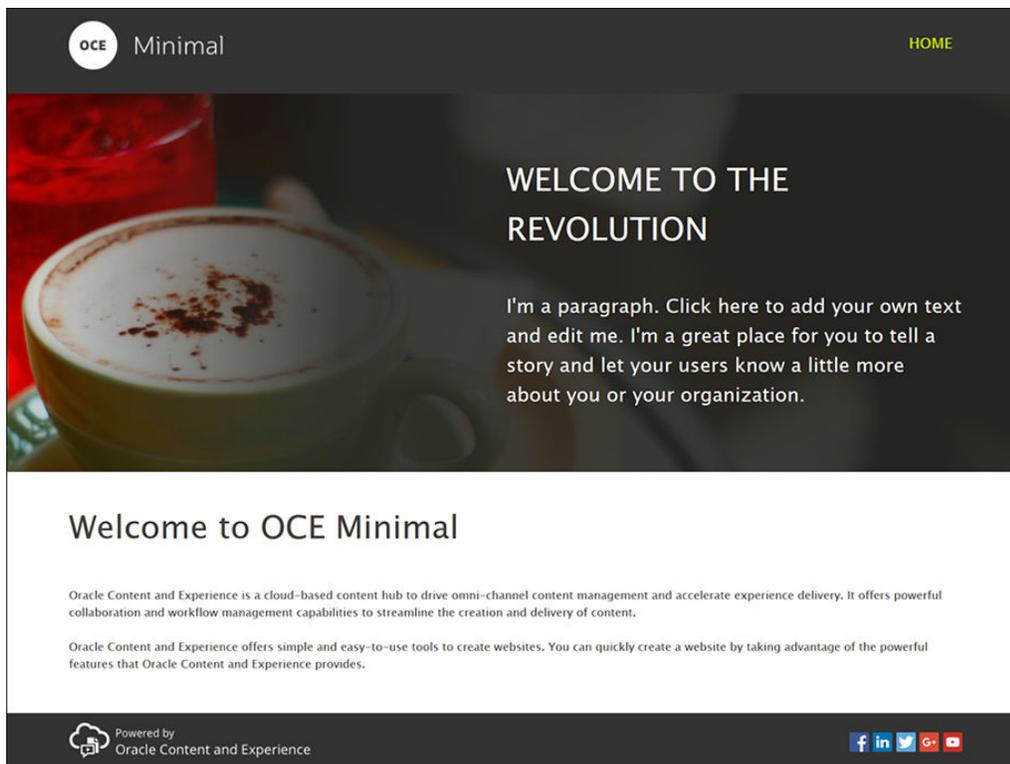
Откройте недавно созданный веб-сайт в Site Builder, выбрав его и нажав **Открыть** в строке меню или в контекстном меню. В Site Builder переведите указатель в положение **Редактировать**. Введите имя обновления и нажмите **ОК**.

В режиме редактирования вы увидите, что на сайте есть три ячейки, которые являются областями, доступными на странице (в зависимости от макета страницы). Наведите указатель на каждый знак "+" на странице, чтобы увидеть такие ячейки, как "Заголовок", "Тело" и "Нижний колонтитул".



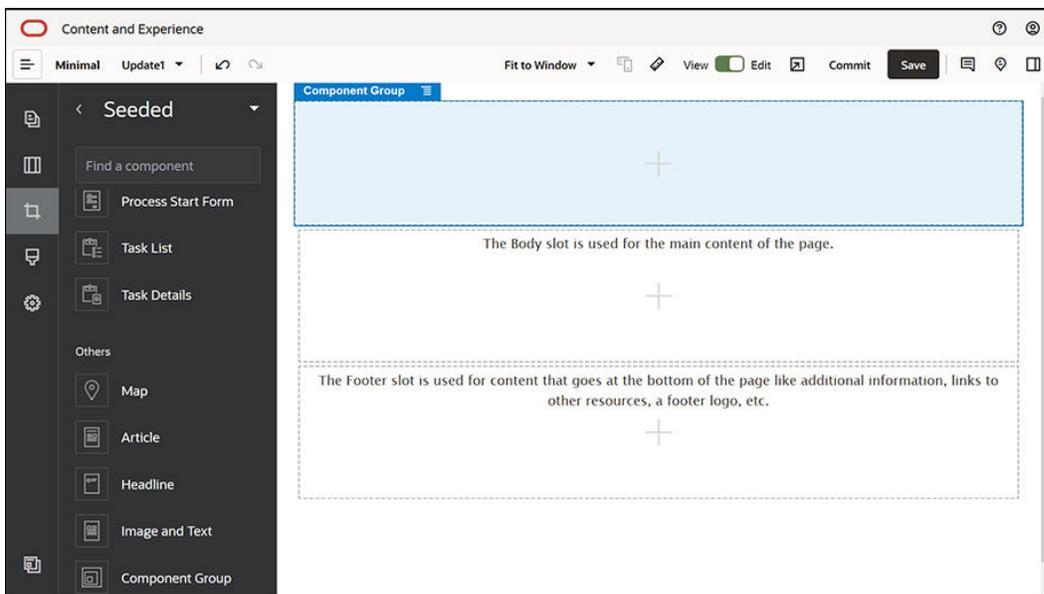
Ячейка "Заголовок" обычно используется для отображения логотипа компании, меню навигации и т. д. Ячейка "Тело" предназначена для основного контента страницы, а ячейка "Нижний колонтитул" содержит информацию об авторских правах, ссылки на социальные сети и любую дополнительную информацию.

Во-первых, давайте создадим домашнюю страницу. Вот как домашняя страница будет выглядеть после завершения:

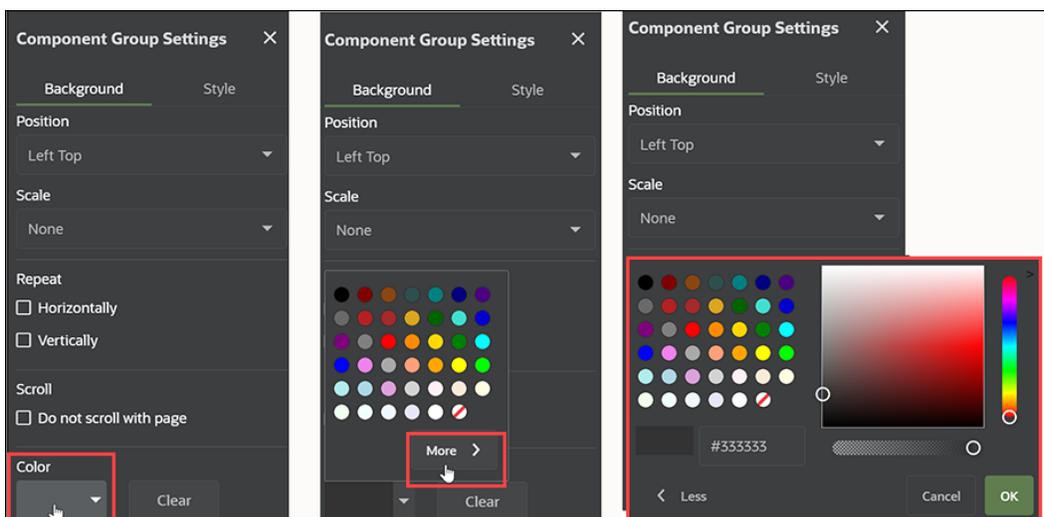


Давайте заполним ячейку "Заголовок" готовыми компонентами:

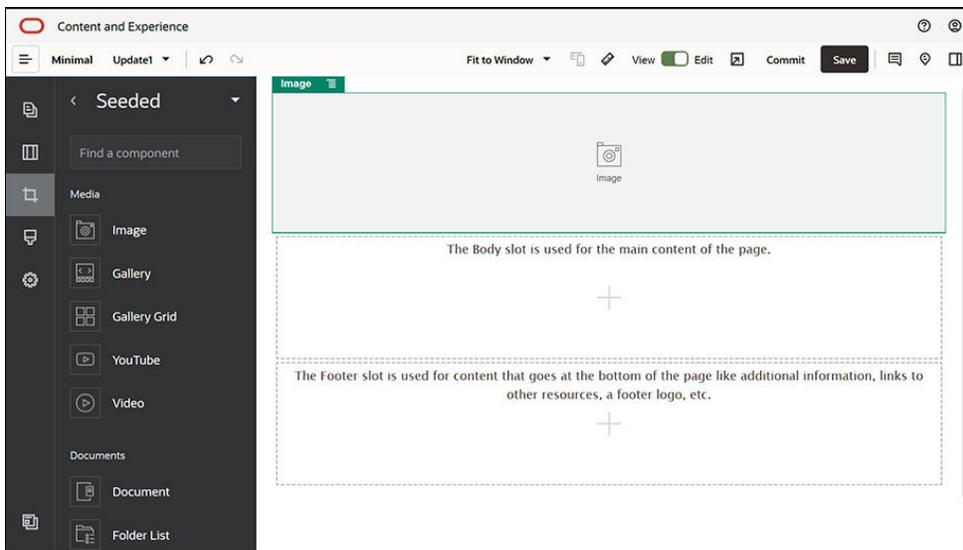
1. На левой боковой панели нажмите , а затем **Заданные предварительно**, чтобы отобразить список готовых компонентов, доступных в Oracle Content Management.
2. На левой боковой панели найдите готовый компонент "Группа компонентов". Перетащите его в ячейку "Заголовок".



3. Нажмите на значок меню группы компонентов , а затем нажмите **Настройки**. В настройках нажмите на раскрывающийся список **Цвет** (доступен в нижней части списка настроек), а затем нажмите **Дополнительно**. Введите **#333333** и нажмите **ОК**.

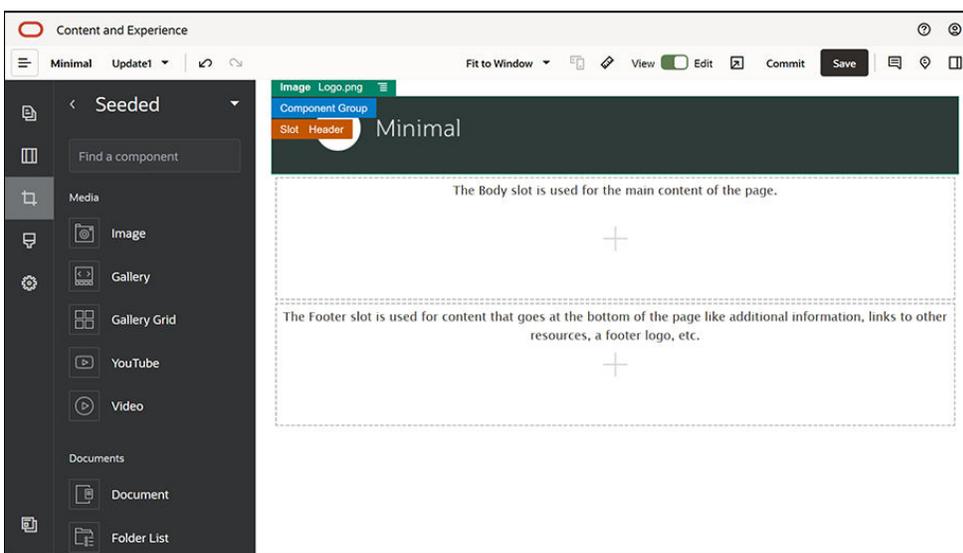


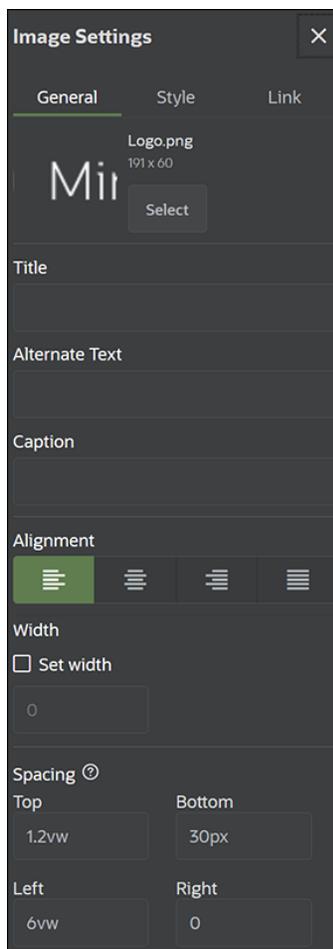
4. Перетащите компонент "Изображение" в группу компонентов.



- Нажмите на значок меню компонента "Изображение" , а затем нажмите **Настройка**. Завершите настройку на вкладке **Общие**.

Свойство	Значение
Выбрать	Logo.png из папки Minimal-Images
Выравнивание	Слева
Ширина	Отмените выбор Задать ширину
Первые	1.2vw
Внизу	30px
Слева	6vw
Справа	0

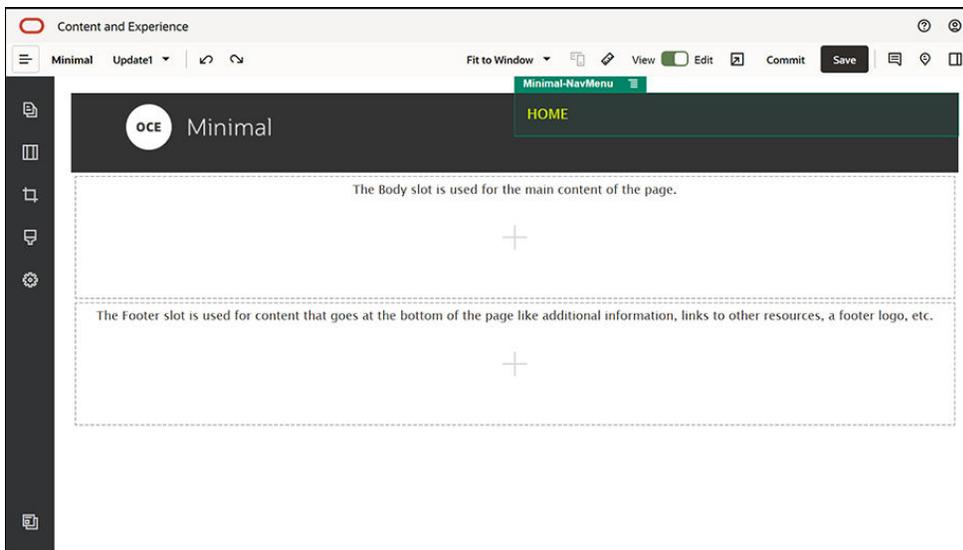




- Давайте свяжем это изображение логотипа с домашней страницей. Задайте настройки компонента "Изображение" на вкладке **Ссылка**.

Свойство	Значение
Выберите тип ссылки	Страница сайта
Страница	Домашняя страница
Назначение	Открыть в том же окне

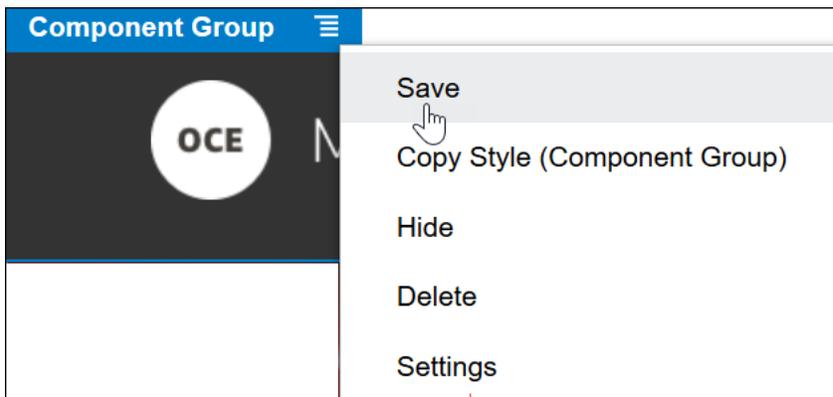
- На левой боковой панели нажмите  и **Пользовательский**, чтобы отобразить список пользовательских компонентов.
- Теперь давайте добавим меню навигации на домашнюю страницу, используя пользовательский компонент `Minimal-NavMenu`. Перетащите компонент `Minimal-NavMenu` в группу компонентов справа от компонента "Изображение". Нажмите на заголовок компонента `Minimal-NavMenu`, чтобы убедиться, что его родительский компонент — это группа компонентов, которую вы добавили ранее. Это полезный способ узнать местоположение любого компонента в структуре веб-страницы.



Завершите настройку на вкладке **Общие**.

Свойство	Значение
Выравнивание	Справа
Первые	1.2vw
Внизу	0
Слева	0
Справа	6vw

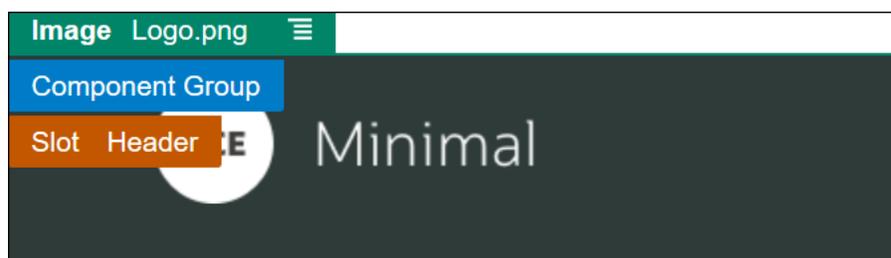
- Теперь заголовок готов. Давайте сохраним эту группу компонентов как пользовательскую группу компонентов, чтобы использовать ее позже на других страницах веб-сайта. Нажмите на заголовок группы компонентов, а затем нажмите на значок ее меню  и нажмите **Сохранить**. В диалоговом окне "Сохранение группы компонентов" в поле **Имя** введите "Minimal-Header", а затем нажмите кнопку **Сохранить**.



 **Совет.**

Если нажать имя группы компонентов, отображается заголовок компонента "Изображение" или Minimal-NavMenu, а не заголовок группы компонентов. Снова нажмите заголовок компонента "Изображение" или Minimal-NavMenu и вы увидите заголовок группы компонентов. Теперь можно нажать заголовок группы компонентов и выполнить указанный выше шаг.

На изображении ниже показана родительская структура компонента "Изображение" в ячейке "Заголовок".

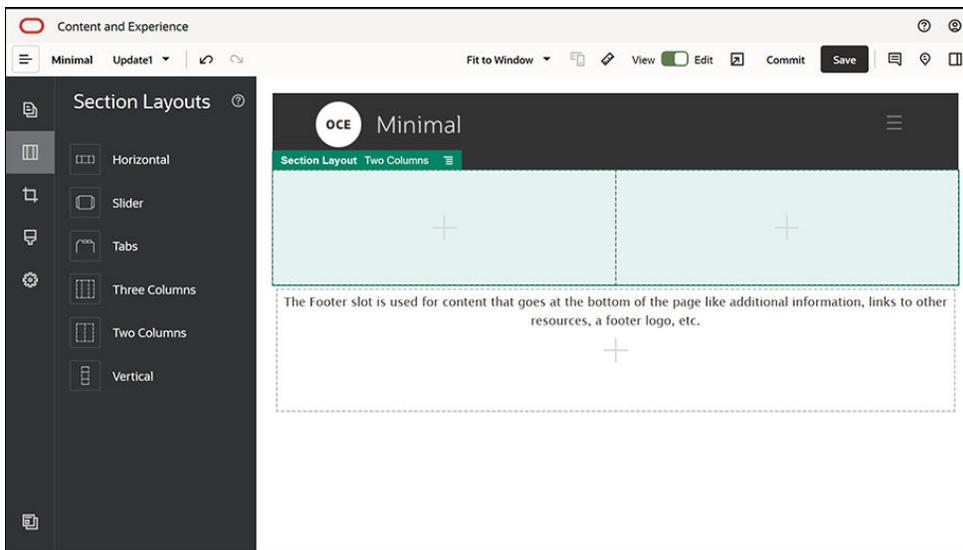


10. Нажмите **Сохранить** в правом верхнем углу окна Site Builder, чтобы сохранить свои изменения. Теперь ячейка "Заголовок" должна выглядеть следующим образом:



Давайте перейдем к ячейке "Тело":

1. На левой боковой панели нажмите , затем выберите **Заданные предварительно**.
2. Из левой боковой панели перетащите группу компонентов в ячейку "Тело". С помощью этой группы компонентов и компонентов (которые мы добавим в нее) создадим баннер.
3. На левой боковой панели нажмите .
4. Перетащите макет раздела "Два столбца" в группу компонентов.



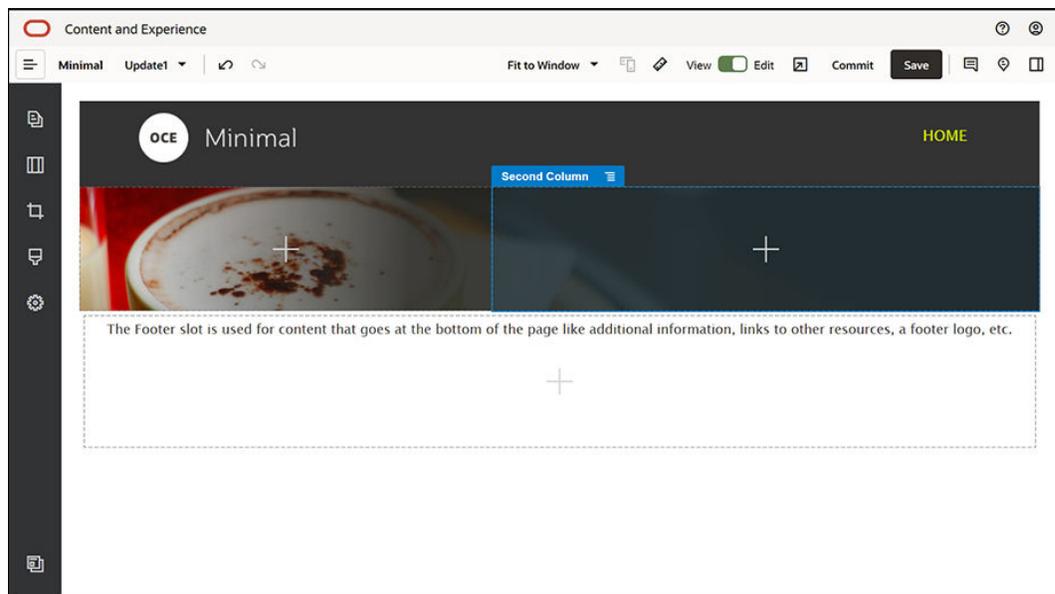
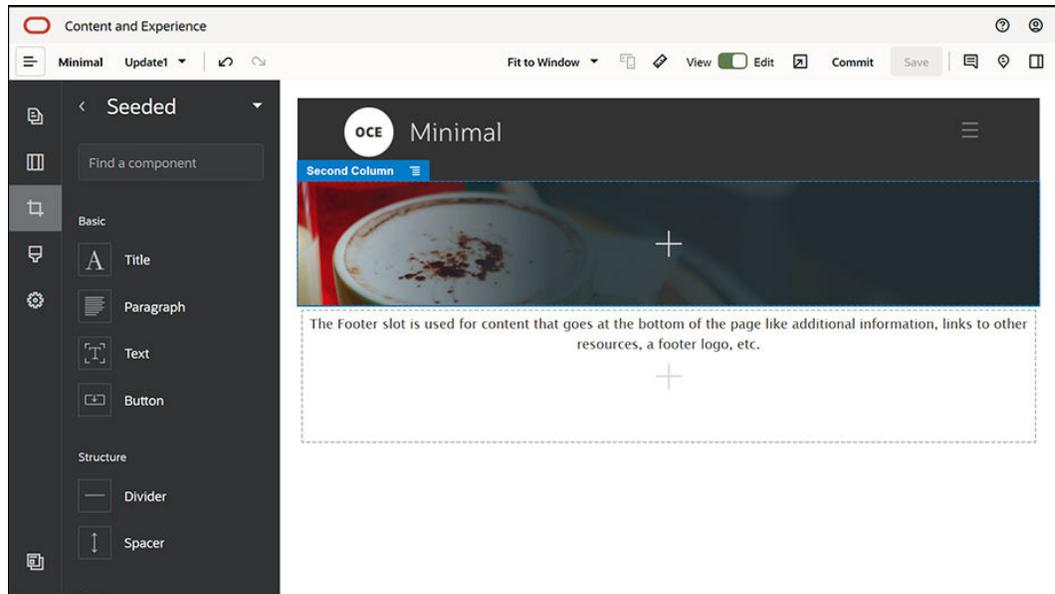
5. Задайте настройки макета раздела на вкладке **Общие**. Нажмите **Пользовательские настройки**, чтобы указать приведенные ниже настройки.

Свойство	Значение
Ширина первого столбца (%)	43
Ширина второго столбца (%)	57
Точка останова при ответе (пикс.)	1,023
Поведение при ответе	Скрыть первый столбец

Завершите настройку на вкладке **Фон**.

Свойство	Значение
Изображение	Banner1.jpg из папки Minimal-Images
Позиция	Посередине по центру
Масштаб	Растянуть

6. На левой боковой панели нажмите , чтобы отобразить список предварительно заданных компонентов.
7. Из списка предварительно заданных компонентов перетащите компонент "Заголовок" во второй столбец макета "Два столбца".



- Нажмите компонент "Заголовок" и введите "WELCOME TO THE REVOLUTION". В текстовом редакторе выделите текст и задайте цвет шрифта **Белый**. Завершите задание настроек компонента "Заголовок" на вкладке **Общие**.

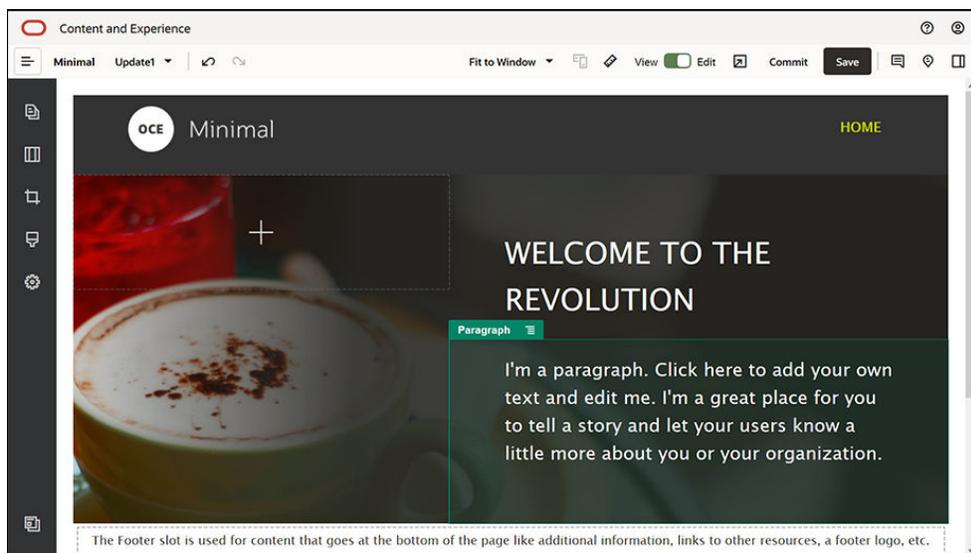
Свойство	Значение
Первые	6vw
Внизу	1.8vw
Слева	6vw
Справа	6vw

9. Из списка предварительно заданных компонентов на левой боковой панели перетащите компонент "Абзац" под компонент "Заголовок" во втором столбце макета "Два столбца". Завершите настройку на вкладке **Общие**.

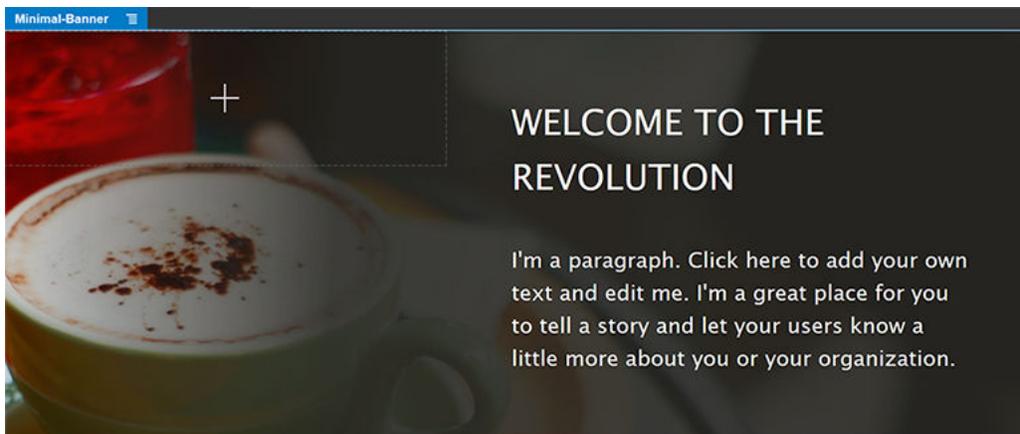
Свойство	Значение
Первые	1.8vw
Внизу	6vw
Слева	6vw
Справа	6vw

10. Нажмите в компоненте "Абзац", чтобы ввести следующий текст: "I'm a paragraph. Click here to add your own text and edit me. I'm a great place for you to tell a story and let your users know a little more about you or your organization." (Я — абзац. Нажмите здесь, чтобы добавить свой текст и отредактировать меня. Я отлично подхожу для того, чтобы вы могли рассказать историю и сообщить своим пользователям немного больше о себе или своей организации.)

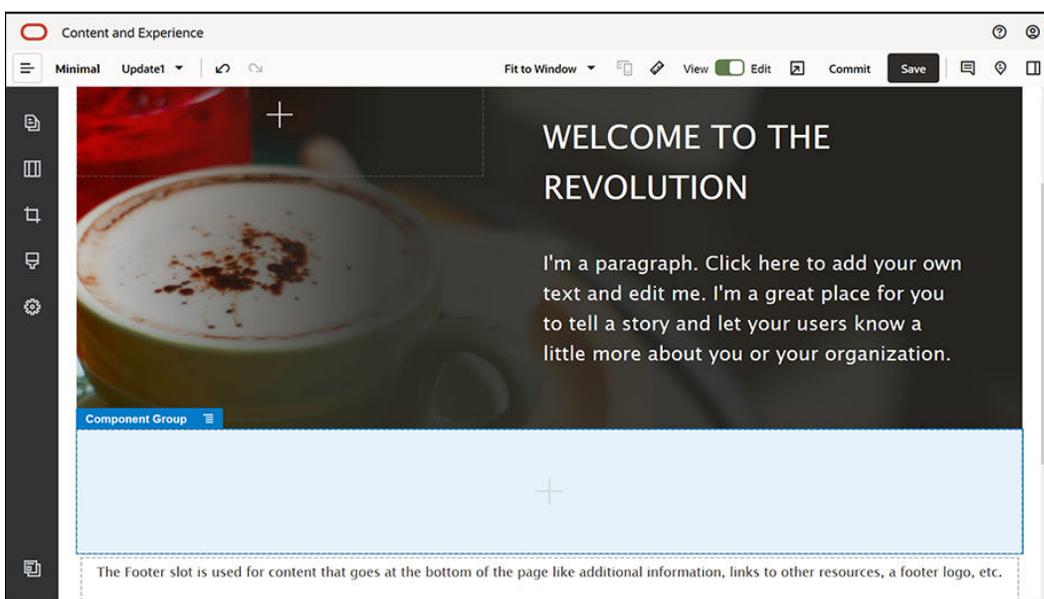
В текстовом редакторе выделите текст и задайте размер 24. Также в текстовом редакторе задайте цвет шрифта **Белый**.



11. Теперь баннер готов. Давайте сохраним эту группу компонентов как пользовательскую группу компонентов, чтобы использовать ее позже на других страницах веб-сайта. Нажмите на значок меню группы компонентов , а затем нажмите **Сохранить**. В диалоговом окне "Сохранение группы компонентов" в поле **Имя** введите "Minimal-Banner", а затем нажмите кнопку **Сохранить**. Вы заметите, что данное имя (**Minimal-Banner**) теперь отображается для группы компонентов.

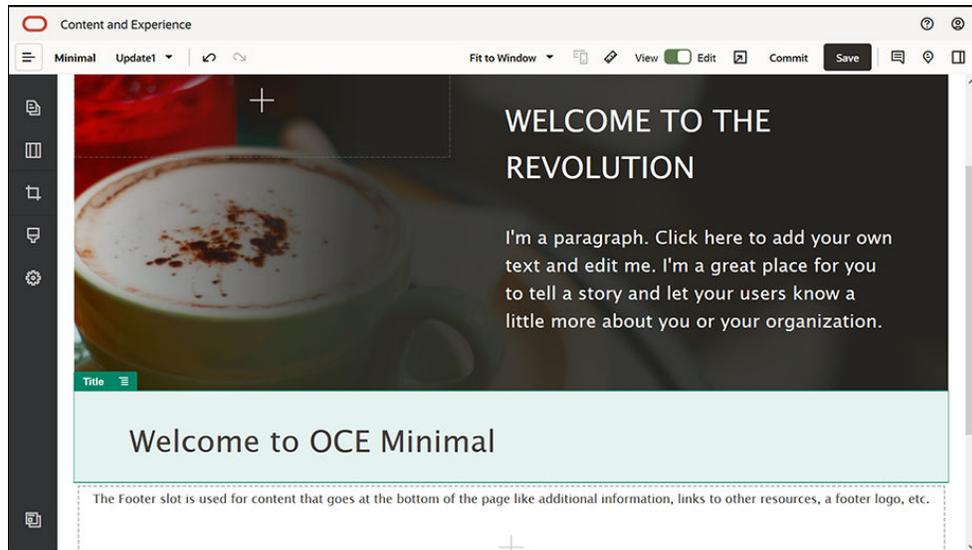


- Из левой боковой панели перетащите еще одну группу компонентов в ячейку "Тело" под уже добавленной группой компонентов Minimal-Banner.



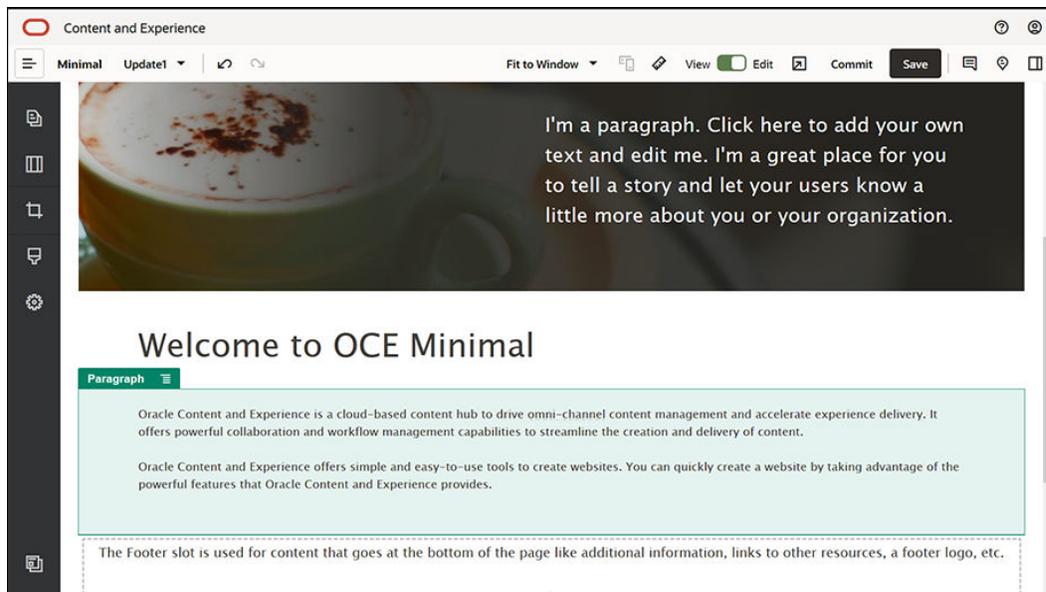
- Перетащите компонент "Заголовок" в данную группу компонентов.
- Нажмите компонент "Заголовок" и введите "Welcome to OCE Minimal".
- Завершите задание настроек компонента "Заголовок" на вкладке **Общие**.

Свойство	Значение
Первые	3vw
Внизу	1.8vw
Слева	6vw
Справа	6vw

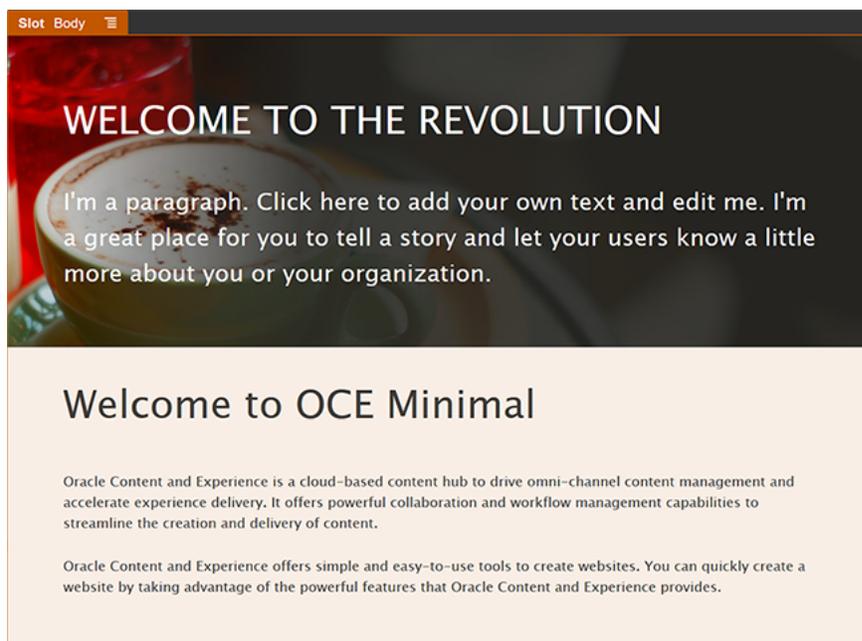


16. Из левой боковой панели перетащите компонент "Абзац" в группу компонентов под компонент "Заголовок". Нажмите в компоненте "Абзац", чтобы ввести следующий текст:
 "Oracle Content Management — централизованная платформа контента в облаке для управления многоканальным контентом и ускорения доставки компонентов взаимодействия. It offers powerful collaboration and workflow management capabilities to streamline the creation and delivery of content."
 (Oracle Content and Experience — централизованная платформа контента в облаке для управления многоканальным контентом и ускорения его доставки. Она предоставляет мощные возможности управления потоками операций и совместной работой, позволяющие упростить создание и доставку контента.)
 "Oracle Content Management предлагает простые и удобные в использовании инструменты для создания веб-сайтов. Вы можете быстро создать веб-сайт, используя преимущества мощных функций, предоставляемых Oracle Content Management".
17. Завершите задание настроек компонента "Абзац" на вкладке **Общие**.

Свойство	Значение
Первые	20px
Внизу	50px
Слева	6vw
Справа	6vw

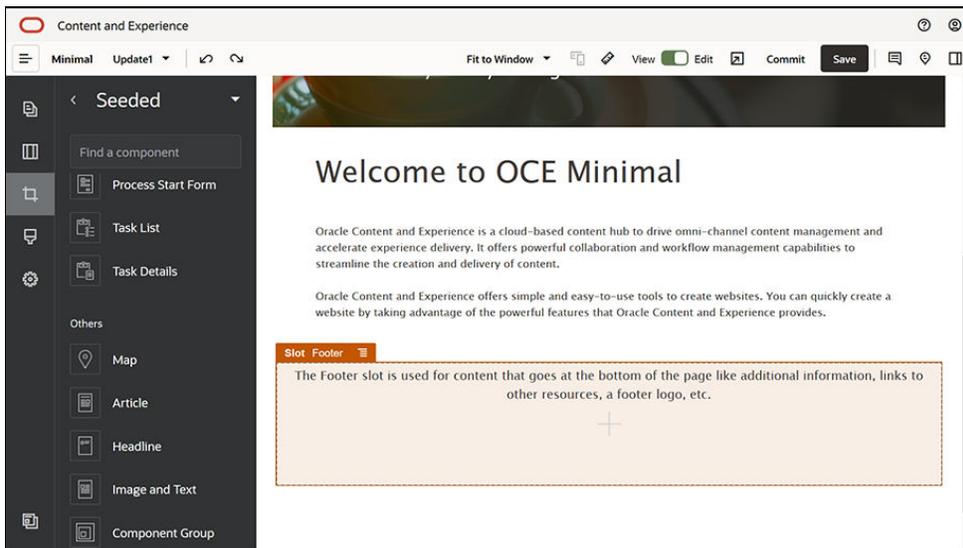


18. Создание ячейки "Тело" завершено. Давайте сохраним эту группу компонентов как пользовательскую группу компонентов, чтобы использовать ее позже на других страницах веб-сайта. Нажмите на значок меню группы компонентов , а затем нажмите **Сохранить**. В диалоговом окне "Сохранение группы компонентов" в поле **Имя** введите "Minimal-Body", а затем нажмите кнопку **Сохранить**.
19. Нажмите **Сохранить** в правом верхнем углу окна Site Builder, чтобы сохранить свои изменения. Теперь ячейка "Тело" должна выглядеть следующим образом:



Давайте завершим создание ячейки "Нижний колонтитул":

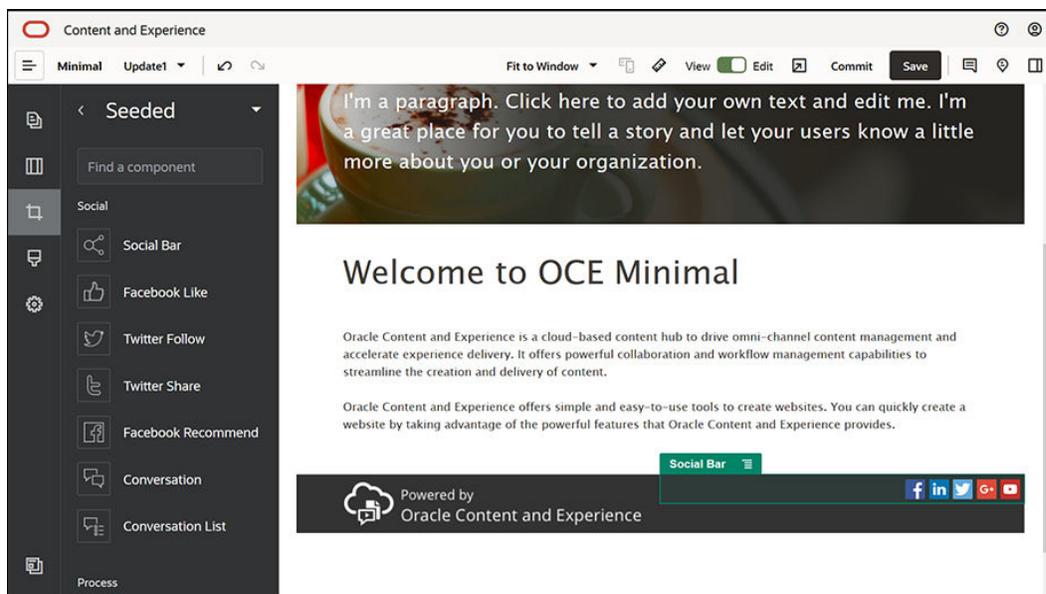
1. Из левой боковой панели перетащите группу компонентов в ячейку "Нижний колонтитул". В настройках группы компонентов в поле **Цвет** введите значение **#333333**.



2. Перетащите компонент "Изображение" в группу компонентов и задайте ее настройки на вкладке **Общие**.

Свойство	Значение
Выбрать	Powered_by_OCE.png из папки Minimal-Images
Выравнивание	Слева
Ширина	Отмените выбор Задать ширину
Первые	0.9vw
Внизу	0.9vw
Слева	6vw
Справа	0

3. Из левой боковой панели перетащите компонент "Социальная панель" в группу компонентов справа от компонента "Изображение".



Задайте настройки компонента "Социальная панель" на вкладке **Общие**.

Свойство	Значение
Первые	1.8vw
Внизу	1.8vw
Слева	0.3vw
Справа	6vw

На вкладке **Общие** нажмите **Значки**, а затем нажмите имя значка, чтобы завершить настройку.

Свойство	Значение
URL	<ul style="list-style-type: none"> • https://www.facebook.com/Oracle/ (для Facebook) • https://www.linkedin.com/company/oracle/ (для LinkedIn) • https://twitter.com/Oracle (для Twitter) • https://www.youtube.com/oracle/ (для YouTube)
Назначение	Открыть в новом окне

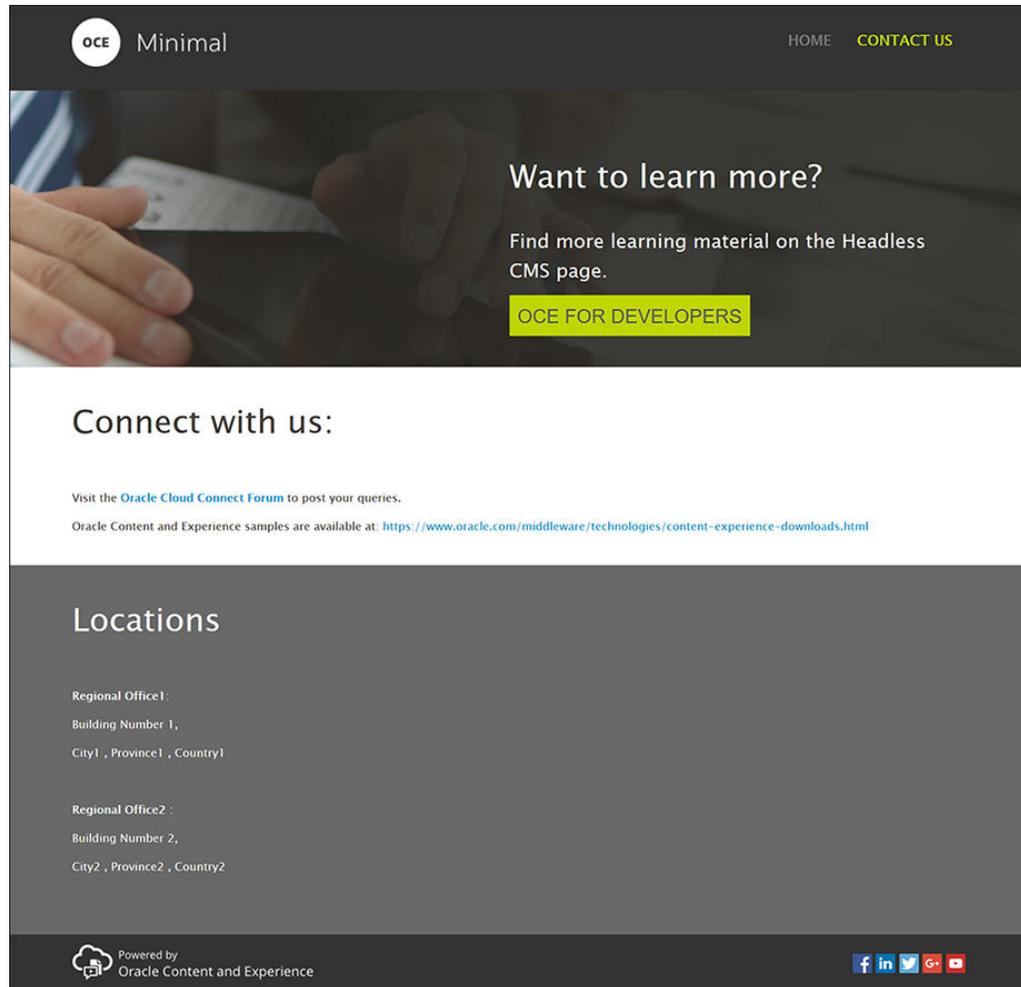
4. Теперь нижний колонтитул готов. Давайте сохраним эту группу компонентов как пользовательскую группу компонентов, чтобы использовать ее позже на других страницах веб-сайта. Нажмите на значок меню группы компонентов , а затем нажмите **Сохранить**. В диалоговом окне "Сохранение группы компонентов" в поле **Имя** введите "Minimal-Footer", а затем нажмите кнопку **Сохранить**. Данная группа компонентов должна выглядеть следующим образом:



5. Нажмите **Сохранить** в правом верхнем углу окна Site Builder, чтобы сохранить свои изменения.

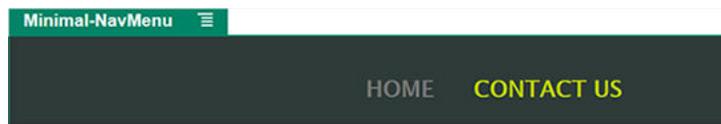
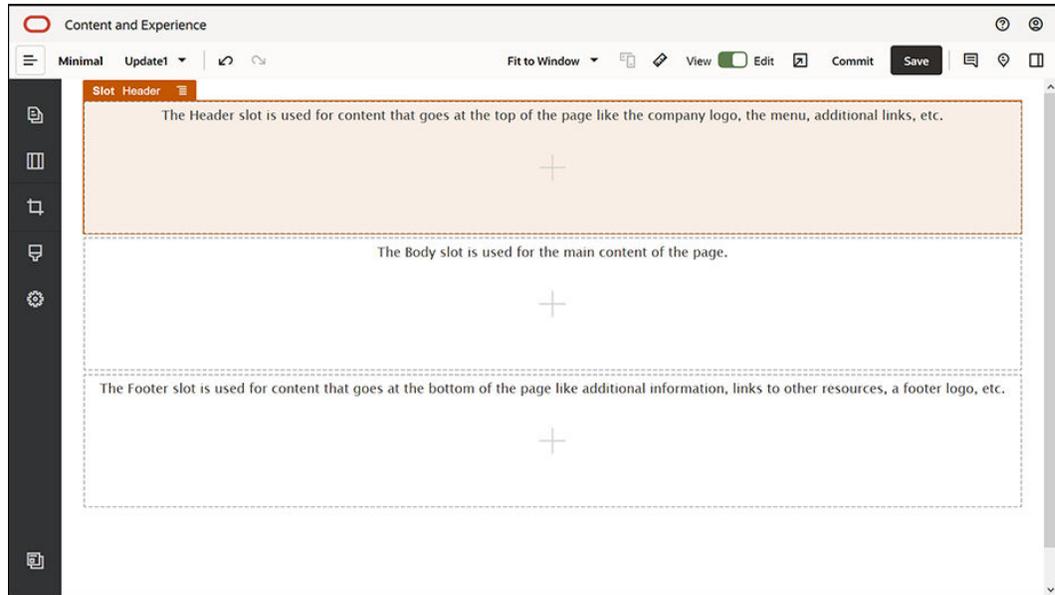
Выполните предварительный просмотр первой страницы своего веб-сайта, нажав  в правом верхнем углу окна Site Builder. Сайт до сих пор не опубликован и его не могут просматривать другие пользователи.

Создание домашней страницы завершено. Давайте создадим страницу CONTACT US (Контакты). Вот как страница контактов будет выглядеть после завершения:

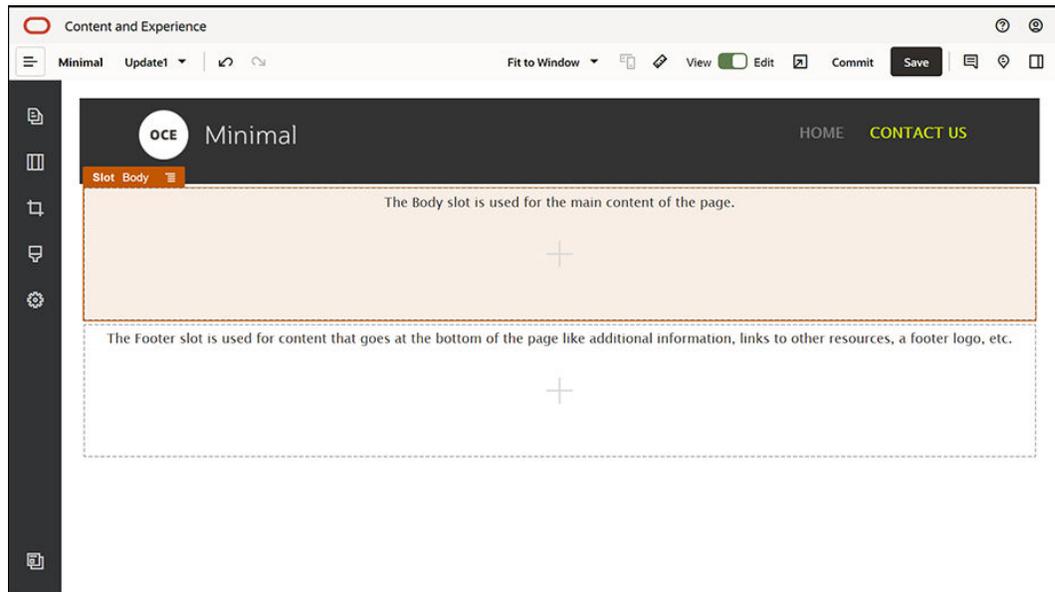


Давайте добавим компоненты в различные ячейки:

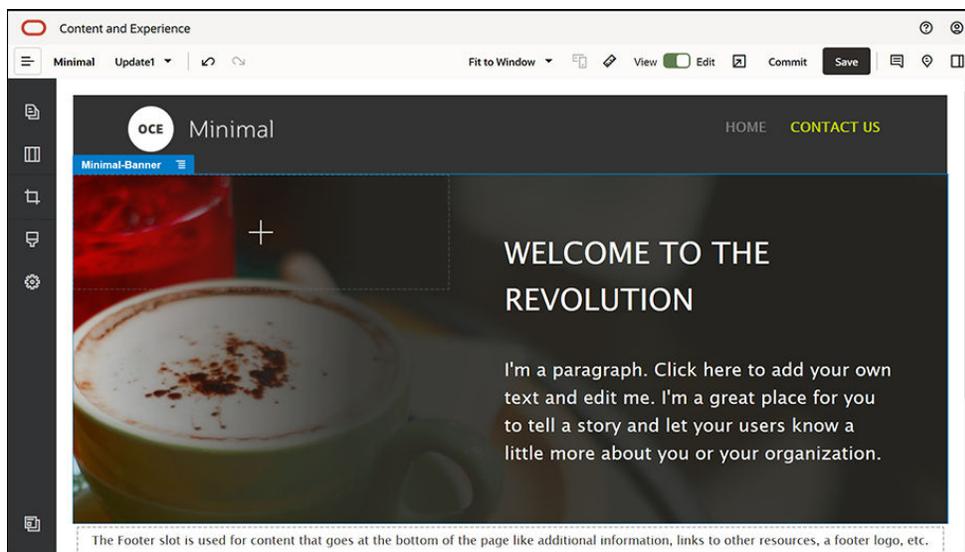
1. На левой боковой панели нажмите , а затем нажмите **Добавить страницу**.
2. В поле **Имя страницы** введите "CONTACT US" (Контакты) и нажмите **Закрыть**. К вашему веб-сайту добавлена новая страница.
3. На левой боковой панели нажмите , затем выберите **Пользовательские**.
4. Перетащите компонент Minimal-Header (созданный и сохраненный ранее) в ячейку "Заголовок".



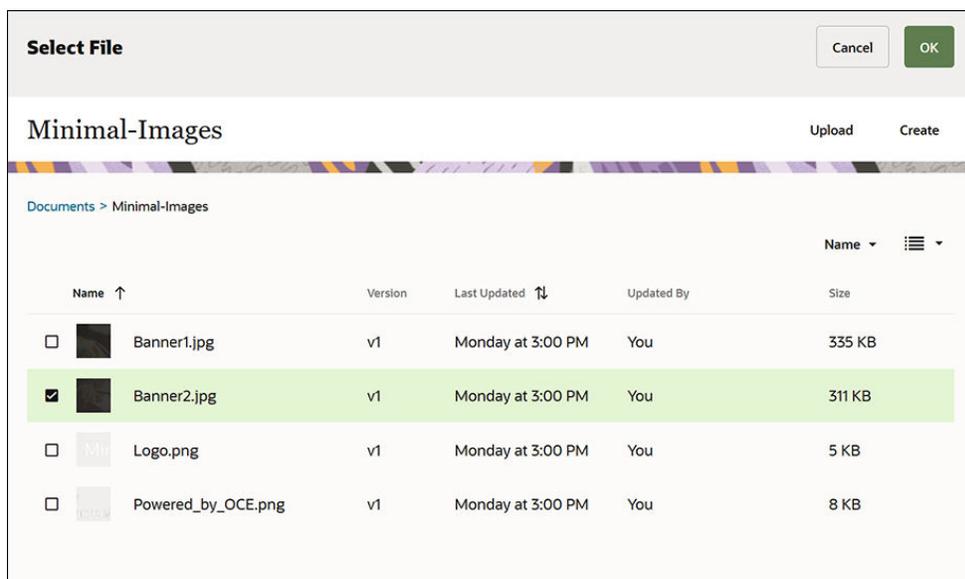
Обратите внимание, что компонент Minimal-NavMenu автоматически выбрал только что созданную, новую страницу CONTACT US (Контакты).



5. Из левой боковой панели перетащите компонент Minimal-Banner в ячейку "Тело".



6. Давайте изменим баннер, чтобы он отличался от баннера домашней страницы и соответствовал странице CONTACT US (Контакты). В группе компонентов измените настройки макета раздела "Два столбца". На вкладке **Фон** в поле **Изображение** нажмите **Выбрать изображение**, а затем выберите **Banner2.jpg** в папке **Minimal-Images**, созданной ранее в разделе **Документы**.



7. В макете раздела "Два столбца" измените текст в компонентах "Заголовок" и "Абзац".
 - В компоненте "Заголовок" введите "Хотите узнать больше?".
 - В компоненте "Абзац" введите "Find more learning material on the Headless CMS page." (Найдите дополнительные учебные материалы на странице автономной CMS.).

8. На левой боковой панели нажмите , затем выберите **Заданные предварительно**.
9. На изображении страницы CONTACT US (Контакты) с состав баннера входит кнопка **OCE FOR DEVELOPERS** (OCE для разработчиков), поэтому давайте добавим эту кнопку на баннер. В группе компонентов Minimal-Banner перетащите компонент "Кнопка" во второй столбец макета раздела "Два столбца" (под недавно добавленным компонентом "Абзац"). Завершите задание настроек компонента "Кнопка" на вкладке **Общие**.

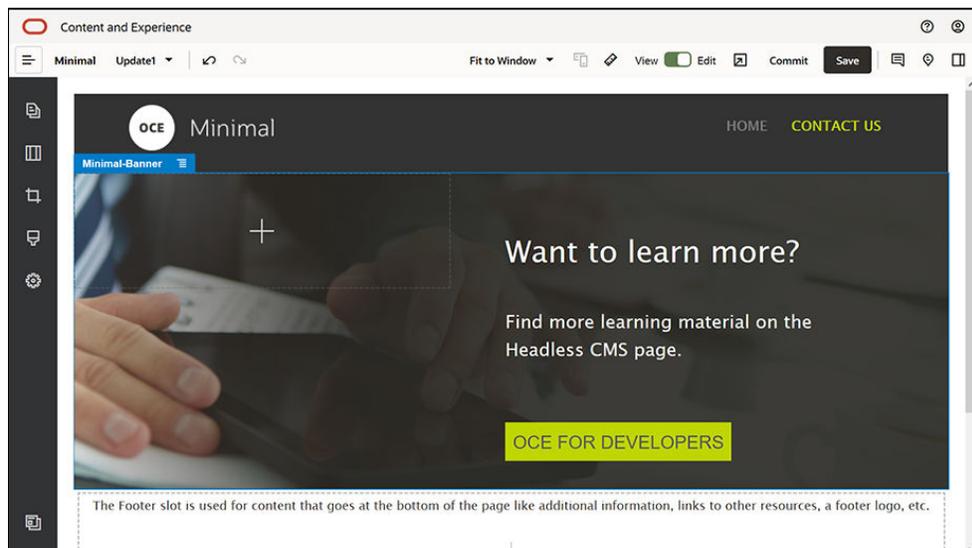
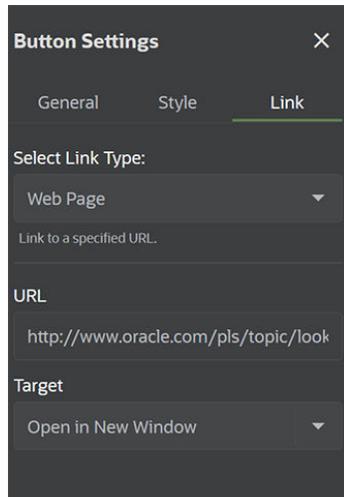
Свойство	Значение
Подпись	OCE FOR DEVELOPERS
Первые	0.3vw
Внизу	3vw
Слева	6vw
Справа	0.3vw

На вкладке **Стиль** выберите **Настройка** и завершите настройку.

Свойство	Значение
Цвет фона	#c0d600
Шрифт	<ul style="list-style-type: none"> • Введите 24 в качестве размера. • Введите цвет #58595b.
Рамка	Нет
Цвет при наведении курсора	<ul style="list-style-type: none"> • Задайте Фон = #e1fa00. • Задайте Шрифт = #58595b. • Задайте Граница = #2222dd.
Углы	0

Завершите настройку на вкладке **Ссылка**.

Свойство	Значение
Выберите тип ссылки	Веб-страница
URL	http://www.oracle.com/pls/topic/lookup?ctx=cloud&id=content-cloud-headless
Назначение	Открыть в новом окне



10. На левой боковой панели нажмите , затем выберите **Пользовательские**.
11. Перетащите компонент Minimal-Body ниже компонента Minimal-Banner в ячейку "Тело".
12. В компоненте Minimal-Body измените текст в компонентах "Заголовок" и "Абзац".
 - a. В компоненте "Заголовок" введите "Контакты:".
 - b. В компоненте "Абзац" введите подробные сведения, такие как адрес электронной почты и другие ссылки, связанные со службой поддержки: "Посетите [Oracle Cloud Connect Forum](https://www.oracle.com/middleware/technologies/content-experience-downloads.html), чтобы разместить свои запросы."
"Примеры Oracle Content Management доступны по адресу: <https://www.oracle.com/middleware/technologies/content-experience-downloads.html>"

13. Из левой боковой панели перетащите еще один компонент Minimal-Body в ячейку "Тело", расположенную ниже компонента Minimal-Body, добавленного ранее.
14. В настройках группы компонентов на вкладке **Фон** в поле **Цвет** введите значение **#696969**.
15. В компоненте Minimal-Body измените компоненты "Заголовок" и "Абзац".
 - a. В компоненте "Заголовок" введите "Местонахождения". В текстовом редакторе выделите текст и задайте цвет шрифта **Белый**.
 - b. В компоненте "Абзац" введите указанный ниже текст.
"Regional Office1:
Building Number 1,
City1, Province1, Country1
Regional Office2:
Building Number 2,
City2, Province2, Country2"
В текстовом редакторе выделите текст и задайте цвет шрифта **Белый**.
16. Теперь давайте добавим нижний колонтитул на страницу CONTACT US (Контакты). Перетащите компонент Minimal-Footer в ячейку "Нижний колонтитул".
17. Нажмите **Сохранить** в правом верхнем углу окна Site Builder, чтобы сохранить свои изменения.

Выполните предварительный просмотр веб-сайта, нажав  в правом верхнем углу окна Site Builder, чтобы убедиться, что все выглядит хорошо. Убедитесь в том, что переход по изображению логотипа на странице CONTACT US (Контакты) возвращает на домашнюю страницу. Проверьте меню, чтобы убедиться, что навигация между страницами сайта работает правильно.

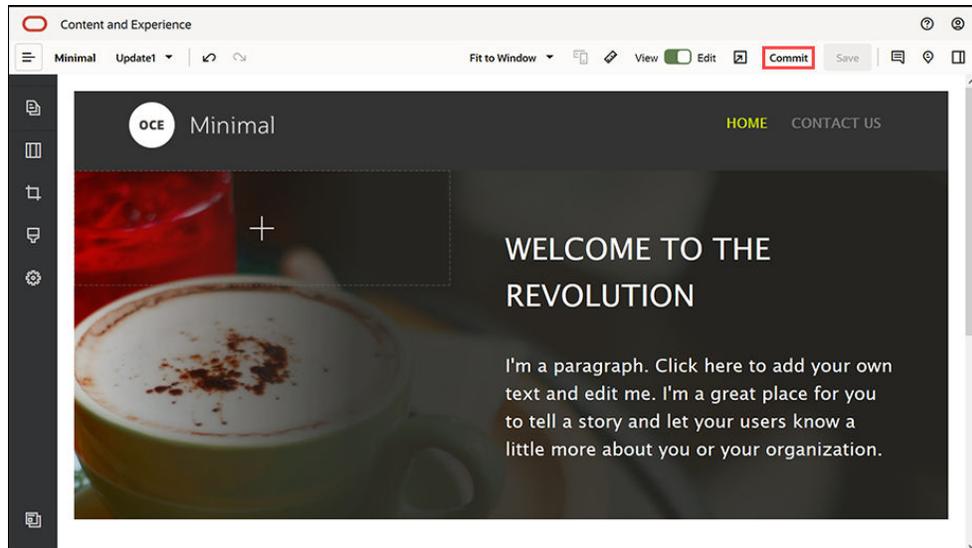
Ваш сайт готов к публикации.

[Следующий шаг: публикация веб-сайта](#)

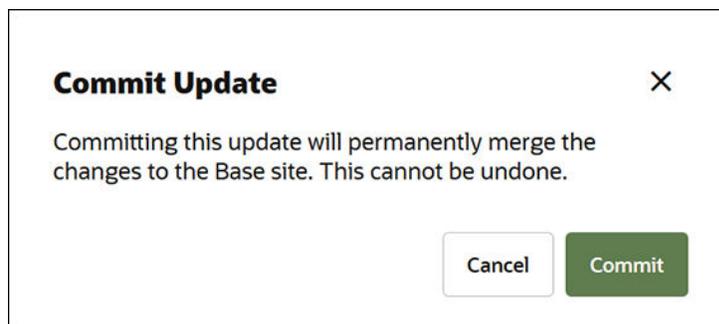
Шаг 3. Публикация веб-сайта

Теперь, после успешного создания веб-сайта вы можете его опубликовать, чтобы сделать доступным для пользователей.

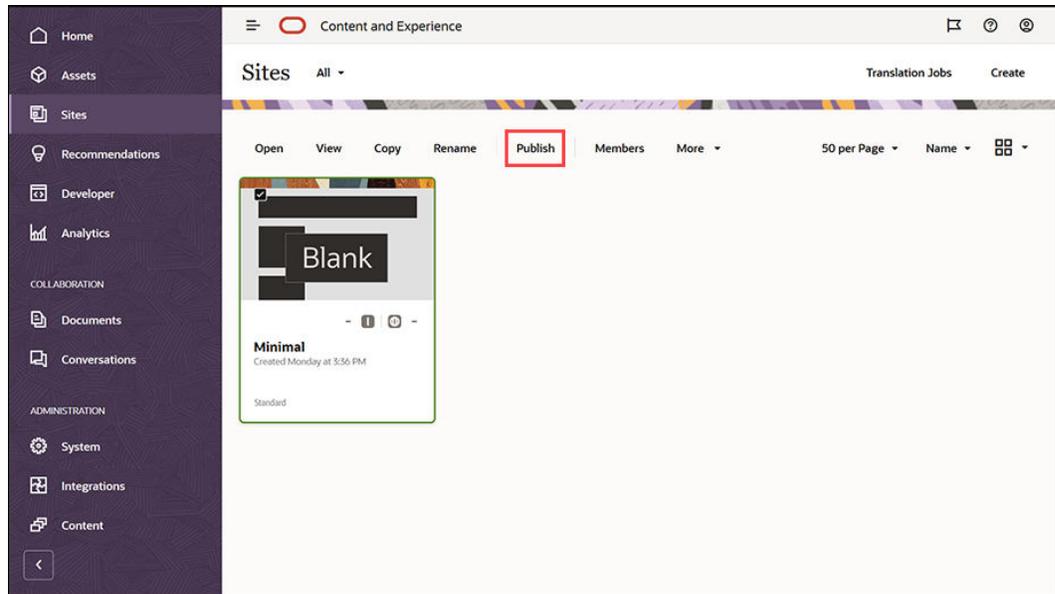
1. Если все выглядит хорошо, вы можете зафиксировать свои изменения на базовом веб-сайте, нажав в правом верхнем углу Site Builder кнопку **Фиксировать**.



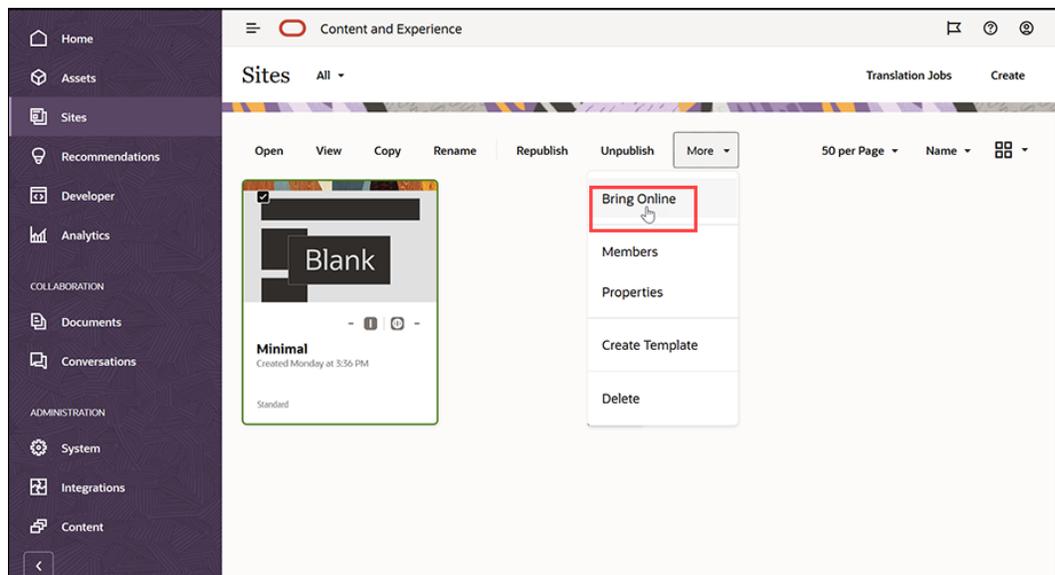
2. В диалоговом окне "Фиксация обновления", нажмите **Фиксировать**.

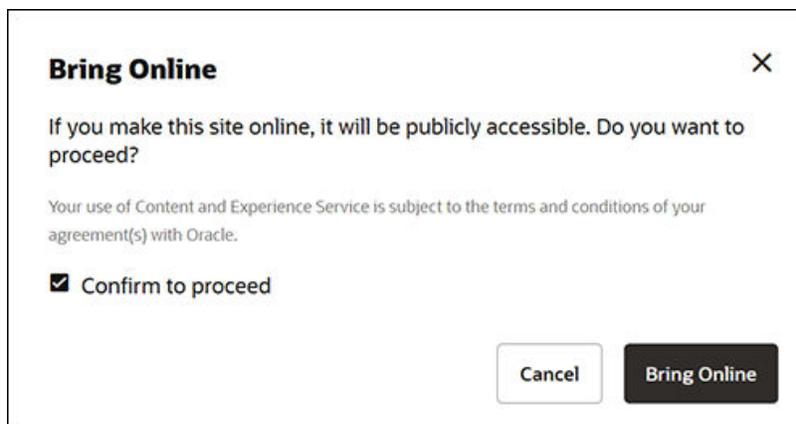


3. После фиксации изменений веб-сайт готов к публикации.
4. Закройте Site Builder
5. Нажмите **Сайты** в левом меню навигации в веб-интерфейсе Oracle Content Management и выберите веб-сайт.
6. Выберите **Опубликовать** в строке меню или контекстном меню.



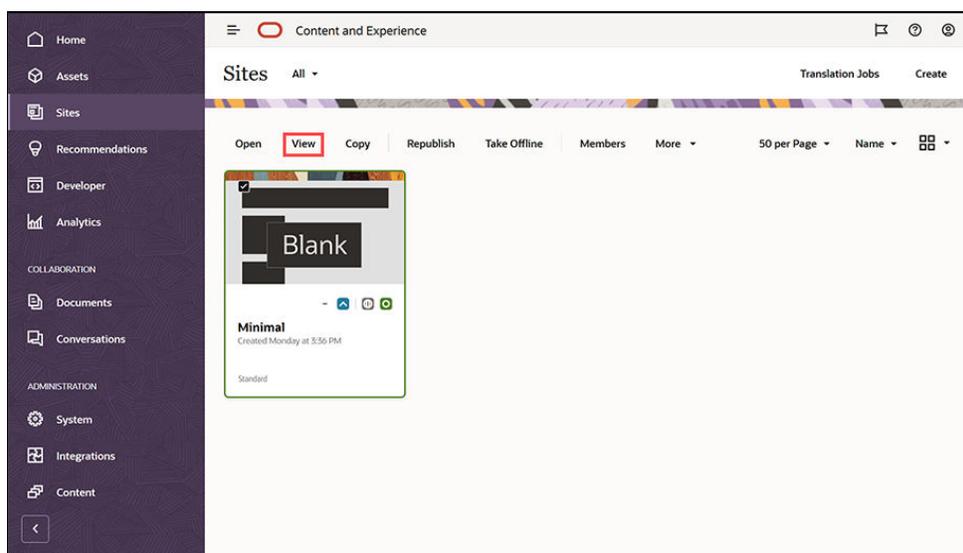
7. После публикации веб-сайта в верхней части страницы отображается уведомление о публикации веб-сайта.
8. На странице "Сайты" снова выберите веб-сайт, а затем выберите **Перевести в онлайн-режим** в строке меню или в меню, вызываемом нажатием правой кнопкой мыши. В диалоговом окне "Перевод в онлайн-режим" выберите **Подтвердить, чтобы продолжить** и нажмите **Перевести в онлайн-режим**.





9. Вот и все. Ваш веб-сайт работает в онлайн-режиме, и другие пользователи могут его видеть.

Чтобы просмотреть свой общедоступный веб-сайт, выберите **Вид** в строке меню или контекстном меню.



Дополнительные действия

Вы можете настроить свой веб-сайт в соответствии с потребностями вашей организации. Вот несколько ссылок, которые помогут начать работать:

- [Использование стилей и форматирование](#)
- [Редактирование рекомендаций и советов](#)
- [Управление сайтами и настройки сайтов](#)
- [Загрузка файлов сайта](#)
- [Добавление страниц](#)
- [Перемещение страниц](#)

- Организация содержимого страницы
- Изменение настроек страницы
- Изменение макета страницы
- Работа с таблицами
- Настройка свойств поисковых систем

Часть II

Создание и редактирование сайтов

В этой части подробно описано, как создавать сайты с помощью шаблонов и тем, а также редактировать сайты с помощью компонентов и макетов для организации и добавления контента. Здесь содержатся указанные ниже главы.

- [Создание сайтов](#)
- [Редактирование сайтов](#)
- [Использование шаблонов и тем для сайтов](#)
- [Управление пользовательскими компонентами и макетами](#)
- [Работа со страницами сайта](#)
- [Организация содержимого страницы](#)
- [Использование встроенных компонентов](#)

4

Создание сайтов

Любой пользователь, имеющий соответствующие разрешения, может создать веб-сайт с помощью Oracle Content Management. Для этого не требуются проприетарные инструменты, код и программное обеспечение. Графический пользовательский интерфейс интуитивно понятен и удобен в использовании.

Чтобы создать сайт, необходимо выбрать шаблон. Шаблон содержит все необходимые элементы для создания сайта, в том числе кодовую платформу сайта, стандартный сайт с примерами страниц и содержимого, тему со стилями, ресурсы (например, изображения), а также пользовательские компоненты. См. [Общие сведения о процессе создания сайта](#).

Управление сайтами Oracle Content Management упрощает и ускоряет предоставление компонентов взаимодействия бизнес-пользователям, обеспечивая для ИТ-отделов простой способ управления и отслеживания компонентов взаимодействия из централизованного местоположения с возможностью полного управления всем жизненным циклом таких компонентов, снижая затраты на создание и поддержание каждого нового компонента, необходимого компании. Управление встроено в ядро Oracle Content Management; ваш администратор просто должен включить его. См. раздел Конфигурирование настроек сайтов и активов в документе *Administering Oracle Content Management*.

- [Общие сведения о процессе создания сайта](#)
- [Создание сайтов](#)
- [Копирование сайтов](#)
- [Управление запросами сайта](#)
- [Изменение сведений о запросе сайта](#)
- [Просмотр политик запросов сайта](#)

Создание сайтов

Чтобы создать сайт, выберите шаблон, присвойте имя сайту и добавьте контент. Если включено управление, то перед добавлением контента необходимо отправить запрос сайта, а затем после его утверждения добавить контент.

Чтобы пользователи могли создавать сайты, администратору необходимо включить функцию создания сайтов и сделать доступными шаблоны. Если вы видите параметр **Создать** на странице Сайты или нет доступных шаблонов, обратитесь к администратору. См. [Начало работы с сайтами](#) и [Общие сведения об управлении сайтами](#).

При создании корпоративного сайта, позволяющего использовать активы и многоязычные сайты, сайт будет связан с репозиторием, политикой локализации и, возможно, языком по умолчанию. Если репозиторий и политика локализации недоступны, создайте их или попросите администратора содержимого создать их.

 **Примечание.**

В версии Oracle Content Management Starter переводы недоступны, управление не может быть включено, и можно создать только один сайт. Для получения полного набора функций и неограниченного количества сайтов выполните обновление до версии Oracle Content Management Premium.

1. Нажмите **Сайты**.
2. Нажмите **Создать**.
3. На странице Выбор шаблона выберите шаблон для своего сайта.
 - Если включено управление, этот шаблон будет определять, является ли созданный сайт стандартным (не использует репозиторий активов или политику локализации) или корпоративным (использует репозиторий активов и политику локализации).
4. На странице Настройка сайта отображается информация о том, какое утверждение необходимо получить до создания сайта, какой минимальный уровень безопасности необходимо установить для сайта, а также показаны изображения на страницах сайта. Если утверждение разрешено только определенным лицам, нажмите **Показать утверждающих**, чтобы узнать, кто может утвердить ваш запрос сайта.
 - Если управление не включено, выберите для создаваемого сайта тип **Стандартный** или **Корпоративный**.
 - Если вы создаете стандартный сайт, нажмите **Далее**, чтобы перейти на следующую страницу.
 - Если вы создаете корпоративный сайт, выполните указанные ниже действия.
 - a. Выберите репозиторий по умолчанию, используемый для управления контентом сайта. Если вы — администратор репозитория и нет доступных репозиториях или требуется использовать новый репозиторий, выберите **Создать новый репозиторий** и выполните соответствующую процедуру, чтобы создать репозиторий. Сведения об использовании нескольких репозиториях на узле см. Предоставление доступа к нескольким репозиториям на уровне узла в разделе *Управление активами с помощью Oracle Content Management*.
 - b. Выберите политику локализации для данного сайта. Если вы — администратор репозитория и нет доступных политик или требуется использовать новую политику, выберите **Создать новую политику локализации** и выполните соответствующую процедуру, чтобы создать политику.
 - c. Если в политике локализации шаблона не задан язык по умолчанию, выберите его сейчас. Вы видите только те языки, которые требуются в соответствии с политикой локализации.
 - d. Нажмите **Далее** для перехода на следующую страницу.
5. На странице Добавление сведений укажите следующую информацию.

- Введите имя сайта. Это имя используется в URL-адресе сайта. Можно употреблять буквы, цифры, символы подчеркивания (_) и дефисы (-). URL-адрес следует вводить с учетом регистра. Введенный пробел автоматически заменяется дефисом. Не используйте для создаваемого сайта имя уже существующего сайта, отличающееся только регистром букв. Например, если существует сайт с именем *ABC*, не следует создавать другой сайт с именем *Abc*.

 **Примечание.**

Не используйте для шаблонов, тем, компонентов, сайтов и страниц сайтов следующие имена: `authsite`, `content`, `pages`, `scstemplate_*`, `_comps`, `_components`, `_compsdelivery`, `_idcservice`, `_sitescloud`, `_sitesclouddelivery`, `_themes`, `_themesdelivery`. Следующие имена можно использовать только для страниц сайтов (не для шаблонов, тем, компонентов и сайтов): `documents`, `sites`.

- Если вы создаете корпоративный сайт, политика шаблона может разрешать вам изменить префикс сайта по умолчанию на понятные значения URL. Данный префикс будет добавлен к кодовым значениям элемента контента (части URL-адреса, относящейся к странице или активу).
- Дополнительно можно ввести описание сайта.
- Введите обоснование для запроса сайта (необязательно). Это поможет администратору сайта понять, должен ли запрос быть утвержден.

6. Нажмите кнопку **Готово**.

Если запрос сайта требует утверждения, тому, кто должен его утвердить, по электронной почте отправляется сообщение. В случае утверждения вы получите электронное сообщение с уведомлением о том, что сайт утвержден и будет создан. Если запрос отклонен, вы получите по электронной почте уведомление об этом. Ожидающие запросы можно посмотреть на странице Сайты, выбрав **Запросы** в меню фильтра. Если запрос отклонен, должно отобразиться сообщение о том, почему он отклонен, чтобы можно было исправить проблему и снова отправить запрос.

Если запрос сайта настроен на автоматическое утверждение, сайт создается автоматически. В индикаторе выполнения отображается имя нового сайта и состояние создания. Когда сайт создан, его имя появляется в списке сайтов, а его статус будет "автономный".

Если вы создали корпоративный сайт, в выбранном вами репозитории создается соответствующая коллекция сайтов. Если предоставить доступ к сайту другому пользователю, этот пользователь получит такие же разрешения в отношении связанной коллекции.

После создания сайта вы можете предоставить общий доступ к нему, изменить параметры безопасности, добавить и редактировать контент, опубликовать сайт или перевести его в онлайн-режим. См. [Управление сайтами и настройки сайтов](#) и [Редактирование сайтов](#).

Копирование сайтов

Возможность копирования сайтов упрощает создание новых сайтов. Все элементы исходного сайта, в том числе тема, все неопубликованные обновления, страницы, контент страниц, рекомендации, все остальные активы, такие как изображения, и политики копируются в новый сайт с указанными именами.

Примечание.

При копировании сайта, использующего контент из нескольких репозиториях, необходимо сделать это с помощью инструментария ОСЕ. См. разделы [Разработка с помощью инструментария ОСЕ](#) и [Использование утилиты командной строки СЕС](#).

1. Нажмите **Сайты**.
2. Выберите сайт, который требуется скопировать, и выберите **Копировать** в контекстном меню или  на панели действий.
3. На странице Настройка сайта отображается информация о том, какое утверждение необходимо получить до создания сайта, какой минимальный уровень безопасности необходимо установить для сайта, а также показаны изображения на страницах сайта. Выполните следующие шаги:
 - a. В раскрывающемся списке **Копировать** выберите, следует ли копировать только базовый сайт или также включать обновления. Если вы включите обновления, ваш скопированный сайт будет содержать все отложенные обновления с исходного сайта. Если вы копируете стандартный сайт, нажмите на стрелку для перехода на следующую страницу и сразу перейдите к шагу 4.
 - b. Если вы копируете корпоративный сайт, то репозиторий активов по умолчанию, политика локализации, требуемые языки и язык по умолчанию устанавливаются на основе исходного сайта. При желании вы можете выбрать другой репозиторий для управления контентом сайта.
 - c. Нажмите **Далее** для перехода на следующую страницу.
4. На странице Добавление сведений укажите следующую информацию.
 - Введите имя сайта. Это имя используется в URL-адресе сайта. Можно употреблять буквы, цифры, символы подчеркивания (_) и дефисы (-). URL-адрес следует вводить с учетом регистра. Введенный пробел автоматически заменяется дефисом. Не используйте для создаваемого сайта имя уже существующего сайта, отличающееся только регистром букв. Например, если существует сайт с именем ABC, не следует создавать другой сайт с именем Abc.

 **Примечание.**

Не используйте для шаблонов, тем, компонентов, сайтов и страниц сайтов следующие имена: authsite, content, pages, scstemplate_*, _comps, _components, _compsdelivery, _idcservice, _sitescloud, _sitesclouddelivery, _themes, _themesdelivery. Следующие имена можно использовать только для страниц сайтов (не для шаблонов, тем, компонентов и сайтов): documents, sites.

- Если вы копируете корпоративный сайт, вы можете при желании изменить префикс сайта по умолчанию на понятные значения URL. Данный префикс будет добавлен к кодовым значениям элемента контента (части URL-адреса, относящейся к странице или активу).
- Добавьте описание сайта (необязательно).
- Введите обоснование для запроса сайта (необязательно). Это поможет администратору сайта понять, должен ли запрос быть утвержден.

5. Нажмите кнопку **Готово.**

Если запрос сайта требует утверждения, сайт будет создан после того, как администратор сайта утвердит его. Ожидающие запросы можно посмотреть на странице Сайты, выбрав **Запросы** в меню фильтра. Если запрос отклонен, отобразится сообщение о том, почему он отклонен, чтобы можно было исправить проблему и снова отправить запрос.

Если запрос сайта настроен на автоматическое утверждение, сайт создается автоматически. В индикаторе выполнения отображается имя нового сайта и состояние создания. Когда сайт создан, его имя появляется в списке сайтов, а его статус будет "автономный".

Если вы скопировали корпоративный сайт, в выбранном вами репозитории создается соответствующая коллекция сайтов. Если предоставить доступ к сайту другому пользователю, этот пользователь получит такие же разрешения в отношении связанной коллекции.

После создания сайта вы можете предоставить общий доступ к нему, изменить параметры безопасности, добавить и редактировать контент, опубликовать сайт или перевести его в онлайн-режим. При совместном использовании сайта, который имеет доступ к нескольким репозиториям, общий доступ предоставляется только для репозитория по умолчанию. Для совместного использования какого-либо дополнительного репозитория необходимо открыть общий доступ к нему отдельно. См. разделы [Управление сайтами и настройки сайтов](#) и [Редактирование сайтов](#).

Управление запросами сайта

Если управление сайтами включено, может потребоваться утверждение сайтов до их создания. Ожидающие запросы сайтов можно просмотреть на странице "Сайты", используя "Запросы".

 **Примечание.**

Версия Oracle Content Management Starter не поддерживает управление. Для получения полного набора функций и неограниченного количества сайтов выполните обновление до версии Oracle Content Management Premium.

См. [Общие сведения об управлении сайтами](#).

Чтобы просмотреть ожидающие запросы сайта, на странице Сайты в меню фильтра выберите **Запросы**. Если вы являетесь *администратором сайта*, все запросы сайта будут отображены в системе, включая запросы с требованием подтверждений от других пользователей. Если вы являетесь *утверждающим*, отобразятся все запросы сайта, утверждающим которого вы являетесь. *Все пользователи* видят отправленные ими запросы сайта.

В списке запросов отображается эскиз сайта, имя сайта, имя запрашивающего, дата и время запроса, описание сайта, статус запроса и значок, показывающий, требуется ли на сайте выполнять вход или нет.

В зависимости от роли можно выполнять следующие действия:

- Для просмотра дополнительных сведений о запросе, таких как минимальные требования безопасности, используемый шаблон, необязательное обоснование для сайта и эскизы, нажмите на имя сайта или выберите запрос, а затем нажмите **Просмотреть**.
- Если вы являетесь администратором сайта или утверждающим, запрос можно утвердить, выбрав его и нажав **Утвердить**. Сайт будет автоматически создан после его утверждения.
- Если вы являетесь администратором сайта или утверждающим, вы можете отклонить запрос, просмотрев сведения и нажав **Отклонить**. Введите причину отклонения запроса и нажмите **Отклонить**. Можно также отклонить запрос, выбрав его в списке запросов и нажав **Отклонить**, но при этом вы не сможете указать причину отклонения запроса.
- Если ваш запрос не выполнен или был отклонен, вы можете просмотреть сведения, отредактировать запрос, а затем **Повторно отправить** его.
- Если вы являетесь создателем сайта, вы можете удалить свой запрос, выбрав его и нажав **Удалить**.

 **Примечание.**

Если источник запроса сайта удален, администратор сайта может удалить запрос сайта.

После создания сайта вы можете предоставить общий доступ к нему, изменить параметры безопасности, добавить и редактировать контент, опубликовать сайт или перевести его в онлайн-режим. См. [Управление сайтами и настройки сайтов](#) и [Редактирование сайтов](#).

Изменение сведений о запросе сайта

На вкладке сведений о запросе сайта отображаются имя сайта, описание, используемый шаблон, обоснование, владелец (лицо, создающее запрос) и изображения сайта.

Вы можете обновить сведения о запросе сайта, если это вы создали запрос (вы — владелец запроса).

Чтобы просмотреть или изменить сведения о запросе сайта, выполните указанные ниже действия.

1. Нажмите **Сайты**, а затем в меню "Сайты" нажмите **Запросы**.
2. Нажмите запрос, который необходимо просмотреть или отредактировать.
3. Утверждающее лицо может **утвердить** или **отклонить** запрос. Инициатор запроса может изменить данную информацию необходимым образом, а затем **повторно отправить** запрос.

Просмотр политик запросов сайта

На вкладке политик запросов сайтов отображаются тип сайта (стандартный или корпоративный), те, у кого есть доступ к опубликованному сайту, а также все утверждения, необходимые для создания сайта. Для корпоративных сайтов также отображаются репозиторий активов, политика локализации, требуемый язык и язык, по умолчанию используемый для сайта.

Чтобы просмотреть политики запросов сайтов, выполните указанные ниже действия.

1. Нажмите **Сайты**, а затем в меню "Сайты" нажмите **Запросы**.
2. Нажмите запрос, который необходимо просмотреть или отредактировать, затем перейдите на вкладку **Политики**.
3. Утверждающее лицо может **утвердить** или **отклонить** запрос. Инициатор запроса может изменить репозиторий активов необходимым образом, а затем **повторно отправить** запрос.

5

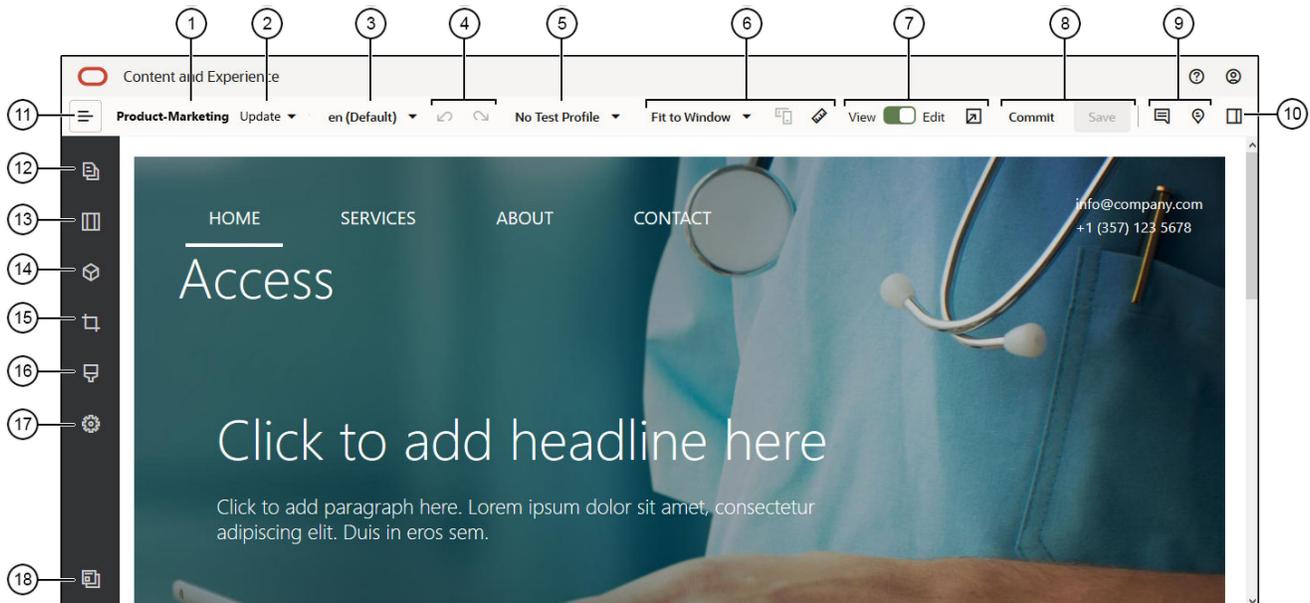
Редактирование сайтов

В этом разделе содержится краткий обзор редактора и его функций.

- [Вводные сведения о странице "Site Builder"](#)
- [Общие сведения об обновлениях сайтов](#)
- [Использование обновления](#)
- [Редактирование рекомендаций и советов](#)
- [Использование стилей и форматирование](#)
- [Добавление пользовательских свойств сайта](#)
- [Настройки сайта](#)
- [Работа с таблицами](#)
- [Загрузка файлов сайта](#)

Вводные сведения о странице "Site Builder"

При редактировании существующего обновления или создании обновления для сайта, обновление открывается в Site Builder. Выделите несколько минут, чтобы познакомиться с макетом данной страницы и доступными инструментами.



Следует отметить несколько аспектов.

Выноска	Описание
1	Отображается имя вашего сайта.
2	Проверьте обновление , чтобы узнать имя обновления, с которым вы работаете в настоящее время. Если имеется несколько обновлений, можно переключиться на другое обновление.
3	Проверьте язык , чтобы узнать, с какой версией вы работаете в данный момент.
4	Используйте значок Отменить  для отмены последнего изменения, внесенного с помощью редактора. Используйте значок Вернуть  , чтобы повторно применить последнее изменение, отмененное с помощью операции отмены. Операцию "Отменить" можно выполнить несколько раз для отмены целой серии изменений в текущем обновлении, включая изменения в содержимом, стиле и порядке страницы.
5	Тестовые профили дают возможность создавать профили с различными атрибутами аудитории и использовать эти профили с целью представления различных посетителей сайта для тестирования рекомендаций. Тестовые профили настраиваются, когда администратор сайта создает рекомендацию.
6	Используйте параметры предварительного просмотра , чтобы увидеть, как содержимое будет выглядеть на разных устройствах. Можно перейти к любой странице и посмотреть, как будет выглядеть базовый сайт после применения текущего обновления. Нажмите По размеру окна и выберите измерение, чтобы страница отображалась так, как на экране устройства с выбранным размером. Имеется несколько predefined размеров, а также можно создать собственные. Нажмите  , чтобы просмотреть метки. Нажимайте на деления линейки, чтобы посмотреть, как сайт выглядит на экранах разных размеров. Можно также нажать  , чтобы посмотреть, как сайт будет выглядеть на мобильном устройстве при повороте экрана. При использовании тем с адаптивным дизайном содержимое страниц автоматически упорядочивается для оптимального использования на экранах выбранного размера.

 **Примечание.**

Ссылки на страницы сайта не работают в режиме предварительного просмотра.

Выноска	Описание
7	Используйте переключатель в верхней части страницы для переключения между режимами просмотра и редактирования. Нажмите  для предварительного просмотра сайта. В этом случае страница отображается так, как ее видят посетители вашего сайта, без ячеек, границ компонентов и других визуальных вспомогательных элементов, используемых во время редактирования. Для перемещения по сайту можно использовать ссылки на странице, включая ссылки в меню навигации, ссылки в тексте и т. д. Ссылки на другие сайты открываются в новом окне из соображений безопасности. Ссылки на страницы текущего сайта открываются в том же окне или в новом окне в зависимости от указанного объекта ссылки.
8	Для обработки изменений используйте доступные параметры . Нажмите Фиксировать , чтобы объединить свои изменения с основным сайтом , или нажмите Сохранить , чтобы сохранить свои изменения в обновлении .
9	Используйте управление аннотациями , чтобы добавлять или просматривать существующие аннотации. <ul style="list-style-type: none"> Чтобы добавить аннотацию, нажмите , выберите компонент или нажмите в том месте, где требуется добавить аннотацию, затем введите комментарий в текстовом поле и нажмите Поместить. См. раздел Добавление аннотаций в документе <i>Collaborating on Documents with Oracle Content Management</i>. Для просмотра аннотации нажмите . При нажатии аннотации заметка с аннотацией становится активной в диалоге.
10	Нажмите  , чтобы отобразить (или скрыть) панель диалогов , которая позволяет создать обсуждение этого сайта или принять участие в обсуждении этого сайта. См. раздел Работа с диалогами в документе <i>Collaborating on Documents with Oracle Content Management</i> .
11	Нажмите  , чтобы отобразить (или скрыть) инструменты на боковой панели.

 **Примечание.**

Чтобы использовать инструменты на боковой панели, переключатель  необходимо перевести в положение **Редактировать**.

Нажмите на значок (12–18) на боковой панели, чтобы перейти к управлению страницами, добавить компоненты, изменить тему и т. д.

Выноска	Щелкните...	Чтобы...
12		Редактировать и добавлять страницы, управлять структурами вложенных страниц, реорганизовывать страницы и изменять настройки страниц.

Выноска	Щелкните...	Чтобы...
13		Добавить макеты разделов, например горизонтальные или вертикальные.
14		Добавить активы на сайт. Можно выбрать любые активы в репозитории, указанном при создании сайта.
15		Вставить компоненты на сайт. Можно выбрать различные типы компонентов, например тематические, пользовательские, заданные предварительно, или просмотреть все доступные компоненты.
16		Выбрать другую тему для сайта или изменить фоновое изображение страницы или области.
17		Установить настройки сайта, например ключевые слова для поисковой оптимизации (SEO) и перенаправлений сайта.
18		Вернуться к странице "Сайты" для управления своими сайтами.

Общие сведения об обновлениях сайтов

Обновление — это именованная коллекция изменений для текущего базового сайта. Изменения остаются в обновлении, пока вы не примените их и не обновите базовый сайт окончательно.

Всякий раз при просмотре или редактировании сайта в редакторе используется обновление. Внесенные в редактор изменения становятся частью этого обновления. Можно работать с одним или несколькими обновлениями и продолжать добавлять изменения в обновления. Обновления предоставляют гибкие возможности управления правками для сайта. Например, можно выбрать несколько человек, которые работают над собственными обновлениями для разных частей сайта. Вы можете просматривать и изменять отдельные обновления, а также применять обновления к базовому сайту в удобное для вас время. Вам по-прежнему необходимо публиковать сайт, чтобы обновления стали доступны в онлайн-режиме.

Обновления можно упорядочить несколькими способами:

- уникальные изменения страниц в именованном обновлении;
- изменения, внесенные конкретным пользователем, в обновлении с именем пользователя;
- изменения, внесенные в течение заданного дня или за конкретный контрольный период проекта.

 **Примечание.**

В обновлении представлены изменения базового сайта. В любой момент времени можно просматривать только одно обновление, но в редакторе легко переключаться между доступными обновлениями. Если для указанной страницы существует несколько обновлений, вы можете не знать, что в одну и ту же область содержимого внесено несколько изменений. Чтобы предотвратить конфликт изменений на странице, отдельные обновления следует ориентировать на конкретные страницы или области сайта.

При применении изменений из текущего обновления изменения вносятся на основном сайте, а само обновление удаляется. Вам необходимо опубликовать сайт (переведя его в онлайн-режим или повторно опубликовав), чтобы эти изменения были видны всем пользователям, имеющим доступ к сайту.

Далее описывается порядок редактирования сайта.

1. Выберите сайт в списке и выберите **Открыть** в контекстном меню или  на панели действий.
2. Редактор открывается в режиме предварительного просмотра. Чтобы внести изменения или воспользоваться средствами навигации на боковой панели, убедитесь, что переключатель "Редактировать"  переведен в положение **Редактировать**.
3. Если это первое обновление сайта, введите имя и необязательное описание, затем нажмите кнопку **ОК**. В имени можно использовать буквы, цифры, символы подчеркивания (_) и дефисы (-). Введенный пробел автоматически заменяется дефисом. Если для сайта уже есть обновления, выберите обновление из списка и щелкните .
4. Чтобы отредактировать определенную страницу, найдите ее в дереве сайта на левой боковой панели, выполняя поиск страницы, или с помощью собственной системы навигации сайта. Чтобы отобразить дерево сайта, нажмите  слева, а затем .
5. Добавьте и настройте содержимое страницы. Выберите **По размеру окна** и, изменяя параметры макета, просмотрите, как страница будет отображаться на различных устройствах и на экранах разных размеров. Имеется несколько предопределенных размеров, а также можно создать собственные. Нажмите , чтобы просмотреть метки. Нажимайте на деления линейки, чтобы посмотреть, как сайт выглядит на экранах разных размеров. Можно также нажать , чтобы посмотреть, как сайт будет выглядеть на мобильном устройстве при повороте экрана.
6. После завершения редактирования сайта сохраните изменения одним из описанных ниже способов.
 - Нажмите кнопку **Сохранить**, чтобы сохранить изменения в текущем обновлении. Вы сможете продолжить работу с текущим обновлением или вернуться к нему позднее.
 - Нажмите **Фиксировать**, чтобы применить изменения к основному сайту.

Чтобы пользователи веб-сайта увидели эти изменения, по-прежнему необходимо опубликовать сайт. См. разделы "[Перевод сайта в онлайн-режим или автономный режим](#)" или "[Публикация изменений сайта](#)".

При применении изменений из текущего обновления изменения вносятся на основном сайте, а само обновление удаляется. Поскольку при каждом просмотре или изменении сайта в редакторе необходимо использовать активное обновление, происходит возврат к списку сайтов, где можно создать очередное обновление.

При наличии обновления процесс носит линейный характер.

- Существует исходный (основной) сайт. Назовем его "Версия 1".
- Создается обновление. Во время объединения данных обновления (с помощью команды "Фиксировать") обновление безвозвратно записывается на существующий сайт. Теперь есть новая версия базового сайта (версия 2).
- Если создать и объединить еще одно обновление, оно безвозвратно записывается в существующий сайт. Теперь есть новая версия базового сайта (версия 3).

Во время работы с обновлениями следует помнить о том, что:

текущий базовый сайт + объединение с обновлением = новая версия базового сайта

Использование обновления

При каждом редактировании сайта в редакторе используется обновление. Внесенные в редакторе изменения становятся частью этого обновления.

Чтобы создать и использовать обновление:

1. На странице "Сайты" выберите сайт и выберите **Открыть** в контекстном меню или нажмите  на панели действий.
2. Установите переключатель "Редактировать"  на Редактирование.
3. Если это первое обновление сайта, введите имя и необязательное описание, затем нажмите кнопку **ОК**. Можно употреблять буквы, цифры, символы подчеркивания (_) и дефисы (-). Введенный пробел автоматически заменяется дефисом. Чтобы использовать существующее обновление, нажмите на него в списке.
4. В редакторе имена сайта и текущего обновления отображаются на верхней панели. Вы можете переключаться между обновлениями, выбирая нужное вам в списке обновлений.
5. Добавьте и настройте содержимое страницы. Используйте различные параметры размера вывода, чтобы увидеть, как страница будет выглядеть на разных устройствах.
6. По завершении нажмите **Сохранить**. Если изменения готовы к объединению с основным сайтом, можно применить обновление. Чтобы пользователи веб-сайта увидели эти изменения, по-прежнему необходимо опубликовать сайт. См. [Публикация изменений сайта](#).

Редактирование рекомендаций и советов

Ниже приведены некоторые советы, которые помогут вам начать работу с редактором.

- [Скрытие или отображение боковой панели](#)
- [Первое знакомство](#)
- [Управление страницами](#)
- [Редактирование с перетаскиванием](#)
- [Компоненты](#)
- [Настройка размера компонентов и пространства вокруг них](#)
- [Редактируемое содержимое макета](#)
- [Стили и форматирование](#)
- [Отмена изменений](#)

Скрытие или отображение боковой панели

Нажмите , чтобы отобразить параметры управления страницами и содержимым и их редактирования.

Переведите переключатель  в положение **Редактировать**, чтобы получить доступ к боковой панели. Нажмите , чтобы скрыть боковую панель и увеличить область предварительного просмотра страницы.

Первое знакомство

Для выбора другой страницы с помощью дерева сайта нажмите  на боковой панели. Для поиска страницы используйте поле поиска в верхней части списка страниц. Также можно использовать средства навигации по сайту и ссылки на самих страницах.

При переключении на другую страницу все изменения страницы сохраняются. Чтобы сохранить изменения в текущем обновлении, также можно нажать кнопку **Сохранить**.

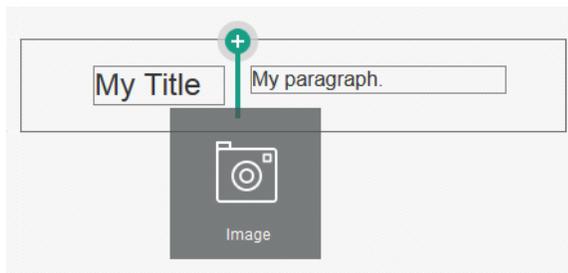
Управление страницами

- Чтобы добавить страницу, выберите уровень или ветвь, в которую необходимо добавить страницу, затем нажмите **Добавить страницу** или, чтобы добавить дочернюю страницу, нажмите , затем нажмите . Внизу дерева сайта добавляется запись "Новая страница" и предлагается ввести имя страницы и указать другие настройки.
- Для удаления страницы выберите ее и нажмите .
- Чтобы переместить страницу, выберите ее и перетащите в новое положение. Также можно вырезать и вставить страницу в новое место. Чтобы вырезать страницу, нажмите , затем . Чтобы вставить страницу, выберите ветвь, в которую требуется ее вставить, нажмите , затем .

- Чтобы скопировать страницу, выберите ее, нажмите , затем . Чтобы вставить страницу, выберите ветвь, в которую требуется ее вставить, нажмите , затем .
- Чтобы изменить настройки страницы, такие как метаданные, верхний и нижний колонтитулы и другие параметры, нажмите . См. [Изменение настроек страницы](#).

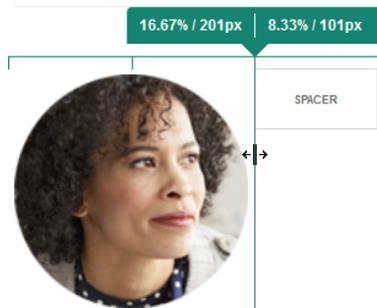
Редактирование с перетаскиванием

Чтобы добавить компонент с боковой панели или переместить его в другое место на странице, выберите и перетащите элемент на нужную позицию. При перетаскивании элемента на странице отображаются границы доступных ячеек и всех имеющихся элементов. Панель размещения  указывает место размещения нового содержимого (выше, ниже, слева, справа).



В одной ячейке может быть несколько элементов. Элементы можно перемещать по странице, просто перетаскивая в новое место. Также можно настроить относительную ширину двух компонентов в ячейке, нажав и перетащив границу между двумя компонентами. Компонент привязывается к следующей линии сетки, указанной в линейке, изображенной над компонентами.

Размер каждого компонента отображается в пикселях и в процентах от доступного пространства в области. Чтобы настроить значения ширины, отличные от указанных в сетке, нажмите и удерживайте клавишу **Ctrl** при перетаскивании границы компонента.



Компоненты

После размещения компонента на странице можно изменить параметры выравнивания, интервалов и другие свойства. Выберите элемент, нажмите на

значок  и выберите **Настройки**. Если выбрать имя компонента вместо значка меню, отображается значок меню для ячейки и группы компонентов (при условии, что текущий компонент входит в группу). Если выбрать другую вкладку, отображается значок меню, и вы можете изменить свойства элемента:



Разработчики тем могут выбирать компоненты, которые будут включены в тему, поэтому компоненты, доступные на одном сайте, не всегда доступны на других сайтах. Разработчики тем также могут указать, какие компоненты будут разрешены в конкретной области на конкретном макете страницы. Если компонент не разрешен в определенной области, строка размещения меняет цвет и обозначение (знак минуса)

 и отображается следующее сообщение:



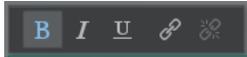
Настройка размера компонентов и пространства вокруг них

Можно выполнять настройку компонентов, таких как галереи или изображения, в различных единицах CSS, а также настраивать пространство вокруг них. Пример:

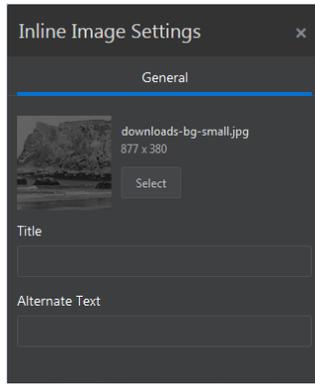
- Значение по умолчанию: px (пиксели). Если указано только числовое значение, то используются пиксели.
- % (процент): указывает процентное отношение элемента к размеру его родительского элемента HTML. Пример: 25%
- em: указывает размер компонента в em-пространстве. Пример: 20em
- vw: указывает размер в процентном отношении к ширине области просмотра и зависит от размера области просмотра. Пример: 10vw

Редактируемое содержимое макета

Разработчики тем могут добавить на макеты страниц "встроенные" элементы содержимого (например, уведомления об авторском праве), которые невозможно изменить в редакторе. Разработчики тем также могут сделать редактируемыми элементы простого текста и изображений, включая цифровые активы. Это позволит пользователю с ролью "Соавтор" изменять текст и изображения, но не расположение и другие атрибуты макета. Для редактируемых текстовых элементов доступно меню, с помощью которого можно применить к тексту полужирный шрифт, курсив или нижнее подчеркивание, а также изменить или удалить ссылку:



Для редактируемых графических элементов доступен значок настроек  и панель, с помощью которой можно выбрать изображение, ввести заголовок и добавить текст в изображение.



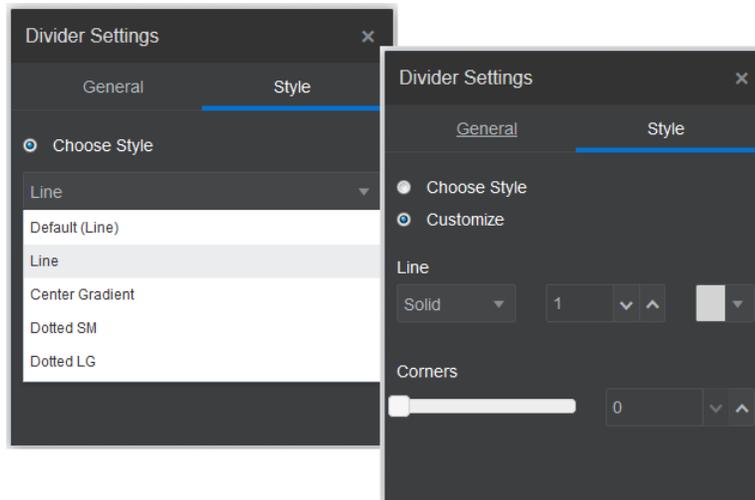
Рамка вокруг текста или изображения изменяется в зависимости от длины текста или размера изображения.

Примечание.

Изменения применяются только к текущей странице. Исходное содержимое сохраняется с макетом в теме и используется по умолчанию при первом применении макета к странице.

Стили и форматирование

Для большинства компонентов тема определяет один или несколько базовых стилей, которые указывают аспекты внешнего вида компонента. Можно с легкостью менять стили и переопределять текущий стиль. Чтобы выбрать один из доступных стилей, откройте панель "Настройки" для компонента, перейдите на вкладку **Стиль**, нажмите **Выбрать стиль** и выберите стиль в меню. Чтобы указать собственные значения свойств, заданных в определении стиля, щелкните **Настроить** и укажите параметры форматирования.



Отмена изменений

Используйте значок  для отмены последнего изменения, внесенного с помощью редактора. Операцию отмены можно выполнить несколько раз, чтобы отменить серию изменений.

Используйте значок , чтобы повторно применить последнее изменение, отмененное с помощью операции отмены. Операцию повтора можно выполнить несколько раз, если операция отмены использовалась несколько раз подряд.

В текущем обновлении можно отменить изменения в содержимом, стиле или организации страницы. Некоторые операции не входят в цепочку операций отмены.

- При переходе на другое обновление цепочка операций отмены сбрасывается и отменить изменения, внесенные в ранее используемом обновлении, невозможно. В пределах обновления изменения можно отменить даже после сохранения.
- Изменения режима просмотра в редакторе (переключение страниц, изменение размера заданной страницы и т. д.) отменяются вручную.
- При редактировании текстовых компонентов (заголовков, абзацев и т. д.) текстовый редактор имеет собственную цепочку операций отмены. После выхода из текстового редактора отменить такие изменения невозможно.

Использование стилей и форматирование

Для большинства компонентов тема определяет один или несколько базовых стилей, которые указывают аспекты внешнего вида компонента. Вы можете легко переключиться с одного стиля на другой или переопределить стиль с помощью выбранных средств.

1. Перейдите к странице, которую хотите изменить, и убедитесь, что переключатель  переведен в положение **Редактировать**.
2. Чтобы задать базовый стиль для компонента, нажмите на значок меню компонента  и выберите **Настройки**. Перейдите на вкладку **Стиль**.

- Чтобы использовать стиль, предоставленный вместе с темой сайта, нажмите **Выбрать стиль** и в меню выберите нужный стиль. Стили определяются для отдельных компонентов, поэтому в списке могут быть представлены разные стили. Например, стиль для изображения отличается от стиля для абзаца.
 - Чтобы указать собственные параметры базового форматирования, щелкните **Настроить** и укажите параметры форматирования.
3. Чтобы скопировать и вставить базовый стиль в один или несколько похожих компонентов, нажмите на значок меню компонента  и выберите **Копировать стиль компонента**. Щелкните значок меню подобного компонента  и выберите пункт **Вставить стиль компонента**.
 4. Чтобы отформатировать текст в компоненте-заголовке или компоненте-абзаце, нажмите на соответствующий компонент. Появится панель инструментов с функциями форматирования. Выделите текст, который требуется отформатировать, а затем выберите любой из вариантов, такой как шрифт, цвет или выравнивание. Вносимые изменения применяются немедленно. Чтобы удалить форматирование, выделите текст и нажмите . Изменения форматирования накладываются поверх базового стиля. В случае изменения базового стиля переопределения сохраняются.

Настройки сайта

Можно выполнить настройку сайта, чтобы добавить значки сайта, файл контроллера, карту сайта, файлы для поискового робота, вспомогательные файлы и указать поставщика карт.

Эти настройки хранятся в обновлении до момента его публикации. После публикации файлы хранятся в корневой папке темы и применяются ко всем сайтам, где используется эта тема.

Изменение значков и поведения ссылок для всего сайта:

1. Откройте сайт для редактирования.
2. Нажмите  на боковой панели, а затем нажмите  **Сайт**.
3. В разделе **Значки избранного** выберите изображение, которое будет использоваться при сворачивании сайта в браузере или на другой платформе, например на мобильном устройстве. Этот значок должен храниться в виде доступного для вас цифрового актива. Нажмите **Выбрать файл для загрузки**, перейдите к значку и выберите его, затем нажмите **ОК**. Чтобы значок избранного работал во всех браузерах, необходимо его настроить.
 - **Для Chrome и Safari** в файле контроллера должна быть ссылка на значок избранного, включая префикс сайта, как показано в следующем примере:

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

Для Internet Explorer 11 и Firefox необходимо добавить похожую запись в шаблоны страниц. Однако шаблон страницы является частью темы и

может использоваться на нескольких сайтах. Поэтому в данном случае вместо фиксированного префикса сайта следует использовать маркер. См. следующий пример:

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

При открытии страницы маркер заменяется префиксом сайта.

Обратите внимание, что Internet Explorer и Firefox загружают значок избранного из контроллера и еще раз из шаблона страницы. Поэтому значок мигает, если контроллер и шаблон страницы ссылаются на разные значки. В Chrome и Safari загружается значок, на который ссылается контроллер.

4. В разделе **Файл контроллера** можно добавить файл, изменяющий метод обработки запросов на ссылки в браузере. В дополнение к ссылкам на значки избранного можно также добавить теги OpenGraph или метаданных для проверки сайта веб-мастером или предоставления доступа к сайту в социальных сетях. Выгрузите файл контроллера по умолчанию и отредактируйте его, или загрузите собственный файл. Этот файл должен храниться в виде доступного для вас цифрового актива. Нажмите **Выбрать файл для загрузки**, перейдите к файлу и выберите его, затем нажмите **ОК**.

Например, если сайт содержит кнопку Facebook "Поделиться", можно добавить метаданные, которые будут использоваться для отображения сведений о вашем сайте в Facebook, как в следующем примере:

```
<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" />
```

Можно добавить тег проверки веб-мастером Google, как в следующем примере:

```
<meta name="google-site-verification"  
content="GCVURS9d2fP6jev5upt0Yt1AIp7lC9D__ALqS8pg" />
```

5. В разделе **Карта сайта и файлы поискового для робота** можно загрузить пользовательскую карту сайта и файлы робота. *Карта сайта* – это файл XML, в котором перечислены URL-адреса веб-сайта и приведены сведения о каждом из этих URL-адресов (например, дата последнего обновления). *Файл робота* – это текстовый файл с инструкциями для поисковых систем, обеспечивающими правильную индексацию страниц. Этот файл должен храниться в виде доступного для вас цифрового актива. Нажмите **Выбрать файл для загрузки**, перейдите к файлу и выберите его, затем нажмите **ОК**.
6. В разделе **Вспомогательные файлы** можно загрузить другие файлы, например те, которые нужны для проверки принадлежности сайта. Этот файл должен храниться в виде доступного для вас цифрового актива. Нажмите **Выбрать файл для загрузки**, перейдите к файлу и выберите его, затем нажмите **ОК**.
7. Вы можете выбрать поставщика для компонента-карты и используемых ссылок. Выберите **Oracle Maps** или **Google Карты**.
8. При публикации обновления также будут опубликованы и применены внесенные изменения.

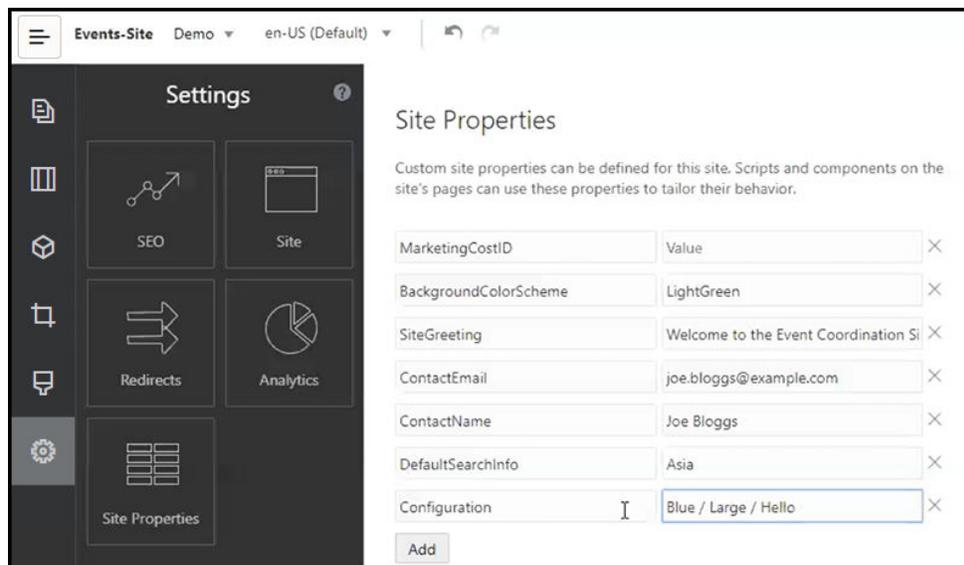
Добавление пользовательских свойств сайта

На сайты можно добавлять пользовательские свойства в виде пар "имя/значение". Эти свойства сохраняются на сайте и доступны для сценариев и компонентов на страницах сайта. Они позволяют параметризовать или настраивать сайт без изменения базовых сценариев и кода компонента.

Например, пользовательские свойства можно использовать для изменения цвета фона страницы, уточнения результатов поиска, заполнения списков и управления любыми другими переменными, относящимися к сайту.

Пользовательские свойства сайта добавляются с использованием панели настроек при редактировании сайта.

1. Откройте сайт для редактирования.
2. Нажмите  на боковой панели и выберите  **Свойства сайта**.
3. Нажмите **Добавить**.
4. Введите имя и значение пользовательского свойства сайта. Можно добавить до 50 пользовательских свойств сайта. В поле имени можно ввести не более 200 символов, а в поле значения — не более 2000 символов.



5. Чтобы удалить пару "имя/значение", нажмите символ **X** рядом с ней.
6. По завершении добавления или удаления пользовательских свойств сайта нажмите кнопку **Фиксировать**.

Примечание.

Изменения не объединяются с существующими пользовательскими свойствами сайта. При внесении изменений в пользовательские свойства сайта все существующие пользовательские свойства сайта на базовом сайте перезаписываются.

После определения пользовательские свойства сайта можно использовать при создании сценариев для сайта и всех компонентов сайта, например, в нижнем колонтитуле или в поле "Дополнительная строка запроса" в компонентах, поддерживающих расширение SCSSMacro, таких как список контента, или посредством маркеров в компонентах заголовка и абзаца.

Эти сценарии работают с SCSSRenderAPI во время выполнения и проектирования. Кроме того, значения пользовательских свойств сайта доступны для кода макета и компонента в компиляторе шаблонов через интерфейс SCSSCompileAPI. Этот API имеет новую функцию, аналогичную SCSSRenderAPI, функцию getCustomSiteProperty, которая позволяет коду макета и компонента считывать значение пользовательского свойства раздела.

Например, предположим, что необходимо определить пользовательский макет раздела, который использует SCSSRenderAPI для вызова и извлечения пользовательских свойств сайта для заголовка, имени и адреса электронной почты контактного лица. Представленный ниже сценарий позволяет создать строку HTML с использованием пользовательских свойств сайта, которые добавляются к DOM.

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

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

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

Пользовательские свойства сохраняются при создании шаблона на основе сайта с добавленными пользовательскими свойствами сайта и при создании сайта на основе шаблона с пользовательскими свойствами сайта.

Работа с таблицами

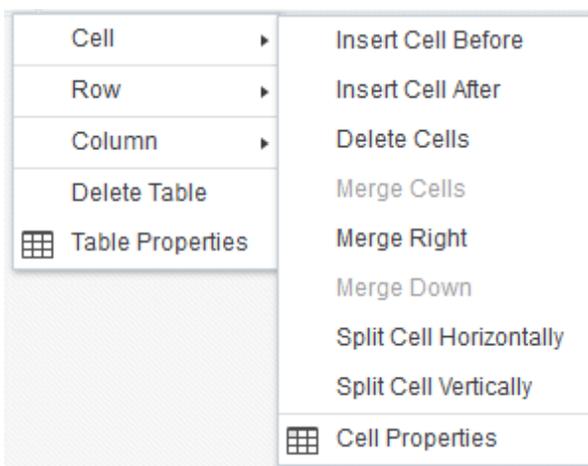
В компонент-абзац можно добавить таблицы, создав их или вставив из существующего источника HTML.

1. Перейдите к странице, которую необходимо изменить, и убедитесь, что переключатель  находится в положении **Редактировать**.
2. Добавьте компонент на страницу или нажмите на существующий компонент-абзац.
3. Для добавления таблицы с другой страницы HTML просто нажмите на нужную таблицу и перетащите ее, а затем скопируйте и вставьте в компонент-абзац. Чтобы создать новую таблицу в месте курсора, нажмите на значок . Выберите число строк и столбцов, высоту, ширину и другие параметры форматирования. Можно добавить дополнительную **подпись** по центру над таблицей и дополнительную **сводку**, предоставляющую контекст для специальных технологий доступности, таких как программы для чтения с экрана.
4. После внесения изменений нажмите **ОК**, чтобы закрыть окно. Если позднее потребуется изменить настройки таблицы, нажмите на нее правой кнопкой и выберите **Свойства таблицы**. Чтобы удалить таблицу, нажмите на нее правой кнопкой и выберите **Удалить таблицу**.

 **Примечание.**

При разработке для дисплеев различных размеров, таких как мобильные экраны, можно использовать код из файла design.css StarterTheme, чтобы [создать адаптивную таблицу](#), что позволяет помещать в стек данные строк при отображении на мобильных устройствах.

5. Чтобы добавить, удалить или изменить отдельные строки, столбцы или ячейки, нажмите на нужную строку, столбец или ячейку правой кнопкой и выберите действие в меню. Пример:

 **Примечание.**

Единовременно можно изменить свойства только одной ячейки.

6. Для создания сложных таблиц можно объединять и разделять ячейки. Пример:

		Split Cells	
Merged cells			

- Чтобы разделить ячейку на две, нажмите на ней правой кнопкой, выберите **Ячейка**, а затем **Разбить ячейки по горизонтали** или **Разбить ячейки по вертикали**.
 - Чтобы объединить две ячейки по горизонтали, нажмите правой кнопкой на левую ячейку, выберите **Ячейка**, а затем **Объединить справа**.
 - Чтобы объединить две ячейки по вертикали, нажмите правой кнопкой на верхнюю ячейку сверху, выберите **Ячейка**, а затем **Объединить вниз**.
7. Используйте общие параметры форматирования, чтобы изменить формат и выравнивание текста внутри ячеек.

Загрузка файлов сайта

С помощью интерфейса Oracle Content Management можно в любой момент загрузить изображения и документы и использовать их на сайте. Можно также загрузить файлы из Site Builder во время работы с фоновыми изображениями, а также с различными компонентами (изображения, галерея и документы).

При использовании файлов изображений и документов на сайтах можно использовать изображения, сохраненные на сайте или в другом расположении, к которому есть доступ. Можно также использовать изображения, к которым вам предоставлен доступ или которые вы загрузили из локального или сетевого расположения файлов.

Загрузка файлов

Чтобы загрузить один или несколько файлов из локального или сетевого каталога, выполните указанные ниже действия.

1. Нажмите **Документы** и перейдите туда, где требуется сохранить файл. Нажмите кнопку **Создать**, чтобы добавить в текущее местоположение новую папку.
2. Нажмите **Загрузить**.
3. Найдите и выберите один или несколько файлов, а затем нажмите кнопку **Открыть**.

Загрузка файлов из редактора

Во время работы с фоновыми изображениями и различными компонентами (документами или изображениями) можно загружать файлы напрямую, если такого файла нет в Oracle Content Management.

Например, чтобы загрузить в сеть один или несколько файлов из локальной или сетевой папки для использования с компонентом-галереей, выполните указанные ниже действия.

1. Чтобы добавить изображению в галерею, нажмите на значок меню , выберите **Настройки**, и нажмите **Изображения** на вкладке **Общие**.
2. Нажмите на значок **Добавить изображения**.
3. Перейдите в местоположение, где требуется сохранить файл, или нажмите кнопку **Создать**, чтобы добавить в текущее местоположение новую папку.
4. Нажмите **Загрузить**.
5. Найдите и выберите один или несколько файлов, а затем нажмите кнопку **Открыть**.
6. Выберите в репозитории одно или несколько изображений и нажмите кнопку **ОК**.

 **Примечание.**

В данном окне отображаются все доступные файлы. Выберите тип файла, соответствующий контексту. Например, выбирая файл изображения, необходимо выбрать допустимый формат изображения (GIF, JPG, JPEG, PNG или SVG). Чтобы добавить ссылку на файл, установите флажок **Использовать ссылку на исходный файл вместо копирования файла на сайт**. В противном случае копия файла сохраняется на самом сайте.

6

Использование шаблонов и тем для сайтов

Шаблон содержит необходимые элементы для создания сайта, в том числе кодовую платформу сайта, стандартный сайт с примерами страниц и содержимого, тему со стилями, ресурсы (например, изображения), а также пользовательские компоненты. Тема определяет общий внешний вид (общий стиль) сайта, в том числе цветовую гамму, размеры и типы шрифтов, а также фон страниц.

Шаблоны

- [Общие сведения о шаблонах](#)
- [Создание шаблона на основе сайта](#)
- [Изменение сведений о шаблоне](#)
- [Изменение политик шаблона](#)
- [Изменение аудитории или статуса шаблона](#)
- [Управление шаблонами](#)
- [Экспорт и импорт шаблонов](#)

Темы

- [Общие сведения о темах](#)
- [Управление темами](#)
- [Публикация тем](#)

Общие сведения о шаблонах

Шаблон содержит все необходимые элементы для создания сайта, в том числе кодовую платформу сайта, стандартный сайт с примерами страниц и содержимого, тему со стилями, ресурсы (например, изображения), а также пользовательские компоненты.

Шаблоны по умолчанию

В Oracle Content Management предоставляется ряд шаблонов, которые можно использовать для создания сайтов. Достаточно выбрать шаблон и указать имя сайта, а затем можно приступить к добавлению содержимого. Как правило, пакет шаблонов устанавливается администратором сервиса во время инициализации.

Шаблон	Описание
Пустой шаблон	Пустой шаблон — это одна страница с ячейками заголовка, тела и нижнего колонтитула, предоставляющая полную свободу использования собственного дизайна.

Шаблон	Описание
Демонстрация рекламных материалов	Шаблон "Демонстрация рекламных материалов" включает в себя выровненное по правому краю горизонтальное меню, простые макеты страниц, страницы верхнего уровня, на которых перечисляются связанные видеоактивы, и вложенные страницы сведений для каждого демонстрируемого видео. Этот шаблон предоставляет стандартный компонент меню и показывает текущего пользователя, выполнившего вход в систему.
Запуск нового продукта	Шаблон "Запуск нового продукта" включает выровненное по правому краю горизонтальное меню, а также макеты примеров, информации о функциональных возможностях, цен, сведений о компании и контактных сведений. На домашней странице размещается изображение баннера, которое можно поворачивать, и текст.
Обзор продуктов и услуг	Шаблон "Обзор продуктов и услуг" включает макеты примеров и информации о продукте, компании, политике конфиденциальности и т. д. На домашней странице размещается изображение баннера, которое можно поворачивать, и текст. Шаблон можно изменять удобным для вас образом.
Начальный	Шаблон "Начальный" используется для создания собственных готовых сайтов. Он предоставляет простую и полнофункциональную заготовку, которую можно изучать и дополнять компонентами и элементами взаимодействия. Он включает в себя среду кода сайта, сайт по умолчанию с примерами страниц и содержимого, тему со стилями, ресурсы (например, изображения) и пользовательский компонент с функциями действий, запускаемых по триггеру. Примеры страниц содержат сведения о создании шаблонов и ссылки на ресурсы, где можно получить более подробную информацию.
Обучение Презентация Демонстрация	Эти шаблоны представляют собой цветовые вариации друг друга. Это гибкие шаблоны, которые включают в себя группу с пользовательским компонентом меню JavaScript. Эти шаблоны содержат пользовательские группы компонентов с некоторым количеством стандартных компонентов. Кроме того, они отображают текущего пользователя, выполнившего вход в систему.
Доступ Знания Связи	Эти шаблоны представляют собой цветовые вариации друг друга. Это гибкие шаблоны, которые включают в себя группу с пользовательским компонентом меню JavaScript. Эти шаблоны содержат пользовательские группы компонентов с некоторым количеством стандартных компонентов.
Общий доступ	Этот шаблон представляет собой сайт с единственной длинной страницей. Это адаптивный шаблон с пользовательским меню JavaScript, предназначенным для перехода к расположениям на странице, а не к отдельным страницам.
Начальный шаблон JET	Oracle JET (JavaScript Extension Toolkit) - это модульный инструментарий с открытым исходным кодом, использующий коллекцию библиотек JavaScript с открытым исходным кодом. Для получения подробных сведений см. Oracle Jet .

Также можно создать шаблон на основе существующего сайта или экспортировать существующий шаблон, внести в него изменения в автономном режиме, а затем импортировать полученный шаблон как новый. Организация может создать шаблоны для своих пользователей.

Как применяются шаблоны

Ниже кратко описана процедура создания сайта на основе шаблона:

- Если управление сайтами включено, может потребоваться утверждение сайтов до их создания. См. [Общие сведения об управлении сайтами](#).
- Содержащийся в шаблоне сайт по умолчанию копируется в новый сайт и служит в качестве отправной точки для создания страниц.
- Все необходимые сопутствующие файлы копируются на новый сайт.
- Если тема шаблона отсутствует в папке "Темы", она копируется в эту папку. Сайт ссылается на тему, сохраненную в папке "Темы". Если тема уже существует, в новом сайте создается ссылка на эту тему.
- Если шаблон содержит пользовательские компоненты, отсутствующие в папке "Компоненты", они копируются в эту папку. Сайт ссылается на компоненты, сохраненные в папке "Компоненты". Если какой-либо из компонентов уже существует, в новом сайте создается ссылка на этот компонент.

Создание пользовательских шаблонов

Шаблоны, предоставляемые веб-разработчикам, содержат все необходимые компоненты для создания сайтов, включая сайт, макет, навигацию, примеры содержимого и т. д. В структуру сайта можно добавить компоненты и способы взаимодействия, позволяющие адаптировать готовый сайт к требованиям организации.

Шаблон представляет собой структуру папок, с которыми можно работать так же, как и с любыми другими папками. На некоторые элементы шаблона (например, темы и пользовательские компоненты) представлена ссылка на их связанные расположения в Oracle Content Management. Например, шаблон ссылается на связанную с ним тему в списке доступных тем практически так же, как сайт.

При создании шаблона на основе существующего сайта копия сайта в новом шаблоне используется в качестве сайта по умолчанию. Шаблон содержит ссылки на используемую тему сайта и дополнительные компоненты, размещенные на страницах сайта. Тема и пользовательские компоненты не копируются в шаблон. Шаблон ссылается на них так же, как и сайт.

Примечание.

Шаблон, созданный на основе сайта, отражает состояние этого сайта на момент создания шаблона. Изменения, внесенные на сайте после создания шаблона, не влияют на сайт, сохраненный в этом шаблоне.

Можно создавать шаблоны содержимого, используемые для обмена моделями содержимого. Модели содержимого включают в себя макеты содержимого, элементы содержимого и необходимые цифровые активы (например, образцы содержимого). Шаблоны содержимого следует создавать на основе сайтов с опубликованными элементами содержимого и цифровыми активами. Обратите внимание, что при импорте шаблона сайта из пакета с содержимым не создаются типы содержимого. Они создаются, когда сайт создается на основе шаблона содержимого. Следовательно, типы содержимого не принадлежат пользователю, который импортирует шаблон сайта. Они принадлежат пользователю, который создает первый сайт на основе этого шаблона. При необходимости этот пользователь может поделиться шаблоном с

другими. Эта функция доступна не во всех типах подписки на Content Management Cloud. Ее доступность также зависит от начальной даты предоставления сервиса.

При экспорте шаблона все его элементы, включая копию темы и компоненты, объединяются в пакет, доступный для выгрузки и работы в автономном режиме.

При импорте шаблона, измененного в автономном режиме, действует следующее правило: если имя шаблона, имя темы, имена или идентификаторы пользовательских компонентов уже существуют, отображается запрос на устранение конфликтов. В этом случае предлагается создать новый шаблон, тему или пользовательский компонент либо заменить существующий шаблон, тему или компонент импортируемой версией. См. [Разработка шаблонов](#).

Шаблон также можно импортировать в определенный репозиторий. В этом случае предлагается обновить или дублировать существующие активы. См. [Импорт шаблонов в определенный репозиторий](#).

Общий доступ к шаблонам

Если шаблон создается путем импорта, копирования или на основе сайта, никто не сможет использовать его, пока вы явно не предоставите общий доступ к шаблону.

Примечание.

Это также применимо к шаблонам, которые предоставляются с Oracle Content Management и устанавливаются администратором. Если вы не видите шаблонов, обратитесь к администратору. Возможно, вам не предоставлен доступ к этим шаблонам.

При первом предоставлении пользователю доступа к шаблону становятся доступными связанные с ним тема и пользовательские компоненты. Пользователю также назначается роль "Выгрузка", чтобы он мог получить эти компоненты для создания сайта на основе шаблона. Последующие изменения роли пользователя не влияют на сведения об общем доступе для связанных тем и пользовательских компонентов.

Если управление сайтами включено, доступность шаблонов устанавливается с использованием политик шаблонов. См. [Общие сведения об управлении сайтами](#) и [Изменение политик шаблона](#).

Создание шаблона на основе сайта

Если вы хотите использовать сайт в качестве отправной точки для других сайтов, можно создать шаблон на основе этого сайта.

Примечание.

Создавать шаблоны на основе сайтов могут только пользователи с ролью "Выгрузка", "Соавтор" и "Управление". Кроме того, администратор должен включить соответствующий параметр в меню **Создать**. Если вы не видите меню **Создать** на странице шаблонов, обратитесь к администратору.

Далее описано, как создать шаблон на основе сайта. Также можно импортировать пакет шаблона, созданный или измененный в автономном режиме. См. [Экспорт и импорт шаблонов](#).

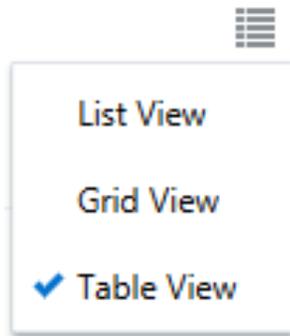
Примечание.

При создании шаблона на основе сайта, который использует контент из нескольких репозиториях, необходимо это сделать с помощью инструментария ОСЕ. См. [Разработка с помощью инструментария ОСЕ](#) и [Использование утилиты командной строки СЕС](#).

Чтобы создать шаблон на основе сайта с помощью веб-интерфейса Oracle Content Management, выполните указанные ниже действия.

1. Нажмите **Сайты** и выберите сайт, который требуется использовать.
2. Выберите **Создать шаблон** в контекстном меню или нажмите  на панели действий.
3. Введите имя шаблона. Можно употреблять буквы, цифры, символы подчеркивания (_) и дефисы (-). Введенный пробел автоматически заменяется дефисом.
Не используйте для шаблонов, тем, компонентов, сайтов и страниц сайтов следующие имена: authsite, content, pages, scstemplate_*, _comps, _components, _compsdelivery, _idcservice, _sitescloud, _sitesclouddelivery, _themes, _themesdelivery. Следующие имена можно использовать только для страниц сайтов (не для шаблонов, тем, компонентов и сайтов): documents, sites.
4. Введите описание шаблона (необязательно).
5. Выберите, надо ли включать в шаблон неопубликованные элементы содержимого и цифровые активы.
6. По завершении нажмите кнопку **Создать шаблон**.

После создания шаблона его имя появляется в списке шаблонов. Чтобы просмотреть все шаблоны, нажмите **Разработчик**, а затем **Просмотреть все шаблоны**. Можно управлять отображением шаблонов, нажав на значок представления и выбрав нужный вариант в списке.



Чтобы просмотреть папки и файлы, связанные с шаблоном, выберите **Открыть** в контекстном меню или нажмите  на панели действий. Чтобы просмотреть или изменить имя, описание и другие сведения о шаблоне, нажмите на имя шаблона

или выберите **Сведения** в контекстном меню или нажмите  на панели действий.

Копия сайта в новом шаблоне используется в качестве сайта по умолчанию. Шаблон содержит ссылки на используемую тему сайта и дополнительные компоненты, размещенные на страницах сайта. Тема и дополнительные компоненты не копируются в шаблон. Шаблон ссылается на них так же, как это происходит на сайте.

Если шаблон был создан на основе корпоративного сайта, политика локализации и язык по умолчанию, используемые исходным сайтом, будут выбраны по умолчанию для всех новых сайтов, созданных на основе этого шаблона.

Шаблон, созданный на основе сайта, отражает состояние этого сайта на момент создания шаблона. Будущие изменения, внесенные на сайте после создания шаблона, не влияют на сайт, сохраненный в этом шаблоне.

Изменение сведений о шаблоне

На вкладке сведений о шаблоне отображается имя шаблона, автор, описание, тема шаблона, пользовательские компоненты, включенные в шаблон, а также изображения для предварительного просмотра шаблонов. Если управление сайтом включено, также отображается статус шаблона (доступен ли шаблон для использования при создании сайтов).

При наличии соответствующих разрешений можно изменить или обновить свойства шаблона, такие как имя и описание. Также можно добавить или удалить изображение для предварительного просмотра шаблона.

Вы можете обновить свойства шаблона, если он создан вами (вы владелец шаблона), если вы администратор сайта, а также если вам предоставлен доступ к этому шаблону и назначена роль "Соавтор" или "Управление".

При добавлении изображения для предварительного просмотра файлы сохраняются в папке активов шаблона. Эти файлы не отображаются на связанных веб-сайтах, но позволяют владельцу шаблона предоставить информацию о самом шаблоне.

Просмотр и изменение свойств шаблона:

1. Выберите **Разработчик** и затем нажмите **Смотреть все шаблоны**.

2. Выберите шаблон и нажмите **Сведения** в контекстном меню или на  панели действий.
3. При необходимости измените информацию. Если вы не можете редактировать информацию, значит, у вас нет роли "Соавтор" или "Управление".
4. Чтобы добавить изображение для предварительного просмотра, нажмите . Найдите изображение для использования или нажмите **Загрузить** для загрузки локально сохраненного изображения. Изображение должно иметь размер как минимум 180 x 135 пикселей (соотношение сторон 4:3). Изображения меньшего размера могут отображаться не оптимально, а изображения большего размера могут влиять на производительность.

Если все точки изображения для предварительного просмотра заполнены, перед добавлением нового потребуется удалить изображение. Чтобы удалить изображение для предварительного просмотра, нажмите значок  на изображении.
5. Нажмите **Сохранить**.

Изменение политик шаблона

На вкладке шаблонов политик отображается тип шаблона (стандартный или корпоративный), требуемое утверждение и минимальный уровень безопасности для сайтов, созданных на основе шаблона. Также отображается статус шаблона (доступен ли шаблон для использования при создании сайтов).

Вкладка политик шаблонов отображается только в том случае, если включено управление сайтом и вы являетесь администратором сайта. См. [Общие сведения об управлении сайтами](#).

Просмотр и изменение политик шаблона:

1. Выберите **Разработчик** и затем нажмите **Смотреть все шаблоны**.
2. Выберите шаблон и нажмите **Сведения** в контекстном меню или на  панели действий.
3. Выберите вкладку **Политики**.
4. Выберите, требуется ли утверждение запросов сайта, созданных на основе этого шаблона, администратором сайта, или они утверждаются автоматически, или требуется утверждение конкретными пользователями. Если вы хотите разрешить утверждение определенным пользователям, начните с ввода имени или адреса электронной почты пользователя или группы, которые вы хотите добавить в качестве утверждающего, затем выберите пользователя или группу из результатов поиска. Чтобы удалить утверждающего, нажмите **X** рядом с его именем.
5. Выберите минимальный уровень безопасности для сайтов, созданных из этого шаблона. При необходимости создатели сайта могут выбрать более высокий уровень безопасности для своего сайта.
 - **Определенные пользователи сервиса.** Сайт доступен только выбранным пользователям, которые могут войти в этот экземпляр Oracle Content Management. Создатель сайта выбирает пользователей Oracle Content Management после создания сайта. См. [Изменение настроек безопасности сайта](#).

- **Определенные пользователи облачного сервиса** - сайт доступен только выбранным пользователям, которые могут войти в ваш домен. Создатель сайта выбирает пользователей облачного сервиса после создания сайта. См. [Изменение настроек безопасности сайта](#).
 - **Пользователи сервиса**. Сайт доступен любым пользователям, которые могут войти в этот экземпляр Oracle Content Management.
 - **Пользователи облачного сервиса** - сайт доступен любым пользователям, которые могут войти в ваш домен.
 - **Все** - сайт доступен любым пользователям без входа в систему.
6. Выберите политику истечения срока действия, чтобы определить истечение срока действия сайта, использующего данный шаблон. Администраторы сайтов могут указать, будут ли сайты с истекшим сроком действия переведены в автономный режим или удалены. Владельцы и менеджеры сайта получают уведомление по электронной почте до истечения срока действия сайта и могут продлить его срок действия. Если они не продлевают срок действия сайта, они будут уведомлены по электронной почте о том, что сайт был переведен в автономный режим или удален. В это время они могут продлить срок действия и при необходимости вернуть сайт в интерактивный режим или восстановить его из корзины. Параметры политики истечения срока действия
- **Никогда**
 - **1 год**
 - **2 года**
 - **Настраиваемый**
Выбор варианта **Настраиваемый** позволяет установить срок действия менее 1 года или более 2 лет.

 **Примечание.**

После создания сайта администраторы сайта могут изменить политику истечения срока действия сайта в диалоговом окне **Свойства сайта**.

7. Если вы редактируете шаблон предприятия, вы можете выбрать, как создать префикс сайта, чтобы он содержал понятные значения URL. Данный префикс будет добавлен к кодовым значениям элемента контента (части URL-адреса, относящейся к странице или активу). Вы можете создать префикс автоматически на основе имени сайта или позволить пользователю ввести префикс.
8. При редактировании корпоративного шаблона выберите политику локализации, используемую для определения необходимых языков сайта.
9. По завершении нажмите кнопку **Сохранить**.

Изменение аудитории или статуса шаблона

На вкладке аудитории шаблонов отображаются статус шаблона (доступность шаблона для использования при создании сайтов) и пользователи, которые могут использовать шаблон для создания сайтов.

Вкладка аудитории шаблонов отображается только в том случае, если включено управление сайтом и вы являетесь администратором сайта. См. [Общие сведения об управлении сайтами](#).

Просмотр и изменение аудитории шаблона:

1. Выберите **Разработчик** и затем нажмите **Смотреть все шаблоны**.
2. Выберите шаблон и нажмите **Сведения** в контекстном меню или на  панели действий.
3. Откройте вкладку **Аудитория**.
4. Чтобы изменить статус шаблона, нажмите переключатель статуса. Необходимо сделать шаблон активным для создания сайтов пользователями на основе этого шаблона. По умолчанию при активации шаблона он доступен всем создателям сайта. Если вы хотите изменить список пользователей, которые могут использовать шаблон, измените настройку на этой вкладке.
5. Выберите пользователей, которые могут использовать данный шаблон для создания сайтов.
 - **Отдельные пользователи.** Этот шаблон для создания сайтов могут использовать только выбранные пользователи. Начните с ввода имени или адреса электронной почты пользователя или группы, которым вы собираетесь разрешить использование данного шаблона, затем выберите пользователя или группу из результатов поиска. Чтобы удалить пользователя или группу, нажмите **X** рядом с их именем.
 - **Все.** Любой пользователь, который может создавать сайты, может использовать этот шаблон.
6. По завершении нажмите кнопку **Сохранить**.

Управление шаблонами

Папку шаблонов можно копировать, переименовывать и удалять, как любую другую папку, однако при импорте шаблонов и предоставлении доступа к ним следует учесть несколько соображений.

Откройте страницу "Шаблон", выбрав **Разработчик**, а затем **Просмотреть все шаблоны**. Для выполнения указанных ниже действий используйте строку меню или контекстное меню на странице "Шаблоны".

Задача	Описание
Создание шаблона	Если вы хотите использовать сайт в качестве отправной точки для других сайтов, можно создать шаблон на основе этого сайта.

 **Примечание.**

У вас должна быть роль "Выгрузка", "Соавтор" или "Управление" для сайта, и ваш администратор должен разрешить эту функцию. Если вы не видите меню **Создать** на странице шаблонов, обратитесь к администратору. Кроме того, при создании шаблона на основе сайта, который использует контент из нескольких репозиториях, необходимо это сделать с помощью инструментария ОСЕ. См. [Разработка с помощью инструментария ОСЕ](#) и [Использование утилиты командной строки СЕС](#).

На странице шаблонов нажмите **Создать**, а затем **Создать на основе существующего сайта**. Выберите сайт, добавьте имя шаблона и нажмите **Создать**. См. [Создание шаблона на основе сайта](#).

Просмотр и редактирование файлов шаблона	<p>Можно просмотреть папки и файлы, из которых создан шаблон, выбрав шаблон и выбрав Открыть в контекстном меню или нажав  на панели действий.</p> <p>Если вы являетесь веб-разработчиком, то можете выгружать и изменять отдельные файлы шаблонов, а также использовать настольное приложение и синхронизировать файлы после внесения изменений на локальном компьютере. Вы также можете экспортировать пакет шаблона, который включает связанную тему и пользовательские компоненты, и работать с ним в автономном режиме в предпочитаемой среде разработки. Шаблон представляет собой структуру папок, с которыми можно работать так же, как и с любыми другими папками. При экспорте шаблона все его элементы, включая копию темы и компоненты, объединяются в пакет, доступный для выгрузки и работы в автономном режиме. См. Разработка шаблонов.</p>
Создание сайта	<p>Чтобы создать сайт из шаблона, выберите шаблон и выберите Создать сайт в контекстном меню или нажмите  на панели действий. См. Создание сайтов.</p>
Переименование шаблона	<p>Выберите шаблон, который необходимо переименовать, и выберите Переименовать в контекстном меню или нажмите  на панели действий. Введите новое имя шаблона, отличное от всех имен шаблонов, существующих на том же сервере. Рекомендации по выбору имен см. в разделе Создание шаблона на основе сайта.</p>

Задача	Описание
Копирование шаблона	<p>Шаблон можно создать, скопировав существующий шаблон и внося изменения в полученную копию.</p>
	<div style="border-left: 2px solid #0070C0; padding-left: 10px;"> <p> Примечание.</p> <p>При копировании шаблона сведения об общем доступе к этому шаблону не копируются.</p> </div>
	<p>Выберите шаблон, который требуется скопировать, и выберите Копировать в контекстном меню или нажмите  на панели действий. Введите новое имя шаблона, отличное от всех имен шаблонов, существующих на том же сервере. Другие рекомендации по выбору имен см. в разделе Создание шаблона на основе сайта.</p> <p>На индикаторе выполнения отображается имя нового шаблона и статус копирования.</p>
Удаление и восстановление шаблона	<p>При наличии соответствующих разрешений можно удалить папку шаблона и ее содержимое. При удалении шаблона соответствующая папка и все связанные с ней папки и файлы перемещаются в корзину.</p> <p>Можно удалить или восстановить шаблон, если он создан вами (вы являетесь владельцем шаблона) или другой пользователь предоставил вам доступ к этому шаблону и назначил роль "Соавтор" или "Управление".</p>
	<div style="border-left: 2px solid #0070C0; padding-left: 10px;"> <p> Примечание.</p> <p>Удаление шаблона не приводит к удалению связанной с ним темы и пользовательских компонентов.</p> </div>
	<p>Чтобы удалить шаблон, выберите шаблон, который необходимо удалить, и выберите Удалить в контекстном меню или нажмите  на панели действий. Отобразится запрос на перемещение шаблона в корзину. Удаленный шаблон остается в корзине до наступления перечисленных ниже событий:</p> <ul style="list-style-type: none"> • Вы восстанавливаете шаблон. • Вы окончательно удаляете шаблон. • Квота корзины превышена. • Корзина автоматически опустошается с интервалом, заданным администратором сервиса. Значение по умолчанию равно 90 дням. <p>Чтобы восстановить шаблон, в раскрывающемся списке "Шаблоны" выберите Корзина, затем выберите шаблон и выберите Восстановить в строке меню или в контекстном меню.</p>

Задача	Описание
Экспорт и импорт шаблона	<p>Экспортированный шаблон можно изменить в автономном режиме, а затем импортировать как новый шаблон или заменить существующий. Можно также экспортировать шаблон, переместить его в другой экземпляр и импортировать в этом экземпляре.</p> <p>При экспорте шаблона выполняется копирование шаблона в папку в качестве файла ZIP. Пакет шаблона можно выгрузить непосредственно из этой папки и распаковать для работы с отдельными файлами. После внесения всех необходимых изменений в файлы создайте файл ZIP, содержащий пакет шаблона, импортируйте его и замените исходный шаблон или создайте новый.</p> <p>См. Экспорт и импорт шаблонов.</p>
Предоставление общего доступа к шаблону и управление ролями участников	<p>Если вы являетесь владельцем или менеджером шаблона, вы можете предоставить другим пользователям общий доступ к шаблону и назначить роли, которые определяют возможные действия пользователей с шаблоном.</p> <p>Если управление <i>не включено</i>, назначение какой-либо роли пользователю дает ему разрешение на создание сайта из шаблона. Если управление <i>включено</i>, необходимо активировать шаблон и убедиться, что пользователь включен в аудиторию шаблона, чтобы предоставить ему разрешение на создание сайта из шаблона. См. Изменение аудитории или статуса шаблона.</p> <p>Тема шаблона и пользовательские компоненты хранятся как отдельные объекты и только ссылаются на шаблон. Поэтому управление принадлежностью к теме и пользовательскому компоненту выполняется отдельно. При предоставлении пользователю общего доступа к шаблону, независимо от роли, присвоенной ему для шаблона, пользователь автоматически получает роль "Выгрузка" для любой связанной темы и пользовательских компонентов для обеспечения доступности пользователю данных объектов при условии создания сайта из шаблона.</p> <p>Выберите шаблон, к которому необходимо предоставить общий доступ, выберите Участники в контекстном меню или на панели действий.</p> <p>Введите одно или несколько имен пользователей или адресов электронной почты и назначьте им одну из описанных ниже ролей.</p> <ul style="list-style-type: none">• Просмотр: пользователи с этой ролью могут просматривать папки и файлы шаблона, но не могут вносить изменения.• Выгрузка: пользователи с ролью "Выгрузка" могут выгружать файлы и сохранять их на своем компьютере.• Соавтор: пользователи с этой ролью также могут редактировать свойства и файлы шаблона, загружать новые файлы, удалять файлы и сам шаблон.• Управление: пользователи с ролью "Управление" также могут добавлять пользователей и назначать им роли. Владелец шаблона (автору) автоматически назначается роль "Управление".

Задача	Описание
Просмотр и редактирование сведений о шаблоне	Чтобы просмотреть сведения о шаблоне, выберите шаблон, а затем выберите Сведения в контекстном меню или нажмите  на панели действий. Сведения о шаблоне включают имя шаблона, автора, файлы поддержки, описание, тему, пользовательские компоненты и эскизы. При наличии соответствующей роли шаблона можно редактировать сведения. См. Изменение сведений о шаблоне .
Просмотр и изменение политик шаблонов (с включенным управлением)	Если управление включено и вы являетесь администратором сайта, также доступен просмотр политик шаблона, например требующееся утверждение, минимальный уровень безопасности, истечение срока действия сайта и локализация. Выберите шаблон, выберите Сведения в контекстном меню или нажмите  на панели действий, затем выберите вкладку Политики . См. Изменение политик шаблона .
Обеспечение доступности шаблона для создания сайта и назначение пользователей, которые могут использовать шаблон для создания сайтов (с включенным управлением)	Если включено управление и вы являетесь администратором сайта, можно сделать шаблон доступным для создания сайта и назначить пользователей, которые могут использовать шаблон для создания сайтов. Рядом с активными шаблонами (доступными для создания сайта) отображается значок  . Чтобы изменить статус или аудиторию, выберите шаблон и выберите Сведения в контекстном меню или нажмите  на панели действий. <ul style="list-style-type: none"> • Чтобы сделать шаблон доступным для создания сайта, нажмите переключатель состояния для его активации. • Чтобы изменить состав пользователей, которые могут использовать шаблон для создания сайтов, нажмите на вкладку Аудитория. См. Изменение аудитории или статуса шаблона.

Экспорт и импорт шаблонов

Экспортированный шаблон можно изменить в автономном режиме, а затем импортировать как новый шаблон или заменить существующий. Можно также экспортировать шаблон, переместить его в другой экземпляр Oracle Content Management и импортировать в этом экземпляре.

Экспорт шаблонов

Экспортируемый шаблон копируется в папку Oracle Content Management как одиночный файл ZIP. Пакет шаблона можно выгрузить непосредственно из этой папки и распаковать для работы с отдельными файлами. После внесения всех необходимых изменений в файлы шаблона создайте файл ZIP, содержащий пакет шаблона, импортируйте его и перезапишите исходный шаблон или создайте новый.

Примечание.

При экспорте шаблона сведения об общем доступе к шаблону не сохраняются.

Экспорт шаблона:

1. Выберите **Разработчик** и нажмите **Смотреть все шаблоны**.
2. Выберите шаблон и нажмите **Экспорт** в строке меню или в контекстном меню.
3. Перейдите в существующую папку или создайте новую папку, нажав **Создать**, укажите имя и добавьте описание (необязательно).
4. Выберите папку, установив для нее флажок, и нажмите **ОК**.

Импорт шаблонов

 **Примечание.**

Если шаблон импортирован на другой сервер, некоторые ссылки на сайте по умолчанию могут не работать в контексте нового сервера. Если на сайте используются ссылки на изображения или другое содержимое вместо непосредственных копий, такое содержимое будет недоступно на новом сервере. Даже если содержимое скопировано на новый сервер, его внутренний идентификатор отличается, поэтому ссылка не работает. При импорте шаблона отображаются уведомления о страницах, содержащих нерабочие ссылки.

Импорт пакета шаблона:

1. Выберите **Разработчик** и нажмите **Смотреть все шаблоны**.
2. Нажмите **Создать** и выберите **Импортировать пакет шаблона**.
3. Если пакет шаблона загружен, перейдите в соответствующую папку. Если пакет шаблона еще не загружен, перейдите в папку, в которую необходимо загрузить шаблон, или создайте новую папку. Нажмите **Загрузить**, найдите и выберите пакет шаблона и нажмите **Открыть**.
4. Чтобы использовать шаблон, установите флажок рядом с именем файла шаблона и нажмите **ОК**. Создаются папки для шаблона, связанной с ним темы и пользовательских компонентов. Если имя или идентификатор шаблона, темы или пользовательского компонента уже существуют, то отображается запрос на устранение конфликтов. Возможно, потребуется создать новый шаблон, тему или пользовательский компонент либо перезаписать существующие.

Импорт шаблонов в определенный репозиторий

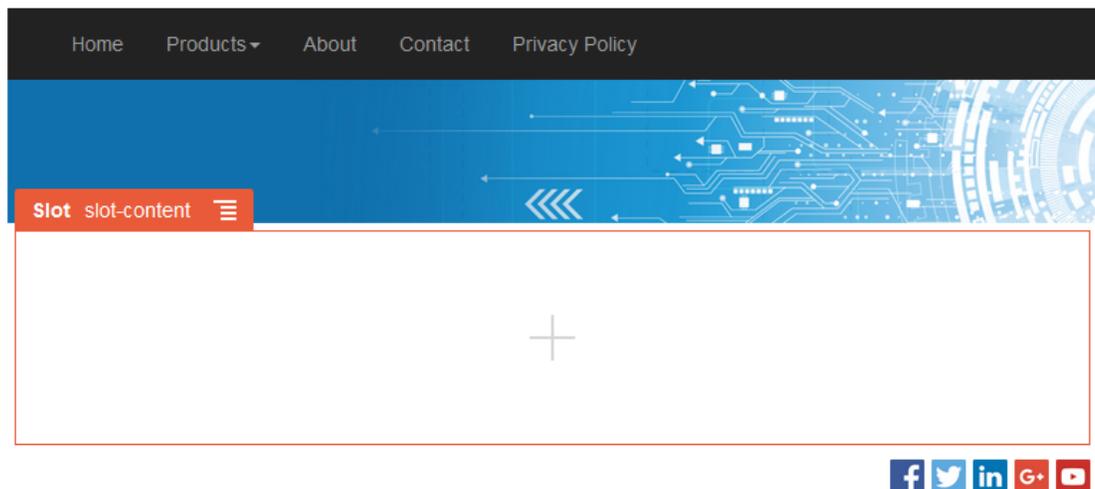
Можно выбрать репозиторий и импортировать в него пакет шаблонов.

1. Нажмите **Контент** и в меню выберите **Репозитории**.
2. Выберите репозиторий, в который необходимо импортировать пакет шаблонов, и нажмите **Импорт контента**.
3. Если пакет шаблона загружен, перейдите в соответствующую папку. Если пакет шаблона еще не загружен, перейдите в папку, в которую необходимо загрузить шаблон, или создайте новую папку. Нажмите **Загрузить**, найдите и выберите пакет шаблона и нажмите **ОК**.

4. Выберите, требуется ли обновить активы, которые уже существуют в репозитории, до новых версий, или создать дубликаты созданных активов, и нажмите **Импорт**.
5. Нажмите **Сведения**, чтобы просмотреть список импортированных активов и типов контента. По завершении нажмите **ОК**.

Общие сведения о темах

Тема определяет общий внешний вид (общий стиль) сайта, в том числе цветовую гамму, размеры и типы шрифтов, а также фон страниц. Темы обеспечивают визуальную согласованность страниц сайта. Вы настраиваете дизайн и добавляете содержимое, чтобы создать сайт, представляющий ваши стиль, бренд и ваше видение.



Тема включает в себя следующие элементы:

- макеты страниц;
- каскадная таблица стилей (CSS);
- вариации на основе таблицы стилей;
- файлы конфигурации;
- фоновый код, определяющий навигацию сайта.

Каждый сайт использует тему. При создании сайта на основе шаблона наследуется тема, связанная с шаблоном. Тему сайта можно изменить в любое время.

Для вашего удобства Oracle Content Management предоставляет ряд шаблонов с темами. Для создания новой темы скопируйте существующую. Можно выгрузить и изменить файлы тем или использовать приложение для ПК и синхронизировать изменения, внесенные в локальной системе. Сведения о дополнительных способах создания тем см. в разделе [Разработка тем](#).

Разработчики также могут создать тему с определенным подмножеством компонентов. При создании сайта с использованием одной из таких тем отображаются только те компоненты, которые связаны с этой темой. Таким образом, вы не задумываетесь о том, допускается ли использование того или иного компонента в выбранной теме. Вам по-прежнему доступны все предварительно заданные компоненты вместе с тематическими компонентами.

Если сайт использует новую неопубликованную тему, она автоматически публикуется вместе с сайтом при его первом переводе в онлайн-режим. Если вы внесли в тему изменения и хотите обновить онлайн-сайты, чтобы отобразить изменения, измененную тему необходимо опубликовать явным образом. Опубликовать тему могут только владельцы темы и пользователи с правами "Управление".

 **Примечание.**

Если опубликовать изменения в теме, они будут отражены на всех онлайн-сайтах, где используется эта тема. Например, если изменить указанный в теме шрифт по умолчанию и опубликовать тему, то на всех сайтах, где используется эта тема, будет применен новый шрифт по умолчанию.

Управление темами

Папки тем можно копировать, переименовывать и удалять так же, как и любые другие, но при этом необходимо соблюдать некоторые правила, поскольку темы могут использоваться одновременно на нескольких сайтах.

Для просмотра всех тем нажмите **Разработчик** и затем выберите **Смотреть все темы**. Для выполнения указанных ниже задач используются кнопки в строке меню или команды контекстного меню на странице "Темы".

Задача	Описание
Создание новой темы	Для создания новой темы скопируйте существующую. Сведения о других способах создания тем см. в разделе Разработка тем .
Копирование темы	Новую тему можно создать, скопировав и изменив существующую тему.

 **Примечание.**

При копировании темы сведения об общем доступе к этой теме не копируются.

Выберите тему для копирования и затем нажмите **Копировать** в строке меню или в контекстном меню. Введите уникальное имя темы. Можно употреблять буквы, цифры, символы подчеркивания (_) и дефисы (-). Введенный пробел автоматически заменяется дефисом. Нажмите **Копировать**.

Не используйте для шаблонов, тем, компонентов, сайтов и страниц сайтов следующие имена: authsite, content, pages, scstemplate_*, _comps, _components, _compsdelivery, _idcservice, _sitescloud, _sitesclouddelivery, _themes, _themesdelivery.

Следующие имена можно использовать только для страниц сайтов (не для шаблонов, тем, компонентов и сайтов): documents, sites.

Задача	Описание
Предоставление общего доступа к теме	Если администратор включил общий доступ и вы являетесь владельцем или менеджером темы, можно поделиться своей темой с другими пользователями. При предоставлении другому пользователю общего доступа к теме необходимо назначить ему роль, определяющую разрешенные для этого пользователя действия.

 **Примечание.**

При предоставлении пользователю общего доступа к шаблону, независимо от роли, присвоенной ему для шаблона, пользователь автоматически получает роль "Выгрузка" для любой связанной темы для обеспечения доступности темы при условии создания сайта из шаблона.

Нажмите правой кнопкой мыши на имя темы, к которой необходимо предоставить общий доступ, выберите **Общий доступ**, затем нажмите **Добавить участников**. Введите одно или несколько имен пользователей или адресов электронной почты и назначьте им одну из описанных ниже ролей.

- **Просмотр:** пользователи с этой ролью могут просматривать папки и файлы темы, но не могут вносить изменения.
- **Выгрузка:** пользователи с ролью "Выгрузка" могут выгружать файлы и сохранять их на своем компьютере.
- **Соавтор:** пользователи с этой ролью также могут редактировать тему, загружать новые файлы, удалять файлы и саму тему.
- **Управление:** пользователи с ролью "Управление" также могут добавлять пользователей и назначать им роли. Автору темы (владельцу) автоматически назначается роль "Управление".

Редактирование темы	Веб-разработчики могут выгружать и изменять отдельные файлы тем или работать в настольном приложении и синхронизировать изменения, внесенные в локальной системе. Также можно экспортировать пакет шаблона, содержащий связанную тему и другие пользовательские компоненты, и работать с этим пакетом в автономном режиме в предпочтительной среде разработки. См. Разработка тем . Если вы внесли изменения в тему, необходимо опубликовать ее, чтобы изменения отображались на сайтах, где используется эта тема.
---------------------	--

Задача	Описание
Публикация темы	<p>Если сайт использует новую неопубликованную тему, она автоматически публикуется вместе с сайтом при его первом переводе в онлайн-режим. Если вы внесли в тему изменения и хотите обновить онлайн-сайты, чтобы отобразить изменения, измененную тему необходимо опубликовать явным образом. Тему можно опубликовать, если вы ее владелец или если вам назначена роль "Управление".</p> <p>Выберите тему в списке, затем нажмите Опубликовать (или Опубликовать повторно для ранее опубликованных тем) в строке меню. Рядом с темой в списке будет добавлен значок "Опубликовано" . См. Публикация тем.</p>
Удаление и восстановление темы	<p>При наличии соответствующих разрешений можно удалить папку темы и ее содержимое. При удалении темы папка темы и все связанные с ней файлы перемещаются в корзину.</p> <p>Тему можно удалить или восстановить, если она создана вами (вы владелец этой темы) или другой пользователь предоставил вам общий доступ к этой теме и роль "Соавтор" или "Управление".</p>

**Примечание.**

Если тема используется на каком-либо сайте, удалить ее невозможно.

Нажмите правой кнопкой мыши на имя темы, которую необходимо удалить, и выберите **Удалить**. Отобразится запрос на перемещение темы в корзину. Удаленная тема остается в корзине до тех пор, пока не наступит одно из указанных ниже событий.

- Восстановление темы
- Окончательное удаление темы
- Квота корзины превышена.
- Корзина автоматически опустошается с интервалом, заданным администратором сервиса. Значение по умолчанию равно 90 дням.

Чтобы восстановить тему, нажмите **Корзина** в строке меню страницы "Тема", затем нажмите на тему в списке правой кнопкой мыши и выберите **Восстановить**.

Публикация тем

Тема определяет общий внешний вид сайта. Вы можете обновить тему, чтобы изменить внешний вид сайтов, где используется эта тема.

Если сайт использует новую неопубликованную тему, она автоматически публикуется вместе с сайтом при его первом переводе в онлайн-режим. Если вы внесли в тему изменения и хотите обновить онлайн-сайты, чтобы отобразить изменения, измененную тему необходимо опубликовать явным образом. Чтобы опубликовать изменения в теме, вы должны быть ее владельцем или обладать ролью "Управление".

 **Примечание.**

Если опубликовать изменения в теме, они будут отражены на всех онлайн-сайтах, где используется эта тема. Прежде чем публиковать обновление темы, протестируйте изменения в автономном режиме и определите, как они повлияют на связанные сайты.

Чтобы опубликовать тему, выполните указанные ниже действия.

1. Нажмите **Разработчик** и затем выберите **Смотреть все темы**.
2. Выберите существующую тему из списка тем.
3. Нажмите **Опубликовать** в строке меню или контекстном меню.
4. Выберите **Подтвердите, чтобы продолжить**, а затем нажмите кнопку **ОК**. Рядом с темой в списке будет добавлен значок "Опубликовано" .

7

Управление пользовательскими компонентами и макетами

Разработчики могут создавать свои компоненты и макеты и управлять ими. Пользователи, которым назначена роль "Соавтор" сайта, могут регистрировать сторонние компоненты (приложения) и группы компонентов.

К пользовательским компонентам относятся группы компонентов, созданные в редакторе, а также локальные и удаленные компоненты, созданные с помощью инструментов, описанных в этом разделе. К пользовательским макетам относятся *макеты разделов*, предназначенные для размещения компонентов в определенных областях страницы, и *макеты содержимого*, определяющие расположение полей в элементе содержимого.

- [Общие сведения о пользовательских компонентах](#)
- [Общие сведения о макетах](#)
- [Использование пользовательских компонентов и макетов](#)
- [Регистрация удаленных компонентов](#)
- [Создание локальных компонентов, макетов, редакторов полей контента или форм контента](#)
- [Экспорт и импорт компонентов или макетов](#)

Информацию об использовании компонентов на сайте см. в разделе [Организация содержимого страницы](#).

Сведения об использовании отдельных компонентов см. в разделе [Использование встроенных компонентов](#).

Подробную информацию о создании собственных компонентов, см. в разделе [Разработка компонентов](#).

Общие сведения о пользовательских компонентах

Компоненты — это отдельные части веб-страницы. Что вы увидите, взглянув на веб-страницу? Вероятно, вы увидите несколько заголовков, абзацы текста и несколько ссылок на другие страницы сайта. Также можно увидеть изображения, кнопки, разделители, карты и галереи. Каждый из этих элементов — компонент.

Чтобы добавить компонент на страницу, убедитесь, что переключатель 

переведен в положение **Редактировать**, нажмите  и выберите тип компонента, который необходимо использовать.

Перетащите компонент с панели в ячейку на странице. Вот и все. Перетаскивайте заголовки, абзацы, изображения и другие компоненты туда, где они должны быть на странице.

Вы можете с легкостью регистрировать и встраивать удаленные компоненты (приложения) и даже создавать собственные компоненты, используя возможности диспетчера компонентов. Выберите **Разработчик**, а затем нажмите **Смотреть все компоненты**.

Нажмите **Создать** и выберите соответствующий параметр, чтобы создать новый локальный компонент или зарегистрировать удаленный компонент. Созданные и опубликованные таким образом компоненты размещаются на панели "Пользовательские компоненты" в редакторе.

Подробную информацию о создании собственных компонентов, см. в разделе [Разработка компонентов](#).

Удаленные компоненты

Чтобы использовать на сайте сторонний компонент (приложение), просто зарегистрируйте его. Вот и все.

Для регистрации удаленного компонента необходимо указать URL-адрес самого компонента и URL-адрес для параметров компонента.

Зарегистрированный удаленный компонент можно сделать доступным для других пользователей. Принадлежащие вам зарегистрированные удаленные компоненты, а также удаленные компоненты, к которым вам предоставлен доступ, отображаются на панели "Пользовательские компоненты" в Site Builder.

Свойства удаленного компонента можно изменить в редакторе так же, как и свойства любого другого компонента. Нажмите на  и выберите **Параметры**. При нажатии кнопки **Пользовательские** открывается URL-адрес параметров, указанный при регистрации удаленного компонента.

Примечание.

Поскольку удаленные компоненты размещаются на другом сервере, отличном от сервера Oracle Content Management, в целях безопасности они заключаются во встроенную рамку (с помощью элемента `iframe`). Во встроенный фрейм можно заключить не все удаленные компоненты. Информацию о возможности заключения компонента во встроенный фрейм уточняйте у поставщика компонента.

Локальные компоненты

Разработчики могут создавать компоненты с доступом к тем же функциям и возможностям, которые предоставляются Oracle Content Management. Компонент можно вставить непосредственно на страницу или заключить во встроенный фрейм с помощью элемента `iframe`.

При нажатии кнопки **Создать** в список компонентов добавляется полнофункциональный образец локального компонента с указанным именем и уникальным идентификатором. Разработчик может внести изменения в этот образец и создать собственное решение.

Подробную информацию о создании собственных компонентов, см. в разделе [Разработка компонентов](#).

Общие сведения о макетах

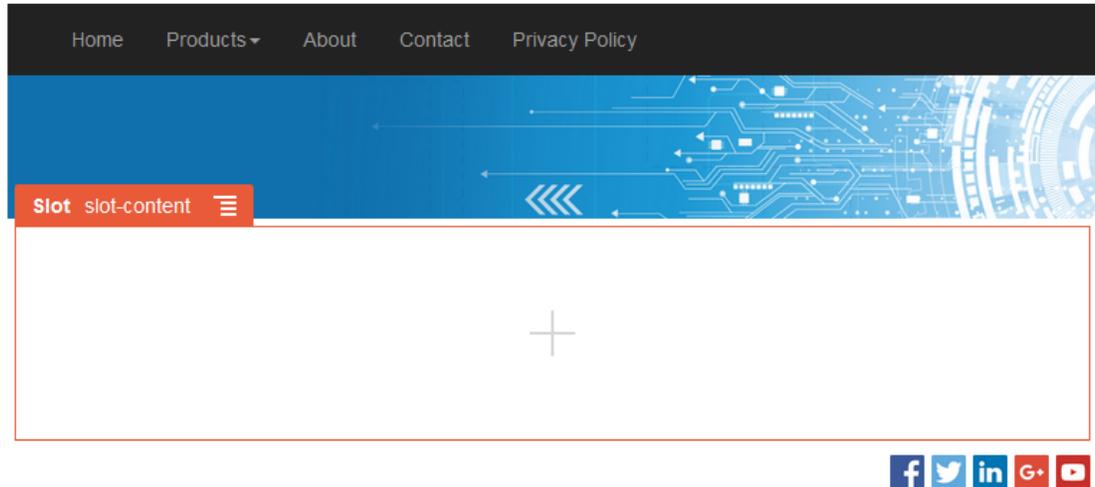
Макет страницы определяет расположение ячеек и содержимого на странице. Макет раздела упорядочивает содержимое в ячейке. Макет содержимого упорядочивает поля в элементе содержимого.

В целом, макет определяет представление содержимого, но не само содержимое. Так как содержимое отделено от представления, можно представлять одно и то же содержимое разными способами или изменять представление, не затрагивая содержимое.

Макет страницы

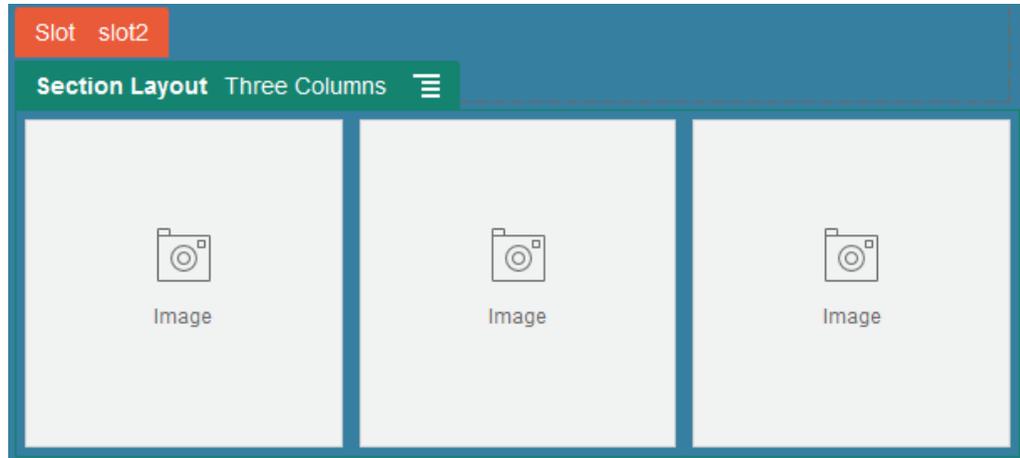
При добавлении страницы на сайт требуется выбрать ее макет. Каждый макет включает в себя области страницы - так называемые ячейки, куда пользователь с ролью "Соавтор" может поместить элементы. Макет страницы определяет количество и расположение ячеек на странице. Кроме того, макет может включать содержимое, заранее определенное и размещенное на странице. Содержимое может быть статическим и недоступным для редактирования (например, логотип компании) или предусматривать минимальные изменения (например, пользователь с ролью "Соавтор" может отредактировать текст заголовка, но не его оформление или расположение).

Макеты страниц хранятся в теме. Тема может содержать один или несколько макетов страниц. Разработчики могут копировать и изменять существующие темы, создавая новые. См. [Разработка макетов](#).



Макет раздела

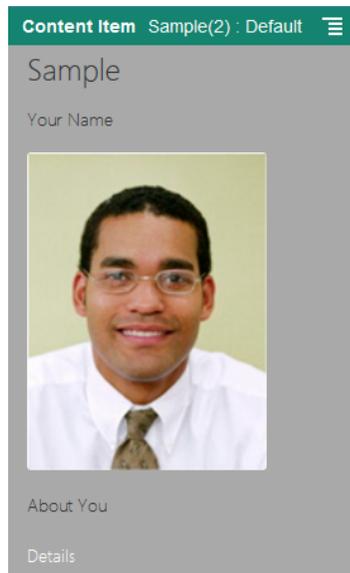
Макет разделов автоматически организует добавляемое в него содержимое, что позволяет пользователям, которым назначена роль "Соавтор", легко добавлять содержимое, не тратя время на его форматирование на странице. Например, макет раздела может автоматически распределять содержимое по нескольким столбцам или представлять его в виде вертикального списка. Пользователь с ролью "Соавтор" может добавить в ячейку один или несколько макетов разделов для структурирования содержимого.



Помимо готовых стандартных макетов разработчики могут создать дополнительные макеты разделов, чтобы устранить определенные проблемы или упростить работу с содержимым. Разработчики тем также могут встраивать макеты разделов в ячейку макета страницы. См. [Разработка макетов](#).

Макет содержимого

Корпоративные пользователи могут создавать и использовать элементы содержимого в соответствии с предоставленными типами содержимого и макетами. Благодаря такой структуре пользователи с ролью "Соавтор" могут собирать контент для элементов вне Site Builder. Если с элементом содержимого связано несколько макетов содержимого, разработчики сайтов могут отображать это содержимое в различных контекстах, не затрагивая собранное содержимое. Разработчики могут создавать новые макеты на основе предоставленных макетов по умолчанию. См. [Разработка макетов](#).



Использование пользовательских компонентов и макетов

Разработчики могут создавать свои компоненты и макеты и управлять ими. Пользователи, которым назначена роль "Соавтор" сайта, могут регистрировать сторонние компоненты (приложения) и группы компонентов.

К пользовательским компонентам относятся группы компонентов, созданные в редакторе, а также локальные и удаленные компоненты, созданные с помощью функций, описанных далее в этом разделе. Чтобы использовать на сайте стороннее приложение (удаленный компонент), достаточно зарегистрировать его. Разработчики также могут создавать локальные компоненты с доступом к тем же функциям и возможностям, которые предоставляются в Oracle Content Management.

К пользовательским макетам относятся *макеты разделов*, предназначенные для размещения компонентов в определенных областях страницы, и *макеты содержимого*, определяющие расположение полей в элементе содержимого.

Для выполнения указанных ниже задач используются кнопки в строке меню или команды контекстного меню на странице "Компоненты".

Задача	Описание
Регистрация удаленного компонента	<p>Для регистрации удаленного компонента необходимо указать URL-адрес самого компонента и URL-адрес для параметров компонента. Чтобы пользователи могли создавать компоненты, администратор должен включить соответствующий параметр в меню Создать. Если вы не видите меню Создать на странице компонентов, обратитесь к администратору.</p> <p>Нажмите Создать и выберите соответствующий параметр, чтобы зарегистрировать удаленный компонент. См. Регистрация удаленных компонентов.</p>
Создание локального компонента или макета	<p>При создании локального компонента или макета предлагается полнофункциональный образец компонента или макета, который можно использовать как основу для нового компонента или макета. Чтобы пользователи могли создавать компоненты или макеты, администратор должен включить соответствующий параметр в меню Создать. Если вы не видите меню Создать на странице компонентов, обратитесь к администратору.</p> <p>Нажмите Создать и выберите соответствующий параметр, чтобы создать новый локальный компонент или макет. См. Создание локальных компонентов, макетов, редакторов полей контента или форм контента.</p> <p>Сведения о других способах создания компонентов см. в разделе Разработка компонентов.</p>

Задача	Описание
Копирование компонента или макета	<p>Чтобы создать новый компонент или макет, можно скопировать существующий компонент или макет и внести изменения в копию.</p>
	<div style="border-left: 2px solid #0070C0; border-right: 2px solid #0070C0; border-bottom: 2px solid #0070C0; padding: 10px; background-color: #E6F2FF;"> <p> Примечание.</p> <p>Изменить имя созданного или зарегистрированного компонента или макета невозможно. Можно скопировать компонент или макет и указать другое имя для копии. Все остальные регистрационные данные, включая значение ключа для удаленных компонентов, сохраняются. Параметры общего доступа не зависят от регистрационных данных и не копируются.</p> </div>
	<p>Нажмите правой кнопкой мыши на имя компонента или макета, который необходимо скопировать, и выберите Копировать. Введите имя и нажмите Копировать. Можно употреблять буквы, цифры, символы подчеркивания (_) и дефисы (-). Введенный пробел автоматически заменяется дефисом.</p>
Создание редактора полей контента	<p>Можно создать редактор полей контента, после чего можно повысить его уровень и использовать при создании типа контента для управления внешним видом поля данных. Поддерживаются все типы данных, кроме <i>Мультимедиа</i> и <i>Ссылка</i>. Чтобы сделать компонент доступным для использования при создании типа контента, необходимо повысить уровень этого компонента.</p>
Предоставление доступа к компоненту или макету	<p>Вы можете предоставить доступ к своему компоненту или макету другим пользователям Oracle Content Management. Вы можете предоставить доступ к компоненту или макету, если вы его владелец или если вам назначена роль "Управление". Чтобы предоставить доступ к компоненту или макету, необходимо назначить пользователю роль, определяющую действия, которые пользователь может выполнять с этим компонентом.</p> <p>Нажмите правой кнопкой мыши на имя компонента или макета, к которому необходимо предоставить общий доступ, выберите пункт Общий доступ, затем нажмите Добавить участников.</p> <p>Введите одно или несколько имен пользователей или адресов электронной почты и назначьте им одну из описанных ниже ролей.</p> <ul style="list-style-type: none"> • Просмотр. Пользователи с ролью "Просмотр" могут просматривать и использовать компонент или макет, доступный в списке пользовательских компонентов или макетов в редакторе. Также они могут просматривать папки и файлы темы, но не могут вносить изменения. • Выгрузка: пользователи с ролью "Выгрузка" могут выгружать файлы и сохранять их на своем компьютере. • Соавтор. Пользователи с ролью "Соавтор" также могут редактировать компонент или макет, загружать новые файлы, удалять файлы, а также удалять сами компоненты или макеты, если они не используются на сайте. • Управление: пользователи с ролью "Управление" также могут добавлять пользователей и назначать им роли. Автору (владельцу) компонента или макета автоматически назначается роль "Управление".

Задача	Описание
Редактирование компонента или макета	<p>Веб-разработчики могут выгружать и изменять отдельные файлы компонентов или макетов, а также использовать настольное приложение и синхронизировать файлы после внесения изменений на локальном компьютере.</p> <p>Также можно экспортировать компонент или макет отдельно или в составе пакета шаблона, содержащего пользовательские компоненты и макеты разделов, и работать с этим компонентом или макетом в автономном режиме в предпочтительной среде разработки.</p> <p>Если вы внесли изменения в компонент или макет, необходимо опубликовать его, чтобы изменения отобразились на сайтах, где используется этот компонент или макет.</p>
Экспорт и импорт компонента или макета	<p>Можно экспортировать компонент или макет для внесения изменений в автономном режиме, а затем импортировать его в качестве нового компонента или макета либо заменить существующий компонент или макет. Экспортированный компонент или макет также можно перенести в другой экземпляр и импортировать в этом экземпляре. Компонент или макет можно экспортировать отдельно или в составе пакета шаблона, содержащего пользовательские компоненты и макеты.</p> <p>Экспортируемый компонент или макет копируется в папку как одиночный файл ZIP. Пакет компонента или макета можно выгрузить непосредственно из этой папки и распаковать для работы с отдельными файлами. После внесения всех необходимых изменений в файлы компонента или макета создайте файл ZIP, содержащий пакет компонента или макета, импортируйте его и замените исходный компонент или макет либо создайте новый.</p> <p>См. Экспорт и импорт компонентов или макетов.</p>
Публикация компонента или макета	<p>Если на сайте используется новый неопубликованный компонент или макет, он автоматически публикуется вместе с сайтом при первом переводе сайта в онлайн-режим. Если вы внесли изменения в компонент или макет и хотите обновить онлайн-сайты, чтобы применить эти изменения, измененный компонент или макет необходимо опубликовать явным образом. Вы можете опубликовать компонент или макет, если вы его владелец или если вам назначена роль "Управление".</p> <p>Выберите компонент или макет в списке, затем нажмите Опубликовать (или Опубликовать повторно для ранее опубликованных компонентов или макетов) в строке меню. Рядом с компонентом или макетом в списке отображается значок "Опубликовано" .</p>

Задача	Описание
Удаление и восстановление компонента или макета	<p>При наличии соответствующих разрешений можно удалить папку компонента или макета его папка и все соответствующие папки и файлы перемещаются в корзину.</p> <p>Компонент или макет можно удалить или восстановить, если он создан вами (вы владелец этого компонента) или другой пользователь предоставил вам доступ к этому компоненту и назначил роль "Соавтор" или "Управление".</p>

 **Примечание.**

Если компонент или макет используется на сайте или в обновлении (включая сайты и обновления, находящиеся в корзине), удалить этот компонент или макет невозможно.

Чтобы удалить компонент или макет, нажмите правой кнопкой мыши на его имя и выберите **Удалить**. Отобразится запрос на перемещение компонента или макета в корзину. Удаленный компонент или макет остается в корзине до тех пор, пока не наступит одно из указанных ниже событий.

- Восстановление компонента или макета
- Окончательное удаление компонента или макета
- Квота корзины превышена.
- Корзина автоматически опустошается с интервалом, заданным администратором сервиса. Значение по умолчанию равно 90 дням.

Чтобы восстановить компонент или макет, нажмите на значок **Корзина**, затем нажмите правой кнопкой мыши на компонент или макет в списке и выберите **Восстановить**.

Регистрация удаленных компонентов

Чтобы использовать удаленный компонент на сайте, сначала необходимо его зарегистрировать в Oracle Content Management.

Можно регистрировать сторонние и собственные удаленные компоненты.

 **Примечание.**

Чтобы пользователи могли регистрировать удаленные компоненты, администратор должен включить соответствующий параметр в меню **Создать**. Если вы не видите меню **Создать** на странице компонентов, обратитесь к администратору.

Чтобы зарегистрировать удаленный компонент для использования в Oracle Content Management:

1. Выберите **Разработчик**, а затем нажмите **Смотреть все компоненты**.
2. Нажмите **Создать** и выберите **Зарегистрировать удаленный компонент**.
3. В окне "Регистрация удаленного компонента" введите или выберите нужные сведения, включая описанные ниже.
 - **Имя**: имя компонента, которое отображается для пользователей.
 - **Описание**: описание компонента, которое отображается для пользователей.
 - **URL-адрес компонента**: конечная точка, которая используется в элементе iframe для визуализации содержимого компонента на странице. Он должен соответствовать протоколу HTTPS.
 - **URL-адрес настроек**: конечная точка, которая используется в элементе iframe для визуализации настроек удаленного компонента, добавленного на страницу. Он должен соответствовать протоколу HTTPS.
 - **Ширина панели настроек**: используемая по умолчанию ширина панели настроек компонента (в пикселях).
 - **Высота панели настроек**: используемая по умолчанию высота панели настроек компонента (в пикселях).
 - **Ключ**: 192-разрядный AES-ключ, который связан с удаленным компонентом и используется для создания подписанной хеш-лексема при развертывании компонента. Он используется для шифрования и обеспечения безопасности во время чтения и записи настроек компонента.

4. Нажмите **Регистрация**.

После создания удаленного компонента его имя появляется в списке компонентов. Чтобы просмотреть файлы, используемые для регистрации компонента, выберите имя компонента в списке компонентов.

Сведения о регистрации компонента сохраняются в каталоге, который используется сайтами, созданными в том же экземпляре Oracle Content Management, но компонент остается удаленным сервисом.

Для владельца компонента на панели "Пользовательские компоненты" в редакторе отображается значок компонента с именем, присвоенным этому компоненту. Если предоставить доступ к компоненту другим пользователям, они также увидят этот компонент на панели "Пользовательские компоненты" в редакторе.

Создание локальных компонентов, макетов, редакторов полей контента или форм контента

При создании локального компонента или макета разработчику предлагается полнофункциональный образец, который можно изменить и преобразовать в собственный компонент или макет.

Чтобы пользователи могли создавать компоненты или макеты, администратор должен включить соответствующий параметр в меню **Создать**. Если вы не видите меню **Создать** на странице компонентов, обратитесь к администратору.

Разработчики могут создавать компоненты с доступом к тем же функциям и возможностям, которые предоставляются Oracle Content Management.

Создание образца локального компонента или макета:

1. Выберите **Разработчик**, а затем нажмите **Смотреть все компоненты**.
2. Нажмите **Создать** и выберите тип компонента или макета, который требуется создать.
3. Введите имя компонента или макета. Имя не должно совпадать с именем существующего компонента или макета.

Можно употреблять буквы, цифры, символы подчеркивания (_) и дефисы (-). Введенный пробел автоматически заменяется символом подчеркивания.

Не используйте для шаблонов, тем, компонентов, сайтов и страниц сайтов следующие имена: authsite, content, pages, scsTEMPLATE_*, _comps, _components, _compsdelivery, _idcSERVICE, _sitescloud, _sitesclouddelivery, _themes, _themesdelivery. Следующие имена можно использовать только для страниц сайтов (не для шаблонов, тем, компонентов и сайтов): documents, sites.

4. При необходимости добавьте описание компонента или макета.
5. Для типа компонента выберите **По умолчанию**, чтобы вставлять компоненты непосредственно на страницу, **В песочнице**, чтобы заключать компоненты во встроенную рамку (с помощью элемента iframe), или **Шаблон**, чтобы выполнять рендеринг только с помощью шаблона.

Примечание.

При выборе параметра **Шаблон** локальный компонент не будет основан на Knockout и поэтому не может использовать вложенные компоненты при размещении на странице. Например, это означает, что редактирование заголовка или основного текста должно выполняться с помощью пользовательских настроек на панели настроек компонента, а не путем выбора текста компонента, отображаемого на странице.

6. Нажмите **Создать**.
На индикаторе выполнения отображается статус создания. После создания компонента, макета или формы имя появляется в списке компонентов. Чтобы просмотреть папки и файлы в составе этого компонента или макета, выберите имя компонента или макета в списке.
7. Выбор значка компонента или макета, отличного от значка по умолчанию:
 - a. Выберите компонент или макет в списке.
 - b. Нажмите **Свойства**.
 - c. Перейдите на вкладку **Логотип компонента**.
 - d. Выберите логотип в галерее логотипов и нажмите **Готово**.

Подробную информацию о создании собственных компонентов см. в разделе [Разработка компонентов](#).

После настройки редактора компонентов, макетов, полей контента или формы контента можно предоставить общий доступ к нему другим пользователям или повысить его уровень, чтобы они могли использовать его указанными ниже способами.

- Пользовательский компонент: во время работы с Site Builder значок этого компонента добавляется на панель "Пользовательские компоненты" с именем, присвоенным компоненту.
- Макет раздела: во время работы с Site Builder значок макета раздела добавляется на панель "Макеты разделов" с именем, присвоенным макету.
- Макет контента: администратор контента может назначить макету один или несколько типов контента. Эти типы контента могут использоваться в качестве представления по умолчанию или добавляться в список макетов в Site Builder. Разработчик сайта может выбрать тип контента и указать, как элементы контента этого типа будут отображаться на странице.
- Редактор полей контента: чтобы сделать редактор полей контента доступным для использования при создании типов контента, выберите его и нажмите **Повысить уровень**, затем подтвердите выбор и нажмите **ОК**. После повышения уровня редакторы полей контента становятся доступными для управления внешним видом всех полей данных, кроме *Мультимедиа* и *Ссылка*, при создании типов контента.
- Форма контента: чтобы сделать форму контента доступной для использования при создании типов контента, выберите ее и нажмите **Повысить уровень**, затем подтвердите выбор и нажмите **ОК**. Формы контента после повышения уровня становятся доступными для использования со всеми типами контента и типами активов.

Экспорт и импорт компонентов или макетов

Экспортированный компонент можно изменить в автономном режиме, а затем импортировать как новый компонент или заменить существующий. Экспортированный компонент также можно перенести в другой экземпляр Oracle Content Management и импортировать в этом экземпляре.

Экспорт

Экспортируемый компонент копируется в папку Oracle Content Management как одиночный файл ZIP. Компонент можно выгрузить непосредственно из этой папки и распаковать для работы с отдельными файлами. После завершения обработки файлов компонента создайте файл ZIP, содержащий папки и файлы компонента, импортируйте его и замените исходный компонент или создайте новый.

Примечание.

При экспорте компонента сведения об общем доступе к компоненту не сохраняются.

Экспорт компонента:

1. Выберите **Разработчик**, а затем нажмите **Смотреть все компоненты**.
2. Выделите компонент или макет и в строке меню или контекстном меню выберите пункт **Экспорт**.
3. Перейдите в существующую папку или создайте новую с помощью кнопки **Создать**. Укажите имя папки, при необходимости добавьте описание и нажмите **Создать**. Чтобы открыть папку, нажмите на значок папки или ее имя.

4. Выберите папку, установив рядом с ней флажок, и нажмите **ОК**.

В выбранной папке создается файл с пакетом компонента или макета. Файл имеет расширение ZIP, а его имя соответствует имени компонента или макета.

Импорт

Чтобы пользователи могли импортировать или создавать редакторы компонентов, макетов или полей контента, администратор должен включить соответствующие параметры в меню **Создать**. Если вы не видите меню **Создать** на странице компонентов, обратитесь к администратору.

Чтобы импортировать редактор компонентов, макетов или полей контента, выполните указанные ниже действия.

1. Выберите **Разработчик**, а затем нажмите **Смотреть все компоненты**.
2. Нажмите **Создать** и выберите **Импортировать компонент**. Выберите соответствующий параметр для импорта компонента, макета раздела или макета содержимого.
3. Если загружен пакет компонента или макета, перейдите в соответствующую папку и откройте ее. Если пакет еще не загружен, перейдите в папку для загрузки или создайте новую. Нажмите **Загрузить**, найдите и выберите пакет компонента или макета, затем нажмите **Открыть**.
4. Установите флажок рядом с именем файла компонента или макета и нажмите **ОК**. Новый компонент или макет будет создан и добавлен в список компонентов. Если компонент или макет с таким же именем или идентификатором уже существует, отображается запрос на устранение конфликтов. В этом случае предлагается создать новый компонент или макет либо заменить существующий компонент или макет импортируемой версией.

8

Работа со страницами сайта

Давайте создадим страницу, выберем макет и расположение страницы на сайте.

- [Переход к странице](#)
- [Просмотр страниц](#)
- [Добавление страниц](#)
- [Перемещение страниц](#)
- [Удаление страниц](#)
- [Изменение настроек страницы](#)
- [Изменение макета страницы](#)
- [Изменение фона или темы](#)

Переход к странице

Чтобы просмотреть иерархию страниц перемещаться по ней, выполните указанные ниже действия.

1. Откройте сайт для редактирования. Убедитесь, что переключатель  переведен в положение "Редактировать".
2. Щелкните , чтобы отобразить возможности организации и редактирования страниц и их содержимого.
3. Нажмите  для отображения списка страниц первого уровня.
4. Чтобы просмотреть страницу, щелкните ее имя.
Страницы со стрелкой (>) имеют вложенный уровень страниц. Щелкните такую страницу, чтобы просмотреть вложенные страницы.
Для поиска необходимой страницы можно также использовать поле поиска в верхней части списка страниц.
5. Чтобы просмотреть и изменить настройки конкретной страницы, щелкните  для этой страницы. См. [Изменение настроек страницы](#).

Просмотр страниц

Чтобы просмотреть страницы на сайте, откройте сайт в редакторе и воспользуйтесь предварительным просмотром, окном просмотра и параметрами ориентации.

Чтобы изменить способ просмотра страниц во время использования редактора, выполните указанные ниже действия.

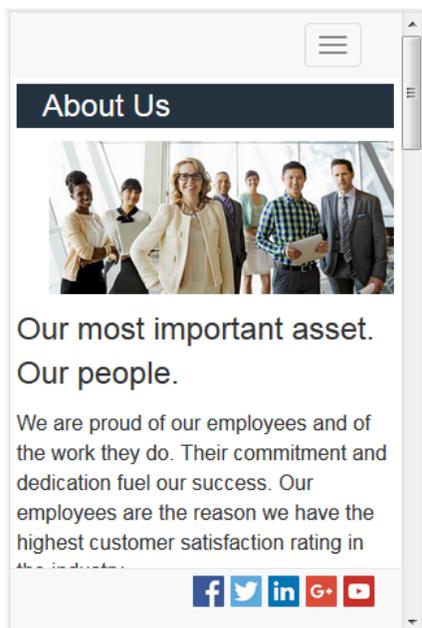
1. Откройте сайт для редактирования.

2. Перейдите к странице.
3. Для предварительного просмотра страницы в редакторе, убедитесь, что переключатель  переведен в положение "Предварительный просмотр". При этом страница отображается так, как ее видят посетители сайта: без визуальных вспомогательных элементов, используемых при редактировании.
4. Для предварительного просмотра текущего обновления, примененного к базовому сайту, в отдельном окне браузера нажмите .

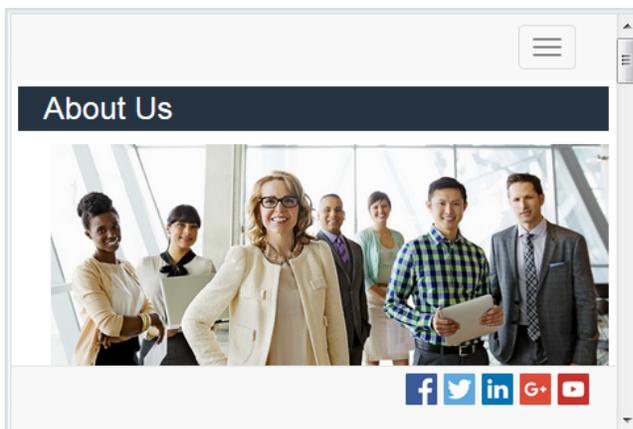
 **Примечание.**

Ссылки на страницы сайта не работают в режиме предварительного просмотра.

5. Для предварительного просмотра страницы в том виде, в каком она отображается на устройстве с конкретным размером экрана, в верхней строке меню **По размеру экрана** и выберите в списке нужные размеры экрана. При использовании тем с адаптивным дизайном содержимое страниц автоматически упорядочивается для оптимального использования на экранах выбранного размера.



6. Чтобы создать пользовательский размер, нажмите **По размеру окна** и выберите **Создать предварительную установку для устройства...** Заполните необходимые поля. Чтобы удалить пользовательский размер, нажмите  рядом с ним. Можно также активировать линейку  и выбрать на ней любой интервал для быстрого просмотра разных размеров.
7. Чтобы переключить портретную ориентацию страницы на книжную и наоборот, щелкните значок .



Добавление страниц

Структура страниц сайта, или *дерево сайта*, похожа на иерархию папок. Вы можете добавить страницу, определить ее настройки и выбрать ее место на сайте.

Можно добавить страницу или ссылку на внешнюю страницу.

- **Веб-страница:** такая страница и ее содержимое находятся на сайте. Необходимо назначить имя странице, определить содержимое, указать ее место на сайте и поведение в контексте текущего сайта.
- **Внешняя ссылка:** данная страница загружается из другого местоположения по URL-адресу. Необходимо назначить имя странице, указать ее место на сайте и поведение в контексте текущего сайта. Так как используется страница действующего сайта, ее содержимое изменить невозможно. В дереве сайта перед именем страницы, которая ссылается на внешний URL-адрес, имеется значок .

Добавление веб-страницы

Чтобы добавить к сайту веб-страницу, выполните указанные ниже действия.

1. Откройте сайт для редактирования.
2. Выберите уровень или ветвь, в которую необходимо добавить страницу, затем нажмите **Добавить страницу** или, чтобы добавить дочернюю страницу, нажмите , затем нажмите . Внизу дерева сайта добавляется запись "Новая страница" и предлагается ввести имя страницы и указать другие настройки.
3. В качестве типа страницы выберите **Веб-страница**. Этот вариант используется, если требуется управлять содержимым страницы, а не повторно использовать страницу другого сайта.
4. Присвойте странице имя. Можно употреблять буквы, цифры, символы подчеркивания () и дефисы (-). Не используйте для шаблонов, тем, компонентов, сайтов и страниц сайтов следующие имена: authsite, content, pages, scstemplate_*, _comps, _components, _compsdelivery, _idcservice, _sitescloud, _sitesclouddelivery, _themes, _themesdelivery. Следующие имена можно использовать только для страниц сайтов (не для шаблонов, тем, компонентов и сайтов): documents, sites.

5. Укажите URL-адрес страницы. По умолчанию имя страницы используется в URL-адресе страницы. Пробелы автоматически заменяются дефисами. Чтобы использовать URL-адрес, отличный от используемого по умолчанию, нажмите **Переопределить** и добавьте имя файла, используемое в URL-адресе. Можно употреблять буквы, цифры, символы подчеркивания (`_`) и дефисы (`-`). Обязательно укажите расширение файла. Расширение файла по умолчанию – `.html`
6. Выберите макет страницы. Он определяет общую структуру страницы, но не содержимое. Макет содержит одну или несколько именованных *ячеек*, куда можно поместить содержимое. Количество и тип макетов страницы зависит от темы, связанной с вашим сайтом.
7. Выберите макет страницы для мобильного устройства. Он определяет общую структуру страницы при просмотре на мобильном устройстве.
8. Укажите заголовок страницы. Он отображается не на самой странице, а в строке заголовка браузера или на вкладке браузера, в которой открыта страница. Чтобы добавить заголовок на страницу, следует в редакторе добавить компонент-заголовок.
9. Добавьте описание (необязательно). Оно не отображается на странице, но позволяет добавить сведения о странице для других соавторов или для себя.
10. Дополнительно можно указать ключевые слова, разделенные запятыми, чтобы помочь поисковым системам индексировать содержимое страницы. Ключевые слова можно использовать для идентификации терминов или концепций, которые не содержатся в тексте страницы или отображаются на изображениях. Ключевые слова не отображаются на странице, но поисковые системы используют их для поиска и идентификации сайта. Хорошие описание, ключевые слова и синонимы повышает посещаемость сайта.
11. При желании в верхний колонтитул страницы можно добавить сценарии или теги в целях анализа или отслеживания. Проверьте любой код, используемый в верхнем или нижнем колонтитуле, чтобы убедиться в том, что он правильно работает и не создает рисков безопасности для сайта.
12. При желании в нижний колонтитул страницы можно добавить сценарии или теги в целях анализа или отслеживания. По умолчанию нижний колонтитул содержит текст всплывающих окон с уведомлением об использовании на сайте файлов `cookie` и ссылкой на политику конфиденциальности. Разработчики могут отредактировать данный текст. Проверьте любой код, используемый в верхнем или нижнем колонтитуле, чтобы убедиться в том, что он правильно работает и не создает рисков безопасности для сайта.
13. Дополнительно можно использовать любой из доступных вариантов страницы.
 - **Страница ошибки:** если запрошенная страница не может отобразиться из-за ошибки, эта страница ошибки отображается вместо страницы ошибки по умолчанию. По умолчанию при выборе этого параметра также выбирается параметр **Скрыть страницу в навигации** и все параметры механизма поиска. Можно отменить выбор параметров, если необходимо включить страницу ошибки в навигацию или в результаты механизма поиска.
 - **Скрыть страницу из системы навигации:** данная страница не включается в автоматически генерируемую систему навигации по сайту.
 - **Страница сведений.** Используется для отображения подробных сведений для выбранного на другой странице элемента контента.

Структурированное содержимое доступно только для корпоративных пользователей. Если выбран этот параметр, по умолчанию выбирается параметр **Скрыть страницу в навигации**. Если вы хотите, чтобы страница сведений отображалась в навигации, снимите этот флажок. Страница сведений открывается, когда для элементов структурированного содержимого настроено отображение сведений и пользователь переходит по ссылке для просмотра подробной информации об элементе.

- **Страница поиска.** На этой странице отображаются результаты поиска. Эту страницу можно выбрать в настройках ссылки компонента поиска содержимого. См. [Поиск в содержимом](#). Если выбран этот параметр, по умолчанию выбирается параметр **Скрыть страницу в навигации**. Можно отменить выбор параметра, если необходимо включить страницу поиска в навигацию.
14. Дополнительно можно выбрать вариант поисковой оптимизации (SEO):
- **Скрыть страницу от поисковых систем:** поисковым системам предписывается не индексировать данную страницу, чтобы она не появлялась в результатах поиска.
 - **Скрыть ссылки страницы от поисковых систем:** поисковым системам предписывается не переходить по ссылкам на странице и не индексировать места назначения ссылок.
 - **Отключить кэширование страниц механизма поиска.** Уведомляет механизмы поиска об отсутствии необходимости кэширования данной страницы.
 - **Скрыть описание страницы от поисковых систем:** поисковым системам предписывается не включать описание (указанное выше) после страницы в результатах поиска.
15. Дополнительно можно указать, будет ли на странице отображаться кнопка совместного просмотра во время сеансов просмотра. Сеанс совместного просмотра использует Oracle Cobrowse Cloud Service для управления совместным использованием экрана с посетителем сайта. См. [Включение интеграции с Cobrowse](#) и [Использование компонента "Совместный просмотр" на странице](#).
16. Дополнительно. Заменяет фрагмент отслеживания аналитики сайта и добавляет характерный для этой страницы фрагмент. Добавление фрагмента отслеживающего кода JavaScript для отслеживания веб-аналитики упрощает интеграцию с внешними поставщиками аналитики, такими как Google, Adobe или Oracle Infinity. См. [Добавление отслеживания аналитики](#).
17. Чтобы сохранить отложенные изменения в текущую тему, нажмите кнопку **Сохранить**.

Добавление связанной страницы

Чтобы добавить к сайту страницу внешней ссылки, выполните указанные ниже действия.

1. Откройте сайт для редактирования.
2. Перейдите к странице на том уровне дерева сайта, на который необходимо добавить страницу. Страницы добавляются на текущий уровень, но их легко можно переместить в другое расположение в дереве сайтов.
3. Нажмите **Добавить страницу**. Внизу дерева сайта добавляется запись "Новая страница" и предлагается ввести имя страницы и указать другие настройки.
4. В качестве типа страницы выберите **Внешняя ссылка**.

5. Присвойте странице имя. Можно употреблять буквы, цифры, символы подчеркивания (_) и дефисы (-). Данное имя используется в дереве сайта для идентификации страницы. Пробелы автоматически заменяются дефисами.
6. Укажите полный URL-адрес страницы. Например: `https://www.example.com/sharedpage.htm`
7. Чтобы страница открывалась в новом окне или вкладке браузера, установите флажок **Открыть ссылку в новом окне**. Если этот флажок не установлен, страница открывается в текущем окне, заменяя страницу и элементы навигации вашего сайта.
8. Дополнительно можно выбрать любой из доступных вариантов.
 - **Страница ошибки:** если запрошенная страница не может отображаться из-за ошибки, эта страница ошибки отображается вместо страницы ошибки по умолчанию. Если выбран этот параметр, по умолчанию выбирается параметр **Скрыть страницу в навигации**. Можно отменить выбор параметра, если необходимо включить страницу ошибки в навигацию.
 - **Скрыть страницу из системы навигации:** данная страница не включается в автоматически генерируемую систему навигации по сайту.
 - **Страница сведений.** Используется для отображения подробных сведений для выбранного на другой странице элемента контента. Структурированное содержимое доступно только для корпоративных пользователей. Если выбран этот параметр, по умолчанию выбирается параметр **Скрыть страницу в навигации**. Если вы хотите, чтобы страница сведений отображалась в навигации, снимите этот флажок. Страница сведений открывается, когда для элементов структурированного содержимого настроено отображение сведений и пользователь переходит по ссылке для просмотра подробной информации об элементе.
 - **Страница поиска.** На этой странице отображаются результаты поиска. Эту страницу можно выбрать в настройках ссылки компонента поиска содержимого. См. [Поиск в содержимом](#). Если выбран этот параметр, по умолчанию выбирается параметр **Скрыть страницу в навигации**. Можно отменить выбор параметра, если необходимо включить страницу поиска в навигацию.
9. Чтобы сохранить отложенные изменения в текущую тему, нажмите кнопку **Сохранить**.

Перемещение страниц

Страницы можно перетащить в другое место на текущем уровне дерева сайта или вырезать и вставить на другом уровне.

В иерархии всегда есть верхняя страница, главная страница сайта, под которой перечислены все остальные страницы. Вы можете перетащить страницы на заданный уровень, чтобы изменить их порядок. Также можно вырезать страницы и вставить их в другое место иерархии.

Страницы могут иметь вложенные страницы. При перемещении страницы вложенные в нее страницы перемещаются вместе с ней, сохраняя вложение. Если используемая тема содержит автоматические средства навигации, при

изменении порядка страниц также меняется порядок их перечисления в меню навигации.

Чтобы переместить страницу в другое место в дереве сайта, выполните следующие действия.

1. Откройте сайт для редактирования.
2. Перейдите на страницу, которую нужно переместить.
3. Чтобы переместить страницу в другое место на текущем уровне дерева сайта, щелкните и перетащите имя страницы в другое место в списке страниц.
4. Чтобы переместить страницу на другой уровень в дереве сайта, нажмите на страницу и нажмите  для нее. Нажмите , чтобы вырезать страницу из текущего расположения, или , чтобы оставить копию страницы в текущем расположении. Перейдите на уровень, на который нужно переместить страницу, и нажмите . Если выбрать страницу на том же уровне, то вставленная страница будет размещена под этим уровнем. Щелкните , чтобы вставить страницу в текущее местоположение. Чтобы изменить порядок страниц, перетащите страницу в другое место на текущем уровне дерева сайта.

Удаление страниц

Вы можете удалять страницы как часть обновления.

Удаление страницы регистрируется как часть обновления при нажатии кнопки **Сохранить**. Страница не удаляется с основного сайта до тех пор, пока не будет опубликовано обновление. Страницы могут удалять все пользователи с ролями "Соавтор" или "Управление".

Примечание.

При публикации обновления страница удаляется без возможности восстановления. Если вы удалите страницу с многоязычного сайта, переведенная версия страницы также будет удалена при публикации сайта. Если вам сложно контролировать удаление или добавление страниц сайта, рекомендуется изолировать эти действия с помощью специальных обновлений.

Чтобы удалить страницу из сайта, выполните указанные ниже действия.

1. Откройте сайт для редактирования.
2. Перейдите к странице, которую хотите удалить, и нажмите .
3. Для подтверждения удаления нажмите кнопку **ОК**. Нажмите **Сохранить**, чтобы сохранить отложенные изменения в текущем обновлении.

Изменение настроек страницы

Можно изменить такие настройки страницы, как имя, заголовок, URL-адрес, верхние колонтитулы, нижние колонтитулы и другие параметры.

1. Откройте сайт для редактирования.
2. Нажмите на страницу, для которой необходимо изменить настройки, затем нажмите на .
3. Измените необходимые настройки:
 - **Тип страницы:**
 - **Веб-страница:** такая страница и ее содержимое находятся на сайте. Необходимо назначить имя странице, определить содержимое, указать ее место на сайте и поведение в контексте текущего сайта.
 - **Внешняя ссылка:** данная страница загружается из другого местоположения по URL-адресу. Необходимо назначить имя странице, указать ее место на сайте и поведение в контексте текущего сайта. Так как используется страница действующего сайта, ее содержимое изменить невозможно. В дереве сайта перед именем страницы, которая ссылается на внешний URL-адрес, имеется значок .

При выборе "Веб-страница" отображаются следующие настройки:

- **Имя страницы.** Можно использовать буквы, цифры, символы подчеркивания (_) и дефисы (-). Не используйте для шаблонов, тем, компонентов, сайтов и страниц сайтов следующие имена: authsite, content, pages, scsTEMPLATE_*, _comps, _components, _compsdelivery, _idcSERVICE, _sitescloud, _sitesclouddelivery, _themes, _themesdelivery. Следующие имена можно использовать только для страниц сайтов (не для шаблонов, тем, компонентов и сайтов): documents, sites.
- **URL-адрес страницы.** По умолчанию имя страницы используется в URL-адресе страницы. Пробелы автоматически заменяются дефисами. Чтобы использовать URL-адрес, отличный от используемого по умолчанию, нажмите **Переопределить** и добавьте имя файла, используемое в URL-адресе. Можно употреблять буквы, цифры, символы подчеркивания (_) и дефисы (-). Обязательно укажите расширение файла. Расширение файла по умолчанию – .html
- **Макет страницы.** Макет страницы определяет общую структуру страницы, но не контента. Макет содержит одну или несколько именованных *ячеек*, куда можно поместить содержимое. Количество и тип макетов страницы зависит от темы, связанной с вашим сайтом.
- **Макет страницы для мобильных устройств.** Макет страницы для мобильных устройств определяет общую структуру страницы при просмотре на мобильном устройстве.
- **Заголовок страницы.** Заголовок страницы отображается не на самой странице, а в строке заголовка браузера или на его вкладке, на которой открыта страница. Чтобы добавить заголовок на страницу, следует в редакторе добавить компонент-заголовок.

- **Описание страницы.** Описание не отображается на странице, но позволяет добавить сведения о странице для других соавторов или для себя.
- **Ключевые слова.** Дополнительно можно указать ключевые слова, разделенные запятыми, чтобы помочь поисковым системам индексировать контент страницы. Ключевые слова можно использовать для идентификации терминов или концепций, которые не содержатся в тексте страницы или отображаются на изображениях. Ключевые слова не отображаются на странице, но поисковые системы используют их для поиска и идентификации сайта. Хорошее описание, ключевые слова и синонимы повышает посещаемость сайта.
- **Верхний колонтитул страницы.** В верхний колонтитул страницы можно добавить сценарии или теги в целях анализа или отслеживания. Проверьте любой код, используемый в верхнем или нижнем колонтитуле, чтобы убедиться в том, что он правильно работает и не создает рисков безопасности для сайта.
- **Нижний колонтитул страницы.** В нижний колонтитул страницы можно добавить сценарии или теги в целях анализа или отслеживания. По умолчанию нижний колонтитул содержит текст всплывающих окон с уведомлением об использовании на сайте файлов cookie и ссылкой на политику конфиденциальности. Разработчики могут отредактировать данный текст. Проверьте любой код, используемый в верхнем или нижнем колонтитуле, чтобы убедиться в том, что он правильно работает и не создает рисков безопасности для сайта.
- **Параметры страницы:**
 - **Страница ошибки:** если запрошенная страница не может отобразиться из-за ошибки, эта страница ошибки отображается вместо страницы ошибки по умолчанию. По умолчанию при выборе этого параметра также выбирается параметр **Скрыть страницу в навигации** и все параметры механизма поиска. Можно отменить выбор параметров, если необходимо включить страницу ошибки в навигацию или в результаты механизма поиска.
 - **Скрыть страницу из системы навигации:** данная страница не включается в автоматически генерируемую систему навигации по сайту.
 - **Страница сведений.** Используется для отображения подробных сведений для выбранного на другой странице элемента контента. Структурированное содержимое доступно только для корпоративных пользователей. Если выбран этот параметр, по умолчанию выбирается параметр **Скрыть страницу в навигации**. Если вы хотите, чтобы страница сведений отображалась в навигации, снимите этот флажок. Страница сведений открывается, когда для элементов структурированного содержимого настроено отображение сведений и пользователь переходит по ссылке для просмотра подробной информации об элементе.
 - **Страница поиска.** На этой странице отображаются результаты поиска. Эту страницу можно выбрать в настройках ссылки компонента поиска содержимого. См. [Поиск в содержимом](#). Если выбран этот параметр, по умолчанию выбирается параметр **Скрыть страницу в навигации**. Можно отменить выбор параметра, если необходимо включить страницу поиска в навигацию.
- **Параметры SEO:**
 - **Скрыть страницу от поисковых систем:** поисковым системам предписывается не индексировать данную страницу, чтобы она не появлялась в результатах поиска.

- **Скрыть ссылки страницы от поисковых систем:** поисковым системам предписывается не переходить по ссылкам на странице и не индексировать места назначения ссылок.
- **Отключить кэширование страниц механизма поиска.** Уведомляет механизмы поиска об отсутствии необходимости кэширования данной страницы.
- **Скрыть описание страницы от поисковых систем:** поисковым системам предписывается не включать описание (указанное выше) после страницы в результатах поиска.
- **Совместный просмотр.** Если администратор включил функцию совместного просмотра для вашей системы и он включен для этого сайта, отобразится параметр выбора при условии, что на странице будет представлена кнопка "Совместный просмотр" для использования в сеансах просмотра. Сеанс совместного просмотра использует Oracle Cobrowse Cloud Service для управления совместным использованием экрана с посетителем сайта. См. разделы [Включение интеграции с Cobrowse](#) и [Использование компонента "Совместный просмотр" на странице](#).
- **Параметры аналитики.** Если данный сайт содержит фрагмент отслеживания аналитики, его можно переопределить для этой страницы. Нажмите переключатель переопределения и измените сценарий. Подробнее см. в разделе [Добавление отслеживания аналитики](#) После внесения изменений во фрагмент сайта, отобразится сообщение об изменении сценария. Чтобы удалить настройки, нажмите **Восстановить последний сценарий сайта**.

При выборе "Внешняя ссылка" отображаются следующие настройки:

- **Имя страницы.** Можно использовать буквы, цифры, символы подчеркивания (_) и дефисы (-). Данное имя используется в дереве сайта для идентификации страницы. Пробелы автоматически заменяются дефисами.
- **URL-адрес ссылки.** Указывает полный URL-адрес страницы. Например: `https://www.example.com/sharedpage.htm`.
- **Открыть ссылку в новом окне.** Если необходимо открыть страницу в новом окне или вкладке браузера, выберите этот параметр. Если этот флажок не установлен, страница открывается в текущем окне, заменяя страницу и элементы навигации вашего сайта.
- **Параметры страницы:**
 - **Страница ошибки:** если запрошенная страница не может отобразиться из-за ошибки, эта страница ошибки отображается вместо страницы ошибки по умолчанию. Если выбран этот параметр, по умолчанию выбирается параметр **Скрыть страницу в навигации**. Можно отменить выбор параметра, если необходимо включить страницу ошибки в навигацию.
 - **Скрыть страницу из системы навигации:** данная страница не включается в автоматически генерируемую систему навигации по сайту.
 - **Страница сведений.** Используется для отображения подробных сведений для выбранного на другой странице элемента контента. Структурированное содержимое доступно только для корпоративных

пользователей. Если выбран этот параметр, по умолчанию выбирается параметр **Скрыть страницу в навигации**. Если вы хотите, чтобы страница сведений отображалась в навигации, снимите этот флажок.

Страница сведений открывается, когда для элементов структурированного содержимого настроено отображение сведений и пользователь переходит по ссылке для просмотра подробной информации об элементе.

- **Страница поиска.** На этой странице отображаются результаты поиска. Эту страницу можно выбрать в настройках ссылки компонента поиска содержимого. См. [Поиск в содержимом](#). Если выбран этот параметр, по умолчанию выбирается параметр **Скрыть страницу в навигации**. Можно отменить выбор параметра, если необходимо включить страницу поиска в навигацию.

4. Чтобы сохранить отложенные изменения в текущую тему, нажмите кнопку **Сохранить**.

Изменение макета страницы

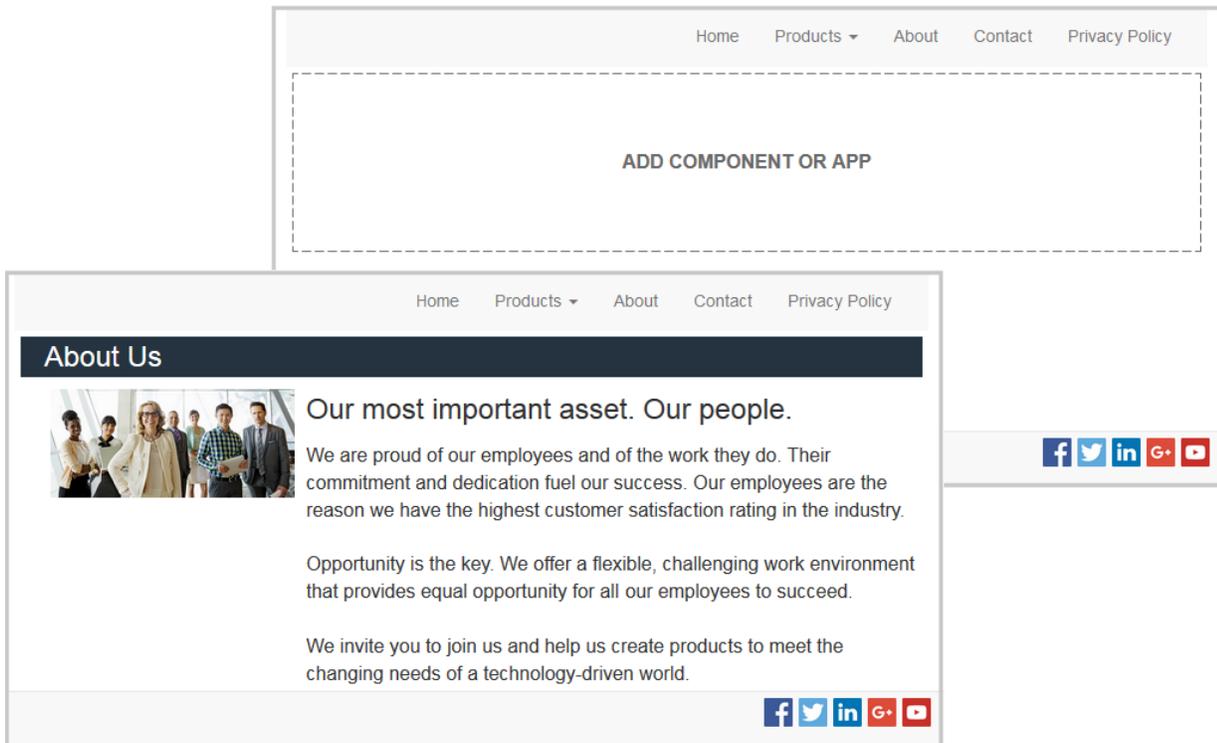
Макет определяет организацию содержимого на странице. Разные макеты могут содержать разное количество именованных *ячеек*, которые представляют собой области, занимающие всю ширину страницы. Ячейка может вмещать содержимое одного или нескольких типов.

В каждой теме есть несколько макетов страниц. При добавлении страницы на сайт требуется выбрать ее макет. Каждый макет содержит ячейки, в которые можно перетащить содержимое. Вы сами определяете содержимое ячеек. Это может быть все что угодно - от заголовков, текста и разделителей до мультимедиа, галерей и социальных сетей. Можно упорядочить содержимое в ячейке, но нельзя изменить количество или компоновку ячеек на странице. Для этого необходимо выбрать новый макет страницы.

Примечание.

Текущий макет можно заменить другим. Однако при этом следует соблюдать осторожность. Если выбрать макет с меньшим количеством ячеек или по-другому именованными ячейками, существующее содержимое в других ячейках не будет отображаться в новом макете. Содержимое не удаляется, оно просто не будет отображаться, пока у выбранного макета не будет ячейки с таким же именем.

Пример макета для страницы показан ниже. На рисунке отображаются пустая ячейка макета страницы и заполненная страница с добавленными в ячейку заголовком, изображением и текстом.



Чтобы выбрать другой макет страницы, выполните указанные ниже действия.

1. Откройте сайт для редактирования.
2. Перейдите к странице и нажмите .
3. Перейдите к полю "Макет страницы" и в меню выберите другой макет. Количество и тип макетов страницы зависит от темы, связанной с вашим сайтом.
4. Чтобы сохранить отложенные изменения в текущую тему, нажмите кнопку **Сохранить**.

Изменение фона или темы

Цвет и изображение фона можно указать для страницы, отдельных ячеек страницы и макетов разделов в ячейке. Вы также можете изменить тему всего сайта.

Ячейка – это область, которая занимает всю ширину страницы и может вмещать содержимое одного или нескольких типов. Указанный для ячейки фон применяется ко всей ячейке и всем компонентам в ней.

Изменение фона

Фоны накладываются друг на друга. Если указать фон для ячейки, он накладывается на фон, заданный для страницы. Для большинства изображений и цветов верхний уровень фактически переопределяет нижний уровень. Если к цвету или изображениям фона применена частичная прозрачность, цвета более низких слоев могут проступить или смешиваться с цветами на верхних уровнях.

Некоторые компоненты, например абзацы и заголовки, могут определять цвет фона в рамках предварительно определенного или настраиваемого стиля. Если для компонента указан цвет фона, он также накладывается на фон макета разделов, ячеек или страницы. См. [Использование стилей и форматирование](#).

Страницы, ячейки и макеты разделов имеют сходные параметры фона, но способы доступа к этим параметрам немного различаются.

1. Откройте сайт для редактирования.
2. Перейдите к странице, которую хотите изменить, и убедитесь, что переключатель  переведен в положение **Редактировать**.
3. Изменение фона страницы или ячейки:
 - Нажмите  на боковой панели, а затем – .
 - Чтобы указать фон для всей страницы, нажмите **Вся страница** и . Чтобы указать фон для ячейки на странице, выделите ячейку и нажмите .
4. Изменение фона макета разделов:
 - Нажмите на значок меню  для макета разделов и выберите **Настройки**.
 - Нажмите **Фон**. Страницы, ячейки и макеты разделов имеют одинаковые параметры фона.
5. Чтобы использовать изображение в качестве фона, щелкните **Выбрать изображение**. Можно использовать изображения из любого расположения, к которому у вас есть доступ. Можно также использовать изображения, к которым вам предоставлен доступ или которые вы загрузили из локального или сетевого расположения файлов.
6. Выберите изображение и нажмите **ОК**. В данном окне отображаются все доступные файлы. Выберите тип файла, соответствующий контексту. Например, выбирая файл изображения, необходимо выбрать допустимый формат изображения (GIF, JPG, JPEG, PNG или SVG).
7. Чтобы изменить настройки изображения, выполните указанные ниже действия.
 - С помощью инструмента **Положение** разместите изображение на странице или в ячейке.
 - Выберите инструмент **Масштаб**, чтобы настроить представление изображений.
 - **Вписать**: изображение масштабируется так, чтобы оно целиком вписалось в доступное пространство без искажения.
 - **Растянуть**: наибольшее из двух измерений (ширина или высота) масштабируется, чтобы вписать изображение в доступное пространство, а наименьшее измерение растягивается до заполнения доступного пространства.
 - **Нет**: изображение используется с полным разрешением в пикселях и равномерно обрезается, если оно не вписывается в доступное пространство.
 - Выберите **Повторить**, чтобы создать мозаику из изображения, которое не заполняет доступное пространство. Этот вариант не применим, если для параметра **Масштаб** выбрано значение **Растянуть**.

- Установите флажок **Не прокручивать вместе со страницей**, чтобы изображение оставалось в покое, пока пользователь прокручивает содержимое страницы.
8. Чтобы выбрать цвет фона, выберите цвет в меню **Цвет** или нажмите кнопку **Дополнительно**, чтобы выбрать цвет из всего диапазона доступных цветов.
- Щелкните на панели спектра справа, чтобы выбрать цвет и отобразить вариации цвета на экране диапазона цветов. В правом верхнем углу отображается выбранный цвет и его оттенки с переходом в белый по направлению влево и переходом в черный по направлению вниз.
 - Чтобы выбрать вариацию цвета, щелкните на экране диапазона цветов или щелкните и перетащите точку выбора в положение нового цвета. Выбранный цвет отображается в палитре и с помощью шестизначного кода цвета.
 - Чтобы настроить прозрачность (альфа-канал), щелкните и перетащите ползунок настройки прозрачности влево, чтобы увеличить прозрачность. Альфа-значение 0 % соответствует прозрачному цвету. Это позволяет полностью видеть цвет и содержимое нижних уровней. Альфа-значение 100 % соответствует полностью непрозрачному цвету. Все цвета и все содержимое нижних уровней закрыты. При других значениях изображения и цвета из более низких уровней проступают. Их цвета в разной степени смешиваются с цветами верхних уровней.
 - Чтобы применить текущий выбор цвета, нажмите кнопку **Выбрать**.

Изменение темы

Изменение темы регистрируется как часть обновления при нажатии кнопки **Сохранить**. Измененная тема не отображается на сайте до тех пор, пока обновление не будет опубликовано.

Примечание.

Не все темы совместимы друг с другом. Разные темы могут содержать разные элементы навигации, имена стилей и сведения о макете страницы. Например, если выбрана тема с другими названиями макетов, то существующие страницы могут использовать макеты, которые недоступны в теме. Такие страницы будут пустыми. Содержимое страниц не удаляется, но не отображается, если в выбранной теме отсутствуют макет и ячейки с такими же именами. Если выбрана несовместимая тема, можно вернуть исходную тему и восстановить содержимое страниц. Убедитесь в том, что переключаетесь на совместимую тему. Если вы не уверены, обратитесь за консультацией к разработчику темы.

Чтобы изменить используемую для сайта тему, выполните следующие действия.

1. Откройте сайт для редактирования.
2. Нажмите  на боковой панели, а затем – .

3. Выберите тему из списка доступных тем и нажмите кнопку **ОК**. Выбранная тема применяется к текущему обновлению. Чтобы сохранить отложенные изменения в текущую тему, нажмите кнопку **Сохранить**.

Организация содержимого страницы

Компоненты и элементы содержимого предоставляют пользователям нужные функции и данные, а макеты разделов позволяют автоматически структурировать эти элементы.

- [Добавление компонентов и макетов разделов](#)
- [Работа с активами и элементами содержимого](#)
- [Использование триггеров и действий](#)
- [Использование горизонтальных макетов разделов](#)
- [Использование вертикальных макетов разделов](#)
- [Использование макетов с двумя и тремя столбцами](#)
- [Использование макетов разделов с вкладками](#)
- [Использование макетов разделов со слайдами](#)

Сведения об использовании отдельных компонентов см. в разделе [Использование встроенных компонентов](#).

Сведения об управлении компонентами и макетами см. в разделе [Управление пользовательскими компонентами и макетами](#).

Подробную информацию о создании собственных компонентов, см. в разделе [Разработка компонентов](#).

Добавление компонентов и макетов разделов

Компоненты — это отдельные части веб-страницы. *Макет разделов* автоматически организует добавляемое в него содержимое, что позволяет пользователям, которым назначена роль "Соавтор", легко добавлять содержимое, не тратя время на его форматирование на странице.

Добавление компонентов

Добавление компонента на страницу:

1. Перейдите к странице, которую хотите изменить, и убедитесь, что переключатель  переведен в положение **Редактировать**.
2. Нажмите , а затем выберите один из следующих типов компонентов:
 - Нажмите **Тематические**, чтобы отобразить список компонентов, которые были выбраны для темы, связанной с сайтом.
 - Нажмите **Пользовательские**, чтобы отобразить список пользовательских компонентов, к которым предоставлен общий доступ.
 - Нажмите **Заданные предварительно**, чтобы отобразить список компонентов по умолчанию, доступные для сервиса.

- Нажмите **Все**, чтобы отобразить все компоненты, к которым вам предоставлен доступ.
3. Нажмите и удерживайте определенный значок, затем перетащите его в нужное место на странице. При перетаскивании элемента на страницу отображаются границы доступных ячеек, макетов разделов и существующих элементов. Значок размещения  или вертикальная черточка  (поместите ее слева или справа) указывают место размещения нового содержимого. Сплошная граница вокруг макета раздела или элемента содержимого означает, что сюда можно перетащить элемент, который будет размещен автоматически. В одной ячейке может быть несколько элементов. Элементы можно перемещать по странице, просто перетаскивая в новое место.
 4. Перетащив элемент в нужное место на странице, отпустите кнопку мыши.
 5. Чтобы настроить свойства элемента, щелкните его значок меню  и выберите пункт **Настройки**. В зависимости от элемента необходимо добавить текст, указать ссылку на изображение, URL-адрес на другой сайт или карту, изменить размер элемента или выполнить другие действия.

У каждого компонента есть настройки, например размер, выравнивание, промежутки, цвет и границы. Эти настройки определяют внешний вид и поведение компонента. Например, настройки абзаца включают тип и размер шрифта, а также определяют представление текста.

Для настройки свойств компонента выберите нужный компонент, нажмите  **Меню** и выберите **Настройки**. Настройки различаются в зависимости от компонента.

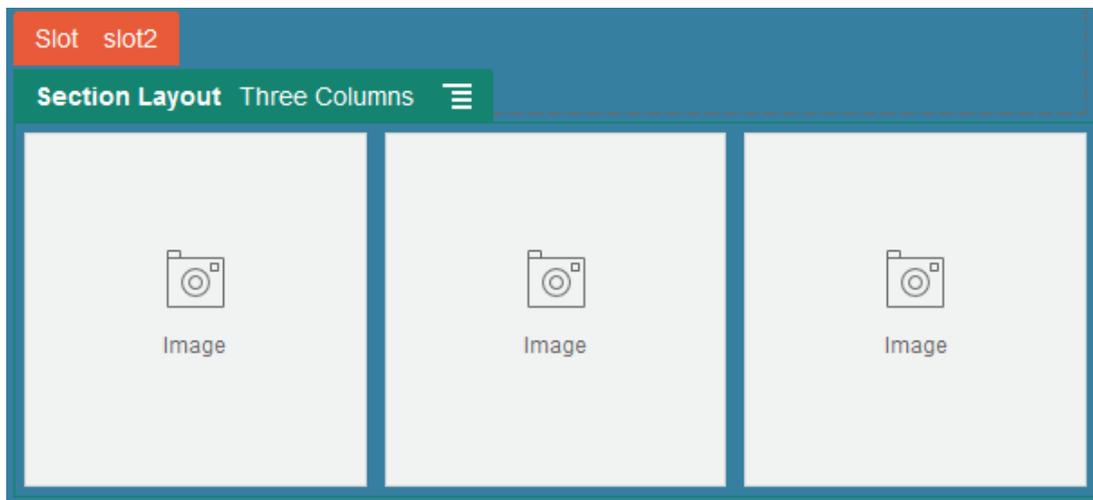
Настройки	Описание
Общие	К общим настройкам относятся интервал, выравнивание и другие настройки, общие для компонентов.
Стиль	Стиль представляет собой именованный набор стандартных значений, определяющих внешний вид. Стили определяются темой. Различные темы могут включать различные стили. Точно так же внутри каждой темы различные стили могут применяться к различным компонентам. Можно вручную задать настройки стиля для определенного экземпляра компонента.
Ссылка	К настройкам ссылки относятся расположения изображений, документов и других ресурсов, используемых компонентом. В случае с кнопками и другими компонентами, поддерживающими выполнение действий, настройки ссылки также включают триггеры и действия, поддерживаемые компонентом.
Компоненты	Компоненты, которые включают в себя другие стандартные компоненты (например, компонент статьи), предоставляют список отдельных компонентов и доступ к настройкам каждого из этих компонентов.

Настройки	Описание
Пользовательские	Пользовательские настройки уникальны для определенных компонентов и размещаются отдельно от стандартных групп настроек. Например, параметры удаленных компонентов могут храниться в расположении с уникальным URL-адресом.

Использование макетов разделов

Макет раздела может автоматически организовать содержимое в несколько столбцов, вертикальный список или набор областей с вкладками. Пользователь с ролью "Соавтор" может добавить в ячейку один или несколько макетов разделов для структурирования содержимого. Добавление макета на страницу:

1. Перейдите к странице, которую хотите изменить, и убедитесь, что переключатель  переведен в положение **Редактировать**.
2. Нажмите , чтобы вставить макет раздела на страницу. Выберите тип макета, который необходимо использовать, и для размещения элемента следуйте инструкциям по размещению компонента.



Значок размещения  указывает место размещения раздела (выше, ниже, слева, справа). Сплошная рамка вокруг макета раздела или элемента содержимого показывает, что сюда можно перетащить элемент, который будет размещен автоматически:



Если вам назначена роль "корпоративный пользователь" и вы работаете со структурированными элементами содержимого, можно назначить макет содержимого во

время добавления списка элементов содержимого на страницу, и этот макет содержимого автоматически отформатирует элементы на странице.

Помимо готовых стандартных макетов разработчики могут создать дополнительные макеты разделов, чтобы устранить определенные проблемы или упростить работу с содержимым. Разработчики тем также могут встраивать макеты разделов в ячейку макета страницы. См. [Разработка макетов](#).

Работа с активами и элементами содержимого

Сайт корпоративных пользователей может содержать цифровые активы и элементы контента, хранящиеся в *коллекции сайта* или связанном репозитории. Коллекция представляет собой подмножество активов в репозитории, которые можно использовать на сайте.

Чтобы добавить на страницу цифровой актив или элемент контента, убедитесь,

что переключатель  переведен в положение **Редактировать**, и нажмите



. Если для вашего сайта используется несколько репозиториях, выберите репозиторий, который будет использоваться. Сведения об использовании нескольких репозиториях на узле см. Предоставление доступа к нескольким репозиториям на уровне узла в разделе *Управление активами с помощью Oracle Content Management*.

Можно применить фильтр, чтобы найти необходимые активы. Нажмите  и выберите критерии. Если активы не отображаются, возможно, они не входят в коллекцию сайта. Измените фильтр, чтобы отобразить все коллекции в репозитории, а не только коллекцию сайтов, чтобы узнать, помогает ли это решить проблему. Для получения подробных сведений см. Поиск, фильтрация и сортировка активов.

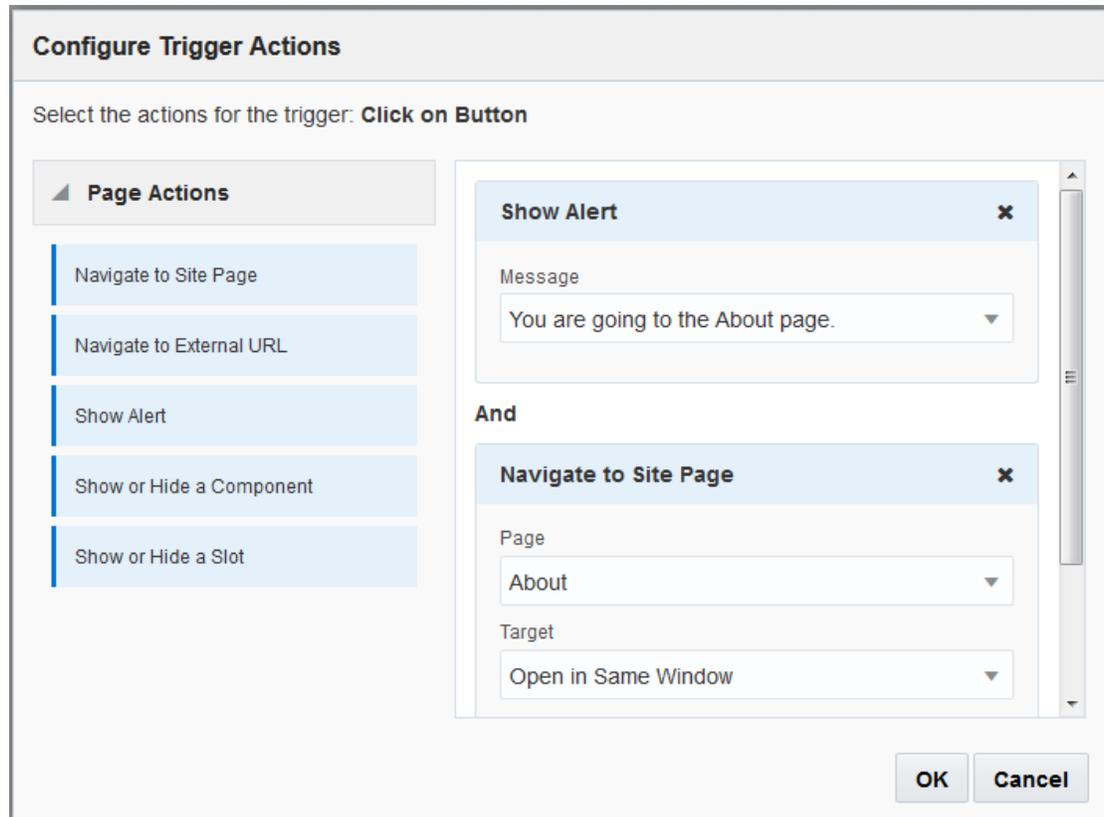
Перетащите цифровой актив или элемент содержимого с панели в ячейку на странице. Изображения можно внедрить непосредственно в компонент абзаца в месте курсора. При этом доступны такие параметры, как ввод альтернативного текста, настройка высоты, ширины и выравнивания изображения.

Элемент, добавленный из коллекции сайта на страницу, автоматически размещается в соответствующем компоненте. Например, добавленный цифровой актив изображения автоматически размещается в компоненте изображения. Добавленный элемент содержимого автоматически размещается в компоненте элемента содержимого. Также можно сначала добавить компонент изображения или элемента содержимого, а затем перетащить элемент содержимого с панели "Содержимое" в этот компонент. Или можно выбрать **Настройки** для компонента, а затем нажать **Выбрать** для выбора изображения в папке активов или списке документов.

Использование триггеров и действий

Компоненты-кнопки могут инициировать одно или несколько действий, например, отображение или скрытие компонентов-страниц и отображение сообщений. Определенные компоненты (например, списки папок и файлов) могут инициировать действия в сопутствующем компоненте в зависимости от выбранных пользователем параметров.

Например, можно настроить кнопку таким образом, чтобы при нажатии кнопки (триггер) пользователь перенаправлялся на другую страницу или внешний URL-адрес и отображалось оповещение, уведомляющее пользователя об изменении (действия). Отображение изменяется в зависимости от используемых компонентов.



При использовании нескольких действий следует учитывать порядок выполнения, выбирая действия в нужном порядке. В приведенном примере сначала необходимо указать действие оповещения. Таким образом у пользователя будет время прочитать и закрыть сообщение, прежде чем он будет перенаправлен на страницу. Если сначала указать действие перенаправления, новая страница может открыться раньше, чем пользователь сможет прочитать сообщение.

Действия со страницами

Все компоненты, поддерживающие триггеры и действия, поддерживают следующие действия со страницами:

- **Перейти на страницу сайта:** выберите страницу текущего сайта.
- **Перейти по внешнему URL-адресу:** укажите полный URL-адрес на внешнюю страницу или сайт.
- **Показать предупреждение:** отображение в окне указанного сообщения.
- **Показать или скрыть компонент:** из списка компонентов на текущей странице выберите компонент, который требуется отобразить, скрыть или переключить.
- **Показать или скрыть ячейку:** из списка ячеек на текущей странице выберите ячейку, которую требуется отобразить, скрыть или переключить.

Действия с компонентами

Помимо действий со страницами компоненты могут определять собственные действия. Можно разрешить компоненту обмениваться данными с другими компонентами и инициировать действия в сопутствующих компонентах. Например, когда пользователь выбирает папку в компоненте "Список папок", компонент "Список файлов" может отображать файлы в выбранной папке. В этом случае компонент "Список папок" поддерживает триггер **Выбранная папка**, а компонент "Список файлов" поддерживает действие **Показать файлы**.

При добавлении на страницу компонентов, поддерживающих действия, эти компоненты и все поддерживаемые действия включаются в список доступных действий. Некоторые компоненты поддерживают только действия или определенные действия. Некоторые компоненты предоставляют триггеры, но при этом не поддерживают действия.

Сведения о создании собственных компонентов см. в разделе [Разработка компонентов](#).

Например, чтобы указать одно или несколько действий для компонента кнопки.

1. Щелкните значок меню кнопки  и выберите пункт **Настройки**.
2. На панели "Настройки" щелкните **Ссылка**.
3. Щелкните **Выбрать тип ссылки** и выберите **Действия триггера**.
4. В разделе "Действия триггера" щелкните **Нажать на кнопку**.
5. В окне "Настройка действия триггера" щелкните и перетащите название действия из столбца слева в ячейку с меткой **Сделать что-то**.

Использование горизонтальных макетов разделов

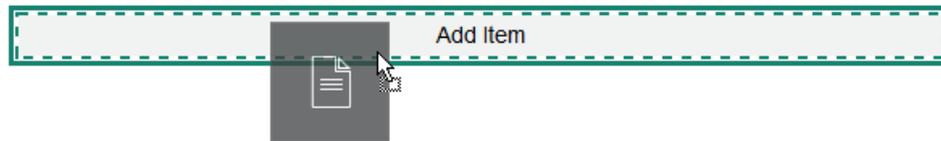
Макет раздела автоматически определяет интервалы и расположение добавленных в него компонентов.

Элементы, добавленные в горизонтальный макет, располагаются горизонтально друг за другом. Пропорции макета меняются при уменьшении или увеличении ширины страницы. По умолчанию элементы укладываются в одной строке, и каждому из них выделяется одинаковое пространство по горизонтали.

Добавление макета на страницу:

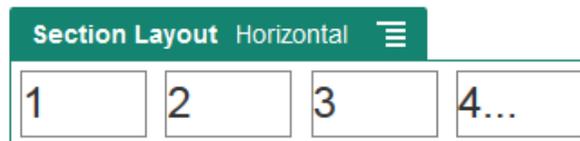
1. Перейдите к странице, которую хотите изменить, и убедитесь, что переключатель  переведен в положение **Редактировать**.
2. Добавьте макет на страницу.
3. Чтобы поместить другое содержимое в макет раздела, перетащите содержимое в макет.

Макет выделяется сплошной рамкой, и на нем появляется баннер с надписью "Добавить элемент".



Можно добавить в макет дополнительные элементы, и к ним будет автоматически применено соответствующее форматирование. Для создания более сложных макетов можно добавить другие макеты разделов.

Ниже представлен горизонтальный макет с пронумерованными текстовыми компонентами, отражающими последовательность элементов в макете:



4. Для редактирования компонента и его внешнего вида нажмите на значок меню приложения  и выберите **Настройки**. Чтобы не использовать пропорции по умолчанию, можно задать ширину для отдельных областей. Также можно задать выравнивание (по левому краю, по центру или по правому краю).
5. На вкладке "Общие" можно изменить параметры отдельных компонентов макета. Выберите имя компонента, чтобы посмотреть его параметры.
6. На вкладке "Фон" можно изменить параметры фона макета.
См. [Изменение фона или темы](#).

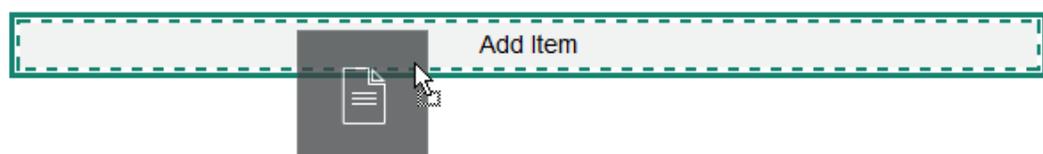
Использование макетов с двумя и тремя столбцами

Для автоматического определения интервалов и упорядочения компонентов, добавленных в макет, можно использовать макет раздела с несколькими столбцами.

Добавление макета на страницу:

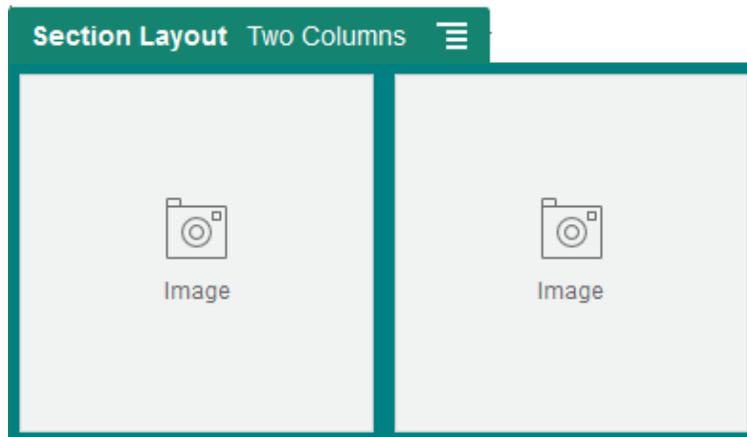
1. Перейдите к странице, которую хотите изменить, и убедитесь, что переключатель  переведен в положение **Редактировать**.
2. Добавьте макет на страницу.
3. Чтобы поместить другое содержимое в макет раздела, перетащите содержимое в макет.

Макет выделяется сплошной рамкой, и на нем появляется баннер с надписью "Добавить элемент".



Можно добавить в макет дополнительные элементы, и к ним будет автоматически применено соответствующее форматирование. Также можно добавлять другие макеты разделов для создания более сложных макетов или добавлять группы компонентов.

Ниже показан макет с двумя столбцами, содержащий компоненты изображения в качестве заполнителей:



Если добавлено несколько элементов, они будут отображаться в дополнительных строках, при этом в каждой строке будет максимум два элемента.

4. Для редактирования компонента и его внешнего вида нажмите на значок меню приложения  и выберите **Настройки**.
5. На вкладке "Общие" можно изменить параметры отдельных компонентов макета.
Выберите имя компонента, чтобы посмотреть его параметры.
6. На вкладке "Фон" можно изменить параметры фона макета.
7. Выберите **Пользовательские настройки**, чтобы задать дополнительные параметры по умолчанию для отображаемого содержимого.
 - **Ширина первого столбца (%)**. Укажите ширину столбца в процентах от пространства, доступного в макете.
 - **Ширина второго столбца (%)**. Укажите ширину столбца в процентах от пространства, доступного в макете.
 - **Ширина третьего столбца (%)**. Укажите ширину столбца в процентах от пространства, доступного в макете.
 - **Точка останова при ответе (в пикс.)**. Этот параметр предназначен для адаптивных страниц, которые автоматически регулируют форматирование содержимого на экранах разного размера. Укажите ширину в пикселах, чтобы макет раздела переключался между стандартным макетом с двумя столбцами и значениями параметров **Поведение при ответе**.
 - **Поведение при ответе**. Укажите, как должен меняться макет, когда размер экрана меньше значения **Точка останова при ответе**.

- **Никаких действий.** Поведение макета не меняется.
- **Набор столбцов.** Столбцы располагаются в порядке сверху вниз в едином столбце, при этом все элементы из первого столбца располагаются над элементами второго столбца и т. д.
- **Скрыть первый столбец.** Содержимое в первом столбце будет скрыто, чтобы освободить дополнительное пространство для остальных столбцов.
- **Скрыть второй столбец.** Содержимое во втором столбце будет скрыто, чтобы освободить дополнительное пространство для остальных столбцов.
- **Скрыть третий столбец.** Содержимое в третьем столбце будет скрыто, чтобы освободить дополнительное пространство для остальных столбцов.
- **Скрыть оба столбца.** Будет скрыто все содержимое в макете.
- **Поместить второй столбец под первым.** Элементы будут расположены в одном столбце, при этом все элементы из первого столбца будут располагаться над элементами второго столбца.
- **Поместить первый столбец под вторым.** Элементы будут расположены в одном столбце, при этом все элементы из второго столбца будут располагаться над элементами первого столбца.

Использование вертикальных макетов разделов

Макет раздела автоматически определяет интервалы и расположение добавленных в него компонентов.

Элементы, добавленные в вертикальный макет, располагаются вертикально друг над другом.

Добавление макета на страницу:

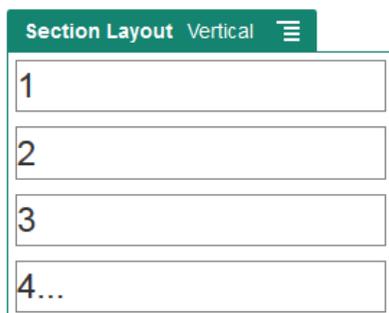
1. Перейдите к странице, которую хотите изменить, и убедитесь, что переключатель  переведен в положение **Редактировать**.
2. Добавьте макет на страницу.
3. Чтобы поместить другое содержимое в макет раздела, перетащите содержимое в макет.

Макет выделяется сплошной рамкой, и на нем появляется баннер с надписью "Добавить элемент".



Можно добавить в макет дополнительные элементы, и к ним будет автоматически применено соответствующее форматирование. Для создания более сложных макетов можно добавить другие макеты разделов.

Ниже представлен вертикальный макет с пронумерованными текстовыми компонентами, отражающими последовательность элементов в макете:



4. Для редактирования компонента и его внешнего вида нажмите на значок меню приложения  и выберите **Настройки**.
5. На вкладке "Общие" можно изменить параметры отдельных компонентов макета.
Выберите имя компонента, чтобы посмотреть его параметры.
6. На вкладке "Фон" можно изменить параметры фона макета.

Использование макетов разделов с вкладками

Чтобы создать интервалы и упорядочить компоненты, добавленные в макет, можно использовать макет раздела с вкладками.

Добавление макета на страницу:

1. Перейдите к странице, которую хотите изменить, и убедитесь, что переключатель  переведен в положение **Редактировать**.
2. Добавьте макет на страницу. Для макета по умолчанию добавляется одна вкладка. Чтобы добавить еще, нажмите **Новая вкладка**.
3. Чтобы добавить во вкладку содержимое, просто перетащите его.
Можно добавить в макет дополнительные элементы, и к ним будет автоматически применено соответствующее форматирование. Также можно добавлять другие макеты разделов для создания более сложных макетов или добавлять группы компонентов.
4. Для редактирования вкладки и ее внешнего вида нажмите на значок меню  и выберите **Настройки**.
5. Настройки фона вкладки можно изменить в разделе "Фон".
Можно добавить изображение для вкладки, изменить его положение и т. д. См. [Изменение фона или темы](#).
6. Для изменения других настроек внешнего вида вкладки используйте раздел "Стиль". Выберите стиль, например тонкую линию, рамку и т. п., или настройте границу и углы вкладки.

Использование макетов разделов со слайдами

Макет раздела со слайдами можно использовать для создания содержимого, которое остается на месте, пока не будет перемещено в другую позицию. Так освобождается место для нового содержимого. Для перехода между слайдами используются точки навигации под областью со слайдами.

Чтобы добавить макет со слайдами на страницу, выполните следующие действия:

1. Перейдите к странице, которую хотите изменить, и убедитесь, что переключатель  переведен в положение **Редактировать**.

2. Добавьте макет на страницу. По умолчанию вместе с макетом добавляется один слайд. Чтобы добавить еще, нажмите **Новый слайд**.

3. Чтобы поместить в слайд содержимое, выберите нужный слайд, а затем перетащите в него компонент или элемент содержимого.

Можно добавить на слайд дополнительные элементы и изменить их расположение. Также можно добавить другие макеты разделов для создания более сложных макетов или группы компонентов, позволяющие размещать содержимое в нужных местах.

4. Для редактирования макета со слайдами и его внешнего вида, нажмите на значок меню  и выберите **Настройки**.

5. Настройки фона для всей области слайдов можно изменить в разделе "Фон".

См. [Изменение фона или темы](#). Вы можете использовать изображение, изменить его положение и т. д.

6. Для изменения других настроек внешнего вида используйте раздел "Стиль". Выберите стиль, например тонкую линию, рамку и т. п., или настройте границу и углы для слайда.

7. Чтобы изменить настройки для каждого слайда по отдельности, нажмите на имя слайда и откройте его настройки. При необходимости измените для каждого слайда фон, стиль и т. д.

Использование встроенных компонентов

Компоненты, макеты разделов, цифровые активы и структурированное содержимое – это удобные инструменты для предоставления пользователям нужных функций и данных.

В данном разделе представлена информация об использовании отдельных компонентов. Для получения дополнительных сведений о компонентах см.:

- Чтобы узнать, как использовать компоненты на сайте, см. [Организация содержимого страницы](#).
- Информацию об управлении компонентами см. в разделе [Управление пользовательскими компонентами и макетами](#).
- Сведения о создании собственных компонентов см. в разделе [Разработка компонентов](#).

Основные компоненты

- [Заголовки](#)
- [Разделы](#)
- [Обычный текст](#)
- [Кнопки](#)

Структурные компоненты

- [Разделители](#)
- [Разделители](#)

Медиа-компоненты

- [Изображения](#)
- [Галереи](#)
- [Сетка галереи](#)
- [Видео YouTube](#)
- [Видео](#)

Компоненты документов

- [Документы](#)
- [Список папок](#)
- [Списки файлов](#)
- [Диспетчер документов](#)
- [Библиотека проекта](#)

Социальные компоненты

- Социальная панель
- Кнопки "Нравится" и "Рекомендую" на Facebook
- Как поделиться и подписаться в Twitter
- Компонент диалога
- Список диалогов

Компоненты процесса

- Форма запуска процесса
- Список задач процесса
- Форма сведений о задаче

Элементы содержимого

- Компонент элемента содержимого
- Заполнитель содержимого
- Список содержимого
- Поиск в содержимом
- Рекомендация

Другие компоненты

- Карты
- Заголовки
- Статьи
- Изображения с текстом
- Группы компонентов
- Использование компонента "Совместный просмотр" на странице
- Oracle Intelligent Advisor
- Oracle Visual Builder

Основные компоненты

Основные компоненты позволяют создавать текстовые блоки.

- Заголовки
- Разделы
- Обычный текст
- Кнопки

Заголовки

Порядок добавления и форматирования компонента-заголовка

1. Перейдите к странице, которую хотите изменить, и убедитесь, что переключатель  переведен в положение **Редактировать**.
2. Добавьте компонент на страницу.
3. Щелкните компонент-заголовок, чтобы ввести текст заголовка. На данный текст распространяется форматирование стиля по умолчанию данного компонента.
4. Чтобы добавить ссылку в тексте заголовка, выполните следующие действия.
 - a. Введите и выберите текст, который требуется использовать в качестве текста ссылки, а затем нажмите .
 - b. Щелкните **Выбрать тип ссылки** и выберите один из описанных ниже вариантов.
 - **Веб-страница:** укажите полный URL-адрес внешней страницы или сайта и выберите, где следует открыть ссылку.
 - **Страница сайта:** используйте средство выбора страниц, чтобы выбрать страницу на текущем сайте, и укажите, где следует открыть ссылку. Можно указать дополнительные параметры URL-адреса в формате `key1=value1&key2=value2`. Поддерживаются пустые значения; например, `key1=&key2=value2`. Также можно указать URL-привязку, но необходимо добавить специальный макет раздела "Привязка" к месту на целевой странице сайта, где требуется разрешить ссылку на привязку, и указать такое же имя привязки в настройках макета, которые использовались при определении действия триггера.

 **Примечание.**

Макет раздела "Привязка", необходимый для использования URL-привязок, предоставляется в инструментарии OCE. Сведения о том, как получить набор инструментов, см. в разделе Разработка с помощью инструментария OCE.

- **Выгрузка файла:** выгрузите указанный файл из репозитория. Выберите файл.
Выбрав цифровой актив, можно выбрать определенное представление. Если не выбирать представление, будет использоваться исходный размер. Чтобы вместе с сайтом публиковалась последняя версия актива, установите флажок **Использовать последнюю версию актива**. Если не установлен флажок **Использовать последнюю версию актива**, то используется последняя *опубликованная* версия, а не более новая черновая версия, если таковая существует.
- **Предварительный просмотр файлов:** предпросмотр (или выгрузка) выбранного файла, наложенного на полутемную неактивную версию страницы.
- **Элемент контента.** Выберите элемент контента в связанном репозитории активов, выберите страницу сведений, которую требуется отобразить, и укажите страницу, которую нужно открыть в том же или новом окне.
- **Электронная почта:** укажите действительный адрес электронной почты и, при необходимости, тему. Результирующее сообщение открывается или

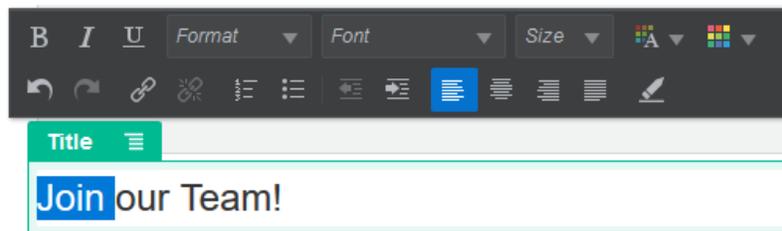
отправляется с помощью клиента электронной почты, используемого по умолчанию.

- **Карта:** введите действительный адрес или координаты и выберите, где должна открываться карта, в браузерах для настольных компьютеров и мобильных устройств.
- **Телефонный вызов:** введите действительный номер телефона.

Чтобы удалить ссылку, нажмите в любом месте текста ссылки и нажмите



5. Чтобы изменить форматирование по умолчанию, выделите текст, который нужно форматировать, затем выберите на панели инструментов форматирования любой из вариантов форматирования, такой как шрифт, цвет или выравнивание. В зависимости от темы могут отображаться не все эти параметры.



Вносимые изменения применяются немедленно. Такие изменения форматирования накладываются поверх базового стиля. В случае изменения базового стиля переопределения сохраняются.

6. Чтобы удалить форматирование, примененное с помощью этих инструментов, выберите текст и щелкните

Разделы

Порядок добавления и форматирования компонента-абзаца

1. Перейдите к странице, которую хотите изменить, и убедитесь, что переключатель переведен в положение **Редактировать**.
2. Добавьте компонент на страницу.
3. Нажмите на компонент-абзац, чтобы ввести текст заголовка. На данный текст распространяется форматирование стиля по умолчанию данного компонента.
4. Чтобы добавить в абзац видео (), изображение () или элемент контента () нажимайте на соответствующий значок. Выберите актив, чтобы добавить и задать настройки нужным образом. Например, вставьте элемент контента и выберите разные виды, чтобы управлять сложными макетами в абзаце.
5. Чтобы добавить абзац в компоненте абзаца, просто нажмите клавишу Enter.
6. Порядок добавления ссылки в абзаце:

- a. Введите и выберите текст, который требуется использовать в качестве текста ссылки, а затем нажмите .
- b. Щелкните **Выбрать тип ссылки** и выберите один из описанных ниже вариантов.
- **Веб-страница:** укажите полный URL-адрес внешней страницы или сайта и выберите, где следует открыть ссылку.
 - **Страница сайта:** используйте средство выбора страниц, чтобы выбрать страницу на текущем сайте, и укажите, где следует открыть ссылку. Можно указать дополнительные параметры URL-адреса в формате `key1=value1&key2=value2`. Поддерживаются пустые значения; например, `key1=&key2=value2`. Также можно указать URL-привязку, но необходимо добавить специальный макет раздела "Привязка" к месту на целевой странице сайта, где требуется разрешить ссылку на привязку, и указать такое же имя привязки в настройках макета, которые использовались при определении действия триггера.

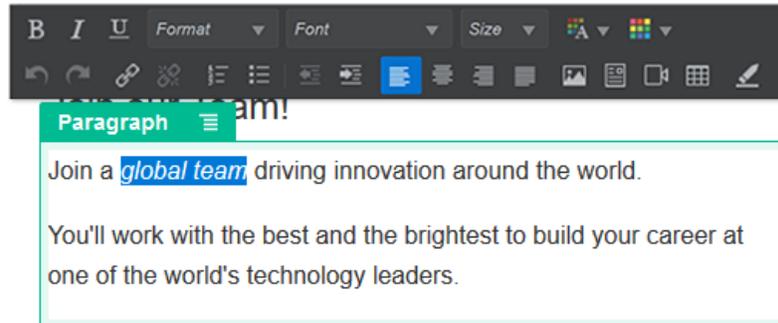
 **Примечание.**

Макет раздела "Привязка", необходимый для использования URL-привязок, предоставляется в инструментарии OCE. Сведения о том, как получить набор инструментов, см. в разделе Разработка с помощью инструментария OCE.

- **Выгрузка файла:** выгрузите указанный файл из репозитория. Выберите файл.
Выбрав цифровой актив, можно выбрать определенное представление. Если не выбирать представление, будет использоваться исходный размер. Чтобы вместе с сайтом публиковалась последняя версия актива, установите флажок **Использовать последнюю версию актива**. Если не установлен флажок **Использовать последнюю версию актива**, то используется последняя *опубликованная* версия, а не более новая черновая версия, если таковая существует.
- **Предварительный просмотр файлов:** предпросмотр (или выгрузка) выбранного файла, наложенного на полутемную неактивную версию страницы.
- **Элемент контента.** Выберите элемент контента в связанном репозитории активов, выберите страницу сведений, которую требуется отобразить, и укажите страницу, которую нужно открыть в том же или новом окне.
- **Электронная почта:** укажите действительный адрес электронной почты и, при необходимости, тему. Результирующее сообщение открывается или отправляется с помощью клиента электронной почты, используемого по умолчанию.
- **Карта:** введите действительный адрес или координаты и выберите, где должна открываться карта, в браузерах для настольных компьютеров и мобильных устройств.
- **Телефонный вызов:** введите действительный номер телефона.

Чтобы удалить ссылку, нажмите в любом месте текста ссылки и нажмите .

7. Чтобы изменить форматирование по умолчанию любой части текста, выделите текст, который нужно форматировать, затем выберите на панели инструментов форматирования любой из вариантов форматирования, такой как шрифт, цвет или выравнивание. В зависимости от темы могут отображаться не все эти параметры.



Вносимые изменения применяются немедленно. Такие изменения форматирования накладываются поверх базового стиля. В случае изменения базового стиля переопределения сохраняются.

8. Чтобы удалить форматирование, примененное с помощью данных параметров, выделите текст и нажмите .

Обычный текст

Компонент-текст позволяет добавлять текст на страницу и форматировать его с помощью стилей, заданных в теме. Панель инструментов форматирования недоступна при редактировании содержимого, поэтому невозможно перезаписать выбранный стиль пользовательским форматированием.

Это полезно, если используется стандартное форматирование для таких элементов, как заголовки. Можно обновить стили, определенные в теме, и автоматически обновить связанный текст на сайте без перезаписи формата.

Добавление компонента-текста:

1. Перейдите к странице, которую хотите изменить, и убедитесь, что переключатель  переведен в положение **Редактировать**.
2. Добавьте компонент на страницу.
3. Нажмите на компонент-абзац, чтобы ввести текст заголовка. К тексту применяется форматирование стиля компонента по умолчанию (<p> или элемента абзаца).

Чтобы добавить еще один абзац, нажмите Enter.

Примечание.

Можно копировать и вставлять в компонент-текст содержимое из других источников, но изображения и базовые теги HTML удаляются.

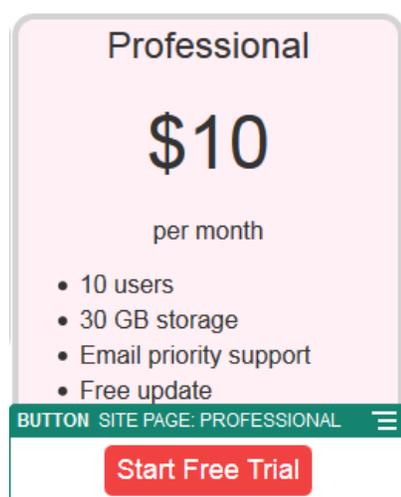
4. Чтобы изменить базовый стиль компонента-текста, выполните следующие действия.
 - a. Щелкните значок меню компонента  и выберите пункт **Настройки**.
 - b. Нажмите **Выбрать стиль** и выберите стиль в меню.

Список включает стандартные теги HTML для абзацев и заголовков. Выбранный стиль назначает связанный тег содержимому.

Вид компонента отражает выбранный базовый стиль.

Кнопки

Кнопка позволяет сделать ссылку или другую функциональную возможность более заметной на странице.



Чтобы добавить кнопку на страницу, выполните указанные ниже действия.

1. Перейдите к странице, которую хотите изменить, и убедитесь, что переключатель  переведен в положение **Редактировать**.
2. Добавьте компонент на страницу.
3. Чтобы настроить свойства кнопки, щелкните ее значок меню  и выберите пункт **Настройки**.

Вы можете указать текст на кнопке (метка), размер, выравнивание и другие настройки отображения кнопки.

Примечание.

Если в полях **Ширина** и **Высота** задать значение 0 (ноль), размер кнопки автоматически подстраивается под текст, указанный для метки.

Чтобы задать цвет фона, шрифт, границу и другие настройки, перейдите на вкладку **Стиль**. Можно выбрать один из predetermined стилей в текущей теме. В

зависимости от темы можно также нажать **Настройка** для указания собственных значений.

4. Чтобы связать с кнопкой ссылку или другие действия, выполните указанные ниже действия.
 - a. На панели "Настройки" щелкните **Ссылка**.
 - b. Щелкните **Выбрать тип ссылки** и выберите один из описанных ниже вариантов.
 - **Нет ссылки:** при нажатии кнопки никакие действия не выполняются.
 - **Веб-страница:** укажите полный URL-адрес внешней страницы или сайта и выберите, где следует открыть ссылку.
 - **Страница сайта:** используйте средство выбора страниц, чтобы выбрать страницу на текущем сайте, и укажите, где следует открыть ссылку. Можно указать дополнительные параметры URL-адреса в формате `key1=value1&key2=value2`. Поддерживаются пустые значения; например, `key1=&key2=value2`. Также можно указать URL-привязку, но необходимо добавить специальный макет раздела "Привязка" к месту на целевой странице сайта, где требуется разрешить ссылку на привязку, и указать такое же имя привязки в настройках макета, которые использовались при определении действия триггера.

 **Примечание.**

Макет раздела "Привязка", необходимый для использования URL-привязок, предоставляется в инструментарии OCE. Сведения о том, как получить набор инструментов, см. в разделе Разработка с помощью инструментария OCE.

- **Выгрузка файла:** выгрузите указанный файл из репозитория. Выберите файл.

Выбрав цифровой актив, можно выбрать определенное представление. Если не выбирать представление, будет использоваться исходный размер. Чтобы вместе с сайтом публиковалась последняя версия актива, установите флажок **Использовать последнюю версию актива**. Если не установлен флажок **Использовать последнюю версию актива**, то используется последняя *опубликованная* версия, а не более новая черновая версия, если таковая существует.
- **Предварительный просмотр файлов:** предпросмотр (или выгрузка) выбранного файла, наложенного на полутемную неактивную версию страницы.
- **Элемент контента.** Выберите элемент контента в связанном репозитории активов, выберите страницу сведений, которую требуется отобразить, и укажите страницу, которую нужно открыть в том же или новом окне.
- **Электронная почта:** укажите действительный адрес электронной почты и, при необходимости, тему. Результирующее сообщение открывается или отправляется с помощью клиента электронной почты, используемого по умолчанию.

- **Действия триггера:** выберите одно или несколько действий страницы, которое будет выполняться при нажатии кнопки. Далее описываются возможные действия на странице.
 - **Перейти на страницу сайта:** выберите страницу текущего сайта.
 - **Перейти по внешнему URL-адресу:** укажите полный URL-адрес на внешнюю страницу или сайт.
 - **Показать предупреждение:** отображение в окне указанного сообщения.
 - **Показать или скрыть компонент:** выберите компонент текущей страницы, который требуется отобразить, скрыть или переключить.
 - **Показать или скрыть ячейку:** выберите на текущей странице ячейку, которую требуется отобразить, скрыть или переключить.
- **Карта:** введите действительный адрес или координаты и выберите, где должна открываться карта, в браузерах для настольных компьютеров и мобильных устройств.
- **Телефонный вызов:** введите действительный номер телефона.

См. [Использование триггеров и действий](#).

Структурные компоненты

Структурные компоненты разделяют различные виды содержимого на странице.

- [Разделители](#)
- [Разделители](#)

Разделители

Разделитель (горизонтальная линия) используется для создания визуального разрыва в столбце или на странице.

Чтобы добавить разделитель на страницу, выполните указанные ниже действия.

1. Перейдите к странице, которую хотите изменить, и убедитесь, что переключатель  переведен в положение **Редактировать**.
2. Добавьте компонент на страницу.

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>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>
	

3. Чтобы настроить свойства разделителя, щелкните его значок меню  и выберите пункт **Настройки**.

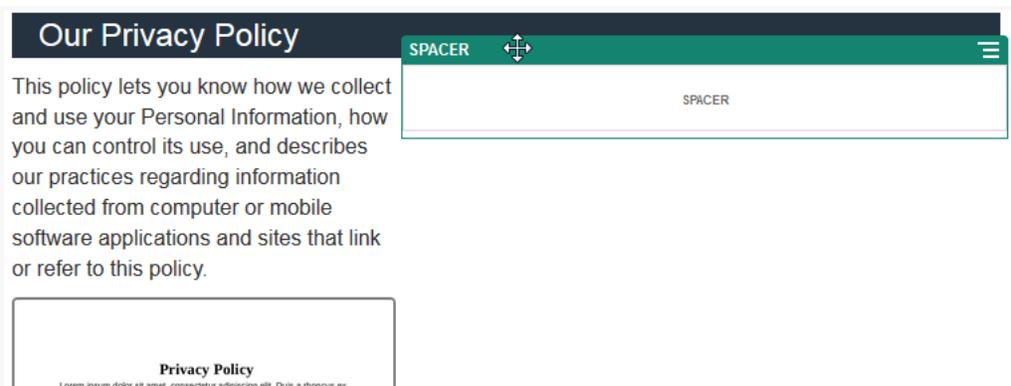
Горизонтальное размещение и длину разделителя можно указать, регулируя параметры интервала слева и справа. Цвет, толщина и другие параметры линии определяются стилем компонента. Вы можете выбрать предварительно определенные стили в текущей теме или нажмите кнопку **Настроить**, чтобы указать собственные значения.

Разделители

Контролируя "свободное пространство" на страницах, можно повысить их визуальную привлекательность и удобство чтения. При использовании компонента "Разделитель символов" добавляется пустое место без переопределения разбивки, заданной в стилях или других компонентах.

Чтобы между компонентами на странице добавить пустое пространство по вертикали, выполните указанные ниже действия.

1. Перейдите к странице, которую хотите изменить, и убедитесь, что переключатель  переведен в положение **Редактировать**.
2. Добавьте компонент на страницу.



3. Чтобы настроить высоту разделителя, нажмите на его значок меню  и выберите пункт **Настройки**.

Медиа-компоненты

Медиа-компоненты позволяют размещать на страницах изображения и видео.

- [Изображения](#)
- [Галереи](#)
- [Сетка галереи](#)
- [Видео YouTube](#)
- [Видео](#)

Изображения

Изображения могут привлекать внимание посетителей и побуждать их к изучению разных областей содержимого на странице.

Чтобы добавить изображение на страницу, выполните указанные ниже действия.

1. Перейдите к странице, которую хотите изменить, и убедитесь, что переключатель  переведен в положение **Редактировать**.
2. Добавьте компонент на страницу. Для компонента-изображения заполнитель отображается, пока не будет выбрано изображение.



3. Чтобы выбрать изображение, щелкните значок меню , выберите **Настройки** и щелкните **Выбрать** рядом с полем "Изображение".
4. В репозитории сайтов в папке документов выберите изображение, к которому вам был предоставлен общий доступ, или загрузите изображение в папку документов.

Примечание.

В данном окне отображаются все доступные файлы. Необходимо выбрать тип файла, соответствующий контексту. Например, выбирая файл изображения, необходимо выбрать допустимый формат изображения (GIF, JPG, JPEG, PNG или SVG).

- a. Найдите и выберите изображение, которое требуется использоваться

Если цифровые активы не отображаются, нажмите  и измените фильтр по коллекциям на вариант **Все**.

- b. Выбрав изображение в папке документов, можно создать ссылку на файл, не копируя его на сайт. Чтобы добавить ссылку на файл, установите флажок **Использовать ссылку на исходный файл вместо копирования файла на сайт**. В противном случае копия файла сохраняется на самом сайте. Добавление ссылки на исходный файл позволяет избежать дублирования содержимого. Кроме того, ссылка дает посетителям возможность посмотреть содержимое, даже если изменяются разрешения на доступ к файлу или просмотр файла.
- c. Нажмите **ОК**.
- d. Выбрав цифровой актив, можно выбрать определенное представление. Если не выбирать представление, будет использоваться исходный размер. Чтобы

вместе с сайтом публиковалась последняя версия актива, установите флажок **Использовать последнюю версию актива**.

5. Чтобы отредактировать изображение, нажмите , и отредактируйте изображение, выполнив любое из следующих действий:
 - Чтобы обрезать изображение, нажмите  **Обрезать**. Выберите на панели инструментов обрезки одно из предварительно определенных соотношений сторон изображения или перетащите маркеры обрезки на изображение по своему усмотрению. Добившись нужного результата, на панели инструментов обрезки нажмите **Обрезать**.
 - Чтобы повернуть или отразить изображение, нажмите  **Повернуть**. На панели инструментов поворота введите произвольный угол поворота в градусах или с помощью кнопок поверните изображение влево или вправо или выберите вариант отражения изображения: по горизонтали или по вертикали.
 - Чтобы добавить водяной знак на изображение, нажмите  **Водяной знак**. Добавьте текст к изображению, изменив его размер, стиль, цвет и непрозрачность с помощью инструментов водяного знака.
 - Чтобы изменить формат изображения, нажмите  **Параметры**, затем в раскрывающемся списке **Формат** выберите новый формат.
 - Чтобы изменить цвет фона, нажмите  **Параметры**, затем в раскрывающемся меню **Цвет фона** выберите нужный вариант.
 - При редактировании файлов .jpg или .webp (доступно в браузерах Google Chrome) можно изменить качество изображения, чтобы создать файл меньшего размера. Нажмите  **Параметры** и введите новое процентное значение в поле (**Качество**).
 - Чтобы отменить или повторить изменение, нажмите  или . Чтобы удалить все внесенные изменения, нажмите **Сброс**.
 - Чтобы изменить размер представления изображения, используйте элементы управления масштабированием (  ).
6. Панель **Общие** меню **Настройки**  позволяет добавить подпись или изменить интервалы, выравнивание, стиль и другие параметры.
7. С изображением можно связать ссылку или другие действия.
 - a. На панели "Настройки" щелкните **Ссылка**.
 - b. Выберите один из следующих вариантов:
 - **Нет ссылки**: по щелчку изображения никакие действия не выполняются.
 - **Веб-страница**: укажите полный URL-адрес внешней страницы или сайта и выберите, где следует открыть ссылку.
 - **Страница сайта**: используйте средство выбора страниц, чтобы выбрать страницу на текущем сайте, и укажите, где следует открыть

ссылку. Можно указать дополнительные параметры URL-адреса в формате `key1=value1&key2=value2`. Поддерживаются пустые значения; например, `key1=&key2=value2`. Также можно указать URL-привязку, но необходимо добавить специальный макет раздела "Привязка" к месту на целевой странице сайта, где требуется разрешить ссылку на привязку, и указать такое же имя привязки в настройках макета, которые использовались при определении действия триггера.

 **Примечание.**

Макет раздела "Привязка", необходимый для использования URL-привязок, предоставляется в инструментари ОСЕ. Сведения о том, как получить набор инструментов, см. в разделе Разработка с помощью инструментария ОСЕ.

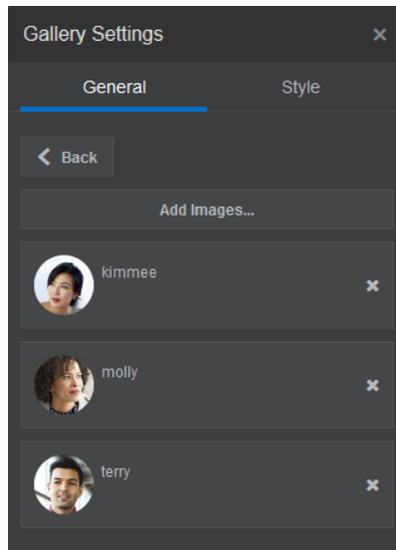
- **Выгрузка файла:** выгрузите указанный файл из репозитория. Выберите файл.
Выбрав цифровой актив, можно выбрать определенное представление. Если не выбирать представление, будет использоваться исходный размер. Чтобы вместе с сайтом публиковалась последняя версия актива, установите флажок **Использовать последнюю версию актива**. Если не установлен флажок **Использовать последнюю версию актива**, то используется последняя *опубликованная* версия, а не более новая черновая версия, если таковая существует.
- **Предварительный просмотр файла:** выбранный файл отображается в виде наложения на текущую страницу.
- **Элемент контента.** Выберите элемент контента в связанном репозитории активов, выберите страницу сведений, которую требуется отобразить, и укажите страницу, которую нужно открыть в том же или новом окне.
- **Электронная почта:** укажите действительный адрес электронной почты и, при необходимости, тему. Результирующее сообщение открывается или отправляется с помощью клиента электронной почты, используемого по умолчанию.
- **Предварительный просмотр изображения:** выбранное изображение будет отображаться в виде наложения на текущую страницу.
- **Карта:** введите действительный адрес или координаты и выберите, где должна открываться карта, в браузерах для настольных компьютеров и мобильных устройств.
- **Телефонный вызов:** введите действительный номер телефона.

Галереи

Серии изображений отображаются в галерее. Можно включить автоматический циклический просмотр изображений или разрешить пользователю просматривать их вручную.

Чтобы добавить галерею изображений на страницу, выполните указанные ниже действия.

1. Перейдите к странице, которую хотите изменить, и убедитесь, что переключатель  переведен в положение **Редактировать**.
2. Добавьте компонент на страницу. Для компонента-галереи заполнитель отображается, пока не будут выбраны изображения.
3. Чтобы добавить в галерею одно или несколько изображений, щелкните ее значок меню , выберите **Настройки** и на вкладке **Общие** щелкните **Изображения**.
4. Нажмите на значок **Добавить изображения**.



5. Выберите одно или несколько изображений.

 **Примечание.**

В данном окне отображаются все доступные файлы. Необходимо выбрать тип файла, соответствующий контексту. Например, выбирая файл изображения, необходимо выбрать допустимый формат изображения (GIF, JPG, JPEG, PNG или SVG).

- a. Найдите и выберите изображения, которые хотите использовать.
Если цифровые активы не отображаются, нажмите  и измените фильтр по коллекциям на вариант **Все**.
- b. Выбрав изображение в папке документов, можно создать ссылку на файл, не копируя его на сайт. Чтобы добавить ссылку на файл, установите флажок **Использовать ссылку на исходный файл вместо копирования файла на сайт**. В противном случае копия файла сохраняется на самом сайте. Добавление ссылки на исходный файл позволяет избежать дублирования содержимого. Кроме того, ссылка дает посетителям возможность посмотреть содержимое, даже если изменяются разрешения на доступ к файлу или просмотр файла.
- c. Нажмите **ОК**.

- d. Выбрав цифровой актив, можно выбрать определенное представление. Если не выбирать представление, будет использоваться исходный размер. Чтобы вместе с сайтом публиковалась последняя версия актива, установите флажок **Использовать последнюю версию актива**.

Выбранные изображения добавляются в список изображений. Порядок изображений в списке можно изменить перетаскиванием. По умолчанию в качестве заголовка каждого изображения используется имя файла без расширения.

6. Чтобы изменить заголовок, описание или другие настройки конкретного изображения, нажмите на изображение в списке и внесите нужные изменения.

С изображением в галерее также можно связать ссылку или другие действия.

- a. На панели "Настройки" для конкретного изображения нажмите на поле "Ссылка".
- b. Выберите один из следующих вариантов:
 - **Нет ссылки:** по щелчку изображения никакие действия не выполняются.
 - **Веб-страница:** укажите полный URL-адрес внешней страницы или сайта и выберите, где следует открыть ссылку.
 - **Страница сайта:** используйте средство выбора страниц, чтобы выбрать страницу на текущем сайте, и укажите, где следует открыть ссылку. Можно указать дополнительные параметры URL-адреса в формате `key1=value1&key2=value2`. Поддерживаются пустые значения; например, `key1=&key2=value2`. Также можно указать URL-привязку, но необходимо добавить специальный макет раздела "Привязка" к месту на целевой странице сайта, где требуется разрешить ссылку на привязку, и указать такое же имя привязки в настройках макета, которые использовались при определении действия триггера.

Примечание.

Макет раздела "Привязка", необходимый для использования URL-привязок, предоставляется в инструментари ОСЕ. Сведения о том, как получить набор инструментов, см. в разделе Разработка с помощью инструментария ОСЕ.

- **Выгрузка файла:** выгрузите указанный файл из репозитория. Выберите файл.

Выбрав цифровой актив, можно выбрать определенное представление. Если не выбирать представление, будет использоваться исходный размер. Чтобы вместе с сайтом публиковалась последняя версия актива, установите флажок **Использовать последнюю версию актива**. Если не установлен флажок **Использовать последнюю версию актива**, то используется последняя *опубликованная* версия, а не более новая черновая версия, если таковая существует.

- **Элемент контента.** Выберите элемент контента в связанном репозитории активов, выберите страницу сведений, которую требуется отобразить, и укажите страницу, которую нужно открыть в том же или новом окне.
- **Электронная почта:** укажите действительный адрес электронной почты и, при необходимости, тему. Результирующее сообщение открывается или

- отправляется с помощью клиента электронной почты, используемого по умолчанию.
- **Карта:** введите действительный адрес или координаты и выберите, где должна открываться карта, в браузерах для настольных компьютеров и мобильных устройств.
 - **Телефонный вызов:** введите действительный номер телефона.
- c. Чтобы вернуться к панели настроек изображения, нажмите **Назад**. Еще раз нажмите **Назад**, чтобы вернуться к списку изображений и выбрать для обновления другое изображение.
- d. Завершив обновление отдельных изображений, нажмите **Назад**, чтобы указать параметры галереи.
7. Чтобы изменить размер и масштаб изображений в галерее, выполните указанные ниже действия.
- a. В поле **Ширина** укажите в пикселях ширину галереи в ячейке. Чтобы указать ширину, необходимо выбрать любой вариант выравнивания, кроме **Заполнение**. После задания ширины вариант **Заполнение** можно использовать для растягивания изображения на указанную ширину.
- b. Выберите вариант **Масштабирование**, чтобы настроить представление изображений в галерее.
- **Обрезать:** наименьшее из двух измерений (ширина или высота) масштабируется, чтобы вписать изображение в доступное пространство, а наибольшее измерение обрезается, чтобы предотвратить растягивание.
 - **Вписать:** каждое изображение масштабируется так, чтобы оно целиком вписалось в доступное пространство без искажения.
 - **Растянуть:** наибольшее из двух измерений (ширина или высота) масштабируется, чтобы вписать изображение в доступное пространство, а наименьшее измерение растягивается до заполнения доступного пространства.
 - **Нет:** изображение используется с полным разрешением в пикселях и равномерно обрезается, если оно не вписывается в доступное пространство.
8. Чтобы помочь пользователям перемещаться по галерее, выполните указанные ниже действия.
- a. Выберите метод **навигации**:
- **Эскизы.** Под документом отображается последовательность его страниц. Чтобы перейти к нужному изображению в галерее, нажмите на эскиз.
 - **Индексатор.** Отображает последовательность кнопок  под галереями, представляющих каждое изображение в галерее. Чтобы перейти к нужному изображению в галерее, нажмите соответствующую кнопку.
 - **Нет:** визуальные средства навигации не предоставляются. Чтобы перейти к следующему или предыдущему изображению, необходимо провести пальцем по экрану вправо или влево. Этот ручной метод всегда доступен.

- b. Нажмите **Показывать стрелки "Предыдущая/следующая"**, чтобы на каждом изображении отображались стрелки перехода к предыдущему или следующему изображению в галерее.
9. Название изображения можно скрыть, установив флажок **Показать заголовок**.
10. Чтобы включить автоматический циклический просмотр изображений в галерее, выберите **Автоспроизведение**, затем укажите значения для параметров **Время перехода** и **Время отображения**.

Для пользователей также будут доступны все настроенные функции ручной навигации.

Сетка галереи

Сетка галереи используется для одновременного представления набора изображений в строках и столбцах.

Чтобы добавить галерею изображений на страницу, выполните указанные ниже действия.

1. Перейдите к странице, которую хотите изменить, и убедитесь, что переключатель  переведен в положение **Редактировать**.
2. Добавьте компонент на страницу. Для компонента-сетки галерея заполнитель отображается, пока не будут выбраны изображения.
3. Чтобы добавить в галерею одно или несколько изображений, щелкните ее значок меню , выберите **Настройки** и на вкладке **Общие** щелкните **Изображения**.
4. Нажмите на значок **Добавить изображения**.
5. Выберите одно или несколько изображений.

Примечание.

В данном окне отображаются все доступные файлы. Необходимо выбрать тип файла, соответствующий контексту. Например, выбирая файл изображения, необходимо выбрать допустимый формат изображения (GIF, JPG, JPEG, PNG или SVG).

- a. Найдите и выберите изображения, которые хотите использовать.

Если цифровые активы не отображаются, нажмите  и измените фильтр по коллекциям на вариант **Все**.

- b. Выбрав изображение в папке документов, можно создать ссылку на файл, не копируя его на сайт. Чтобы добавить ссылку на файл, установите флажок **Использовать ссылку на исходный файл вместо копирования файла на сайт**. В противном случае копия файла сохраняется на самом сайте. Добавление ссылки на исходный файл позволяет избежать дублирования содержимого. Кроме того, ссылка дает посетителям возможность посмотреть содержимое, даже если изменяются разрешения на доступ к файлу или просмотр файла.
- c. Нажмите **ОК**.

- d. Выбрав цифровой актив, можно выбрать определенное представление. Если не выбирать представление, будет использоваться исходный размер. Чтобы вместе с сайтом публиковалась последняя версия актива, установите флажок **Использовать последнюю версию актива**.

Выбранные изображения добавляются в список изображений. Порядок изображений в списке (и в сетке) можно изменить перетаскиванием. По умолчанию в качестве заголовка каждого изображения используется имя файла без расширения.

- 6. Чтобы изменить заголовок, описание или другие настройки конкретного изображения, щелкните изображение в списке и внесите нужные изменения.

С изображением в галерее также можно связать ссылку или другие действия.

- a. На панели "Настройки" для конкретного изображения щелкните поле "Ссылка".
- b. Выберите один из следующих вариантов:
 - **Нет ссылки:** по щелчку изображения никакие действия не выполняются.
 - **Веб-страница:** укажите полный URL-адрес внешней страницы или сайта и выберите, где следует открыть ссылку.
 - **Страница сайта:** используйте средство выбора страниц, чтобы выбрать страницу на текущем сайте, и укажите, где следует открыть ссылку. Можно указать дополнительные параметры URL-адреса в формате `key1=value1&key2=value2`. Поддерживаются пустые значения; например, `key1=&key2=value2`. Также можно указать URL-привязку, но необходимо добавить специальный макет раздела "Привязка" к месту на целевой странице сайта, где требуется разрешить ссылку на привязку, и указать такое же имя привязки в настройках макета, которые использовались при определении действия триггера.

 **Примечание.**

Макет раздела "Привязка", необходимый для использования URL-привязок, предоставляется в инструментарии OCE. Сведения о том, как получить набор инструментов, см. в разделе Разработка с помощью инструментария OCE.

- **Выгрузка файла:** выгрузите указанный файл из репозитория. Выберите файл.

Выбрав цифровой актив, можно выбрать определенное представление. Если не выбирать представление, будет использоваться исходный размер. Чтобы вместе с сайтом публиковалась последняя версия актива, установите флажок **Использовать последнюю версию актива**. Если не установлен флажок **Использовать последнюю версию актива**, то используется последняя *опубликованная* версия, а не более новая черновая версия, если таковая существует.

- **Элемент контента.** Выберите элемент контента в связанном репозитории активов, выберите страницу сведений, которую

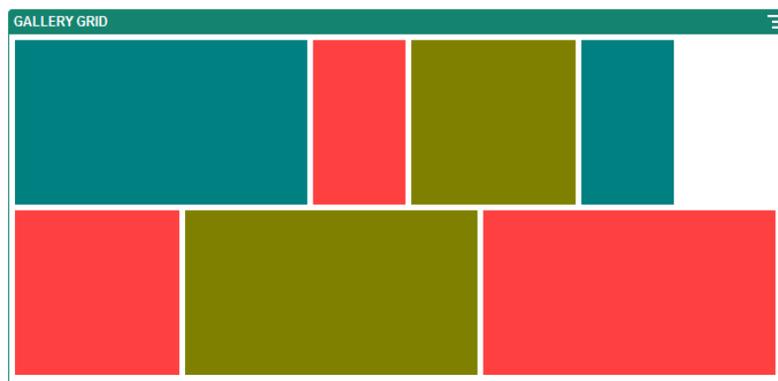
требуется отобразить, и укажите страницу, которую нужно открыть в том же или новом окне.

- **Электронная почта:** укажите действительный адрес электронной почты и, при необходимости, тему. Результирующее сообщение открывается или отправляется с помощью клиента электронной почты, используемого по умолчанию.
 - **Предварительный просмотр изображения:** выбранное изображение будет отображаться в виде наложения на текущую страницу.
 - **Карта:** введите действительный адрес или координаты и выберите, где должна открываться карта, в браузерах для настольных компьютеров и мобильных устройств.
 - **Телефонный вызов:** введите действительный номер телефона.
- c. Чтобы вернуться к панели настроек изображения, нажмите кнопку **Назад**. Еще раз нажмите кнопку **Назад**, чтобы вернуться к списку изображений и выбрать для обновления другое изображение.
- d. Завершив обновление отдельных изображений, нажмите кнопку **Назад**, чтобы указать параметры галереи.
7. Для упорядочивания изображений в сетке используются разные варианты **макета**.
- Каменная кладка
 - Столбцы
 - Пользовательские

Эти варианты описываются в приведенных ниже действиях.

8. Выберите макет **Каменная кладка**, чтобы изображения автоматически располагались в строках с учетом доступного пространства.

Получающиеся в результате строки обладают одинаковой высотой, но не имеют определенных столбцов.



- a. Укажите значение **Высота**, чтобы пропорционально масштабировать все изображения к указанной высоте в пикселях.
- b. Укажите **Интервал между изображениями**, чтобы увеличить или уменьшить интервал между изображениями в строке.
9. Выберите макет **Пользовательский**, чтобы упорядочить изображения по строкам и столбцам.

a. Выберите элемент **Масштабирование**, чтобы настроить представление изображений в сетке.

- **Обрезать:** наименьшее из двух измерений (ширина или высота) масштабируется, чтобы вписать изображение в доступное пространство, а наибольшее измерение обрезается, чтобы предотвратить растягивание.
- **Вписать:** каждое изображение масштабируется так, чтобы оно целиком вписалось в доступное пространство без искажения.

Например, в показанной ниже сетке с четырьмя столбцами шесть изображений масштабируются, чтобы вписаться в доступное пространство.



Такая же сетка с обрезанными изображениями:



b. Укажите **характеристическое отношение**, чтобы определить форму ячеек сетки.

- **Квадрат:** характеристическое отношение 1:1.
- **Альбомная ориентация:** характеристическое отношение 16:9.
- **Портрет:** характеристическое отношение 9:16.
- **Пользовательское:** укажите собственные числовые значения для характеристического отношения.

c. Укажите число **столбцов**.

Сетка настраивается автоматически, чтобы создать столбцы одинаковой ширины.

- d. Укажите **Интервал между изображениями**, чтобы увеличить или уменьшить интервал между изображениями в строке и столбцах.
10. Измените макет **Пользовательский**, чтобы упорядочить изображения по строкам и столбцам на основе указанных размера изображения и ширины.
- a. Выберите элемент **Масштабирование**, чтобы настроить представление изображений в сетке.
 - **Обрезать**: наименьшее из двух измерений (ширина или высота) масштабируется, чтобы вписать изображение в доступное пространство, а наибольшее измерение обрезается, чтобы предотвратить растягивание.
 - **Вписать**: каждое изображение масштабируется так, чтобы оно целиком вписалось в доступное пространство без искажения.
 - b. Укажите значения **Высота изображения** и **Высота изображения**, чтобы определить форму ячеек сетки.

Сетка настраивается автоматически, чтобы создать ячейки указанных измерений.
 - c. Укажите **Интервал между изображениями**, чтобы увеличить или уменьшить интервал между изображениями в строке и столбцах.
11. Укажите параметры **Выравнивание**, **Ширина** и **Интервал**, чтобы задать положение сетки в ячейке.

В поле **Ширина** укажите в пикселях ширину галереи в ячейке. Чтобы указать ширину, необходимо выбрать любой вариант выравнивания, кроме **Заполнение**. После задания ширины вариант **Заполнение** можно использовать для растягивания изображения на указанную ширину.

Видео YouTube

Используйте потоковое видео YouTube, чтобы добавить анимацию и визуальные эффекты на страницу.

Для работы с видеороликами, которые не размещены на сайте YouTube, см. раздел [Видео](#).

Для добавления видео YouTube на страницу выполните следующие действия.

1. Перейдите к странице, которую хотите изменить, и убедитесь, что переключатель  переведен в положение **Редактировать**.
2. Добавьте компонент на страницу. Пока не выбрано видео для использования, на компоненте показывается замещающее изображение.
3. Чтобы указать видео YouTube для использования и настроить свойства его отображения, нажмите на значок меню  и выберите **Настройки**.
4. Найдите видео, которое необходимо использовать, скопируйте URL-адрес и вставьте его в поле **YouTube URL**.
5. Определите параметры отображения.
 - **Показать элементы управления**: включить параметры на экране и устройстве, чтобы управлять воспроизведением видео вручную.

- **Показать сведения:** временно включить описание видео в левом верхнем углу.
 - **Автовоспроизведение:** автоматически начинать воспроизведение видео.
 - **Цикл:** автоматически повторять видео.
6. Используйте **Пропорции**, определенные для видео (**Авто**), или выберите другие пропорции, определяющие форму отображения видео.
 7. Укажите тип выравнивания и интервалы для размещения видео.

Видео

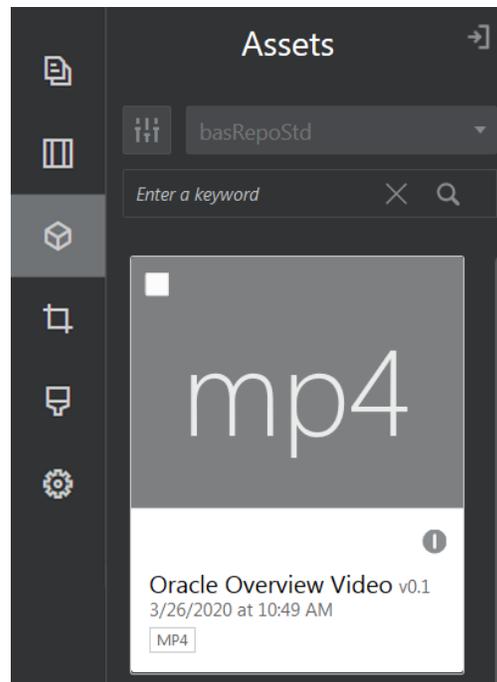
Чтобы сделать страницу более эффектной и визуально привлекательной, можно добавить видео из Oracle Content Management.

Oracle Content Management предлагает несколько вариантов добавления видео на страницу. Вы можете встроить [Видео YouTube](#), размещенное на сайте, или добавить видео, хранимое, управляемое и доставляемое Oracle Content Management, чтобы воспользоваться преимуществами автоматического транскодирования и оптимизированного потокового видео Video Plus. Это полезно, когда сайты обслуживаются на нескольких устройствах с различными возможностями. Если системный администратор не включил функцию Video Plus, можно использовать стандартное видео.

Для использования видеороликов, размещенных на сайте YouTube, см. раздел [Видео YouTube](#).

Чтобы использовать видео, хранящееся в документах или репозитории активов Oracle Content Management, выполните указанные ниже действия.

1. Перейдите на страницу, которую требуется отредактировать, и убедитесь, что переключатель редактирования () переведен в положение **Редактировать**.
 - a. Чтобы использовать видеоролик, сохраненный как цифровой актив, откройте панель **Активы** и с ее использованием найдите требуемый видеоролик, а затем выберите его и перетащите на страницу.



- b. Альтернативный вариант: можно добавить на эту страницу компонент "Видео" из списка компонентов. Пока не выбрано видео для использования, на компоненте показывается замещающее изображение.
- 2. При перетаскивании актива с панели "Активы" или видеокомпонента из списка компонентов используйте значок меню () для настройки свойств отображения видео или перейдите к видеоролику для этого видеокомпонента. Нажмите на меню и выберите **Настройки**.
- 3. Чтобы выбрать видео для компонента видео, щелкните **Выбрать** для открытия вкладок **Документы** и **Цифровые активы** и перейдите к видео, которое собираетесь добавить из Oracle Content Management. Выберите его и нажмите **ОК**.

 - a. Если включена функция Video Plus, можно выбирать различные репозитории, выполнять поиск каналов, коллекций и ключевых слов, а также фильтровать активы, чтобы сузить поиск при переходе к видеоролику.
 - b. При использовании стандартного видео выберите видеофайл типа MP4. Формат MP4 поддерживается всеми браузерами. Кроме того, некоторые стандартные видеоролики имеют предварительно определенное изображение для предварительного просмотра. Если такое изображение отсутствует, вместо него отображается пустой экран. Чтобы использовать графический элемент вместо пустого экрана, щелкните **Выбрать** в диалоговом окне настроек видео и выберите изображение предварительного просмотра из Oracle Content Management.
- 4. После того как вы выбрали видео, перетащив его с панели активов или выбрав его в диалоговом окне настроек, укажите в этом окне любые параметры отображения:

 - **Показать элементы управления:** включить параметры на экране и устройстве, чтобы управлять воспроизведением видео вручную.
 - **Автовоспроизведение:** автоматически начинать воспроизведение видео.
 - **Цикл:** автоматически повторять видео.
 - **Скрытые:** автоматическое скрытие при загрузке страницы.

5. Укажите любые параметры ширины, выравнивания или интервала для размещения видео.
6. Нажмите **Стиль**, чтобы выбрать или определить пользовательский стиль для границы видео.

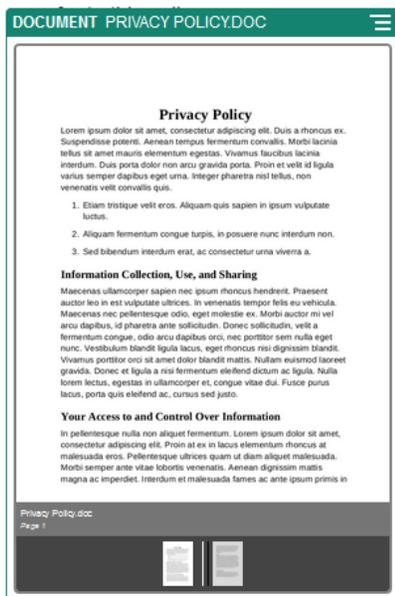
Компоненты документов

В этом разделе описаны некоторые компоненты, позволяющие предоставить доступ к файлам и папкам и отобразить их.

- [Документы](#)
- [Список папок](#)
- [Списки файлов](#)
- [Диспетчер документов](#)
- [Библиотека проекта](#)

Документы

Многостраничные документы и презентации слайдов можно просматривать прямо со страницы.



Чтобы добавить документ на страницу, выполните указанные ниже действия.

1. Перейдите к странице, которую хотите изменить, и убедитесь, что переключатель  переведен в положение **Редактировать**.
2. Добавьте компонент на страницу. Для компонента-документа заполнитель отображается, пока не будет выбран документ.
3. Чтобы выбрать документ, щелкните его значок меню , выберите **Настройки** и щелкните **Выбрать** рядом с полем "Документ".

Данный документ должен храниться в репозитории сайта или другом репозитории, к которому у вас есть доступ. Можно также использовать документы, к которым вам предоставлен доступ или которые вы загрузили из локального или сетевого расположения файлов.

4. Выделите документ и нажмите **ОК**.

 **Примечание.**

В данном окне отображаются все доступные файлы. Необходимо выбрать тип файла, соответствующий контексту. Например, выбирая файл документа, необходимо выбрать допустимый формат документа, такой как TXT или DOC.

5. Панель "Настройки" позволяет добавить подпись или изменить интервалы, выравнивание, стиль и другие параметры презентации.
6. Чтобы помочь пользователям перемещаться по документу, выполните указанные ниже действия.
 - a. Установите флажок **Показывать номера страниц**, чтобы под каждой страницей отображался ее номер.
 - b. Выберите метод **навигации**:
 - **Эскизы**: под документом отображается последовательность его страниц. Чтобы перейти к нужной странице, следует нажать на эскиз.
 - **Индексатор**: под документом отображается серия кнопок , чтобы представить каждую страницу документа. Чтобы перейти к соответствующей странице, следует нажать кнопку.
 - **Нет**: визуальные средства навигации не предоставляются. Чтобы перейти к соседней странице, необходимо провести пальцем по экрану вправо или влево. Этот ручной метод всегда доступен.
 - c. Нажмите **Показывать стрелки "Предыдущая/следующая"**, чтобы на каждой странице отображались стрелки перехода к предыдущей или следующей странице документа.

Список папок

Список папок используется для отображения папок в указанной папке учетной записи Oracle Content Management.

Если этот компонент используется в сочетании с компонентами списка файлов или диспетчера документов на странице, эти компоненты могут автоматически отображать содержимое папки, выбранной в списке папок.

Добавление компонента "Список папок" на страницу:

1. Перейдите к странице, которую хотите изменить, и убедитесь, что переключатель  переведен в положение **Редактировать**.
2. Добавьте компонент на страницу.

3. Для редактирования компонента и его внешнего вида нажмите на значок меню приложения  и выберите **Настройки**.
4. Выберите **Пользовательские настройки** для настройки сведений по умолчанию об отображаемом содержимом.
5. Нажмите **Выбрать** рядом с надписью **Выбор папки**, чтобы изменить папку, используемую для отображения. Требуется выбрать папку, отличную от домашней папки репозитория документов. Нажмите **Назад** по завершении.

Примечание.

Список папок предоставляет всем пользователям полномочия роли "Выгрузка". Пользователи могут просматривать и выгружать файлы независимо от своей роли. Если посетителю сайта предоставлены более широкие полномочия, чем указанные для компонента, то последние будут перезаписаны.

6. Выберите один из следующих вариантов, чтобы настроить дополнительные значения по умолчанию для отображаемого содержимого.
 - **Выбор значений по умолчанию:** при выборе **Oracle Documents Folder** выберите папку (при наличии) для отображения в соответствии с выбранными в списке значениями.
 - **Показывать заголовок с именем папки:** выберите для отображения имени папки в заголовке встроенного компонента.
 - **Сортировка папок:** выберите первоначальный способ отображения элементов - в алфавитном порядке по имени или по дате последнего обновления.
 - **Показать подпапки:** с помощью ползунка можно ограничить количество отображаемых вложенных папок.
7. На вкладке "Общие" можно изменить интервалы, тип выравнивания и другие параметры представления.
8. Отформатировать фрейм можно на вкладке "Стиль", которая содержит компонент с предопределенными стилями или с пользовательскими вариантами выбора.

Списки файлов

Можно использовать список файлов для просмотра файлов из указанной папки в учетной записи Oracle Content Management.

Если компонент используется в сочетании с компонентами "Список папок" на странице, компонент "Список файлов" может автоматически отображать содержимое папки, выбранной в списке папок. Можно также настроить компонент для выполнения одного или нескольких действий, когда пользователь нажимает на имя файла в списке. Например, выбранный файл может быть открыт для предварительного просмотра в отдельном окне браузера, в Lightbox или даже в компоненте "Диспетчер документов" на данной странице.

Добавление компонента "Список файлов" на страницу:

1. Перейдите к странице, которую хотите изменить, и убедитесь, что переключатель  переведен в положение **Редактировать**.
2. Добавьте группу компонентов на страницу.
3. Для редактирования компонента и его внешнего вида нажмите на значок меню приложения  и выберите **Настройки**.
4. Выберите **Пользовательские настройки** для настройки сведений по умолчанию об отображаемом содержимом.
5. Нажмите **Выбрать** рядом с надписью **Выбор папки**, чтобы изменить папку, используемую для отображения. Необходимо выбрать папку, отличную от домашней страницы документа. Нажмите **Назад** по завершении.

 **Примечание.**

Список файлов предоставляет всем пользователям полномочия роли "Выгрузка". Пользователи могут просматривать и выгружать файлы независимо от своей роли. Если посетителю сайта предоставлены более широкие полномочия, чем указанные для компонента, то последние будут перезаписаны.

6. Выберите один из следующих вариантов, чтобы настроить дополнительные значения по умолчанию для отображаемого содержимого.
 - **Параметры отображения:** выберите сведения, которые будут отображаться вместе с файлами в списке.
 - **Заголовок с именем папки:** показывать имя папки в заголовке.
 - **Описание файла:** показывать описание файла (при наличии).
 - **Разделители файлов:** отделить каждый файл линией.
 - **Значок "Выгрузить":** добавить значок выгрузки, чтобы пользователи могли добавить файл, если потребуется.
 - **Последнее обновление:** показывать дату последнего обновления файла.
 - **Размер файла:** показывать размер файла.
 - **Изображение:** показывать эскиз содержимого файла.
 - **Триггеры и действия:** выберите, чтобы автоматически добавлять в список файлов содержимое папки, выбранной в компоненте "Список папок". Кроме того, при выборе файла можно включить триггер "Выбранный файл". На вкладке "Ссылка" можно связать действия с триггером **Выбранный файл**.
 - **Сортировка файлов:** выберите первоначальный способ отображения элементов - в алфавитном порядке по имени или по дате последнего обновления.
 - **Показать файлы:** выберите для отображения всех файлов или ограничения количества файлов с помощью ползунка.
7. На вкладке "Общие" можно изменить интервалы, тип выравнивания и другие параметры представления.
8. Отформатировать фрейм можно на вкладке "Стиль", которая содержит компонент с predefined стилями или с пользовательскими вариантами выбора.

9. На вкладке "Ссылка" можно связать действия с триггером **Выбранный файл**. Например, чтобы настроить для компонента "Список файлов" предварительный просмотр выбранного файла в Lightbox:
 - a. Обязательно выберите параметр **Активировать триггер при выборе файла** в окне "Настройки списка файлов".
 - b. На вкладке "Ссылка" в окне "Настройки списка файлов" включите триггер **Выбранный файл**.
 - c. Нажмите и перетащите действие со страницей **Предварительный просмотр в Lightbox** в список действий.
 - d. В поле "Идентификатор или URL-адрес файла" выберите **Ссылка на файл**.

Ссылка файла – ссылка с полномочиями "Выгрузка". С помощью этой ссылки все посетители сайта могут просмотреть или выгрузить файл. Если выбрать **Идентификатор файла**, используется ссылка участника. Ссылка участника предназначена только для зарегистрированных пользователей и позволяет предварительно просматривать или выгружать выбранный файл.

Когда пользователь нажимает на имя файла в списке файлов, поверх полутемной, неактивной версии страницы открывается предварительный просмотр файла.

Диспетчер документов

Диспетчер документов позволяет просматривать информацию на домашней странице или файлы в Oracle Content Management.

Чтобы добавить на страницу компонент диспетчера документов, выполните следующие действия.

1. Перейдите к странице, которую хотите изменить, и убедитесь, что переключатель  переведен в положение **Редактировать**.
2. Добавьте компонент на страницу.
3. Для редактирования компонента и его внешнего вида нажмите на значок меню приложения  и выберите **Настройки**.
4. Выберите **Пользовательские настройки** для настройки сведений по умолчанию об отображаемом содержимом.
5. Нажмите **Выбрать** рядом с надписью **Выбор папки**, чтобы изменить папку, используемую для отображения.

Для выбора папки установите флажок рядом с именем нужной папки. Чтобы открыть папку, нажмите на ее имя. Нажмите на имя папки в пути для возврата к этой папке или нажмите **Домашняя папка** для возврата в домашнюю папку. Нажмите **Назад** по завершении.

6. Нажмите **Выбрать доступ к папке** и укажите роль доступа для посетителей. Посетители могут просматривать содержимое папки и работать с ним в зависимости от роли и **параметров просмотра**, активируемых на следующем шаге.

- **Доступ для участника:** посетители смогут использовать все функции, доступные участникам папки, например просмотр диалогов, аннотаций и пользовательских свойств элементов.
- **Просмотр:** пользователи с ролью "Просмотр" могут просматривать файлы и папки, но не могут их изменять.
- **Выгрузка:** пользователи с ролью "Выгрузка" могут выгружать файлы и сохранять их на своем компьютере.
- **Соавтор:** пользователи с ролью "Соавтор" могут изменять, обновлять и удалять файлы, а также загружать новые файлы.

Учитывайте следующее при настройке доступа к папке:

- Автор сайта не может предоставить доступ к папке, если для доступа к папке требуются более широкие полномочия, чем у автора. Например, если автору предоставлена роль доступа "Выгрузка" для папки, то он не может предоставить посетителям сайта полномочий соавтора для этой папки.
 - Полномочия, установленные для папки в компоненте, могут дополнять полномочия посетителя. Например, если посетителю предоставляются полномочия роли "Просмотр" (или вообще не предоставляются полномочия) при доступе к папке, то компонент может предоставить более широкие полномочия для выбранной роли. Эти расширенные полномочия применимы только в компоненте.
 - Если посетителю сайта предоставлены более широкие полномочия, чем указанные для компонента, то последние будут перезаписаны.
 - Полномочия, предоставленные на уровне папки, применимы ко всем вложенным папкам и файлам в этой папке.
7. Выберите один из следующих вариантов, чтобы настроить дополнительные значения по умолчанию для отображаемого содержимого.
- **Макет:** выберите исходный макет сетки, списка или компактного списка для папок и файлов. Пользователи могут изменить макет при просмотре готового встроенного компонента.
 - **Цветовая схема:** выберите одну из доступных цветовых схем для встроенного списка папок.
 - **Порядок сортировки:** выберите порядок отображения элементов: по алфавиту или по времени последнего обновления.
 - **Параметры просмотра:** выберите параметры, которые будут доступны пользователям при выборе элемента. Например, можно разрешить пользователям просматривать файлы, выгружать их, делиться ими, копировать файлы и папки, а также удалять файлы. Чтобы ограничить доступные для пользователей действия с файлами и папками, отмените выбор соответствующего параметра в списке. Если в предыдущем шаге выбрана роль "Доступ для участника", то можно отобразить боковую панель, где показываются диалоги, аннотации и пользовательские свойства.
 - **Параметры средства просмотра:** выберите, как пользователи будут просматривать файлы. Можно разрешить просмотр файлов во встроенном фрейме или в другой вкладке (или в окне в зависимости от настроек браузера). Также можно скрыть или отобразить эскизы и настроить способ просмотра видео.

- **Показать элементы управления масштабированием:** выберите, следует ли показывать ползунок и элементы управления масштабированием во встроенном представлении папки.
 - **Режим просмотра:** показывать файлы на всю страницу, по ширине страницы или в исходном размере.
 - **Триггеры и действия:** можно обновить список файлов, если он также используется в компоненте списка папок.
8. На вкладке "Общие" можно изменить интервалы, тип выравнивания и другие параметры представления.
 9. Отформатировать фрейм можно на вкладке "Стиль", которая содержит компонент с предопределенными стилями или с пользовательскими вариантами выбора.

После добавления диспетчера документов посетители могут увидеть представление выбранной папки, встроенной во фрейм на сайте. С помощью имеющихся параметров отображения пользователи могут изменить представление списка папок и файлов. Выбирая файл или папку, пользователь может изменить параметры в строке меню или в контекстном меню, чтобы выполнить любое разрешенное действие или использовать заданные **Параметры просмотра**.

Если с элементом связан диалог, то рядом с элементом отображается значок . Нажмите на него, чтобы открыть панель диалога, где можно просмотреть аннотации и комментарии. Независимые диалоги (не связанные с папкой) необходимо добавлять с помощью компонента диалогов.

Если элемент содержит пользовательские свойства, то их можно просмотреть на соответствующей панели. Нажмите **Дополнительно** и выберите **Пользовательские свойства**, чтобы открыть панель свойства.

Библиотека проекта

Библиотеку проекта можно использовать для сбора папок из разных расположений репозитория без изменения исходной папки или ее расположения. Компонент библиотеки проекта также позволяет назначить каждой папке разные уровни разрешений, чтобы учесть различные роли участников проекта.

Например, если папке в библиотеке проекта назначить одну из ролей посетителя ("Просмотр", "Выгрузка" или "Соавтор"), посетители смогут просматривать содержимое папки и взаимодействовать с ним на уровне полномочий, связанных с данной ролью. Если указать доступ для участников, эта папка будет отображаться в библиотеке проекта только для участников. Участники взаимодействуют с содержимым папки с полномочиями, заданными для них в исходной папке.

Если этот компонент используется в сочетании с компонентами списка файлов или диспетчера документов на странице, то эти компоненты могут автоматически отображать содержимое папки, выбранной в библиотеке проекта.

Чтобы добавить на страницу компонент библиотеки проекта, выполните следующие действия.

1. Перейдите к странице, которую хотите изменить, и убедитесь, что переключатель  переведен в положение **Редактировать**.

2. Добавьте компонент на страницу.
3. Для редактирования компонента и его внешнего вида нажмите на значок меню приложения  и выберите **Настройки**.
4. Нажмите **Пользовательские настройки** для выбора диалогов и параметров отображения.
5. Укажите заголовок. Заголовок по умолчанию – "Библиотека проекта". Можно отобразить или скрыть заголовок с помощью флажка **Заголовок**.
6. Выберите **Цветовая схема**.

 **Примечание.**

Если библиотека проекта связана с компонентом диспетчера документов посредством триггеров или действий, цветовая схема, настроенная для этой библиотеки проекта, также применяется к папке, выбранной в компоненте диспетчера документов. Это не относится к случаям, когда библиотека проекта связана с компонентами списка папок или файлов, так как эти компоненты не поддерживают цветовые схемы.

7. Нажмите **Добавить** рядом со списком папок, чтобы добавить существующую папку или создать новую.
 - a. Найдите и выберите одну или несколько папок либо нажмите **Создать**, чтобы создать папку. Каждая выбранная в этом окне папка добавляется в библиотеку проекта.

Можно использовать любые функции на панели инструментов, включая просмотр и сортировку.
 - b. Нажмите **Назад** по завершении.
8. Чтобы упорядочить список, выберите папку и нажмите **Переместить вверх** или **Переместить вниз** для изменения ее места в списке или нажмите **Удалить** для удаления папки из списка.
9. Чтобы настроить доступ к папке на основе роли пользователя, выберите папку в списке папок, нажмите **Выбрать разрешение папки** и выберите роль доступа.
 - **Доступ для участника:** папку, указанную в библиотеке проекта, будут видеть только зарегистрированные пользователи с разрешениями для данной папки. Участники взаимодействуют с содержимым папки с полномочиями, заданными для них в исходной папке.
 - **Просмотр:** пользователи с ролью "Просмотр" могут просматривать файлы и папки, но не могут их изменять.
 - **Выгрузка:** пользователи с ролью "Выгрузка" могут выгружать файлы и сохранять их на своем компьютере.
 - **Соавтор:** пользователи с ролью "Соавтор" могут изменять, обновлять и удалять файлы, а также загружать новые файлы.

Учитывайте следующее при настройке доступа к папке:

- Автор сайта не может предоставить доступ к папке, если для доступа к папке требуются более широкие полномочия, чем у автора. Например, если автору предоставлена роль доступа "Выгрузка" для папки, то он не может предоставить посетителям сайта полномочий соавтора для этой папки.

- Полномочия, установленные для папки в компоненте, могут дополнять полномочия посетителя. Например, если посетителю предоставляются полномочия роли "Просмотр" (или вообще не предоставляются полномочия) при доступе к папке, то компонент может предоставить более широкие полномочия для выбранной роли. Эти расширенные полномочия применимы только в компоненте.
 - Если посетителю сайта предоставлены более широкие полномочия, чем указанные для компонента, то последние будут перезаписаны.
 - Полномочия, предоставленные на уровне папки, применимы ко всем вложенным папкам и файлам в этой папке.
10. Для папок с разрешением **Доступ для участника** можно отобразить диалоги, связанные с папкой или содержимым, нажав **Показать панель диалога в диспетчере документов**.
- Если установить этот флажок и настроить компонент диспетчера документов на странице для отображения выбранной папки, то пользователь может нажать на значок диалога, чтобы отобразить все диалоги, связанные с данной папкой. Если флажок не установлен, значок диалога не отображается.
11. По завершении закройте окно.
12. На вкладке "Общие" можно изменить интервалы, тип выравнивания и другие параметры представления.
13. Отформатировать фрейм можно на вкладке "Стиль", которая содержит компонент с предопределенными стилями или с пользовательскими вариантами выбора.
14. На вкладке "Ссылка" можно связать действия с триггером **Выбранная папка**. Например, если на страницу также добавлен компонент диспетчера документов, он может отображать содержимое папки, выбранной в папке проекта.
- a. На вкладке "Ссылка" в настройках библиотеки проекта нажмите на триггер **Выбранная папка**.
 - b. В окне "Настройка действий триггеров" нажмите "Диспетчер документов".
 - c. Нажмите и перетащите действие **Отобразить документы** в список действий.
 - d. В поле "Идентификатор или URL-адрес папки" выберите **Выбранная папка**.

Когда пользователь нажимает на папку в библиотеке проекта, ее содержимое отображается на странице в компоненте диспетчера документов.

Социальные компоненты

Социальные компоненты помогают пользователям поддерживать связь друг с другом.

- [Социальная панель](#)
- [Кнопки "Нравится" и "Рекомендую" на Facebook](#)
- [Как поделиться и подписаться в Twitter](#)
- [Компонент диалога](#)

- [Список диалогов](#)

Социальная панель

Вы можете легко добавлять значки и ссылки на популярные социальные сети, такие как Facebook и Twitter. Социальная панель включает некоторые элементы по умолчанию, но вы можете добавлять элементы на панель и удалять с нее.



Чтобы добавить и изменить социальную панель, выполните указанные ниже действия.

1. Перейдите к странице, которую хотите изменить, и убедитесь, что переключатель  переведен в положение **Редактировать**.

2. Добавьте компонент на страницу. На социальной панели отображаются значки, включенные в нее по умолчанию.

3. Чтобы добавить или изменить значки на социальной панели, щелкните ее значок меню , выберите **Настройки** и наверху вкладки **Общие** щелкните **Значки**.

Чтобы удалить значок социальной сети, щелкните значок x рядом с ее именем. Чтобы добавить значок, он должен храниться в репозитории сайта или другом репозитории, к которому у вас есть доступ. Можно также использовать изображения, к которым вам предоставлен доступ или которые вы загрузили из локального или сетевого расположения файлов.

4. Чтобы загрузить один или несколько значков из локального или сетевого каталога, выполните указанные ниже действия.
 - a. Щелкните **Добавить значки** в верхней части панели.
 - b. Перейдите к расположению в репозитории, где планируется сохранить изображение, или щелкните , чтобы загрузить новую папку в текущее расположение.
 - c. Нажмите .
 - d. Найдите и выберите один или несколько файлов изображений, а затем нажмите кнопку **Открыть**.

Выбранные файлы изображений загружаются в текущее расположение в репозитории.

5. Выберите в репозитории одно или несколько изображений и нажмите кнопку **ОК**.
 - a. Найдите и выберите изображения, которые хотите использовать.
 - b. Чтобы добавить ссылку на файл в репозитории, установите флажок **Использовать ссылку на исходный файл вместо копирования файла на сайт**. В противном случае копия файла сохраняется на самом сайте. Добавление ссылки на исходный файл позволяет избежать дублирования содержимого. Кроме того, ссылка дает посетителям возможность посмотреть содержимое, даже если изменяются разрешения на доступ к файлу или просмотр файла.

с. Нажмите **ОК**.

Выбранные изображения добавляются в список изображений. Порядок изображений в списке можно изменить перетаскиванием. По умолчанию в качестве заголовка каждого изображения используется имя файла без расширения.

- Чтобы изменить целевой URL-адрес, описание или другие настройки конкретного изображения, щелкните изображение в списке и внесите нужные изменения.

Завершив обновление отдельных значков, нажмите кнопку **Назад**, чтобы указать общие параметры.

- Укажите размер, промежутки, ориентацию и выравнивание для всех значков на социальной панели.

Кнопки "Нравится" и "Рекомендую" на Facebook

Можно добавить кнопку Facebook "Нравится" на страницу, чтобы пользователи могли оценить сайт на Facebook.

- Перейдите к странице, которую хотите изменить, и убедитесь, что переключатель  переведен в положение **Редактировать**.
- Добавьте компонент на страницу.
- Для редактирования приложения и его внешнего вида нажмите на значок меню приложения  и выберите **Настройки**.
- Выберите **Настройки кнопки Facebook "Нравится"** или **Настройки кнопки Facebook "Рекомендую"**, чтобы указать URL-адрес и при необходимости добавить кнопку **Поделиться**, которая позволяет опубликовать ссылку на сайт на странице Facebook.
- На вкладке "Общие" можно изменить интервалы, тип выравнивания и другие параметры представления.
- Отформатировать фрейм можно на вкладке "Стиль", которая содержит приложение с предопределенными стилями или с пользовательскими вариантами выбора.

Как поделиться и подписаться в Twitter

Можно добавить кнопку Twitter "Поделиться" на страницу, чтобы пользователи могли быстро поделиться ссылкой в учетной записи Twitter.

- Перейдите к странице, которую хотите изменить, и убедитесь, что переключатель  переведен в положение **Редактировать**.
- Добавьте компонент на страницу.
- Для редактирования приложения и его внешнего вида нажмите на значок меню приложения  и выберите **Настройки**.
- Выберите **Настройки кнопки Twitter "Подписаться"**, чтобы указать имя пользователя Twitter и выбрать, следует ли отображать имя пользователя, а также определить размер кнопки **Подписаться**. Выберите **Настройки**

кнопки Twitter "Поделиться" для настройки значений по умолчанию. Пользователи могут изменить эти значения при использовании кнопки.

- **Поделиться URL:** URL-адрес сайта, которым может поделиться пользователь.
 - **Текст твита:** сообщение о странице.
 - **От @:** учетная запись пользователя Twitter, используемая для публикации твита.
 - **Рекомендовать @:** учетная запись пользователя Twitter, используемая для рекомендации в Twitter.
 - **Хештег #:** хештег, который вы хотите добавить к публикации.
 - **Количество:** отображение количества публикаций, вертикально или горизонтально.
 - **Большая кнопка:** выбор размера кнопки для приложения.
5. На вкладке "Общие" можно изменить интервалы, тип выравнивания и другие параметры представления.
 6. Отформатировать фрейм можно на вкладке "Стиль", которая содержит приложение с предопределенными стилями или с пользовательскими вариантами выбора.

Компонент диалога

Диалоги можно использовать прямо на сайте для обсуждения различных тем.

Примечание.

Чтобы диалог работал на сайте, сайт должен быть безопасным и доступным только для указанных пользователей или для участников с ролью "Пользователь Oracle Content Management". См. [Изменение настроек безопасности сайта](#).

Добавление компонента-диалога на страницу:

1. Перейдите к странице, которую хотите изменить, и убедитесь, что переключатель  переведен в положение **Редактировать**.
2. Добавьте компонент на страницу.
3. Для редактирования компонента и его внешнего вида нажмите на значок меню приложения  и выберите **Настройки**.
4. Выберите **Пользовательские настройки**, чтобы выбрать диалог и цветовую схему.
5. Нажмите **Выбрать** рядом с надписью **Выбрать диалог**, чтобы выбрать существующий диалог или создать новый.
 - a. Выберите диалог в списке доступных диалогов или нажмите **Создать**, чтобы создать новый диалог и присвоить ему имя.

В списке перечислены все диалоги, к которым у вас есть доступ.

Список можно отфильтровать:

- **Все:** отображение всех диалогов, кроме скрытых.
- **Избранное:** отображение диалогов, добавленных в избранные.

- **Скрытые:** отображение скрытых диалогов, исключенных из списка диалогов.
- **Закрытые:** отображение закрытых диалогов.

Также можно отсортировать список:

- **Последнее обновление:** сортировка диалогов по времени обновления.
- **Имя:** сортировка диалогов по алфавиту.
- **Не прочитано:** сортировка диалогов, при которой первыми отображаются диалоги с наибольшим количеством непрочтенных сообщений.

- Нажмите **Назад** по завершении.
- Если выбран независимый диалог (не основанный на файле или папке), то можно задать разрешения доступа для пользователей, просматривающих диалог.
- Выберите **Цветовая схема**.
- Чтобы использовать компонент "Диалог" в сочетании с компонентом "Список диалогов" на странице, выберите **Автоматически обновлять диалог, выбранный в списке диалогов**.

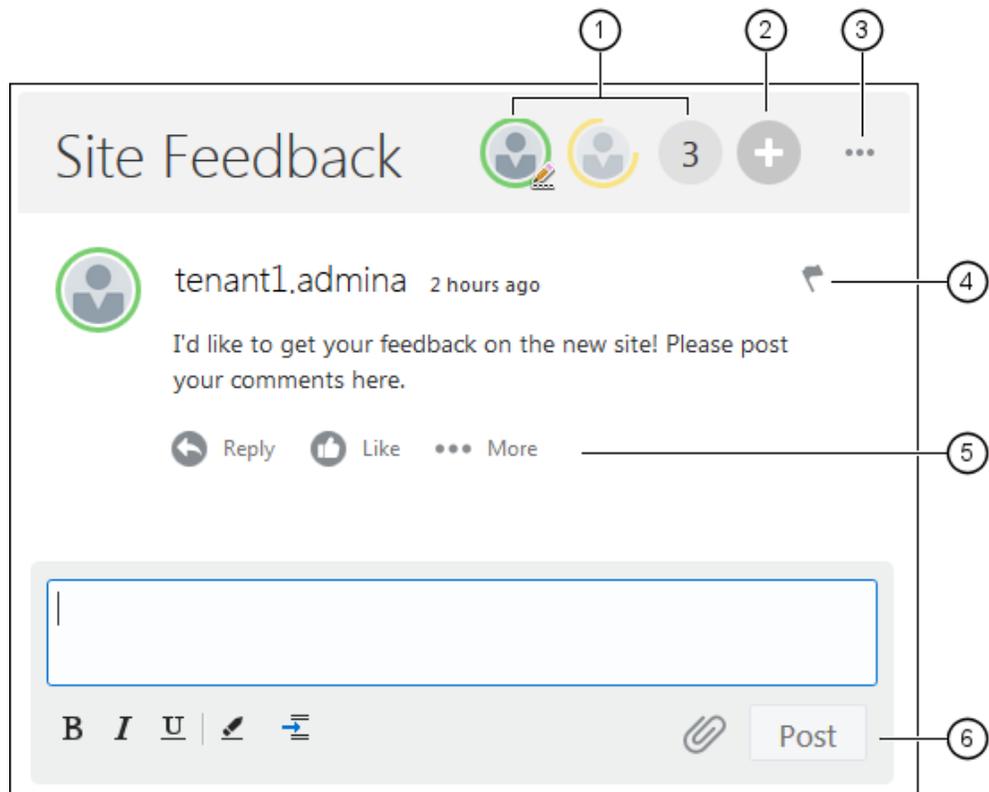
Если пользователь выбирает диалог в списке, то диалог отображается в соответствующем компоненте.

- По завершении закройте окно.
- На вкладке "Общие" можно изменить интервалы, тип выравнивания и другие параметры представления.
 - Отформатировать фрейм можно на вкладке "Стиль", которая содержит компонент с предопределенными стилями или с пользовательскими вариантами выбора.

На опубликованном сайте посетители увидят диалоги в рамке вашего сайта. Пользователи могут переключиться между диалогами, чтобы прочесть сообщения и ответить на них.

Примечание.

Если посетитель не добавлен в список участников диалога, то он сможет читать сообщения и отвечать на них, но ему будут недоступны элементы с 1 по 4, показанные на изображении ниже.



- **Строка меню (1)** содержит сведения об участниках диалога. Нажмите на значок пользователя, чтобы просмотреть сведения о статусе и варианты взаимодействия с пользователем. Значок с цифрой показывает общее число участников. Нажмите на значок, чтобы просмотреть весь список.
- Нажмите **Добавить пользователей (2)** для добавления участников в диалог.
- Нажмите на значок **Дополнительные параметры (3)** для просмотра параметров работы с диалогом. Список параметров зависит от роли пользователя. Для диалогов, связанных с папками и файлами, список параметров также зависит от разрешений, установленных для папок и файлов.
- Нажмите на значок **Флаг (4)** для назначения флага уведомления конкретному пользователю. Пользователь получит уведомление по электронной почте в соответствии с установленными им параметрами.
- Используйте **параметры комментариев (5)** для выполнения действий с определенным сообщением в диалоге, например для ответа, оценки, редактирования или удаления комментария.
- При добавлении и редактировании комментария используйте **параметры редактирования (6)** для добавления и удаления базового форматирования (полужирный шрифт, подчеркивание), а также для добавления вложения и публикации сообщения в диалоге. Если посетитель сайта не является участником диалога, то он не сможет добавлять вложения.

Список диалогов

Диалоги можно использовать прямо на сайте для обсуждения различных тем.

Если компонент используется в сочетании с компонентами "Список диалогов" на странице, компонент "Список диалогов" может автоматически отображать содержимое диалога, выбранного в списке папок.

Примечание.

Чтобы диалог или список диалогов работали на сайте, сайт должен быть безопасным и доступным только для указанных пользователей или для участников с ролью "Пользователь Oracle Content Management". См. [Изменение настроек безопасности сайта](#).

Чтобы добавить компонент "Список диалогов" на страницу, выполните следующие действия.

1. Перейдите к странице, которую хотите изменить, и убедитесь, что переключатель  переведен в положение **Редактировать**.
2. Добавьте компонент на страницу.
3. Для редактирования компонента и его внешнего вида нажмите на значок меню приложения  и выберите **Настройки**.
4. Нажмите **Пользовательские настройки** для выбора диалогов и параметров отображения.
5. Укажите заголовок. Заголовок по умолчанию – "Список диалогов". Можно отобразить или скрыть заголовок с помощью флажка **Заголовок**.
6. Нажмите **Добавить** рядом со списком диалогов, чтобы добавить существующий диалог или создать новый.
 - a. Выберите диалоги в списке доступных диалогов или нажмите **Создать**, чтобы создать новый диалог и присвоить ему имя.
 - b. Нажмите **Назад** по завершении.
 - c. Выберите разрешения для пользователей, которые будут просматривать список диалогов.
7. Чтобы упорядочить список диалогов, выберите диалог и нажмите **Переместить вверх** или **Переместить вниз** для изменения его места в списке или нажмите **Удалить** для удаления диалога из списка.
8. Выберите **Цветовая схема** и дополнительно **Параметры отображения**. Выбранные параметры отображения показываются под именем диалога в списке.
9. По завершении закройте окно.
10. На вкладке "Общие" можно изменить интервалы, тип выравнивания и другие параметры представления.

11. Отформатировать фрейм можно на вкладке "Стиль", которая содержит компонент с предопределенными стилями или с пользовательскими вариантами выбора.

На следующем изображении показан список диалогов с названием *Отзыв о сеансе* и всеми включенными параметрами отображения.



Компоненты процесса

Компоненты процесса позволяют пользователям сайта запускать задачи для предварительно определенных процессов и управлять ими.

Примечание.

Для работы с компонентами процесса требуется Oracle Process Cloud Service 17.1.3 или более поздняя версия.

- [Форма запуска процесса](#)
- [Список задач процесса](#)
- [Форма сведений о задаче](#)

Форма запуска процесса

Форму запуска процесса можно использовать для начала процесса, определенного с помощью Oracle Process Cloud Service.

Чтобы форма запуска процесса работала на сайте, должны быть выполнены следующие условия.

- Чтобы связанные процессы и формы запуска процессов отображались в этом компоненте, их необходимо определить с помощью Oracle Process Cloud Service. См. раздел *Разработка структурированных процессов* в документе *Использование процессов в Oracle Integration*.
- Работать с формами запуска процессов могут пользователи, которым присвоена роль, связанная с дорожкой процесса, содержащего эту форму запуска.
- Администратор должен настроить интеграцию между Oracle Process Cloud Service и Oracle Content Management. См. раздел *Интеграция с Oracle Process Cloud Service* в документе *Administering Oracle Content Management*. Для интеграции сервисов

необходимо использовать функцию единого входа, поэтому оба сервиса должны размещаться в одном домене идентификационных данных.

Чтобы добавить компонент "Форма запуска процесса" на страницу, выполните следующие действия.

1. Перейдите к странице, которую хотите изменить, и убедитесь, что переключатель  переведен в положение **Редактировать**.
2. Добавьте компонент на страницу.
3. Для редактирования компонента и его внешнего вида нажмите на значок меню приложения  и выберите **Настройки**.
4. Нажмите **Пользовательские настройки**, чтобы выбрать форму запуска процесса и задать настройки по умолчанию.
 - a. Если ваш сайт будет общедоступным, выберите прокси-службу.
 - b. Выберите раздел для формы запуска. Можно выбрать тестовый раздел, чтобы проверить правильность функционирования процесса, или производственный раздел, чтобы развернуть процесс для общего применения.
 - c. Если вы хотите всегда использовать версию процесса, выбранную по умолчанию, выберите **Использовать версию процесса по умолчанию**. Если вы не выбираете этот параметр, необходимо будет выбрать определенную версию, а при обновлении процесса потребуется обновить выбор в данных настройках.
 - d. Выберите процесс.
 - e. Выберите начальную форму. Синтаксис форм выглядит следующим образом: `process type:version:processname:start`. Пример: `Basic Approval:1.0:Process:Start Basic Approval`.

Сообщение "Нет подключения к Process Cloud Service" в верхней части окна может означать, что не настроена интеграция между Oracle Process Cloud Service и Oracle Content Management. Обратитесь к администратору.

Чтобы процесс появился в списке, разработчик процесса должен добавить вас в качестве инициатора процесса. Разработчик процесса должен добавить всех посетителей сайтов в качестве инициаторов процесса. В противном случае посетители смогут заполнить форму, но не смогут запустить процесс.
 - f. Выберите следующие варианты, чтобы задать дополнительные значения по умолчанию для формы.
 - **Заголовок формы:** при необходимости замените заголовок формы по умолчанию на собственный.
 - **Имя кнопки "Отправить":** дополнительно кнопку **Отправить** можно переименовать по своему выбору.
 - **Показать кнопку "Отправить":** дополнительно можно отобразить или скрыть кнопку **Отправить** на компоненте (по умолчанию она отображается). Кнопку **Отправить** можно скрыть, если похожая кнопка предоставляется в самой форме.

- **Подтверждение отправки:** дополнительно сообщение подтверждения по умолчанию можно заменить собственным сообщением.
 - **Показать подтверждение отправки:** выбор отображения или скрытия сообщения подтверждения при нажатии кнопки **Отправить**. По умолчанию это сообщение отображается в компоненте формы запуска процесса.
 - **Показать кнопку "Сохранить":** дополнительно можно отобразить или скрыть кнопку **Сохранить** на компоненте (по умолчанию она отображается). Можно выбрать отображение кнопки **Сохранить**, если связанный процесс позволяет сохранить работу и вернуться к ней позже.
 - **Показать кнопку "Удалить":** дополнительно можно отобразить или скрыть кнопку **Удалить** на компоненте (по умолчанию она отображается). Можно выбрать отображение кнопки **Удалить**, если необходимо предоставить пользователю возможность удалить содержимое формы и начать все сначала.
 - **Показать вложения:** дополнительно можно отобразить или скрыть область **Вложения** в форме с возможностью загрузки файлов (по умолчанию она скрыта). Загруженные файлы сохраняются как часть процесса Oracle Process Cloud Service.
 - **Настройка значений по умолчанию.** Можно указать пары имен и значений для предварительного заполнения начальной формы. Нажмите **Добавить поле**, затем введите имя и значение поля. Имя является одним из полей формы, а не подписью, используемой в форме. Само поле заполняется значением. Чтобы вставить данные пользователя, в настоящее время вошедшего в систему, используйте специальные значения `%%username%%` и `%%userid%%`.
5. На вкладке "Общие" можно изменить интервалы, тип выравнивания и другие параметры представления.
 6. Отформатировать фрейм можно на вкладке "Стиль", которая содержит компонент с определенными стилями или с пользовательскими вариантами выбора.
 7. Используйте вкладку "Ссылки" для назначения триггеров, предоставленных компонентом.
 - **Отправка формы запуска:** этот триггер активируется, когда пользователь нажимает кнопку **Отправить**.
 - **Сохранение формы запуска:** этот триггер активируется, когда пользователь нажимает кнопку **Сохранить**.
 - **Удаление формы запуска:** этот триггер активируется, когда пользователь нажимает кнопку **Удалить**.

Нажмите на триггер, чтобы назначить действие. Дополнительные сведения о назначении триггеров и действий см. в разделе [Использование триггеров и действий](#).

Список задач процесса

Список задач процесса можно использовать для выборочного перечисления процессов, определенных в Oracle Process Cloud Service. Вы можете отобразить подробную информацию о таких задачах в компоненте списка задач или использовать компонент

списка задач процессов вместе с компонентом сведений о задаче для упрощения списка и отображения подробной информации только о выбранной задаче.

Для работы списка задач процессов на сайте должны выполняться следующие условия.

- Чтобы связанные процессы отображались в этом компоненте, их необходимо определить с помощью Oracle Process Cloud Service. См. раздел *Разработка структурированных процессов* в документе *Использование процессов в Oracle Integration*.
- Сайт должен быть безопасным и доступным только для указанных пользователей или для участников с ролью "Пользователь Oracle Content Management". См. [Изменение настроек безопасности сайта](#).
- Администратор должен настроить интеграцию между Oracle Process Cloud Service и Oracle Content Management. См. раздел "Интеграция с Oracle Process Cloud Service" в документе *Administering Oracle Content Management*. Для интеграции сервисов необходимо использовать функцию единого входа, поэтому оба сервиса должны размещаться в одном домене идентификационных данных.

Чтобы добавить на страницу компонент списка задач процессов, выполните следующие действия.

1. Перейдите к странице, которую хотите изменить, и убедитесь, что переключатель  переведен в положение **Редактировать**.
2. Добавьте компонент на страницу.
3. Для редактирования компонента и его внешнего вида нажмите на значок меню приложения  и выберите **Настройки**.
4. Выберите **Пользовательские настройки**, чтобы задать параметры отображения и отфильтровать доступные задачи.
5. Чтобы задать параметры отображения списка задач, выберите один из следующих вариантов.

Примечание.

Вы можете задать параметры отображения для компонента, но представление самих задач определяется оформлением в Oracle Process Cloud Service.

- **Показывать сведения:** выберите, чтобы включить в список задач подробные сведения о задаче. Компонент списка задач процессов вместе с компонентом сведений о задаче упрощают перечисление и отображение подробной информации только для выбранной задачи.
- **Показывать поиск:** выберите, чтобы включить отображение панели поиска в верхней части списка задач. Если выбрать **Показывать поиск**, предоставляется дополнительная возможность выбрать настройку **Показывать фильтр** для включения в панель поиска параметров фильтрации. Чтобы задать значения фильтрации по умолчанию, используйте описанные ниже параметры фильтрации.

- **Показать кнопку "Выбрать все"**: выберите, чтобы позволить пользователю выбрать все отображаемые задачи для обработки.
 - **Размер страницы**: укажите максимальное число отображаемых задач. Если число задач меньше указанного максимума, в этом компоненте автоматически устанавливается меньшее число. Если количество задач превышает указанное максимальное значение, компонент показывает максимальное количество задач и добавляет ссылки на дополнительные страницы.
6. Далее описаны параметры фильтрации доступных задач в списке.
- Применяемые фильтры определяют исходный список задач. Если выбрать описанные выше настройки **Показывать поиск** и **Показывать фильтр**, пользователь может регулировать параметры, чтобы изменить фильтрацию для всех параметров, кроме параметра **От пользователя**.
- **Ключевые слова для поиска**: дополнительно можно указать одно или несколько условий поиска. В результаты поиска входят только задачи с заголовками, содержащими все указанные ключевые слова. Если выбран параметр **Показывать поиск**, эти ключевые слова отображаются на панели поиска, и пользователь может их удалять или изменять.
 - **Статус**: выберите одно из доступных значений статуса. По умолчанию задано значение **Назначено**.
 - **Ответственный**: выберите одно из доступных значений. По умолчанию задано значение **Я и мои группы – все**. В этом случае включены все задачи, доступные пользователю и его группе, в том числе задачи, которые затребованы, но не доступны для работы.
 - **От пользователя**: дополнительно выберите одного или нескольких пользователей, предоставивших задачу. Начните вводить имя пользователя, чтобы начать поиск доступных пользователей в связанном сервисе Oracle Process Cloud Service. Указанные вами значения не показываются конечным пользователям и не могут быть удалены. Пользователи могут добавить дополнительные имена, чтобы расширить список пользователей.
 - **Дата выполнения**: дополнительно укажите дату выполнения: выбранная на календаре дата, до, после или между выбранными датами.
 - **Приложение**: дополнительно выберите имя конкретного приложения. Вы можете выбрать одно или несколько приложений в списке приложений, доступных пользователю.
7. На вкладке "Общие" можно изменить интервалы, тип выравнивания и другие параметры представления.
8. Отформатировать фрейм можно на вкладке "Стиль", которая содержит компонент с предопределенными стилями или с пользовательскими вариантами выбора.

 **Примечание.**

Если на странице есть список задач процессов и форма сведений о задаче, компонент сведений о задаче автоматически показывает сведения о задаче, выбранной в списке. Чтобы узнать больше о триггерах и действиях, см. [Использование триггеров и действий](#).

Форма сведений о задаче

Форму сведений о задаче совместно со списком задач процесса можно использовать для отображения сведений о выбранной задаче. Если страница содержит список задач процесса и форму сведений о процессе, компонент сведений о задаче процесса автоматически показывает сведения о задаче, выбранной в списке.

В отношении работы формы сведений о задаче на сайте должно быть верно следующее.

- Чтобы связанные процессы отображались в этом компоненте, их необходимо определить с помощью Oracle Process Cloud Service. См. раздел *Разработка структурированных процессов* в документе *Using Processes in Oracle Integration*.
- Сайт должен быть безопасным и доступным только для указанных пользователей или для участников с ролью "Пользователь Oracle Content Management". См. [Изменение настроек безопасности сайта](#).
- Администратор должен настроить интеграцию между Oracle Process Cloud Service и Oracle Content Management. См. раздел *Интеграция с Oracle Process Cloud Service* в документе *Administering Oracle Content Management*. Для интеграции сервисов необходимо использовать функцию единого входа, поэтому оба сервиса должны размещаться в одном домене идентификационных данных.

Чтобы добавить на страницу компонент сведений о задаче, выполните следующие действия.

1. Перейдите к странице, которую хотите изменить, и убедитесь, что переключатель  переведен в положение **Редактировать**.
2. Добавьте компонент на страницу.
3. Для редактирования компонента и его внешнего вида нажмите на значок меню приложения  и выберите **Настройки**.
4. Выберите **Пользовательские настройки**, чтобы задать параметры отображения.
5. Чтобы задать параметры отображения формы сведений о задаче, выберите один из следующих вариантов.

Примечание.

Можно задать параметры отображения для компонента, но представление сведений о задаче определяется оформлением в Oracle Process Cloud Service.

- **Показывать действия:** выберите, чтобы отображать доступные пользователю действия, такие как "Утвердить", "Отклонить" и т. д.
- **Показывать кнопку "Сохранить":** выберите, чтобы включить отображение кнопки **Сохранить**.

- **Показывать кнопку "Заккрыть"**: выберите, чтобы включить отображение кнопки **Заккрыть**.
 - **Показывать вложения**: выберите, чтобы отобразить в форме сведений раздел **Вложения**.
 - **Показывать комментарии**: выберите, чтобы отобразить в форме сведений раздел **Комментарии**.
 - **Показывать историю**: выберите, чтобы отобразить в форме сведений раздел **История**.
 - **Показывать дополнительные сведения**: выберите, чтобы отобразить в форме сведений раздел **Дополнительные сведения**.
 - **Показывать ссылки**: выберите, чтобы отобразить в форме сведений раздел **Ссылки**.
6. На вкладке "Общие" можно изменить интервалы, тип выравнивания и другие параметры представления.
- По умолчанию компонент сведений о задаче разворачивается, чтобы отобразить все указанные сведения. Нажмите **Установить высоту** и укажите значение высоты, чтобы задать указанную высоту.
7. Отформатировать фрейм можно на вкладке "Стиль", которая содержит компонент с предопределенными стилями или с пользовательскими вариантами выбора.
8. Используйте вкладку "Ссылки" для назначения триггеров, предоставленных компонентом.
- **Отправлены сведения о задаче**: этот триггер активируется, когда пользователь нажимает кнопку **Отправить**.
 - **Утверждена задача**: этот триггер активируется, когда пользователь нажимает кнопку **Утвердить**.
 - **Отклонена задача**: этот триггер активируется, когда пользователь нажимает кнопку **Отклонить**.
 - **Закрыта задача**: этот триггер активируется, когда пользователь нажимает кнопку **Заккрыть**.
 - **Сохранена задача**: этот триггер активируется, когда пользователь нажимает кнопку **Сохранить**.
 - **Добавлен комментарий к задаче**: этот триггер активируется, когда пользователь добавляет текст комментария и нажимает кнопку **Опубликовать комментарий**.

Нажмите на триггер, чтобы назначить действие.

Примечание.

Компонент списка задач процесса не поддерживает ручные триггеры или действия, однако если добавить на страницу компонент сведений о задаче, он автоматически отображает сведения о задаче, выбранной в данном списке. Чтобы узнать больше о триггерах и действиях, см. [Использование триггеров и действий](#).

Информацию об использовании Oracle Process Cloud Service см. в Начало работы с процессом.

Элементы содержимого

Для каждого сайта создается коллекция, которая содержит цифровые активы и элементы содержимого, связанные с этим сайтом. Эти элементы доступны для корпоративных пользователей. Компоненты элементов содержимого упрощают добавление элементов на сайт.

Цифровые активы и элементы содержимого можно перетаскивать непосредственно с панели "Содержимое" в редакторе. При этом для всех элементов (например, цифровых активов изображений или структурированных элементов содержимого) автоматически используются соответствующие компоненты.

Дополнительные компоненты, доступные на панели "Компонент", позволяют динамически отображать элементы содержимого в зависимости от типа содержимого.

- [Компонент элемента содержимого](#)
- [Заполнитель содержимого](#)
- [Список содержимого](#)
- [Поиск в содержимом](#)
- [Рекомендация](#)

Компонент элемента содержимого

Корпоративные пользователи могут использовать компонент элемента содержимого для разметки страницы и настройки взаимодействий на странице. Затем можно добавить сами элементы содержимого.

При перетаскивании элемента контента с панели "Контент" на страницу компонент элемента контента автоматически вставляется для удержания элемента, если только этот элемент контента не относится к типу "пользовательский цифровой актив". Перетаскивание пользовательского цифрового актива на страницу приводит к вставке этого актива в качестве компонента изображения или видео. Сначала необходимо удалить его и добавить компонент элемента контента, а затем перетащить пользовательский цифровой актив или элемент стандартного контента с панели "Контент" на компонент для надлежащего отображения.

Добавление компонента элемента содержимого на страницу:

1. Перейдите к странице, которую хотите изменить, и убедитесь, что переключатель  переведен в положение **Редактировать**.
2. Добавьте компонент на страницу.
3. Для редактирования контента компонента "элемент контента", нажмите на его значок меню  и выберите **Редактирование** или **Настройки**, чтобы открыть панель настроек, и нажмите на значок редактирования .

 **Примечание.**

Если контент, используемый в компоненте элемента контента, относится к типу "пользовательский цифровой актив", а не к типу "стандартный контент", редактировать актив типа "пользовательский цифровой актив" невозможно. В меню компонента не будет параметра **Редактировать**, а на панели настроек не будет значка редактирования.

4. Внесите изменения в элемент контента и нажмите **Сохранить**. Элемент контента сохраняется в репозитории контента в качестве новой версии.
5. Чтобы изменить внешний вид компонента, нажмите на его значок меню  и выберите **Настройки**.
6. На вкладке "Общие" можно изменить интервалы, тип выравнивания и другие параметры представления.

Если с этим компонентом связан элемент содержимого, отображается эскиз для просмотра элементов содержимого. Если элементы содержимого еще не назначены, отображается замещающее изображение и сообщение.

7. Выберите один из следующих вариантов, чтобы настроить дополнительные значения по умолчанию для отображаемого содержимого.
 - **Версия для использования.** Если выбран параметр **Использовать последнюю версию актива** и для данного элемента содержимого существует более новая неопубликованная версия, эта новая версия будет опубликована автоматически во время публикации текущего обновления сайта. По умолчанию все элементы являются черновиками или последними версиями.
 - **Представление элемента.** Выберите макет для отображения элемента содержимого. В предоставленном макете **по умолчанию** отображаются все поля элемента содержимого. Если для элемента содержимого настроены другие, пользовательские макеты, можно выбрать любой доступный макет. Если выбран пользовательский макет, для которого включена функция **Добавить поддержку пользовательских настроек при использовании на сайтах**, можно добавить пользовательские данные в макет, нажав  и введя данные в поле **Пользовательские данные**, а затем нажать кнопку **Назад**, чтобы вернуться на вкладку "Общие".
 - **Страница для отображения отдельного элемента.** Здесь перечисляются страницы, выбранные в качестве страниц сведений. Выберите страницу для отображения подробной информации, когда пользователь переходит по ссылке в элементе содержимого.

 **Примечание.**

Если страница сведений не создана, в стандартном макете элемента содержимого отсутствует ссылка на подробные сведения.

8. Отформатировать фрейм можно на вкладке "Стиль", которая содержит компонент с предопределенными стилями или с пользовательскими вариантами выбора.

Заполнитель содержимого

Компоненты заполнителя содержимого позволяют динамически отображать элементы содержимого одного или нескольких типов.

Например, если на странице сведений создан заполнитель элемента содержимого, то при нажатии на ссылку для просмотра сведений об элементе содержимого автоматически загружается подробное представление для соответствующего элемента. Если на странице размещено несколько статей с заголовками и иллюстрациями, то при нажатии на определенную статью открывается страница сведений с полным текстом статьи (независимо от того, какая выбрана статья).

Добавление компонента заполнителя содержимого на страницу:

1. Перейдите к странице, которую хотите изменить, и убедитесь, что переключатель  переведен в положение **Редактировать**.
2. Добавьте компонент на страницу.
3. Для редактирования компонента и его внешнего вида нажмите на значок меню приложения  и выберите **Настройки**.
4. На вкладке "Общие" можно изменить интервалы, тип выравнивания и другие параметры представления.
5. Выберите один из следующих вариантов, чтобы настроить дополнительные значения по умолчанию для отображаемого содержимого.
 - **Тип содержимого.** Выберите один или несколько доступных типов содержимого. Типы контента — это типы элементов контента в коллекции сайта, которые включают типы пользовательских цифровых активов.
 - **Представление элемента.** Выберите макет для отображения элемента содержимого. В предоставленном макете **по умолчанию** отображаются все поля элемента содержимого. Если для элемента содержимого настроены другие, пользовательские макеты, можно выбрать любой доступный макет.
 - **Страница для отображения отдельного элемента.** Здесь перечисляются страницы, выбранные в качестве страниц сведений. Выберите страницу для отображения подробной информации, когда пользователь переходит по ссылке в элементе содержимого.
6. Отформатировать фрейм можно на вкладке "Стиль", которая содержит компонент с predefined стилями или с пользовательскими вариантами выбора.

Примечание.

Если страница сведений не создана, в стандартном макете элемента содержимого отсутствует ссылка на подробные сведения.

Список содержимого

Корпоративный пользователь может создать компонент списка содержимого, в котором будут динамически отображаться элементы содержимого определенного типа.

Например, если на странице сведений создан список элементов контента, то при нажатии на ссылку для просмотра сведений об элементе контента автоматически загружается подробное представление для соответствующего элемента. Если на странице размещено несколько статей с заголовками и иллюстрациями, то при нажатии на определенную статью открывается страница сведений с полным текстом статьи (независимо от того, какая выбрана статья).

Чтобы добавить компонент "Список контента" на страницу, выполните следующие действия.

1. Перейдите к странице, которую хотите изменить, и убедитесь, что переключатель  переведен в положение **Редактировать**.
2. Добавьте компонент на страницу.
3. Для редактирования компонента и его внешнего вида нажмите на значок меню приложения  и выберите **Настройки**.
4. На вкладке "Общие" можно изменить интервалы, тип выравнивания и другие параметры представления.
5. Выберите один из следующих вариантов, чтобы настроить дополнительные значения по умолчанию для отображаемого содержимого.
 - **Тип содержимого.** Выберите один из доступных типов содержимого. Типы контента — это типы элементов контента в коллекции сайта, которые включают типы пользовательских цифровых активов.
 - **Максимальное количество элементов и Начать с элемента.** Укажите максимальное количество элементов для отображения и элемент, с которого необходимо начать отображение. Дополнительные элементы не отображаются.
 - **Разбивка на страницы.** Укажите, нужно ли включать в список параметры разбивки на страницы. Если этот параметр выбран, можно настроить отображение разбивки на страницы с помощью кнопок или номеров страниц и различных подписей.
 - **Дата.** Используйте параметры в этом разделе, чтобы элементы содержимого отображались в зависимости от даты создания (перед, после, между или в пределах указанных дат или диапазонов).
 - **Категории.** Выберите категории, чтобы отфильтровать список элементов контента, чтобы показать только эти элементы в определенной категории или категориях. Нажмите **Выбрать категории**, чтобы открыть выдвижную панель, которую можно использовать для выбора категорий в репозитории сайта. Если выбрано несколько категорий в одной таксономии, то отображаемым элементам должна быть назначена одна из выбранных категорий. Если выбрано несколько категорий из разных таксономий, то отображаемым элементам должны быть назначены все выбранные категории. Например, если в таксономии **Пункт назначения** выбраны категории **Города** и **Парки**, отображаются элементы, которые являются городами или парками. При выборе категории **Города** в таксономии **Пункт назначения** и категории **Европа** в таксономии **Регионы** отображаются элементы, являющиеся городами Европы.

По умолчанию будут доступны все дочерние элементы выбранной категории. Чтобы в списке отображались только элементы в узле определенной категории, выберите этот узел и снимите флажок **Включить дочерние категории**.

- **Язык.** Выберите язык из списка языков, связанных с репозиторием.
- **Дополнительная строка запроса** (необязательный параметр). Укажите дополнительные параметры запроса для дальнейшего уточнения отображаемого списка элементов. Пример синтаксиса: `field.dept eq "Finance"`. В таблице ниже приведен список доступных операторов.
- **Упорядочить по.** Сортировка элементов по имени или дате по возрастанию или убыванию. Если тип контента содержит другие поля даты, числа или десятичного разделителя, эти поля также можно отсортировать.

Также можно выбрать **Пользовательский** и ввести пользовательское выражение. Например, для сортировки по номеру отделения в порядке возрастания можно ввести `fields.deptno:asc`. Также можно задать параметр URL для сортировки. Например, можно ввести `{{URLParams.sortBy}}`, а затем добавить следующий параметр в конец URL-адреса сайта: `?sortBy=fields.deptno:asc`.

- **Представление элемента.** Выберите макет для отображения элемента содержимого. В предоставленном макете **по умолчанию** отображаются все поля элемента содержимого. Если для элемента содержимого настроены другие, пользовательские макеты, можно выбрать любой доступный макет. Если выбран пользовательский макет, для которого включена функция **Добавить поддержку пользовательских настроек при использовании на сайтах**, можно добавить пользовательские

данные в макет, нажав  и введя данные в поле **Пользовательские данные**, а затем нажать кнопку **Назад**, чтобы вернуться на вкладку "Общие".

- **Страница для отображения отдельного элемента.** Здесь перечисляются страницы, выбранные в качестве страниц сведений. Выберите страницу для отображения подробной информации, когда пользователь переходит по ссылке в элементе содержимого.

Примечание.

Если страница сведений не создана, в стандартном макете элемента содержимого отсутствует ссылка на подробные сведения.

- **Представление списка.** Выберите макет раздела для упорядочения элементов. Этот список может включать в себя пользовательские макеты. Доступны следующие макеты:
 - **Горизонтальный.** Элементы располагаются друг за другом по горизонтали.
 - **Вертикальный.** Элементы располагаются друг за другом по вертикали. Этот макет используется по умолчанию, если не выбран другой.

- **Два столбца.** Элементы отображаются попарно в нескольких строках.
- **Три столбца.** Элементы отображаются по три в нескольких строках.

Нажмите на стрелку вправо рядом с выбранным макетом, чтобы изменить его параметры.

- **Представление пустого списка.** Выберите способ отображения пустого списка.
 - **Параметры.** Укажите, нужно ли выполнять автоматическое обновление поискового запроса, указанного в компоненте поиска содержимого.
 - **Пространство.** Выберите способ отображения элементов.
6. Отформатировать фрейм можно на вкладке "Стиль", которая содержит компонент с predetermined styles or user-selectable options.

Таблица 10-1 Операторы запросов

Оператор	Пример	Поддерживаемые типы данных	Описание
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"	текст, ссылка, число, десятичное число, логическое значение, дата и время	Оператор Equals (eq) соответствует точному значению, указанному в запросе. Этот оператор не применим к многозначным типам данных. Значение, указанное этим оператором, не учитывает регистр, кроме стандартных полей. Этот оператор учитывает даже специальные символы в значении.

Таблица 10-1 (Продолж.) Операторы запросов

Оператор	Пример	Поддерживаемые типы данных	Описание
co	?q=(type eq "Employee" AND name co "john alex") ?q=(type eq "Car" AND fields.features co "manual")	текст, ссылка, число, десятичное время, большой текст	Оператор Contains (co) соответствует каждому слову, указанному в критериях. Слова формируются путем разделения значения на специальные символы. Это дает результаты, которые включают по крайней мере одно из слов (в этом примере, John или Alex, или оба). Этот оператор не учитывает специальные символы в значении при поиске. Этот оператор не выполняет поиск по стоп-словам. Обратитесь к документации Arache Lucene, чтобы узнать больше о стоп-словах. Этот оператор применим к <i>тексту, большому тексту</i> в случае однозначных атрибутов, тогда как для многозначных атрибутов он применим к <i>тексту, ссылке, числу, десятичному числу, дате и времени, большому тексту</i> . Чтобы понять возможные форматы даты и времени, обратитесь к таблице <i>Поддерживаемые форматы даты / даты и времени (24 часа)</i> ниже. Значение, предоставляемое этим оператором, не учитывает регистр.
sw	?q=type eq "Employee" AND name sw "Joh" ?q=type eq "Employee" AND fields.city sw "Los"	текст	Оператор Starts with (sw) соответствует только начальным символьным значениям, заданным в условии поля. Этот оператор не применим к многозначным типам данных. Значение, предоставляемое этим оператором, не учитывает регистр.

Таблица 10-1 (Продолж.) Операторы запросов

Оператор	Пример	Поддерживаемые типы данных	Описание
ge	?q=(type eq "Employee" AND fields.age ge "40") ?q=type eq "DigitalAsset" AND updatedDate ge "20171026"	число, десятичное число, дата и время	Оператор Greater than or equal to (ge) соответствует только числовым значениям и значениям даты и времени. Чтобы понять возможные форматы даты и времени, обратитесь к таблице <i>Поддерживаемые форматы даты / даты и времени (24 часа)</i> ниже. Этот оператор не применим к многозначным типам данных.
le	?q=(type eq "Employee" AND fields.weight le "60.6")	число, десятичное число, дата и время	Оператор Less than or equal to (le) соответствует только числовым значениям и значениям даты и времени. Чтобы понять возможные форматы даты и времени, обратитесь к таблице <i>Поддерживаемые форматы даты / даты и времени (24 часа)</i> ниже. Этот оператор не применим к многозначным типам данных.
gt	?q=(type eq "Employee" AND fields.age gt "20")	число, десятичное число, дата и время	Оператор Greater than (gt) соответствует только числовым значениям и значениям даты и времени. Чтобы понять возможные форматы даты и времени, обратитесь к таблице <i>Поддерживаемые форматы даты / даты и времени (24 часа)</i> ниже. Чтобы понять возможные форматы даты и времени, обратитесь к таблице <i>Поддерживаемые форматы даты / даты и времени (24 часа)</i> ниже. Этот оператор не применим к многозначным типам данных.

Таблица 10-1 (Продолж.) Операторы запросов

Оператор	Пример	Поддерживаемые типы данных	Описание
lt	?q=(type eq "Employee" AND fields.age lt "20") ?q=type eq "Employee" AND createdAt lt "1994/09/26T16:23:45.208"	число, десятичное число, дата и время	Оператор Less than (lt) соответствует только числовым значениям и значениям даты и времени. Чтобы понять возможные форматы даты и времени, обратитесь к разделу Поддерживаемые форматы даты / даты и времени (24 часа) . Этот оператор не применим к многозначным типам данных.

Таблица 10-1 (Продолж.) Операторы запросов

Оператор	Пример	Поддерживаемые типы данных	Описание
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>	текст, большой текст	<p>Оператор фразового запроса или поиска с учетом расположения (matches) (mt) позволяет находить слова, которые находятся на определенном расстоянии друг от друга. Результаты сортируются по лучшему совпадению. Это полезно для поиска элементов содержимого, когда значения, заданные в критериях "бензин 20 км/л", должны обнаружить фактическое содержимое, которое может содержать "пробег на бензиновом топливе составляет 20 км/л по автостраде".</p> <p>Оператор Matches также может использовать подстановочный знак в заданном значении и поддерживает поиск как односимвольных, так и многосимвольных подстановочных знаков для поиска в пределах одного значения. Используйте "?" для подстановочного знака одного символа и "*" для нескольких символов. "John" и "Joan" могут быть найдены с помощью "Jo?n" для одного символа и "Jo*" для нескольких символов.</p> <p>Этот оператор применим как к однозначным, так и к многозначным типам данных. Этот оператор не выполняет поиск по стоп-словам. Обратитесь к документации Apache Lucene, чтобы узнать больше о стоп-словах. Значение, предоставляемое этим оператором, не учитывает регистр.</p>

Таблица 10-1 (Продолж.) Операторы запросов

Оператор	Пример	Поддерживаемые типы данных	Описание
sm	?q=(type eq "Employee" And fields.city sm "Rome")	текст, большой текст	Оператор запроса сходства. Этот оператор позволяет искать значения, которые похожи на заданные критерии. Также называется нечетким поиском, в котором по умолчанию используется не более двух правок для соответствия результату. "Rome" имеет сходство с "Dome". Этот оператор применим как к однозначным, так и к многозначным типам данных. Значение, предоставляемое этим оператором, не учитывает регистр.
AND	?q=(type eq "Employee" AND name eq "John" AND fields.age ge "40")	Н/Д	Оператор AND можно использовать для помещения условия И между несколькими условиями запроса. Он имеет приоритет перед OR.
OR	type eq "Employee" AND name eq "John" OR fields.age ge "40"	Н/Д	Оператор OR может использоваться для помещения условия ИЛИ между несколькими условиями запроса.
()	?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)	Н/Д	Скобки, окружающие оператор для группировки условий в критериях. Они имеют наивысший приоритет, затем следует AND, а затем OR.

Поиск в содержимом

Корпоративные пользователи могут использовать компонент поиска содержимого и указывать действия, возвращаемые в результатах поиска.

Можно вставить настроенную панель поиска для изменения или обновления содержимого, которое отображается на странице, или выбрать другое действие, например открытие результатов поиска или отображение уведомления.

Чтобы добавить на страницу компонент поиска содержимого, выполните следующие действия.

1. Перейдите к странице, которую хотите изменить, и убедитесь, что переключатель  переведен в положение **Редактировать**.
2. Добавьте компонент поиска содержимого на страницу.
3. Для редактирования компонента и его внешнего вида нажмите на значок меню приложения  и выберите **Настройки**.
4. На вкладке "Общие" можно изменить замещающий текст, интервалы, тип выравнивания и другие параметры представления.
5. Отформатировать фрейм можно на вкладке "Стиль", которая содержит компонент с предопределенными стилями или с пользовательскими вариантами выбора шрифтов, границ, цвета фона и т. д.
6. Если компонент поиска содержимого будет использоваться для обновления данных на текущей странице, добавьте на страницу компонент списка содержимого. Перейдите в раздел **Настройки**, затем выберите тип содержимого и задайте другие параметры запроса, например число отображаемых элементов, разбиение результатов на страницы (да/нет) или медленная загрузка при прокрутке (загрузка содержимого при прокрутке страницы). Теперь можно перейти в режим просмотра и выполнить поиск.
7. Если компонент поиска содержимого будет использоваться для обновления данных на текущей странице и на странице есть несколько компонентов списка содержимого, необходимо отключить автоматический запрос для всех списков содержимого, за исключением списка, в котором будут отображаться результаты. Перейдите в раздел настроек списка содержимого и снимите флажок **Автообновление при поисковом запросе в компоненте поиска содержимого**.
8. Если на странице должно быть несколько компонентов поиска содержимого, каждый из которых создает определенный список содержимого для результатов поиска, необходимо снять флажки автообновления в компонентах списка содержимого и связать компонент поиска содержимого с соответствующим компонентом списка в разделе "Триггер/действия".
9. На вкладке "Ссылка" в компоненте поиска можно связать действия с компонентом. Выберите нужный пункт в раскрывающемся списке "Выбрать тип ссылки". Выберите **Действия триггера**, затем нажмите **Для поискового запроса**, чтобы просмотреть доступные триггеры или создать новый. Найдите список контента для поиска, разверните его и перетащите **Поиск контента** в список действий. В разделе **Поиск** выберите **Строка поиска**. Выбранный список контента выбирается в списке **Выполнять это действие в**.
10. Также можно настроить компонент поиска для передачи запроса на страницу результатов поиска. На вкладке "Ссылка" в компоненте поиска выберите **Страница поиска**, чтобы открыть страницу результатов поиска. Можно использовать страницу по умолчанию или создать ссылку на созданную страницу результатов поиска. Выберите страницу в раскрывающемся списке "Страница", а затем выберите действия отображения для этой страницы. Можно адаптировать результаты поиска к конкретному типу содержимого, открыть результаты в новом окне и т. д.

Эти указатели помогут вам создать удобную страницу результатов поиска:

- Создание страницы и присвоение ей функций страницы поиска. По умолчанию она помечена как скрытая, но эту настройку можно изменить в свойствах страницы.
- Редактирование страницы и добавление списка содержимого на страницу. Можно изменить параметры списка содержимого и указать тип содержимого. Тип содержимого также можно задать в настройках ссылки в компоненте содержимого

(см. выше). Настройки ссылки позволяют использовать страницу результатов содержимого, где отображаются результаты для различных типов содержимого, в зависимости от компонента, из которого инициирован поиск.

- Чтобы изменить область отображения, измените настройки, например разбиение на страницы с медленной загрузкой (поскольку эта страница скорее всего будет использоваться только для отображения результатов поиска).
- На странице результатов поиска также можно вставить компонент поиска содержимого. Он будет копией строки поиска, используемой для открытия этой страницы, и пользователь сможет уточнить поиск.

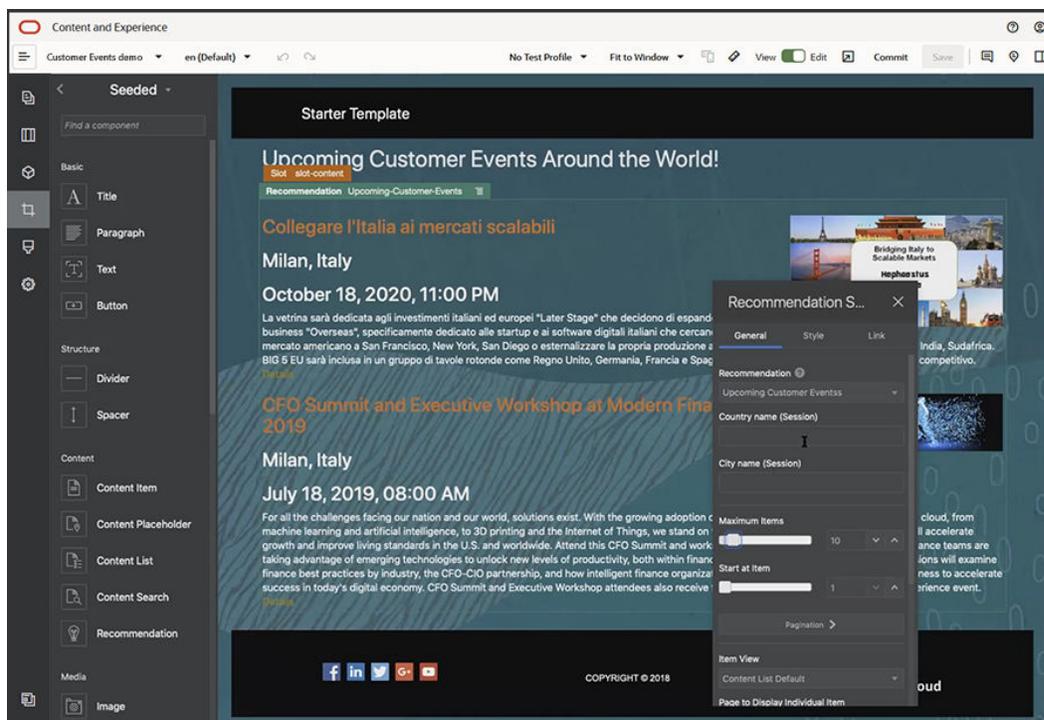
Рекомендация

Как корпоративный пользователь вы можете использовать компонент **Рекомендацию**, чтобы предоставлять посетителям веб-сайта персонализированные возможности, показывая активы на основе местоположения или областей интересов. Когда соавторы репозитория создают рекомендации, они определяют набор правил для поиска активов, соответствующих атрибутам аудитории, таким как информация о местоположении посетителей сайта на земной поверхности. Например, посетители сайта с европейскими IP-адресами могут сначала увидеть на домашней странице сайта объявления о событиях для Европы, а посетители сайта из Северной Америки увидят события в США и Канаде.

Как и цифровые активы, рекомендации связаны с репозиторием и могут редактироваться и перемещаться по рабочему процессу для проверки и публикации любым пользователем с ролью "Соавторы контента" для репозитория. Однако даже если у вас нет прав соавтора, вы по-прежнему можете просматривать и тестировать рекомендации, чтобы узнать, как они работают, прежде чем использовать их на сайте или в автономной среде.

Чтобы добавить рекомендацию на страницу, выполните указанные ниже действия.

1. Нажмите **Сайты** в боковом меню, выберите сайт, на который необходимо добавить рекомендацию, затем выберите **Открыть** в контекстном меню или нажмите  на панели действий.
2. Переключите сайт в режим **редактирования** и выберите данное обновление, чтобы использовать или создать новое.
3. В боковом меню выберите **Компоненты**.
4. Откройте **Заданные предварительно**.
5. Нажмите компонент **Рекомендация** в разделе "Контент" и перетащите его на страницу.
6. Выберите **Настройки** в меню "Рекомендация" (.



7. В настройках рекомендации на вкладке **Общие** выберите рекомендацию для использования. Страница обновляется и отображает рекомендуемые активы.
8. При необходимости введите значения по умолчанию для атрибутов аудитории, используемых данной рекомендацией, нажав стрелку рядом с выбранной рекомендацией. Завершив добавление значений, нажмите **Назад**.
9. Измените нужные дополнительные свойства, например, какую версию использовать, максимальное количество отображаемых элементов или макет контента по умолчанию. Когда закончите, закройте настройки рекомендаций и нажмите **Сохранить**.

Примечание.

Если вы создали тестовые профили с предварительно заданными значениями атрибутов аудитории, в строке меню выберите профиль в меню тестовых профилей, чтобы предварительно просмотреть, как рекомендация реагирует на эти входные данные. Значения тестового профиля переопределяют значения по умолчанию, заданные в настройках компонента.

10. Если в рекомендациях используется текущая дата (системы), чтобы проверить рекомендацию для другой даты, в меню тестовых профиля выберите **Системная дата** и выберите дату тестирования.

Другие компоненты

Рассмотрим некоторые компоненты, которые содержат разные типы содержимого.

- **Карты**

- [Заголовки](#)
- [Статьи](#)
- [Изображения с текстом](#)
- [Группы компонентов](#)
- [Использование компонента "Совместный просмотр" на странице](#)
- [Oracle Intelligent Advisor](#)
- [Oracle Visual Builder](#)

Карты

Добавьте к своему сайту карту, чтобы пользователи могли осматривать область вокруг местоположения в интерактивном режиме.



Чтобы добавить карту на страницу, выполните указанные ниже действия.

1. Перейдите к странице, которую хотите изменить, и убедитесь, что переключатель  переведен в положение **Редактировать**.
2. Добавьте компонент на страницу. Компонент-карта Oracle показывает местоположение, заданное по умолчанию.
3. Чтобы изменить местонахождение на карте и настроить свойства ее отображения, щелкните значок меню карты  и выберите пункт **Настройки**.
4. Введите для карты начальное **Местонахождение**. Можно использовать адрес, почтовый индекс или разделенные запятой значения широты и долготы (например, 40.5,-57.6)
При использовании адреса необходимо предоставить достаточно информации, чтобы установить соответствие с одним местонахождением. Если адрес соответствует нескольким местоположениям, карта остается пустой.
5. Укажите для карты исходный уровень **масштаба**. Дополнительно можно разрешить пользователям регулировать уровень масштабирования с помощью мыши, сенсорной площадки или экранных элементов управления.
6. Укажите стиль, выравнивание и параметры отображения.
 - **Масштабировать**: позволяет пользователям регулировать уровень масштаба карты с помощью элементов управления на экране или устройстве.

- **Панорамировать**: позволяет пользователям перемещать область фокуса карты с помощью элементов управления на экране или устройстве.
- **Экранный маркер**: начальное местоположение помечается значком булавки.
- **Масштаб**: включает индикатор, показывающий уровень текущего масштаба.
- **Обзор**: включает карту-врезку, на которой отображается текущий вид в более крупном контексте.

Заголовки

Чтобы привлечь внимание к определенному месту на сайте, можно добавить заголовок с текстом, вспомогательным изображением и абзацем.

1. Перейдите к странице, которую хотите изменить, и убедитесь, что переключатель  переведен в положение **Редактировать**.
2. Добавьте компонент на страницу.
3. Для добавления текста нажимайте на разные области компонента. Добавьте содержимое в основной заголовок и в абзац под основным заголовком. На данный текст распространяется форматирование стиля по умолчанию данного компонента. Для добавления строк текста нажмите Enter.
4. Чтобы изменить форматирование по умолчанию любой части текста, выделите текст, который нужно форматировать, затем выберите на панели инструментов форматирования любой из вариантов форматирования, такой как шрифт, цвет или выравнивание.
5. Чтобы удалить форматирование, примененное с помощью этих инструментов, выберите текст и щелкните .
6. Для редактирования компонента и его внешнего вида нажмите на значок меню приложения  и выберите **Настройки**.
7. Выберите **Компоненты** для настройки сведений об отображаемом содержимом.
 - **Изображение**:
 - Нажмите **Выбрать** для использования изображения в заголовке. В репозитории сайтов в папке документов выберите изображение, к которому вам был предоставлен общий доступ, или загрузите изображение в папку документов. Найдите и выберите изображение, которое требуется использовать. Если цифровые активы не отображаются, нажмите  и измените фильтр по коллекциям на вариант **Все**. Выбрав изображение в папке документов, можно создать ссылку на файл, не копируя его на сайт. Чтобы добавить ссылку на файл, установите флажок **Использовать ссылку на исходный файл вместо копирования файла на сайт**. В противном случае копия файла сохраняется на самом сайте. Добавление ссылки на исходный файл позволяет избежать дублирования содержимого. Кроме того, ссылка дает посетителям возможность посмотреть содержимое, даже если изменяются разрешения на доступ к файлу или просмотр файла. Нажмите **ОК**. Выбрав цифровой актив, можно выбрать определенное представление. Если не выбирать представление, будет использоваться исходный размер. Чтобы вместе с сайтом публиковалась последняя версия актива, установите флажок **Использовать последнюю версию актива**.

Чтобы отредактировать изображение, нажмите , и отредактируйте изображение, выполнив любое из следующих действий:

- * Чтобы обрезать изображение, нажмите  **Обрезать**. Выберите на панели инструментов обрезки одно из предварительно определенных соотношений сторон изображения или перетащите маркеры обрезки на изображение по своему усмотрению. Добившись нужного результата, на панели инструментов обрезки нажмите **Обрезать**.
 - * Чтобы повернуть или отразить изображение, нажмите  **Повернуть**. На панели инструментов поворота введите произвольный угол поворота в градусах или с помощью кнопок поверните изображение влево или вправо или выберите вариант отражения изображения: по горизонтали или по вертикали.
 - * Чтобы добавить водяной знак на изображение, нажмите  **Водяной знак**. Добавьте текст к изображению, изменив его размер, стиль, цвет и непрозрачность с помощью инструментов водяного знака.
 - * Чтобы изменить формат изображения, нажмите  **Параметры**, затем в раскрывающемся списке **Формат** выберите новый формат.
 - * Чтобы изменить цвет фона, нажмите  **Параметры**, затем в раскрывающемся меню **Цвет фона** выберите нужный вариант.
 - * При редактировании файлов .jpg или .webp (доступно в браузерах Google Chrome) можно изменить качество изображения, чтобы создать файл меньшего размера. Нажмите  **Параметры** и введите новое процентное значение в поле (**Качество**).
 - * Чтобы отменить или повторить изменение, нажмите  или . Чтобы удалить все внесенные изменения, нажмите **Сброс**.
 - * Чтобы изменить размер представления изображения, используйте элементы управления масштабированием ().
- **Название**: введите текст, который будет отображаться в виде всплывающей подсказки.
 - **Альтернативный текст**: введите альтернативный текст, который будет отображаться для поддержки специальных возможностей.
 - **Подпись**: введите подпись, которая указывается под изображением.
 - **Тип выравнивания, ширина и интервалы**: измените макет изображения, если требуется.
- **Заголовок**: изменение интервалов в тексте заголовка.
 - **Абзац**: изменение интервалов в тексте, который отображается под заголовком.

8. На вкладке "Общие" можно изменить интервалы, тип выравнивания и другие параметры представления.
9. На вкладке "Стиль" можно добавить форматирование вокруг текста и настроить цвет фона, шрифты и границы.

Статьи

На сайт можно добавить статью, которая сочетает в себе компоненты заголовка, абзаца и изображения в одном удобном компоненте.

1. Перейдите к странице, которую хотите изменить, и убедитесь, что переключатель  переведен в положение **Редактировать**.
2. Добавьте компонент на страницу.
3. Для добавления текста нажимайте на разные области компонента. Добавьте содержимое в заголовок статьи и в подзаголовок. Затем можно добавить текст статьи после подзаголовка. Весь текст отформатирован в соответствии со стилем, используемым для компонента по умолчанию. Для добавления строк текста нажмите Enter.
4. Чтобы изменить форматирование по умолчанию любой части текста, выделите текст, который нужно форматировать, затем выберите на панели инструментов форматирования любой из вариантов форматирования, такой как шрифт, цвет или выравнивание.
5. Чтобы удалить форматирование, примененное с помощью этих инструментов, выберите текст и щелкните .
6. Для редактирования компонента и его внешнего вида нажмите на значок меню приложения  и выберите **Настройки**.
7. Выберите **Компоненты** для настройки сведений об отображаемом содержимом.
 - **Изображение:**
 - Нажмите **Выбрать**, чтобы использовать изображение в данной статье. В репозитории сайтов в папке документов выберите изображение, к которому вам был предоставлен общий доступ, или загрузите изображение в папку документов. Найдите и выберите изображение, которое требуется использовать. Если цифровые активы не отображаются, нажмите  и измените фильтр по коллекциям на вариант **Все**. Выбрав изображение в папке документов, можно создать ссылку на файл, не копируя его на сайт. Чтобы добавить ссылку на файл, установите флажок **Использовать ссылку на исходный файл вместо копирования файла на сайт**. В противном случае копия файла сохраняется на самом сайте. Добавление ссылки на исходный файл позволяет избежать дублирования содержимого. Кроме того, ссылка дает посетителям возможность посмотреть содержимое, даже если изменяются разрешения на доступ к файлу или просмотр файла. Нажмите **ОК**. Выбрав цифровой актив, можно выбрать определенное представление. Если не выбирать представление, будет использоваться исходный размер. Чтобы вместе с сайтом публиковалась последняя версия актива, установите флажок **Использовать последнюю версию актива**.

Чтобы отредактировать изображение, нажмите , и отредактируйте изображение, выполнив любое из следующих действий:

- * Чтобы обрезать изображение, нажмите  **Обрезать**. Выберите на панели инструментов обрезки одно из предварительно определенных соотношений сторон изображения или перетащите маркеры обрезки на изображение по своему усмотрению. Добившись нужного результата, на панели инструментов обрезки нажмите **Обрезать**.
 - * Чтобы повернуть или отразить изображение, нажмите  **Повернуть**. На панели инструментов поворота введите произвольный угол поворота в градусах или с помощью кнопок поверните изображение влево или вправо или выберите вариант отражения изображения: по горизонтали или по вертикали.
 - * Чтобы добавить водяной знак на изображение, нажмите  **Водяной знак**. Добавьте текст к изображению, изменив его размер, стиль, цвет и непрозрачность с помощью инструментов водяного знака.
 - * Чтобы изменить формат изображения, нажмите  **Параметры**, затем в раскрывающемся списке **Формат** выберите новый формат.
 - * Чтобы изменить цвет фона, нажмите  **Параметры**, затем в раскрывающемся меню **Цвет фона** выберите нужный вариант.
 - * При редактировании файлов .jpg или .webp (доступно в браузерах Google Chrome) можно изменить качество изображения, чтобы создать файл меньшего размера. Нажмите  **Параметры** и введите новое процентное значение в поле (**Качество**).
 - * Чтобы отменить или повторить изменение, нажмите  или . Чтобы удалить все внесенные изменения, нажмите **Сброс**.
 - * Чтобы изменить размер представления изображения, используйте элементы управления масштабированием ().
- **Название**: введите текст, который будет отображаться в виде всплывающей подсказки.
 - **Альтернативный текст**: введите альтернативный текст, который будет отображаться для поддержки специальных возможностей.
 - **Подпись**: введите подпись, которая указывается под изображением.
 - **Тип выравнивания, ширина и интервалы**: измените макет изображения, если требуется.
- **Заголовок**: изменение интервалов в тексте заголовка статьи.
 - **Абзац** (подзаголовок) и **Абзац**: изменение интервалов в тексте, который отображается под заголовком статьи.

8. На вкладке "Общие" можно изменить интервалы, тип выравнивания и другие параметры представления компонента.
9. На вкладке "Стиль" можно добавить форматирование вокруг текста и настроить цвет фона, шрифты и границы.

Изображения с текстом

Вы можете добавить компонент "Изображение и текст", который объединяет компоненты абзаца и изображения в одном удобном компоненте.

1. Перейдите к странице, которую хотите изменить, и убедитесь, что переключатель  переведен в положение **Редактировать**.
2. Добавьте компонент на страницу.
3. Для добавления текста нажмите в области абзаца компонента. Весь текст отформатирован в соответствии со стилем, используемым для компонента по умолчанию. Для добавления строки текста нажмите Enter.
4. Чтобы изменить форматирование по умолчанию любой части текста, выделите текст, который нужно форматировать, затем выберите на панели инструментов форматирования любой из вариантов форматирования, такой как шрифт, цвет или выравнивание.
5. Чтобы удалить форматирование, примененное с помощью этих инструментов, выберите текст и щелкните .
6. Для редактирования компонента и его внешнего вида нажмите на значок меню приложения  и выберите **Настройки**.
7. Выберите **Компоненты** для настройки сведений об отображаемом содержимом.
 - **Изображение:**
 - Нажмите **Выбрать**, чтобы использовать изображение. В репозитории сайтов в папке документов выберите изображение, к которому вам был предоставлен общий доступ, или загрузите изображение в папку документов. Найдите и выберите изображение, которое требуется использовать. Если цифровые активы не отображаются, нажмите  и измените фильтр по коллекциям на вариант **Все**. Выбрав изображение в папке документов, можно создать ссылку на файл, не копируя его на сайт. Чтобы добавить ссылку на файл, установите флажок **Использовать ссылку на исходный файл вместо копирования файла на сайт**. В противном случае копия файла сохраняется на самом сайте. Добавление ссылки на исходный файл позволяет избежать дублирования содержимого. Кроме того, ссылка дает посетителям возможность посмотреть содержимое, даже если изменяются разрешения на доступ к файлу или просмотр файла. Нажмите **ОК**. Выбрав цифровой актив, можно выбрать определенное представление. Если не выбирать представление, будет использоваться исходный размер. Чтобы вместе с сайтом публиковалась последняя версия актива, установите флажок **Использовать последнюю версию актива**.

Чтобы отредактировать изображение, нажмите , и отредактируйте изображение, выполнив любое из следующих действий:

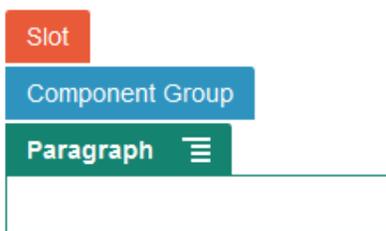
- * Чтобы обрезать изображение, нажмите  **Обрезать**. Выберите на панели инструментов обрезки одно из предварительно определенных соотношений сторон изображения или перетащите маркеры обрезки на изображение по своему усмотрению. Добившись нужного результата, на панели инструментов обрезки нажмите **Обрезать**.
 - * Чтобы повернуть или отразить изображение, нажмите  **Повернуть**. На панели инструментов поворота введите произвольный угол поворота в градусах или с помощью кнопок поверните изображение влево или вправо или выберите вариант отражения изображения: по горизонтали или по вертикали.
 - * Чтобы добавить водяной знак на изображение, нажмите  **Водяной знак**. Добавьте текст к изображению, изменив его размер, стиль, цвет и непрозрачность с помощью инструментов водяного знака.
 - * Чтобы изменить формат изображения, нажмите  **Параметры**, затем в раскрывающемся списке **Формат** выберите новый формат.
 - * Чтобы изменить цвет фона, нажмите  **Параметры**, затем в раскрывающемся меню **Цвет фона** выберите нужный вариант.
 - * При редактировании файлов .jrg или .webp (доступно в браузерах Google Chrome) можно изменить качество изображения, чтобы создать файл меньшего размера. Нажмите  **Параметры** и введите новое процентное значение в поле (**Качество**).
 - * Чтобы отменить или повторить изменение, нажмите  или . Чтобы удалить все внесенные изменения, нажмите **Сброс**.
 - * Чтобы изменить размер представления изображения, используйте элементы управления масштабированием ().
- **Название**: введите текст, который будет отображаться в виде всплывающей подсказки.
 - **Альтернативный текст**: введите альтернативный текст, который будет отображаться для поддержки специальных возможностей.
 - **Подпись**: введите подпись, которая указывается под изображением.
 - **Тип выравнивания, ширина и интервалы**: измените макет изображения, если требуется.
 - **Абзац**: изменение интервалов в тексте.
8. На вкладке "Общие" можно изменить интервалы, тип выравнивания и другие параметры представления компонента.
 9. На вкладке "Стиль" можно добавить форматирование вокруг текста и настроить цвет фона, шрифты и границы.

Группы компонентов

Часто используемые компоненты можно объединять в именованные группы.

Группа компонентов с присвоенным ей именем сохраняется как пользовательский компонент и отображается в редакторе в списке пользовательских компонентов.

1. Перейдите к странице, которую хотите изменить, и убедитесь, что переключатель  переведен в положение **Редактировать**.
2. Добавьте группу компонентов на страницу. Группа компонентов обозначается специальным значком: .
3. Перетащите один или несколько компонентов во фрейм пользовательского компонента.
4. Размещение компонентов в группе и изменение их масштаба осуществляется так же, как и с обычными компонентами в ячейке.
5. Для редактирования компонента и его внешнего вида нажмите на значок меню  и выберите **Настройки**. Если выбрать имя компонента вместо значка меню, отображается значок меню для группы компонентов (или ячейки):



На вкладке "Настройки" для группы компонентов можно указать расположение группы компонентов, выбрать фоновое изображение и задать другие настройки для всей группы компонентов.

6. Чтобы сохранить изменения, внесенные в группу компонентов, выберите ее, нажмите на значок меню , затем выберите **Сохранить**.
 - a. В открывшемся диалоговом окне введите имя группы компонентов. Можно употреблять буквы, цифры, символы подчеркивания () и дефисы (-). Введенный пробел автоматически заменяется дефисом.

Имя новой группы компонентов не должно совпадать с именем существующего пользовательского компонента.

Если добавить на страницу существующую группу компонентов, внести в нее изменения и попытаться сохранить их, отобразится запрос на создание новой группы компонентов. При этом можно ввести имя группы и выбрать параметр **Перезаписать существующую группу компонентов**, позволяющий сохранить изменения в существующей группе.
 - b. Нажмите **Сохранить**.

Группа компонентов сохраняется как пользовательский компонент с указанным именем. Она помещается в собственную папку в диспетчере компонентов и появляется в списке пользовательских компонентов в редакторе, если вы

владелец компонента или вам предоставлен доступ к этому компоненту. Общий доступ к группе компонентов предоставляется так же, как и к любому пользовательскому компоненту.

Использование компонента "Совместный просмотр" на странице

Сервис Oracle Cobrowse Cloud Service представляет собой средство для совместной работы и позволяет предоставлять доступ к экрану или запускать сеансы совместного просмотра с другим пользователем. Например, эту функциональность можно внедрить для формы заказа, чтобы представитель мог просматривать экран заказчика, когда последний размещает заказ.

Чтобы использовать функцию, необходимо сначала включить ее для сайта. Можно использовать два типа сценария средства запуска: с настраиваемой кнопкой (Точка запуска 2) и с кнопкой совместного просмотра по умолчанию (Точка запуска 1). При включении функции для сайта вы определяете, как тип средства запуска использовать, и добавляете необходимый сценарий. Чтобы узнать больше, см. [Включение интеграции с Cobrowse](#).

После включения функции для сайта совместный просмотр можно настроить и использовать для любой страницы сайта.

1. Перейдите к странице, которую хотите изменить, и убедитесь, что переключатель  переведен в положение **Редактировать**.
2. Нажмите . Установите флажок в разделе "Совместный просмотр".
3. Нажмите **Заккрыть**.

Если включен сценарий точки запуска 1, то при следующем посещении страницы вы увидите кнопку совместного просмотра по умолчанию сразу или при нажатии сочетания клавиш (если настроено).

Если включен сценарий точки запуска 2, то потребуется добавить пользовательскую кнопку на страницу.

1. Убедитесь, что переключатель  переведен в положение **Редактировать**.
2. Добавьте компонент на страницу. Компонент средства запуска совместного просмотра находится в списке компонентов в разделе "Интеграция".
3. Для редактирования компонента и его внешнего вида нажмите на значок меню  и выберите **Настройки**. Можно настроить метку кнопки, ее внешний вид, размер и выравнивание. На вкладке "Стиль" можно добавить форматирование вокруг текста и настроить цвет фона, шрифты и границы. Для дополнительного редактирования стиля измените или добавьте классы стилей в файлах `design.json` и `design.css` в папке тем текущего шаблона сайта. Префикс класса стилей – `scs-cobrowse`.

После публикации сайта его посетители смогут использовать кнопку "Совместный просмотр" для запуска совместного сеанса с представителем вашей организации. Когда посетитель нажимает кнопку "Совместный просмотр", ему предоставляется защищенный идентификатор сеанса. Посетитель сообщает этот идентификатор по телефону представителю вашей организации, у которого есть доступ к консоли агента совместного просмотра. Агент использует консоль,

чтобы начать сеанс, который продолжается до тех пор, пока посетитель или агент не завершат его. При использовании точки запуска 2 в компоненте средства запуска используется фиксированный идентификатор `ses-start-cobrowse`. Этот идентификатор следует использовать в консоли совместного просмотра.

Примечания по использованию

При использовании мгновенного режима совместного просмотра (ICB), видео и встроенные фреймы `iFrame` недоступны для просмотра на странице, если для содержимого `iFrame` используется другой идентификатор сайта совместного просмотра. Поэтому некоторые компоненты Oracle Content Management не отображаются в консоли агента совместного просмотра в режиме ICB. Используйте расширенный режим совместного просмотра (ACB), чтобы отобразить следующие компоненты:

- Видео
- Youtube
- Диспетчер документов
- Кнопка Facebook "Нравится"
- Кнопка Twitter "Подписаться"
- Кнопка Twitter "Поделиться"
- Кнопка Facebook "Рекомендовать"
- Диалог

Нестандартные компоненты, которые используют `iFrame` для получения содержимого, также не отображаются в мгновенном режиме.

Чтобы узнать больше о Oracle Cobrowse Cloud Service, см. раздел ["Обзор совместного просмотра"](#) в документе *Руководство по развертыванию и использованию совместного просмотра*. Чтобы узнать больше об использовании совместного просмотра для безопасных сайтов или сайтов, находящихся в разработке, см. [Включение интеграции с Cobrowse](#).

Oracle Intelligent Advisor

Oracle Intelligent Advisor (ранее Oracle Policy Automation) используется для реализации интерактивных сценариев типа "интервью", например отзывов об устранении неполадок или оценки сервисов на соответствие требованиям. Сервис позволяет получать отзывы по различным каналам. Он позволяет записывать правила на естественном языке документов Microsoft Word и Excel и на основе этих правил создавать интерактивные компоненты взаимодействия с заказчиками, называемые "интервью".

Перед использованием функции Intelligent Advisor необходимо включить и настроить ее. Администратор сервиса включает эту функцию для вашего сервиса, в том числе добавляет имя хоста, URL-адрес, имя пользователя и пароль для используемого концентратора Intelligent Advisor Hub. Для интеграции сервисов необходимо использовать функцию единого входа, поэтому оба сервиса должны размещаться в одном домене идентификационных данных. См. раздел "Интеграция с Intelligent Advisor" в документе *Интеграция и расширение возможностей Oracle Content Management*.

На стороне Intelligent Advisor интервью необходимо создавать и хранить на сайте хоста. Кроме того, необходимо авторизовать сервис Oracle Content Management для использования на хосте Intelligent Advisor.

После настройки и включения Intelligent Advisor можно добавить компонент Intelligent Advisor на страницу сайта.

1. Перейдите к странице, которую хотите изменить, и убедитесь, что переключатель  переведен в положение **Редактировать**.
2. Добавьте компонент на страницу. Данный компонент отображается в разделе "Интеграция" списка "Компонент".
3. Для редактирования компонента и его внешнего вида нажмите на значок меню приложения  и выберите **Настройки**. Можно настроить метку компонента, его внешний вид, размер и выравнивание. Вкладка "Стиль" позволяет использовать стиль по умолчанию, связанный с интервью на хосте Intelligent Advisor. Для дополнительного редактирования стиля измените или добавьте классы стилей в файле design.css в папке тем текущего шаблона сайта. Префикс класса стилей - scs-opainterview-.

После публикации сайта его посетители увидят интерактивное интервью, выбранное в компоненте Oracle Intelligent Advisor. Чтобы узнать больше об Intelligent Advisor, см. раздел [Библиотека документации Intelligent Advisor](#).

Oracle Visual Builder

Oracle Visual Builder — это управляемая среда для вашей инфраструктуры разработки приложений. Она предоставляет стандартизированные решения с открытым исходным кодом для разработки приложений, совместной работы над ними и их развертывания в Oracle Cloud.

Начало работы

Перед использованием сервиса Oracle Visual Builder его необходимо включить и настроить. Администратор сервиса включает функции, в том числе добавляет имя хоста, на котором создаются и размещаются приложения. См. раздел Интеграция с Oracle Visual Builder в документе *Integrating and Extending Oracle Content Management*. Для интеграции сервисов необходимо использовать функцию единого входа, поэтому оба сервиса должны размещаться в одном домене идентификационных данных.

Необходимо выполнить следующие действия в Oracle Visual Builder перед использованием этой функции с Oracle Content Management.

- На сайте Oracle Visual Cloud Service необходимо включить технологию CORS (Cross-Origin Resource Sharing).
- Необходимо создать приложения, сделать их доступными для внедрения и настроить для использования с Oracle Content Management.
- Необходимо создать приложения и сделать их доступными для внедрения в iframe. Необходимо импортировать Sites SDK. В веб-приложениях должны быть указаны ссылки на Sites SDK. В веб-приложения необходимо добавить параметр URL-адреса страницы с именем "id".

Создание компонентов Oracle Visual Builder

После включения интеграции и создания настройки приложений и веб-приложений необходимо создать новый компонент для каждого приложения, которое планируется добавить на страницы сайта.

1. В Oracle Visual Builder получите URL-адрес опубликованного веб-приложения. Нажмите имя интерактивного проекта, который включает веб-приложение, затем нажмите имя веб-приложения. Скопируйте URL-адрес из адресной строки.
2. В Oracle Content Management выберите **Разработчик**, а затем щелкните **Смотреть все компоненты**. Отображаются зарегистрированные удаленные компоненты и макеты.
3. Нажмите **Создать** и выберите **Создать компонент Visual Builder**.
4. Введите имя для компонента. Имя не должно совпадать с именем существующего компонента или макета.

Можно употреблять буквы, цифры, символы подчеркивания (_) и дефисы (-). Введенный пробел автоматически заменяется символом подчеркивания.

Не используйте для шаблонов, тем, компонентов, сайтов и страниц сайтов следующие имена: authsite, content, pages, scstemplate_*, _comps, _components, _compsdelivery, _idcservice, _sitescloud, _sitesclouddelivery, _themes, _themesdelivery. Следующие имена можно использовать только для страниц сайтов (не для шаблонов, тем, компонентов и сайтов): documents, sites.

5. Дополнительно можно ввести описание компонента.
6. Вставьте URL-адрес в веб-приложение или классическое приложение.
7. Нажмите **Создать**. После создания компонента его имя появляется в списке компонентов. Чтобы просмотреть папки и файлы в составе этого компонента или макета, нажмите на имя компонента или макета в списке.
8. Выбор значка компонента, отличного от значка по умолчанию:
 - a. Выберите компонент в списке.
 - b. Нажмите **Свойства**
 - c. Перейдите на вкладку **Логотип компонента**.
 - d. Выберите логотип в галерее логотипов и нажмите **Готово**.

Добавление компонента на страницу сайта

Теперь можно добавить компонент для приложения на одну из страниц сайта. Компонент можно добавить на общедоступную или защищенную страницу.

1. Перейдите к странице, которую хотите изменить, и убедитесь, что переключатель  переведен в положение **Редактировать**.
2. Добавьте компонент на страницу. Компонент появится в разделе "Настройка".
3. Для редактирования компонента и его внешнего вида нажмите на значок меню приложения  и выберите **Настройки**. Можно настроить метку компонента, его внешний вид, размер и выравнивание. На вкладке "Стиль" можно задать стиль по умолчанию, связанный с компонентом на хосте Oracle Visual Builder. Для дополнительного редактирования стиля измените или добавьте классы стилей в

файле `design.css` в папке тем текущего шаблона сайта. Префикс класса стилей – `scs-component`.

Во время редактирования сайта можно выполнить предварительный просмотр компонента. После публикации вашего сайта его посетители увидят приложение Oracle Visual Builder, выбранное для использования с этим компонентом и запущенное в `iFrame` на странице.

Часть III

Публикация сайтов и управление ими

В этом разделе подробно описано, как работать с сайтами, обеспечивать их безопасность, улучшать и публиковать их. Здесь содержатся указанные ниже главы.

- [Управление сайтами](#)
- [Публикация сайтов](#)
- [Безопасность сайтов](#)
- [Работа с многоязычными сайтами](#)
- [Использование перенаправление сайтов или сопоставления URL-адресов](#)
- [Повышение производительности сайта](#)

11

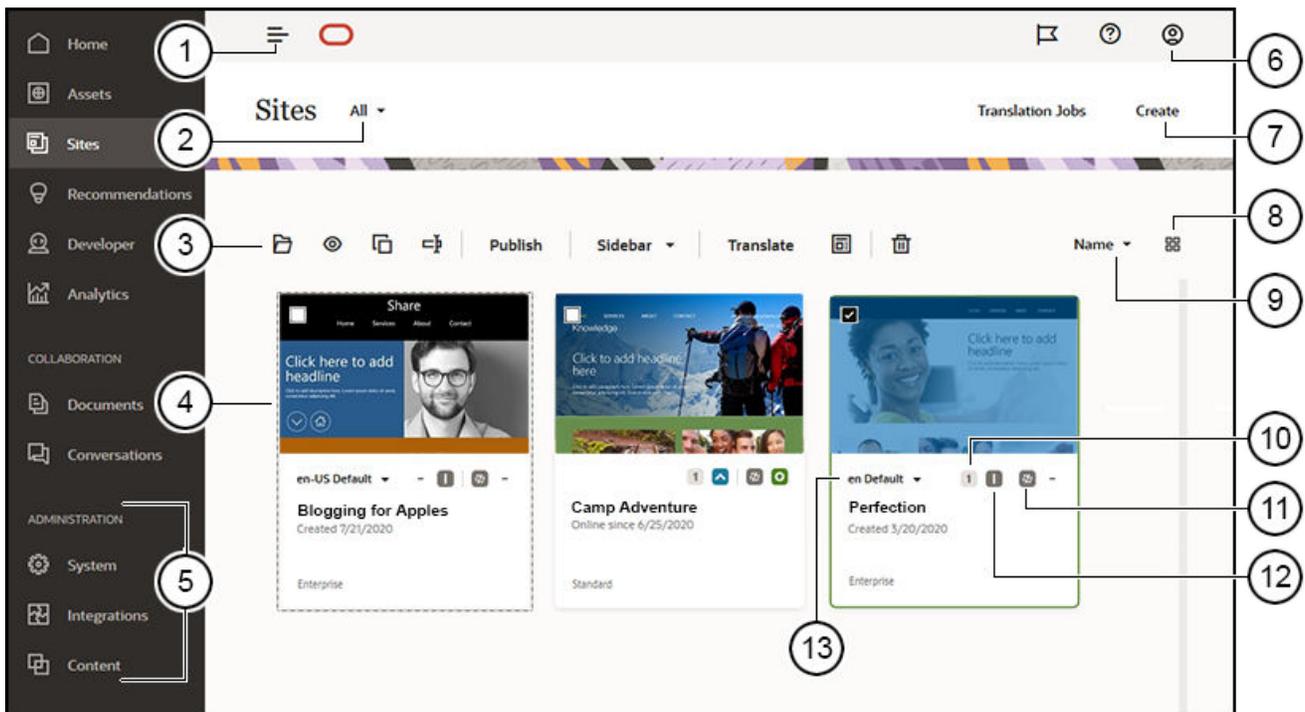
Управление сайтами

Чтобы изменить макет или содержимое сайта, создайте и откройте обновление в редакторе. Для создания самого сайта и управления им и его свойствами используйте возможности диспетчера сайтов.

- [Вводные сведения о странице "Сайты"](#)
- [Управление сайтами и настройки сайтов](#)
- [Перевод сайта в онлайн-режим или автономный режим](#)
- [Изменение описания сайта, логотипа или внедренных свойств](#)
- [Настройка свойств поисковых систем](#)
- [Настройки сайта](#)
- [Включение интеграции с Cobrowse](#)
- [Добавление отслеживания аналитики](#)

Вводные сведения о странице "Сайты"

Страница "Сайты" предоставляет доступ ко всем функциям для работы с веб-сайтами.



Выноска а	Описание
1	Переключатель панели позволяет скрыть и отобразить меню навигации.
2	Меню фильтров позволяет отфильтровать список сайтов, чтобы увидеть все сайты, к которым у вас есть доступ, или их подмножество. Если управление сайтами включено, вы можете просматривать ожидающие запросы сайта.
3	<p>Если выбрать сайт в списке, становится доступной панель действий. Для открытия, просмотра, изменения состояния, переименования сайтов и выполнения других задач используются пункты на панели действий.</p> <p>Доступные инструменты в строке меню зависят от роли, назначенной пользователю для выбранного сайта. Например, автор сайта имеет роль "Управление" и может выполнять все перечисленные задачи. Если кто-то предоставил вам доступ к сайту и назначил роль "Просмотр", вы можете просматривать свойства сайта, но не можете вносить изменения.</p> <p>Доступные пункты панели действий также зависят от текущего статуса сайта. Команды Переименовать и Удалить доступны, только когда сайт находится в автономном режиме.</p> <p>Совет: Хотите узнать сочетание клавиш? Правой кнопкой мыши щелкните значок сайта, чтобы открыть контекстное меню и выбрать команду.</p>
4	В списке сайтов отображаются все принадлежащие вам сайты, а также сайты, к которым вам предоставлен доступ. В этом списке указывается имя сайта, число обновлений, уровень безопасности сайта и статус сайта (онлайн, автономный и отложенные обновления).
5	Меню администрирования отображается, если пользователь выполнил вход от имени администратора контента. Параметры этого меню позволяют изменять настройки сервиса, настраивать интеграции, репозитории и другие элементы, связанные с активами.
6	В пользовательском меню доступны команды настройки параметров, добавления отзывов, доступа к справке и выхода из системы. Нажмите на изображение пользователя, чтобы открыть меню параметров.
7	Нажмите Создать , чтобы запустить процесс создания веб-сайта. Чтобы пользователи могли создавать сайты, администратору необходимо включить функцию создания сайтов и загрузить один или несколько шаблонов. Если параметр Создать не отображается на странице "Сайты" или на странице "Шаблоны" отсутствуют шаблоны, обратитесь к администратору сервиса.
8	Нажмите на значок просмотра , чтобы выбрать, в каком виде будут отображаться сайты, в виде сетки, списка или таблицы.
9	Параметры сортировки позволяют изменить порядок отображения сайтов.
10	Нажмите номер обновления , чтобы изменить существующее обновление сайта или создать новое обновление сайта. Если у сайта нет обновлений, можно создать новое обновление. Для этого откройте сайт и переведите Site Builder в режим редактирования.
11	Нажмите значки Автономный , Онлайн или Опубликовать повторно , чтобы изменить статус сайта или опубликовать изменения, которые применены, но не опубликованы на сайте.

Выноска	Описание
12	Значки безопасности показывают, требуется выполнять вход на сайт или нет.
13	Используйте список языков , чтобы просмотреть языки, указанные в соответствующей политике локализации сайта. Для предварительного просмотра локализованной версии выберите язык и откройте сайт.

Управление сайтами и настройки сайтов

Управление сайтами включает в себя создание и редактирование сайтов, а также управление содержимым этих сайтов. *Site Builder* позволяет создавать, копировать и удалять страницы сайтов и контент страниц. В *диспетчере файлов и папок* можно создавать, копировать и удалять целые сайты, а также предоставлять к ним общий доступ.

Используйте параметры в строке меню или в контекстном меню на странице "Сайты" и в Site Builder, чтобы выполнить следующие задачи.

Задача	Описание
Фильтрация представления сайтов	<p>Чтобы отфильтровать список сайтов, выберите один из следующих параметров в меню фильтра:</p> <ul style="list-style-type: none"> • Все - показывает все сайты, к которым у вас есть доступ. Если вы являетесь администратором сайта и управление сайтом включено, у вас есть доступ ко всем сайтам в вашей среде. • Ваши — отображаются все сайты, которые принадлежат вам. • Общий доступ для вас — отображаются все сайты, к которым вам предоставлен общий доступ. • Онлайн — отображаются сайты, которые находятся в онлайн-режиме (в режиме реального времени). • Автономные — отображаются сайты, которые работают в автономном режиме. • Запросы (доступно только при включенном управлении сайтом) — отображаются ожидающие запросы сайта. Если вы являетесь администратором сайта, вы видите все запросы сайта в вашей системе, в противном случае вы видите только те запросы сайта, которые вы отправили. • Корзина — отображаются удаленные сайты, владельцем которых вы являетесь. Если вы являетесь администратором сайта и функция управления включена, то отображаются также все удаленные сайты.
Создание сайта	<p>Чтобы создать сайт, нажмите Создать. Чтобы создать сайт, необходимо выбрать шаблон. Шаблон содержит все необходимые элементы для создания сайта, в том числе кодовую платформу сайта, стандартный сайт с примерами страниц и содержимого, тему со стилями, ресурсы (например, изображения), а также пользовательские компоненты. См. Создание сайтов.</p> <p>В меню фильтров на странице Сайты можно увидеть, включено ли управление сайтами. Если отображается параметр Запросы, управление сайтом включено. См. Общие сведения об управлении сайтами.</p>

Задача	Описание
Просмотр и управление запросами сайта	Чтобы просмотреть ожидающие запросы сайта (доступно только при включенном управлении сайтами), в меню фильтра выберите Запросы . Если вы являетесь администратором сайта, вы видите все запросы сайта в вашей системе, в противном случае вы видите только те запросы сайта, которые вы отправили. См. Управление запросами сайта .
Редактирование сайта	Чтобы изменить сайт, выберите Открыть в контекстном меню или нажмите  на панели действий. См. Редактирование сайтов .
Обсуждение или добавление аннотации к сайту	Можно обсудить или добавить аннотацию к сайту в Site Builder. Выберите сайт и выберите Открыть в контекстном меню или нажмите  на панели действий. Для обсуждения сайта нажмите  , чтобы открыть панель диалога. Чтобы добавить аннотацию к сайту, нажмите  . См. Вводные сведения о странице "Site Builder" .
Просмотр сайта	Чтобы увидеть, как будет выглядеть сайт в онлайн-режиме, выберите сайт и выберите пункт Просмотр в контекстном меню или нажмите  на панели действий.
Публикация сайта	Чтобы опубликовать сайт, выберите его и выберите Опубликовать в контекстном меню или на панели действий. См. Публикация изменений сайта .
Повторная публикация	Если ранее вы опубликовали сайт, но связанный элемент или политика были изменены (например, соответствующая политика локализации была обновлена для включения дополнительных языков), вы можете Повторно опубликовать сайт, чтобы обновить эти изменения. Выберите сайт и выберите Повторно опубликовать в контекстном меню или на панели действий.
Отмена публикации	Если вы ранее опубликовали сайт и хотите удалить файлы сайта из расположения в Oracle Cloud, выберите сайт и выберите Отменить публикацию в контекстном меню или на панели действий. Для отмены публикации необходимо перевести сайт в автономный режим.
Перевод сайта в онлайн-режим или автономный режим	Сайт должен быть опубликован, прежде чем вы сможете перевести его в онлайн-режим. Чтобы перевести сайт в онлайн-режим или перевести его в автономный режим, выберите сайт и в контекстном меню выберите пункт Перевести в онлайн-режим или Перевести в автономный режим . Также можно нажать  на плитке сайта, чтобы перевести сайт в онлайн-режим, или нажать  на плитке сайта, чтобы перевести сайт в автономный режим. Когда сайт находится в <i>онлайн-режиме</i> , пользователи могут его просматривать с помощью стандартного веб-браузера по назначенному адресу (URL). Когда сайт находится в <i>автономном режиме</i> , он не доступен для публичного просмотра. Сайт можно просматривать только в Oracle Content Management. См. Перевод сайта в онлайн-режим или автономный режим .

Задача	Описание
Переименование сайта	<p>Чтобы переименовать сайт, нажмите на него правой кнопкой мыши и выберите Переименовать в контекстном меню или нажмите  на панели действий. Введите имя, отличное от имен сайтов, существующих на том же сервере.</p> <p>Невозможно переименовать сайт, если он находится в онлайн-режиме. Чтобы перевести сайт в автономный режим, требуется быть владельцем сайта или иметь роль "Управление" для данного сайта.</p>
Копирование сайта	<p>Возможность копирования сайтов упрощает создание новых сайтов с нуля. Все элементы исходного сайта, в том числе тема, все неопубликованные обновления, страницы, контент страниц и все остальные активы, такие как изображения, копируются в новый сайт с указанными именами. Новый сайт находится в автономном режиме и готов к редактированию.</p> <p>Примечание: если управление сайтами включено, может потребоваться утверждение сайта перед созданием его копии.</p> <p>При копировании сайта, использующего контент из нескольких репозиториях, необходимо сделать это с помощью инструментария ОСЕ. См. Разработка с помощью инструментария ОСЕ и Использование утилиты командной строки СЕС.</p> <p>Выберите сайт, который необходимо скопировать, и выберите Копировать в контекстном меню или нажмите  на панели действий. См. Копирование сайтов.</p>
Создание шаблона на основе сайта	<p>Если вы хотите использовать сайт в качестве отправной точки для других сайтов, можно создать шаблон на основе этого сайта. Выберите сайт и выберите Создать шаблон в контекстном меню или нажмите  на панели действий. См. Создание шаблона на основе сайта.</p>

 **Примечание.**

При создании шаблона на основе сайта, который использует контент из нескольких репозиториях, необходимо это сделать с помощью инструментария ОСЕ. См. [Разработка с помощью инструментария ОСЕ](#) и [Использование утилиты командной строки СЕС](#).

Задача	Описание
Удаление и восстановление сайта	<p>При наличии соответствующих разрешений можно удалить сайт и его содержимое из корзины. При удалении сайта в корзину перемещается все содержимое папки сайта, включая все неопубликованные обновления, страницы сайта, содержимое страниц и такие активы, как изображения, добавленные на страницы.</p> <p>Сайт можно удалить или восстановить, если он создан вами (вы владелец этого сайта) или другой пользователь предоставил вам доступ к этому сайту и роль "Соавтор" или "Управление".</p> <p>Невозможно удалить сайт, если он находится в онлайн-режиме. Чтобы перевести сайт в автономный режим, требуется быть владельцем сайта или иметь роль "Управление" для данного сайта.</p> <p>Также невозможно удалить опубликованный сайт. Сначала необходимо отменить публикацию контента. Если на сайте содержится контент из нескольких репозиториях, необходимо отменить публикацию связанного контента в каждом репозитории.</p> <p>Чтобы удалить сайт, выберите его и выберите Удалить в контекстном меню или нажмите  на панели действий. Отобразится запрос на перемещение сайта и всех его обновлений в корзину. Удаленный сайт остается в корзине до перечисленных ниже событий.</p> <ul style="list-style-type: none">• Сайт восстановлен.• Сайт безвозвратно удален.• Квота корзины превышена.• Корзина автоматически опустошается с интервалом, заданным администратором сервиса. Значение по умолчанию равно 90 дням. <p>Чтобы восстановить сайт, в меню "Сайты" выберите Корзина. Выберите сайт в списке и нажмите Восстановить.</p>

Задача	Описание
Добавление участников на сайт	<p>Если администратор разрешил общий доступ, вы можете предоставить доступ к своему сайту другим пользователям Oracle Content Management и разрешить им просматривать сайт, изменять его или управлять им в Oracle Content Management в соответствии с предоставленными разрешениями.</p> <p>Все, у кого есть доступ к вашему сервису, считаются участниками. При предоставлении общего доступа к сайту вы назначаете роль, которая определяет, что участники могут и что не могут делать с вашим сайтом. Участники могут использовать ваш сайт только в соответствии с назначенными им ролями.</p> <p>Выберите сайт, к которому вы хотите предоставить общий доступ, и выберите Участники в контекстном меню или нажмите Боковая панель на панели действий и выберите Участники, чтобы открыть боковую панель участников. Открыв боковую панель участников, нажмите Добавить участников.</p> <p>Введите одно или несколько имен пользователей или адресов электронной почты и назначьте им одну из описанных ниже ролей.</p> <ul style="list-style-type: none"> • Просмотр. Пользователи с этой ролью могут просматривать сайт в редакторе, но не могут вносить изменения. • Выгрузка. Пользователям с этой ролью, заданной для сайта, предоставляются те же права, что и пользователям с ролью "Просмотр". Они могут создавать новый шаблон на основе сайта. • Соавтор. Кроме возможностей, предоставляемых ролью "Просмотр", эта роль позволяет также редактировать сайт, удалять страницы сайта и сам сайт, когда он находится в автономном режиме. • Управление. Кроме возможностей роли "Соавтор", данная роль также позволяет добавлять пользователей и назначать им роли, публиковать изменения на сайте, работающем в онлайн-режиме, и переключать сайт в онлайн-режим и автономный режим. Автору сайта (владельцу) автоматически назначается роль "Управление".
Изменение свойств сайта (описание, логотип и т.д.)	<p>Чтобы изменить описание сайта, добавьте логотип, разрешите встраивание сайта, добавьте неофициальный URL-адрес сайта и просмотрите ценную информацию о сайте, включая URL-адрес сайта, владельца сайта и другие сведения, выберите сайт и выберите Свойства в контекстном меню или нажмите Боковая панель на панели действий и выберите Свойства. См. Изменение описания сайта, логотипа или внедренных свойств.</p>

 **Примечание.**

При совместном использовании сайта, который имеет доступ к нескольким репозиториям, общий доступ предоставляется только для репозитория по умолчанию. Для совместного использования какого-либо дополнительного репозитория необходимо открыть общий доступ к нему отдельно.

Задача	Описание
Перевод сайта	Можно перевести корпоративный сайт на несколько языков, если это позволяет связанный репозиторий, экспортировав файлы сайта, переведя их, а затем импортировав переведенные файлы. См. Перевод сайта .
Задание свойств поисковых систем	В Site Builder можно указать ключевые слова или текст, чтобы помочь поисковым системам идентифицировать контент сайта. См. Настройка свойств поисковых систем
Настройка параметров сайта (значки, файлы контроллера и т. д.)	В Site Builder можно выбрать значки сайта для различных браузеров и платформ или добавить файл контроллера для обработки поведения ссылок. См. Настройки сайта
Совместный просмотр страниц	В Site Builder можно включить Oracle Cobrowse Cloud Service, инструмент для совместной работы, который позволяет делиться экранами или запускать сеанс совместного просмотра с другим пользователем. Например, эту функциональность можно внедрить для формы заказа, чтобы представитель мог просматривать экран заказчика, когда последний размещает заказ. См. Включение интеграции с Cobrowse
Добавление отслеживания аналитики на сайт	В Site Builder можно добавить фрагмент отслеживающего кода JavaScript, который встраивается в сайты. Этот код позволяет отслеживать веб-аналитику и упрощает интеграцию с внешними поставщиками аналитики, такими как Google, Adobe и Oracle Infinity. См. Добавление отслеживания аналитики .
Защита работающего сайта	<p>Когда сайт публикуется и переводится в онлайн-режим, он становится доступным для всех пользователей. Однако владельцы сайта или обладатели роли "Управление" могут ограничить доступ к работающим сайтам, предоставив разрешение только зарегистрированным пользователям или подмножеству пользователей. Чтобы изменить статус сайта, требуется быть его владельцем или иметь роль "Управление".</p> <p>Выберите сайт, который необходимо защитить, и выберите Свойства в контекстном меню или нажмите Боковая панель на панели действий и выберите Свойства, затем перейдите на вкладку Безопасность сайта. См. Изменение настроек безопасности сайта.</p>
Добавление перенаправления сайта	При изменении URL-адреса сайта для исходного URL-адреса устанавливается перенаправление на конечный URL-адрес. Это позволяет сохранить пользовательские закладки и сортировку поисковых систем. См. Добавление перенаправлений сайта .

Изменение описания сайта, логотипа или внедренных свойств

Большая часть свойств сайта задается во время его создания или обновления. При просмотре свойств можно получить важные сведения о сайте, включая URL-адрес, информацию о владельце и т. д.

Изменение описания или логотипа сайта, добавление страницы захвата и включение возможности встраивания:

1. Выберите сайт и выберите **Свойства** в контекстном меню или нажмите **Боковая панель** на панели действий и выберите **Свойства**.
2. Чтобы изменить описание сайта (необязательное), нажмите на вкладку **Свойства сайта** и введите новое описание или измените существующее.
3. Хотя URL-адрес невозможно изменить напрямую, URL-адрес можно выбрать и скопировать, а затем вставить его в документы, презентации и сообщения электронной почты, чтобы предоставить доступ к сайту.

Если сайт находится в онлайн-режиме, нажмите , чтобы напрямую перейти к онлайн-сайту.

4. Чтобы разрешить использовать сайт в качестве вложенного сайта, переключите в состояние **Да** настройку **Встраиваемый сайт**.
5. Чтобы изменить изображение сайта, нажмите на вкладку **Логотип сайта**, а затем **Изменить**. Найдите и выберите нужное изображение. Оно должно быть в формате PNG, JPEG или JPG и иметь соотношение сторон 4:3 (прямоугольник). Оптимальный размер – 300 x 225 пикселей. Изображения меньшего размера могут выглядеть искаженными, а более крупные изображения могут снизить производительность.
6. По завершении нажмите **Заккрыть**.

Настройка свойств поисковых систем

Вы можете предоставить ключевые слова или текст, чтобы помочь поисковым системам идентифицировать содержимое сайта.

Настройки поисковой оптимизации (SEO) можно определить на уровне сайта и на уровне страниц. Настройки уровня сайта дополняют или переопределяют аналогичные настройки для отдельных страниц (см. таблицу ниже).

Параметр	Уровень сайта	Уровень страницы
Описание или описание страницы	Предоставляет общие сведения о сайте, которые не включены в сам сайт. Описание сайта входит в каждую страницу сайта. Это описание также используется для главной страницы в качестве описания на уровне страницы, если для главной страницы не установлено значение в разделе Настройки страницы .	Предоставляет общие сведения о странице, которые не включены в саму страницу. Описание страницы дополняет описание сайта, включенное в каждую страницу сайта.
Ключевые слова	Идентифицирует термины или концепции, применимые ко всем страницам сайта. Эти значения добавляются к ключевым словам, указанным для отдельных страниц.	Идентифицирует термины или концепции, применимые к данной отдельной странице. Ключевые слова страницы можно использовать для идентификации терминов или концепций, которые не содержатся в тексте страницы или отображаются на изображениях.

Параметр	Уровень сайта	Уровень страницы
Заголовок или заголовок страницы	Добавление сценариев или тегов для анализа или отслеживания своего сайта. Содержимое заголовка сайта входит в каждую страницу сайта.	Добавление сценариев или тегов для анализа или отслеживания своего сайта. Содержимое заголовка страницы дополняет содержимое заголовка сайта, которое включено в каждую страницу.
Нижний колонтитул или нижний колонтитул страницы	Добавление сценариев или тегов для анализа или отслеживания своего сайта. Содержимое нижнего колонтитула сайта входит в каждую страницу сайта.	Добавление сценариев или тегов для анализа или отслеживания своего сайта. Содержимое нижнего колонтитула страницы дополняет содержимое нижнего колонтитула сайта, которое включено в каждую страницу.

Если указать следующие параметры исключения при поиске на уровне сайта, настройки применяются ко всем страницам и переопределяют настройки отдельных страниц. Если не выбирать этот параметр на уровне сайта, то он используется только на страницах, на которых указан отдельно.

Параметр	Уровень сайта	Уровень страницы
Скрыть от поисковых систем	Если выбрано, добавьте метатег NOINDEX на все страницы, чтобы поисковые системы не индексировали содержимое страниц на сайте. В этом случае сайт и все его страницы не отображаются в результатах веб-поиска.	Если выбрано, добавьте метатег NOINDEX на текущую страницу, чтобы поисковые системы не индексировали содержимое страницы. В этом случае данная отдельная страница не отображается в результатах веб-поиска..
Скрыть ссылки на страницу от поисковых систем	Если выбрано, добавьте метатег NOFOLLOW на каждую страницу, чтобы поисковые системы не переходили по ссылкам (и не индексировали место назначения) на каждой странице сайта.	Если выбрано, добавьте метатег NOFOLLOW на текущую страницу, чтобы поисковые системы не переходили по ссылкам (и не индексировали место назначения) на текущей странице сайта.
Скрыть описания страницы от поисковых систем	Если выбрано, добавьте метатег NOSNIPPET на каждую страницу, чтобы поисковые системы не включали описание (указанное выше) после страницы в результатах поиска.	Если выбрано, добавьте метатег NOSNIPPET на текущую страницу, чтобы поисковые системы не включали описание (указанное выше) после страницы в результатах поиска.

Чтобы изменить настройки поисковой оптимизации (SEO), выполните указанные ниже действия.

1. Откройте сайт для редактирования.

2. Нажмите  на боковой панели, а затем нажмите  **SEO**.
3. При необходимости укажите дополнительное описание сайта. Описание сайта входит в каждую страницу сайта.

Это описание также используется для главной страницы в качестве описания на уровне страницы, если для главной страницы не установлено значение в разделе **Настройки страницы**.
4. Дополнительно можно указать ключевые слова, разделенные запятыми, чтобы помочь поисковым системам индексировать содержимое сайта.

Ключевые слова сайта идентифицируют термины или концепции, применимые ко всем страницам сайта. Эти значения добавляются к ключевым словам, указанным для отдельных страниц в разделе **Настройки страницы**.
5. При желании добавьте сценарии или теги для анализа или отслеживания своего сайта. Содержимое заголовка входит в каждую страницу сайта. Проверьте код, используемый в заголовке, чтобы убедиться, что он правильно работает и не создает рисков безопасности для сайта.
6. При желании добавьте в нижний колонтитул сценарии или теги для анализа или отслеживания своего сайта. Содержимое нижнего колонтитула входит в каждую страницу сайта. Проверьте код, используемый в нижнем колонтитуле, чтобы убедиться, что он правильно работает и не создает рисков безопасности для сайта.
7. При необходимости выберите один или несколько параметров, чтобы исключить данные из отображаемых результатов поиска, как показано в предыдущей таблице.
8. Чтобы сохранить отложенные изменения в текущую тему, нажмите кнопку **Сохранить**.

Включение сервиса предварительного рендеринга для поисковой оптимизации

Можно разрешить сайтам, созданным с помощью Oracle Content Management для предварительного рендеринга статических HTML-страниц, отвечать на запросы поискового робота.

Если сервис предварительного рендеринга включен и запрос поступает от поискового робота, поиск страницы выполняется в кэше.

- Если страница найдена в кэше и ее возраст меньше 15 дней, запрос обслуживается кэшированной страницей.
- Если страница в кэше старше 15 дней или не найдена в кэше, из Oracle Content Management извлекается новая версия, кэш обновляется и обслуживается новая страница с предварительным рендерингом.
- Если страница не найдена в кэше или в Oracle Content Management, отображается страница ошибки.

Рендеринг страниц с истекшим сроком действия на общедоступных сайтах или новых сайтах, которые еще не кэшированы, выполняется один раз в день. Если страница уже кэширована, кэш не обновляется. Когда сервис предварительного рендеринга включен, кэш с предварительным рендерингом обновляется ежечасно для сайтов, которые были обновлены и повторно опубликованы. Если вы вносите изменения в страницы сайта и хотите проверить, как эти изменения влияют на оптимизацию, вы можете обновить кэш с предварительным рендерингом вручную в свойствах сайта.

1. На боковой панели навигации нажмите **Сайты**.
2. Выберите сайт и выберите **Свойства** в контекстном меню или нажмите **Боковая панель** на панели действий и выберите **Свойства**.
3. Нажмите кнопку **SEO** на боковой панели свойств.
4. Отображаются дата и время последнего обновления кэша. Чтобы обновить кэш, нажмите **Обновить сейчас**. Индикатор выполнения отображает статус кэширования.

 **Примечание.**

Если страница скрыта в поисковых системах в настройках поисковой оптимизации сайта, к этой странице не будет применяться предварительный рендеринг или кэширование.

Включение сервиса предварительного рендеринга в Oracle Content Management.

1. В области администрирования боковой панели навигации нажмите **Интеграции**.
2. В меню настроек системы выберите **SEO для сайтов**.
3. Нажмите **Включено**.
4. Определите всех необходимых дополнительных агентов пользователей.

Таблица 11-1 Жестко закодированные агенты пользователей в Oracle Content Management

Агент пользователя	Агент пользователя	Агент пользователя	Агент пользователя
baiduspider	facebookexternalhit	twitterbot	rogerbot
linkedinbot	embedly	quora link preview	showyoubot
outbrain	pinterest/0.	developers.google.com/+web/snippet	slackbot
vkShare	W3C_Validator	redditbot	Applebot
WhatsApp	flipboard	tumblr	bitlybot
SkypeUriPreview	nuzzel	Discordbot	Google Page Speed
Qwantify	pinterestbot		

Таблица 11-2 Дополнительные предварительно настроенные агенты пользователей в свойстве PrerenderUserAgents файла config.cfg

Агент пользователя	Агент пользователя	Агент пользователя	Агент пользователя
AddSearchBot	AdIdxBot	AdsBot-Google	AdsBot-Google-Mobile-Apps
AppEngine-Google	Baidu-YunGuanCe	Bingbot	BingPreview
DuckDuckBot	DuckDuckGo-Favicons-Bot	endeca webcrawler	Exabot

Таблица 11-2 (Продолж.) Дополнительные предварительно настроенные агенты пользователей в свойстве PrerenderUserAgents файла config.cfg

Агент пользователя	Агент пользователя	Агент пользователя	Агент пользователя
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

Задание параметров доставки статического сайта

Если в вашей компании используются скомпилированные сайты, можно управлять длительностью кэширования статических сайтов и определять, каким мобильным агентам пользователей требуются адаптивные макеты для мобильных устройств, поддерживаемые компиляцией сайта.

- Включение автоматической компиляции при публикации
- Переопределение заголовков управления кэшем по умолчанию для скомпилированных сайтов
- Указание мобильных агентов пользователей для поддержки скомпилированных адаптивных макетов

Включение автоматической компиляции при публикации

В Oracle Content Management имеется встроенный сервис компиляции, который позволяет легко компилировать сайт без дополнительной настройки. В качестве альтернативы администратор сервиса может вручную настроить сервис компиляции для целей тестирования, чтобы проверять компиляцию сайта или использовать пользовательские библиотеки. Независимо от используемого сервиса компиляции можно выбрать компиляцию сайта при его публикации или повторной публикации.

1. После входа в Oracle Content Management в боковом меню навигации выберите пункт **Сайты**.
2. Выберите сайт, который необходимо изменить, и нажмите **Свойства**.
3. Нажмите **Статическая доставка** в диалоговом окне свойств.
4. Установите флажок **Скомпилировать сайт после публикации** в разделе **Включение автоматической компиляции**.

5. По завершении нажмите кнопку **Готово**.

При публикации или повторной публикации сайта с включенной автоматической компиляцией статус публикации отслеживается и отображается на плитке сайта страницы **Сайты**. После завершения публикации процесс компиляции отслеживается в разделе **Статическая доставка** диалогового окна свойств сайта. После завершения компиляции в разделе "Статическая доставка" свойств сайта отображается дата и время последней компиляции, а также ссылка для выгрузки журнала компиляции.

Переопределение заголовков управления кэшем по умолчанию для скомпилированных сайтов

Скомпилированные сайты по умолчанию кэшируются в браузере пользователя в течение 300 секунд (5 минут). Администратор сервиса может изменить это значение по умолчанию, но разработчик сайта может переопределить значение по умолчанию для конкретных сайтов в свойствах сайта.

1. После входа в Oracle Content Management в боковом меню навигации выберите пункт **Сайты**.
2. Выберите сайт, который необходимо изменить, и нажмите **Свойства**.
3. Нажмите **Статическая доставка** в диалоговом окне свойств.
4. В разделе **Заголовки ответов при кэшировании** введите `Cache-control: max-age=`, а затем числовое значение времени в секундах, в течение которого страница должна кэшироваться в браузере пользователя. Например, если задано `Cache-control: max-age=600`, страница будет кэшироваться в течение 10 минут.
5. По завершении нажмите кнопку **Готово**.

Если ваш экземпляр использует Akamai, для сохранения существующих настроек Akamai оставьте поле **Заголовки ответов при кэшировании** пустым. Чтобы переопределить существующие настройки Akamai, введите `Edge-Control: !no-store,max-age=1800,downstream-ttl=1800`, где жирным шрифтом выделены настройки по умолчанию в секундах.

- `!no-store` указывает, что эта настройка должна переопределять соответствующую конфигурацию кэширования Akamai для свойства.
- `max-age` определяет, как долго решение Akamai должно кэшировать эту страницу. Значение по умолчанию: 1800 секунд (30 минут). В течение этого времени решение Akamai будет выполнять запросы этой страницы, не запрашивая ее из Oracle Content Management.
- `downstream-ttl` указывает решению Akamai на необходимость отправить заголовок "Cache-Control: max-age" с ответом на запросы браузеров клиентов и предписывает браузерам кэшировать страницу в течение определенного времени. Значение по умолчанию: 1800 секунд (30 минут).

Указание мобильных агентов пользователей для поддержки скомпилированных адаптивных макетов

При компиляции сайта можно создавать страницы для мобильных устройств для поддержки адаптивных макетов. Можно указать агентов пользователей, которые

будут задавать режим, при котором сервер будет доставлять страницы для мобильных устройств вместо стандартных скомпилированных страниц для сайта. При сопоставлении с заголовками агентов пользователей, отправляемыми браузерами, вводимые здесь значения рассматриваются как подстроки, регистр букв в которых не имеет значения. Обратите внимание, что ключевое слово `Mobile` обычно используется в строках агентов пользователей для браузеров на мобильных устройствах.

1. После входа в Oracle Content Management в боковом меню навигации выберите пункт **Сайты**.
2. Выберите сайт, который необходимо изменить, и нажмите **Свойства**.
3. Нажмите **Статическая доставка** в диалоговом окне свойств.
4. В разделе **Мобильный агент пользователя** введите разделенный запятыми список подстрок "агент пользователя" для агентов пользователей, которым необходимо, чтобы обслуживались страницы для мобильного устройства. Если какая-либо часть подстроки совпадает со строкой агента пользователя браузера, страницы для мобильного устройства будут обслуживаться.

Указание и настройка персонализированных URL-адресов

Настройка персонализированного URL-адреса для сайта позволяет пользователям получать доступ Oracle Content Management к сайту в указанном домене, а не к домену Oracle Content Management экземпляра. Например, хотя URL-адрес экземпляра может иметь вид `https://myinstance.ces.ocp.oraclecloud.com/site/MyCustomerSite/`, более удобный URL-адрес, например `https://www.example.com` легче запомнить, он лучше подходит для брендинга и, как правило, проще в использовании. При необходимости сайт Oracle Content Management можно разместить, используя собственный путь, например `https://www.example.com/store/`

При создании персонализированных доменов требуется выполнить ряд действий.

- Используйте сеть доставки контента (CDN).
- Управляйте персонализированным доменом с помощью системы доменных имен (DNS), чтобы запись канонического доменного имени (CNAME) была сопоставлена с CDN.
- Разверните действительный сертификат в CDN, защищающий персонализированный домен.
- Настройте сайт Oracle Content Management для приема запросов в персонализированном домене.
- Настройте CDN для правильной маршрутизации запросов к Oracle Content Management.

Использование сети доставки контента

Сеть доставки контента (CDN) — это платформа распределенных по всему миру серверов, предназначенная для повышения производительности и безопасности веб-сайтов. CDN минимизирует расстояние между пользователями и серверами, улучшая производительность при запросах контента. Хотя главной целью CDN является улучшение качества работы пользователей, CDN можно также использовать для изменения запросов при передаче. При этом пользователь будет получать нужную информацию, несмотря на усложнение процессов "за кадром".

Для поддержки размещения сайта Oracle Content Management в персонализированном домене необходимо работать с сетью CDN. Ее нужно настроить так, чтобы обрабатывались все запросы из заданного персонализированного домена, выполнялась должным образом их маршрутизация обратно в Oracle Content Management и вносились изменения в запросы для их правильной и безопасной обработки Oracle Content Management.

Управление доменом с помощью системы доменных имен (DNS)

Любой домен можно использовать в качестве персонализированного домена для сайта Oracle Content Management. Необходимо иметь возможность управления любым доменом, который используется в качестве персонализированного, перед его настройкой для использования с сайтом Oracle Content Management.

Из-за ограничений DNS использовать корневой домен, например *example.com*, без *www* или другой субдомен, например *store.example.com*, скорее всего, нельзя. Обратитесь к поставщикам DNS и CDN, чтобы выяснить, возможно ли использование корневого домена.

Поскольку DNS работает на уровне домена, а не на уровне пути, чтобы Oracle Content Management размещался по определенным путям в домене, а другой сервис размещался по другим путям, CDN должна управлять маршрутизацией. DNS можно использовать только для разделения трафика на уровне домена и субдомена.

Развертывание сертификатов

Необходимо создать сертификат, защищающий персонализированный домен, и разместить его в CDN. Сертификат может защищать один домен, несколько доменов и субдоменов, а также субдомены, в именах которых используются подстановочные знаки, например **.example.com*. Для персонализированного домена допустима любая комбинация. Все защищенные домены будут отображаться в сведениях о сертификате, поэтому если общий доступ к этим сведениям нежелателен, следует использовать отдельные сертификаты.



Примечание.

Процесс создания и размещения сертификатов часто зависит от CDN, поставщики CDN должны указать, как это лучше всего сделать.

Настройка сайта Oracle Content Management для персонализированных доменов

Чтобы сайт Oracle Content Management правильно загружался при использовании персонализированного домена, необходимо сначала указать Oracle Content Management, какой URL-адрес сайта использует персонализированный домен. Это выполняется в Oracle Content Management на вкладке свойств сайта.

1. В Oracle Content Management в боковом меню навигации нажмите **Сайты**.
2. Выберите сайт, с которым требуется использовать персонализированный домен, выберите **Свойства** в контекстном меню или **Боковая панель** на панели действий.

3. Введите имя персонализированного домена в поле персонализированного домена и нажмите **Сохранить**.

 **Примечание.**

Oracle Content Management может потребоваться не менее часа, чтобы начать прием запросов в домене. В любое время можно проверить ход выполнения на панели свойств сайта.

Настройка сети доставки контента для маршрутизации запросов

После того, как решение Oracle Content Management будет правильно настроено и готово к приему запросов, запросы для персонализированного домена будут направляться через DNS в CDN, а CDN должна правильно перенаправлять запросы на Oracle Content Management. Например, если сайт Oracle Content Management с URL-адресом `https://myinstance.cec.ocp.oraclecloud.com/site/MyCustomerSite/` настроен на использование персонализированного домена `https://www.example.com/store`, CDN должна быть настроена на выполнение следующих действий:

- распознавать персонализированный домен: `https://www.example.com/store`
- определять экземпляр источника Oracle Content Management с помощью персонализированного домена: `https://myinstance.cec.ocp.oraclecloud.com/`
- добавлять путь к сайту для конкретного сайта: `site/MyCustomerSite/`
- и отправлять полный URL-адрес сайта в экземпляр источника Oracle Content Management: `https://myinstance.cec.ocp.oraclecloud.com/site/MyCustomerSite/`.
- Oracle Content Management получает запрос и отправляет ответ в CDN, который пересылает его в браузер пользователя, отображая в нем только персонализированный домен: `https://www.example.com/store`

Действия по настройке CDN часто зависят от CDN, поэтому для правильной настройки определения источника необходимо работать совместно с поставщиком CDN. Он укажет CDN, как узнать экземпляр источника Oracle Content Management, и правильный путь для добавления к сайту.

 **Примечание.**

Запросы не должны содержать следующие пути:

- `/documents*`
- `/content/published*`
- `/osn*`
- `/pxysvc*`
- `/site*`

Запросы на эти пути не должны включать путь к сайту. Для правильной работы они должны разрешаться в корень исходного экземпляра Oracle Content Management.

Включение интеграции с Cobrowse

Функция совместного просмотра представляет собой средство совместной работы, используемое вместе с Oracle Cobrowse Cloud Service.

Сначала необходимо добавить интеграцию с Cobrowse Cloud Service, как разрешенную администратором сервиса. См. раздел "Интеграция с Oracle Cobrowse Cloud Service" в документе *Интеграция и расширение возможностей Oracle Content Management*.

После включения интеграции с Oracle Cobrowse Cloud Service можно настроить функцию для сайта, а затем добавить ее на отдельные страницы сайта.

Чтобы включить совместный просмотр страниц на сайте, выполните следующие действия.

1. Откройте сайт для редактирования.
2. Нажмите  на боковой панели, а затем – .
3. В разделе "Совместный просмотр" выберите **Разрешить использование Oracle Cobrowse на этом сайте**.
4. Добавьте сценарий средства запуска для сайта. Администратор Oracle Cobrowse Cloud Service может открыть консоль администрирования совместного просмотра, чтобы получить необходимый фрагмент код Javascript для средства запуска. Есть два разных типа средства запуска.
 - Точка запуска 1: кнопка совместного просмотра автоматически добавляется на страницу.
 - Точка запуска 2: позволяет настроить кнопку и интерфейс на странице.
5. Нажмите **Закреть**.

После включения совместного просмотра для сайта можно добавить функцию на страницу или настроить способ ее использования на странице. Подробнее см. в разделе [Использование компонента "Совместный просмотр" на странице](#).

Использование функции совместного просмотра на безопасных сайтах и с Site Builder

Для совместного просмотра доступно два режима: мгновенный (ICB) и расширенный (ACB). Для использования совместного просмотра в режиме ICB на безопасном сайте или при просмотре сайта, находящегося в разработке, необходимо использовать особую конфигурацию. Эта конфигурация создается в консоли администрирования совместного просмотра. См. раздел *Вход в консоль агента* в документе [Использование Standalone Cobrowse](#).

В поле "Пользовательские функции" добавьте функцию, разрешающую агентам просматривать защищенные паролем ресурсы на опубликованном безопасном сайте:

```
function () {  
  return {  
    passwordProtectedPatterns: [  
      "<PROTOCOL>://<DOMAIN>/authsite/*?**",  
      "<PROTOCOL>://<DOMAIN>/documents/*?**",  
    ],  
  };  
}
```

```
"<PROTOCOL>://<DOMAIN>/content/*?*#*"
]
}
}
```

Для использования этой функции и для опубликованных безопасных сайтов, и для просмотра сайтов в разработке добавьте следующий код:

```
function () {
return {
passwordProtectedPatterns: [
"<PROTOCOL>://<DOMAIN>/authsite/*?*#",
"<PROTOCOL>://<DOMAIN>/documents/*?*#",
"<PROTOCOL>://<DOMAIN>/content/*?*#*",
"<PROTOCOL>://<DOMAIN>/sites/*?*#*",
"<PROTOCOL>://<DOMAIN>/_themes/*?*#",
"<PROTOCOL>://<DOMAIN>/_sitescloud/*?*#",
"<PROTOCOL>://<DOMAIN>/_compdelivery/*?*#"
]
}
}
```

Настройка защищенных ресурсов – это новая функция совместного просмотра. Она использует те же шаблоны подстановочных URL-адресов, что и маскирование страниц совместного просмотра. Чтобы узнать больше, см. раздел "[Настройка маскирования страниц](#)" в документе *Руководство по развертыванию и использованию совместного просмотра*.

Добавление отслеживания аналитики

В средстве Site Builder можно добавить фрагмент кода отслеживания JavaScript, который встраивается в сайты или страницы. Этот код позволяет отслеживать веб-аналитику и упрощает интеграцию с внешними поставщиками аналитики, такими как Google, Adobe и Oracle Infinity.

Добавление отслеживания аналитики на сайт:

1. Откройте сайт для редактирования.
2. Нажмите  на боковой панели, а затем нажмите  **Аналитика**.
3. Нажмите на переключатель, чтобы включить отслеживание аналитики.
4. В поле **Фрагмент кода отслеживания JavaScript** добавьте новый фрагмент или измените существующий сценарий.

Администратор мог предоставить фрагмент кода для вашей среды. Если это так, то он будет отображаться в поле. Можно настроить сценарий или добавить собственный. После внесения изменений в предоставленный администратором фрагмент отобразится сообщение об изменении сценария. Чтобы удалить настройки, нажмите кнопку **Восстановить последний сценарий арендатора**.

Ниже приведен пример фрагмента отслеживания 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>
```

Необходимо сохранить и опубликовать это изменение, а при необходимости перевести сайт в онлайн-режим до сбора аналитики для сайта.

Просмотр данных аналитики

После публикации сайта и включения для него онлайн-режима можно просматривать отслеживаемые данные аналитики на сайте поставщика, например на сайте Google Analytics. Если используется фрагмент для отслеживания аналитики Oracle Infinity, перейдите на главную страницу Oracle Infinity и нажмите **Analytics (Аналитика)**, чтобы посмотреть данные и выбрать или создать отчеты.

Публикация сайтов

Давайте узнаем о том, как перевести сайты в оперативный или автономный режим и опубликовать изменения сайтов.

- [Перевод сайта в онлайн-режим или автономный режим](#)
- [Публикация изменений сайта](#)

Перевод сайта в онлайн-режим или автономный режим

Когда сайт находится в *онлайн-режиме*, пользователи с соответствующим доступом могут просматривать его в стандартном веб-браузере по назначенному адресу (URL). Когда сайт находится в *автономном режиме*, он недоступен для публичного просмотра. Вы можете просматривать сайт только в Oracle Content Management.

Прежде чем перевести сайт в онлайн-режим, необходимо опубликовать его. Порядок публикации сайта см. в разделе [Публикация изменений сайта](#).

Значок статуса справа показывает, находится ли сайт в онлайн-режиме или в автономном:

- Если сайт никогда ранее не публиковался, вы увидите тире (-).
- Если сайт находится в онлайн-режиме, вы увидите .
- Если сайт находится в автономном режиме, вы увидите .

Для получения информации о том, кто может получить доступ к интерактивному или автономному сайту, см. [Общие сведения о безопасности сайта](#).

Чтобы изменить статус сайта, необходимо быть владельцем сайта или иметь роль "Управление", либо, если управление сайтами включено, администраторы сайта могут изменять статус любого сайта независимо от наличия у них общего доступа к сайту.

1. На странице "Сайты" выберите нужный сайт в списке.
2. Чтобы перевести сайт в онлайн-режим или перевести его в автономный режим, выберите пункт **Перевести в онлайн-режим** или **Перевести в автономный режим** в контекстном меню. Также можно нажать  на плитке сайта, чтобы перевести сайт в онлайн-режим, или нажать  на плитке сайта, чтобы перевести сайт в автономный режим.

Отображается запрос на подтверждение.

При переводе сайта в онлайн-режим создается полностью визуализированная HTML-версия сайта, которая копируется в соответствующее местоположение в Oracle Cloud. URL-адрес онлайн-сайта отображается под его именем. Формат URL-адреса по умолчанию:

`https://service_name.identity_domain.sites.oraclecloud.com/site_name`

При переводе сайта в автономный режим он сам и все его папки и файлы удаляются из места размещения в Oracle Cloud.

Публикация изменений сайта

Чтобы опубликовать изменения на сайте, вы должны быть его владельцем или обладать ролью "Управление".

При публикации изменений на сайте помните о следующем:

- При публикации изменений на онлайн-сайте все примененные изменения немедленно становятся видимыми всем, у кого есть доступ к сайту.
 - При публикации изменений на сайте можно опубликовать все активы, предназначенные для канала сайта, или только используемые на страницах сайта активы, включая рекомендации. Например, если вы публикуете все активы, предназначенные для канала сайта, и сайт содержит список со ссылками на элементы контента, которые не являются непосредственными компонентами сайта, активы, на которые имеются ссылки, также будут опубликованы.
 - При публикации изменений на многоязычном сайте публикуются только переводы для языков, определенных в связанной с сайтом политике локализации.
 - Если переводы ранее были опубликованы для языков, которые больше не определены в связанной с сайтом политике локализации, переведенные материалы будут удалены с опубликованного сайта.
 - Если удалить страницу с многоязычного сайта, переведенные страницы также будут удалены при публикации сайта.
1. Чтобы опубликовать изменения сайта, выберите его и выберите **Опубликовать** в контекстном меню или на панели действий.
 2. Выберите, что требуется опубликовать: сайт и все его активы, предназначенные для канала публикации сайта, или сайт и все его активы, добавленные на страницы сайта.
 3. Oracle Content Management проверяет, что все строки и активы сайта имеют требуемые утверждения и переводы. Вы не сможете опубликовать сайт, если у вас нет переводов, требуемых согласно политике локализации сайта, если активы не помечены как переведенные или если актив требует утверждения, но не помечен как утвержденный. Разверните элементы, чтобы просмотреть дополнительную информацию. При возникновении проблем исправьте их и повторите попытку публикации. Если все элементы являются допустимыми, нажмите **Опубликовать**. Отображается запрос на подтверждение.
 4. Если сайт еще не находится в онлайн-режиме () , необходимо перевести его в этот режим, чтобы сделать данный сайт доступным для пользователей.

Безопасность сайтов

Когда вы публикуете сайт и делаете его доступным в онлайн-режиме, может потребоваться указать, кто может получить к нему доступ.

В рамках обеспечения безопасности сайта можно настроить, какие группы пользователей могут получить доступ к опубликованному (онлайн) сайту в зависимости от назначенной им роли. Эти роли представляют собой роли, действующие в масштабе сервиса и назначенные администратором экземпляра сервиса.

- [Общие сведения о безопасности сайта](#)
- [Изменение настроек безопасности сайта](#)

Администратор может предпринять дополнительные действия, связанные с безопасностью сайта, такие как включение настраиваемого входа, разрешение общего доступа к сайтам и темам, ограничение создания сайтов, шаблонов и компонентов, включение управления и другие действия. См. раздел "Конфигурирование настроек сайтов и активов" в документе *Администрирование Oracle Content Management*.

Общие сведения о безопасности сайта

Функции безопасности позволяют указывать пользователей, которые могут просматривать опубликованный (онлайн) сайт, взаимодействовать с защищенным содержимым сайта, а также просматривать и редактировать неопубликованный (автономный) сайт.

Безопасность сайта

Когда вы публикуете сайт и делаете его доступным в онлайн-режиме, может потребоваться указать, кто может получить к нему доступ. В зависимости от настройки вашей среды администраторами системы и сайта можно сделать сайт общедоступным, предоставить доступ для зарегистрированных пользователей или только определенных пользователей.

Для изменения настроек безопасности сайта или любых других настроек вы должны быть владельцем сайта или иметь роль "Управление". Чтобы изменить требование входа, необходимо перевести сайт в автономный режим. При изменении пользователей или ролей сайт можно оставить в онлайн-режиме. При переводе сайта в автономный режим он сам и все его папки и файлы удаляются из места размещения в Oracle Cloud.

Доступные параметры безопасности могут быть ограничены политикой шаблона, если управление сайтом включено, или политикой участника, если управление сайтом отключено. См. [Общие сведения об управлении сайтами](#).

В рамках обеспечения безопасности сайта можно настроить, какие группы пользователей могут получить доступ к опубликованному (онлайн) сайту в зависимости от назначенной им роли. Эти роли представляют собой роли, действующие в масштабе сервиса и назначенные администратором экземпляра сервиса.

- **Пользователи Cloud:** аутентифицированные пользователи выполняют вход в экземпляр сервиса, используя имя пользователя и пароль. Это относится ко всем

аутентифицированным пользователям, независимо от того, есть ли у них роли **Посетители** или **Пользователи**.

- **Посетители:** сайт доступен только для пользователей, которым назначена эта роль. Например, эту роль можно назначить пользователям, которые видят опубликованные сайты, но не имеют доступа к папкам и файлам в данном экземпляре Oracle Content Management.

 **Примечание.**

Это не относится к пользователям с ролью **Пользователи** за исключением случаев, когда они являются владельцами сайта или им явным образом предоставлен доступ к сайту.

- **Пользователи сервиса:** сайт доступен только для пользователей, которым назначена эта роль. Например, эту роль можно назначить пользователям, которые видят опубликованные сайты и одновременно имеют доступ к папкам и файлам в этом экземпляре Oracle Content Management.
- **Определенные пользователи:** опубликованный сайт доступен только для пользователей, добавленных в список участников сайта.

Общий доступ к сайту

В разделе *общего доступа к сайту* можно указать отдельных пользователей с правами доступа к неопубликованному (автономному) сайту и разрешить им просмотр, изменение и управление сайтом на основе предоставленных разрешений. Вы можете предоставить общий доступ к сайту, если являетесь владельцем сайта или вам предоставлен доступ к сайту и назначена роль "Управление".

 **Примечание.**

Любая назначенная пользователю роль с правами общего доступа повышает его уровень защиты. Например, если пользователю назначена роль **Посетители**, но при этом ему предоставлен общий доступ к сайту и назначена роль соавтора, он может изменять автономный сайт. Однако остальные пользователи с ролью **Посетители** могут только просматривать онлайн-сайт.

- **Просмотр.** Пользователи с этой ролью могут просматривать сайт в редакторе, но не могут вносить изменения.
- **Выгрузка.** Пользователям с этой ролью предоставляются те же права на работу с сайтом, что и пользователям с ролью "Просмотр".
- **Соавтор.** Кроме возможностей, предоставляемых ролью "Просмотр", эта роль позволяет также редактировать сайт, удалять страницы сайта и сам сайт, когда он находится в автономном режиме.
- **Управление.** Кроме возможностей роли "Соавтор", данная роль также позволяет добавлять пользователей и назначать им роли, публиковать изменения на сайте, работающем в онлайн-режиме, и переключать сайт в онлайн-режим и автономный режим. Автору сайта (владельцу) автоматически назначается роль "Управление".

При создании сайта создается канал, и ему присваивается имя сайта. Чтобы предоставить общий доступ к каналу другим пользователям, необходимо предоставить доступ к сайту и назначить какому-либо пользователю роль не ниже роли "Соавтор", чтобы он мог использовать канал для публикации активов. Для публикации сайта пользователю должна быть предоставлена роль "Управление".

Общий доступ к компонентам

Некоторые компоненты предоставляют доступ к общим ресурсам, включая папки, файлы и диалоги. *Общий доступ к компонентам* подразумевает и обеспечение безопасности сайта (кто может просматривать опубликованный сайт) и обмен ресурсами (кто может просматривать папки, файлы и диалоги, а также работать с ними).

Например, при добавлении компонента диспетчера документов на сайт все посетители смогут просматривать содержимое папки и, в зависимости от роли и разрешений, смогут добавлять, изменять или удалять элементы в папке.

Общие замечания:

- Автор сайта не может предоставить доступ к папке, если для доступа к папке требуются более широкие полномочия, чем у него самого. Например, если автору предоставлена роль доступа "Выгрузка" для папки, то он не может предоставить посетителям сайта полномочий соавтора для этой папки.
- Полномочия, установленные в компоненте, могут дополнять полномочия посетителя. Например, если посетителю предоставляются полномочия роли "Просмотр" (или вообще не предоставляются полномочия) при доступе к папке, то компонент диспетчера документов может предоставить более широкие полномочия для выбранной в компоненте роли. Эти расширенные полномочия применимы только в компоненте.
- Если посетителю сайта предоставлены более широкие полномочия, чем указанные для компонента, то последние будут перезаписаны.
- Полномочия, предоставленные на уровне папки, применимы ко всем вложенным папкам и файлам в этой папке.

Общедоступные сайты:

- Компоненты-диалоги поддерживаются только на безопасных сайтах..
- Компоненты диспетчера документов по умолчанию предоставляют всем посетителям полномочия роли "Выгрузка" для связанной папки. Можно изменить роль в соответствии с вышеописанными рекомендациями, а в настройках компонента можно ограничить полномочия пользователя.
- Компоненты списка папок и списка файлов предоставляют всем пользователям полномочия роли "Выгрузка". Пользователи могут просматривать и выгружать файлы независимо от своей роли.

Безопасные URL-адреса сайтов

При переводе сайта в онлайн-режим полностью визуализированная HTML-версия сайта создается и копируется в соответствующее место размещения в Oracle Cloud. URL-адрес онлайн-сайта отображается под его именем.

Формат URL-адреса по умолчанию для незащищенных сайтов:

`https://service_name.identity_domain.sites.oraclecloud.com/site_name`

Формат URL-адреса по умолчанию для защищенных сайтов:

```
https://service_name.identity_domain.sites.oraclecloud.com/authsite/  
site_name
```

Обратите внимание на добавление слова `authsite` в URL-адрес.

Изменение настроек безопасности сайта

Когда вы публикуете сайт и делаете его доступным в онлайн-режиме, может потребоваться указать, кто может получить к нему доступ. В зависимости от настройки вашей среды администраторами системы и сайта можно сделать сайт общедоступным, предоставить доступ для зарегистрированных пользователей или только определенных пользователей.

Для изменения настроек безопасности сайта или любых других настроек вы должны быть владельцем сайта или иметь роль "Управление". Чтобы изменить требование входа, необходимо перевести сайт в автономный режим. При изменении пользователей или ролей сайт можно оставить в онлайн-режиме. При переводе сайта в автономный режим он сам и все его папки и файлы удаляются из места размещения в Oracle Cloud. Чтобы перевести сайт в автономный режим, выберите сайт на странице "Сайты" и нажмите **Перевести в автономный**

режим в контекстном меню или нажмите  на панели действий. Отображается запрос на подтверждение.

Чтобы изменить безопасность сайта, выполните указанные ниже действия.

1. На странице "Сайты" выберите сайт и выберите **Свойства** в контекстном меню или нажмите на панели действий.

2. Выберите вкладку **Безопасность**.

Доступные параметры на вкладке "Безопасность сайта" зависят от того, как системный администратор задал настройки безопасности сайта, и, если управление сайтами включено, от того, как администратор сайта задал настройки безопасности в шаблоне, на основе которого создан этот сайт. См. [Начало работы с сайтами](#) и [Общие сведения об управлении сайтами](#).

3. Нажмите **Да** рядом с параметром "Требуется выполнить вход", чтобы зарегистрированным пользователям требовалось выполнить вход для просмотра онлайн-сайта. Нажмите **Нет** рядом с параметром "Требуется выполнить вход", чтобы не требовать выполнения входа и сделать онлайн-сайт общедоступным.

4. Выберите, какие группы зарегистрированных пользователей могут получить доступ к онлайн-сайту. Для выбора отдельных групп сначала отмените выбор параметра **Пользователи Cloud**.

- **Пользователи Cloud:** сайт доступен только аутентифицированным пользователям. Аутентифицированные пользователи выполняют вход в ваш домен, указывая имя пользователя и пароль. К ним относятся пользователи с ролью **Посетители Content Management Cloud** или **Пользователи Content Management Cloud**.
- **Посетители:** сайт доступен только для тех пользователей, которым назначена эта роль. Это не относится к пользователям с ролью **Пользователи Oracle Content Management Cloud**.

- **Пользователи сервиса:** сайт доступен только пользователям, которые могут войти в этот экземпляр Oracle Content Management.
 - **Определенные пользователи:** укажите пользователей, которым будет доступен сайт. Нажмите **Добавить участников**. Введите имя пользователя или часть имени пользователя в поле поиска. Выберите пользователя в списке и повторите указанные действия, чтобы добавить дополнительных пользователей. По завершении нажмите **Добавить**. Для удаления пользователя выберите **Удалить** в меню под именем пользователя.
5. Чтобы сохранить изменения и закрыть окно, нажмите кнопку **Сохранить**.
На сайте показано, что он работает в автономном режиме и что требуется вход в систему.
6. Чтобы перевести сайт в онлайн-режим, выберите в контекстном меню команду **Перевести в онлайн-режим** или нажмите  на панели действий. Выберите **Подтвердите, чтобы продолжить**, а затем нажмите кнопку **ОК**.

При переводе сайта в онлайн-режим создается полностью визуализированная HTML-версия сайта, которая копируется в соответствующее местоположение в Oracle Cloud. URL-адрес онлайн-сайта отображается под его именем.

Формат URL-адреса по умолчанию для незащищенных сайтов:

```
https://service_name-identity_domain.cec.ocp.oraclecloud.com/site/site_name
```

Формат URL-адреса по умолчанию для защищенных сайтов:

```
https://service_name-identity_domain.cec.ocp.oraclecloud.com/site/authsite/site_name
```

Обратите внимание на добавление слова `authsite` в URL-адрес.

Можно добавить URL-адрес выхода и реализовать его в виде ссылки, кнопки или страницы, доступной в меню. См. разделы "[Разделы](#)", "[Кнопки](#)" и "[Добавление страниц](#)".

Формат URL-адреса выхода:

```
https://service_name-identity_domain.cec.ocp.oraclecloud.com/cloudgate/logout.html?postlogouturl=%2Fsite%2Fauthsite%2Fsite_name
```

 **Примечание.**

Необходимо, чтобы параметр `postlogouturl` был в кодированном формате, как показано выше.

Работа с многоязычными сайтами

Узнайте о переводах сайтов для поддержки целевых многоязычных ресурсов.

- [Обзор многоязычных сайтов](#)
- [Перевод сайта](#)
- [Управление заданиями перевода сайтов](#)

Обзор многоязычных сайтов

Можно перевести сайт на любой язык, указанный в политике локализации. Чтобы просмотреть языки, указанные в соответствующей политике локализации сайта, в заголовке сайта нажмите на стрелку вниз рядом с языком. Для предварительного просмотра локализованной версии выберите язык и откройте сайт.



Примечание.

В версии Oracle Content Management Starter вы не можете создавать многоязычные сайты. Для получения полного набора функций и неограниченного количества сайтов выполните обновление до версии Oracle Content Management Premium.

Во время выбора сайта для перевода создается ZIP-файл с файлами сайта.

Примечание.

Если сайт содержит активы из нескольких репозиториев, в архив .zip будут включены только активы из репозитория по умолчанию.

Данный ZIP-файл содержит следующие файлы и папки:

- Папка `assets` — эта папка существует, если выбран перевод всего сайта или только целевых активов.
 - Папка `root`
 - * Файл `<contentItem_ID>.json` для каждого элемента содержимого, в том числе имя, описание и переводимые строки из полей в элементе содержимого, а также дополнительная информация об элементе содержимого, которая не подлежит редактированию.
 - `job.json` — это файл с описанием задания перевода. Не редактируйте этот файл.
- Папка `site` — эта папка существует, если выбран перевод всего сайта или только его контента.
 - Папка `root`
 - * Файл `<page_ID>.json` для каждой страницы сайта, в том числе имя, заголовок, описание, ключевые слова, верхний колонтитул, нижний колонтитул и переводимые строки из компонентов на данной странице, а также дополнительная информация о данной странице, которая не подлежит редактированию. Если у вас есть пользовательские компоненты, они могли быть настроены на использование переводимых строк. См. [Разработка переводимых компонентов для многоязычных сайтов](#).
 - * Файл `siteinfo.json` включает описание, ключевые слова, верхний колонтитул и нижний колонтитул.
 - * Файл `structure.json` включает навигацию и структуру сайта.
 - `job.json` — это файл с описанием задания перевода. Не редактируйте этот файл.

Перевод сайта

Если у вас есть активы на странице переведенного сайта, непосредственно или в списке контента, и вы перевели эти активы, то активы будут отображаться на том же языке, что и сайт. Элементы содержимого также можно переводить отдельно от сайта; см. [Локализация элементов содержимого](#).

При переводе сайта, содержащего активы из нескольких репозиториях, в задание на перевод включаются только активы из репозитория по умолчанию

Примечание.

Версия Oracle Content Management Starter не поддерживает работу с переводами. Для получения полного набора функций и неограниченного количества сайтов выполните обновление до версии Oracle Content Management Premium.

Создание задания перевода

При экспорте пакета перевода для перевода вручную или с помощью коннектора перевода необходимо создать задание перевода.

1. Выберите сайт, который необходимо перевести, и нажмите **Перевод**. Может потребоваться нажать кнопку **Дополнительно**, чтобы увидеть команду **Перевести**.
2. Укажите информацию о файлах сайта, которые вы экспортируете — задание перевода, и нажмите **Создать**:
 - Введите имя для задания перевода.
 - Выберите языки, на которые будет переведен сайт.
 - Выберите, нужно ли экспортировать весь сайт — контент сайта (страницы, структура и информация о сайте) и целевые активы — или только целевые активы.
 - Выберите коннектор перевода или экспортируйте пакет перевода для перевода вручную.
 - Нажмите **Создать**.

После завершения задания перевода переведенный пакет импортируется в Oracle Content Management.

Перевод вручную файлов на исходном языке

При переводе контента вручную после завершения задания перевода необходимо выгрузить файл .zip с файлами сайта.

1. На баннере нажмите **Задания перевода**, выберите задание перевода и нажмите **Выгрузить**.
2. Для каждого языка, выбранного в качестве языка перевода, создайте папку в ZIP-файле на том же уровне, что и корневая папка, например, de, es и fr. Можно выполнить перевод на подмножество выбранных языков. Например, можно выполнить перевод на немецкий язык (de) сейчас, а позднее — на испанский (es) и французский (fr) языки.
3. Скопируйте все файлы .json из корневой папки в каждую папку языка.
4. Переведите строки во всех файлах .json на соответствующие языки. Не удаляйте строки из файлов .json и не переименовывайте файлы.
5. Заархивируйте папки активов (если задание на перевод включает активы) и сайта (если задание на перевод включает контент сайта) с job.json, корневой папкой и всеми языковыми папками с переведенными файлами.

Импорт переведенного вручную пакета

Независимо от способа перевода (вручную или автоматически) необходимо импортировать готовый пакет перевода.

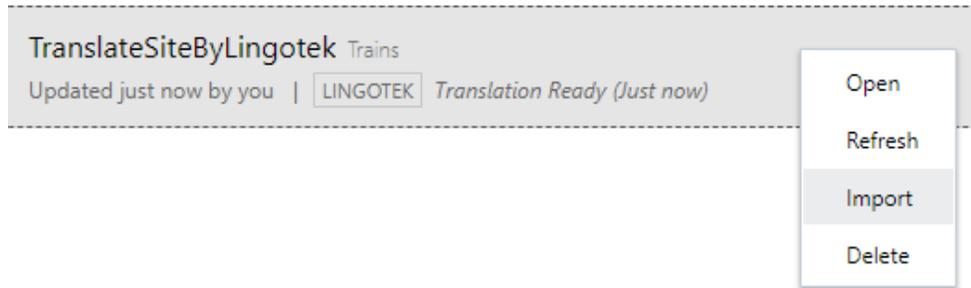
1. На странице **Задания перевода** нажмите **Импорт**.
2. Нажмите **Загрузить**, выберите ZIP-файл с переведенными файлами сайта, а затем нажмите **Открыть**.
3. После завершения загрузки нажмите **ОК**.
4. Oracle Content Management проверяет доступность в ZIP-файле переводов, определенных в задании. Чтобы просмотреть, какие страницы и активы включены в задание перевода, перейдите по ссылке в диалоговом окне.
5. Когда все готово к импорту переводов, нажмите кнопку **Импортировать**.

Над баннером появляется статус импорта. Можно просмотреть сведения о задании, нажав **Сведения**.

Импорт переведенного пакета из коннектора перевода

Независимо от способа перевода (вручную или автоматически) необходимо импортировать готовый пакет перевода.

1. Нажмите **•••** и выберите **Задания перевода**.
2. Правой кнопкой мыши нажмите завершенное задание перевода и выберите **Импорт**.



3. Oracle Content Management проверяет переводы, которые определены в задании. Чтобы просмотреть, какие страницы и активы включены в задание перевода, перейдите по ссылке в диалоговом окне.
4. Когда все готово к импорту переводов, нажмите кнопку **Импортировать**. Над баннером появляется статус импорта. Можно просмотреть сведения о задании, нажав **Сведения**.

Чтобы сделать перевод доступным на вашем сайте, после импорта переводов необходимо опубликовать сайт и убедиться, что он находится в онлайн-режиме.

Если после импорта пакета перевода сервис перевода внес какие-либо изменения в перевод, нажмите **Обновить**, чтобы обновить переводы.

Если изменить сайт после перевода, потребуется перевести все новые или отредактированные строки. При изменении сайта в Site Builder вы изменяете языковую версию сайта по умолчанию. Любые изменения структуры сайта, такие как добавление компонентов или изменение порядка страниц, будут продублированы в локализованных версиях сайта. Затем можно создать новое задание перевода для обновленных строк.

Управление заданиями перевода сайтов

Во время выбора сайта для перевода создается *задание перевода*. Затем можно загрузить файлы для перевода, перевести их и импортировать переведенные файлы.

Инструкции по созданию задания перевода см. в разделе [Перевод сайта](#).

На странице "Задания перевода" перечислены все задания перевода и их статус:

-  (готово) — ZIP-файл с файлами сайта готов к выгрузке.

-  (выполняется) — ZIP-файл выгружен. Статус "Выполняется" сохраняется, пока не будут успешно импортированы все переводы на все целевые языки.
-  (завершено) — переводы на все целевые языки для этого задания успешно импортированы.
-  (сбой) - сбой при выполнении задания перевода. Вы должны были увидеть над баннером сообщение о сбое с информацией о том, почему задание не удалось выполнить. Чтобы снова увидеть это сообщение, можно **повторно отправить** задание на перевод. Исправьте ошибку и **повторно запустите** задание.

Возможно выполнение описанных ниже операций.

- Чтобы просмотреть сведения о задании перевода, откройте его. Подробности включают исходный язык и все выбранные целевые языки, а также статус этих переводов.
- Чтобы выгрузить ZIP-файл с файлами сайта, выберите задание и нажмите **Выгрузить**.
- Для удаления задания выберите задание и нажмите **Удалить**.
- Чтобы импортировать переводы, нажмите **Импорт**, а затем **Загрузить**, выберите ZIP-файл с переведенными файлами сайта и нажмите **ОК**. Oracle Content Management проверяет доступность в ZIP-файле всех переводов, определенных в задании. Чтобы просмотреть, какие страницы и активы сайта включены в задание перевода, перейдите по ссылке в диалоговом окне. Когда все готово к импорту переводов, нажмите кнопку **Импортировать**.

Языковые стандарты для перевода

При отправке элемента на перевод целевой язык идентифицируется кодом, чтобы поставщик переводческих услуг знал, на какой язык перевести элемент и вернуть. Например, **fr** представляет французский язык, а **de** — немецкий.

Эти коды можно расширить на другие региональные диалекты. Например, **de-LI** — это код немецкого языка, на котором говорят в Лихтенштейне, а **de-LU** — это код немецкого языка, на котором говорят в Люксембурге. Но если поставщик переводческих услуг не поддерживает региональный диалект, тогда предоставленный код усекается до двухсимвольного базового языка. Например, усечение кода **de-LI** и **de-LU** дало бы результат **de**.

Если поставщик переводческих услуг поддерживает один региональный диалект, но не все, он может заменить его. Например, **ms-BN** — код малайского языка, на котором говорят в Брунее, однако если поставщик переводческих услуг не поддерживает этот диалект, он может переключиться на поддерживаемый диалект, например, **ms-MY** — код малайского языка, на котором говорят в Малайзии. Если поставщик переводческих услуг не различает диалекты, например диалект английского языка, на котором говорят в Белизе, — **en-BZ** и диалект английского языка, на котором говорят на Ямайке — **en-JM**, тогда код усекается до базового языка, в этом случае до английского языка — **en**.

Пользовательские региональные параметры для перевода

Пользовательские языковые стандарты могут быть созданы разработчиком на основе потребностей организации. Коды пользовательских языковых стандартов включают базовый язык, код регионального диалекта, если применимо, символ **x**, указывающий,

что это пользовательский языковой стандарт, и любые другие идентификационные настройки, требуемые вашей организацией. Например, код пользовательского языкового стандарта для английского языка может иметь вид **en-JM-x-custom**.

Поскольку пользовательский языковой стандарт уникален для вашей организации, коды пользовательских языковых стандартов усекаются при отправке на перевод до обозначений базового языка и регионального диалекта, если это поддерживается поставщиком переводческих услуг. В примере выше код **en-JM-x-custom** был бы усечен до **en-JM** с устранением части кода, характерной для данной настройки. Или если поставщик переводческих услуг не поддерживает код регионального диалекта Ямайки (JM), он может быть усечен только до базового языка **en**.

Настройка псевдонима региональных параметров для перенаправления URL-адреса

Можно легко задать псевдоним для региональных параметров, который будет использоваться в URL-адресе сайта в среде выполнения и в предварительном просмотре среды выполнения. Это особенно полезно, если в вашей организации используются пользовательские региональные параметры, которые могут быть слишком длинными и усложняют URL-адрес.

Например, можно задать пользовательский региональный параметр в виде "en-GB-x-cornish", который в URL-адресе будет выглядеть следующим образом:

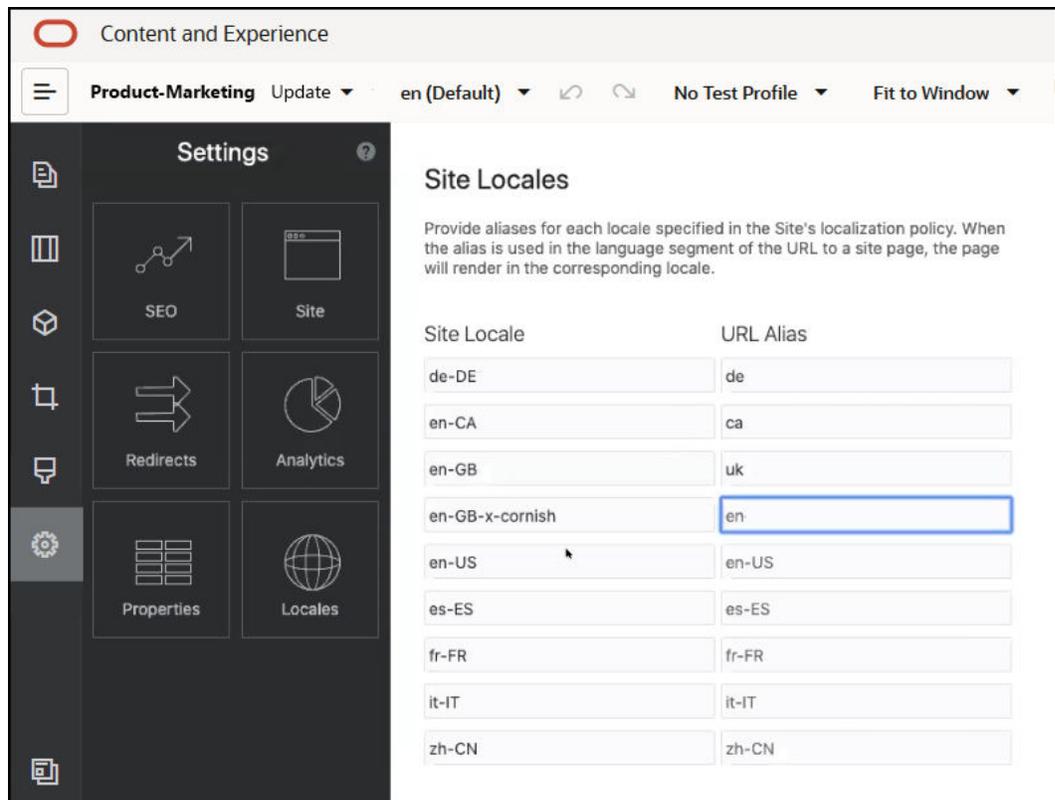
```
https://example.com/site/BlogSite/en-GB-x-cornish/home.html
```

Добавив псевдоним, можно выполнять перенаправление на более простой URL-адрес:

```
https://example.com/site/mysite/en/home.html
```

1. Откройте сайт для редактирования.
2. Нажмите  на боковой панели, а затем выберите **Региональные параметры**.

Рядом с соответствующими полями псевдонимов URL-адресов отображается список всех региональных параметров, используемых на вашем сайте.



3. Введите псевдоним рядом с региональными параметрами каждого сайта, для которого требуется псевдоним, нажмите **Заккрыть**, а затем **Сохранить**.
4. Чтобы просмотреть псевдоним, используемый в URL-адресе во время выполнения, нажмите .
5. При публикации обновления также будут опубликованы и применены внесенные изменения.

Использование перенаправление сайтов или сопоставления URL-адресов

При реструктуризации или перемещении веб-сайта можно перенаправлять запросы пользователей со старых URL-адресов на их текущие. Указание перенаправлений с кодом состояния 30x для URL-адресов позволяет сохранять закладки или опубликованные ссылки в измененных объектах сайта.

Страницы с высоким рейтингом репутации в поисковых системах можно перенести на другие URL-адреса при переходе на сайты, размещенные в Oracle Content Management, с других инфраструктурных технологий. Перенаправления помогают реорганизовать структуру URL-адреса сайта и сохранить рейтинг в поисковой системе.

- [Планирование перенаправлений](#)
- [Добавление перенаправлений сайта](#)
- [Задание правил перенаправлений в файле JSON](#)
- [Загрузка файла правил перенаправления на сайт](#)
- [Сопоставление URL-адреса сайта](#)

Планирование перенаправлений

Можно указать перенаправление ответов HTTP 30x для указанных URL-адресов. Если запрос не соответствует одному из назначенных перенаправлений, выполняется обычная обработка URL-адреса, и страница возвращается обычным способом.

Можно создать файл JSON, указывающий перенаправления, и загрузить этот файл на сервер. Сервер будет использовать этот файл JSON для обработки URL-адресов входящих запросов.

Два типа правил перенаправления позволяют перенаправлять входящие URL-адреса в новые места:

- [Простое сопоставление строк](#)
- [Упрощенное применение символов шаблона](#)

Простое сопоставление строк

Для простого сопоставления и замены строк можно указать явные URL-адреса, а затем перенаправить каждый URL-адрес, сопоставляя его напрямую целевому URL-адресу.

В таблице ниже приведены некоторые примеры сопоставления строк.

URL-адрес источника	URL-адрес целевого местоположения
/index.html	/home.htm
/products/widget	/items/knickknack

URL-адрес источника	URL-адрес целевого местоположения
/index?page=widgets	/items/widgets

Сопоставления строк просты для понимания и тестирования. Правила быстро оцениваются с помощью простых сопоставлений строк и поиска карт.

Однако параметры запроса URL-адреса не обладают большой гибкостью. Они должны точно совпадать. Если есть дополнительные параметры URL-адреса или параметры указаны в другом порядке, это может привести к несоответствию правила.

Упрощенное применение символов шаблона

Упрощенное применение символов шаблона позволяет сопоставить правило нескольким URL-адресам, а также ограничить объем регулярных выражений для обратного отслеживания, необходимых для получения результата.

Поскольку регулярные выражения могут быть сложными для записи, а вычисление плохо построенных выражений может занимать неопределенное количество времени (ReDoS), второй тип правила допускает упрощенный механизм применения символов шаблона. В нем используется подстановочный знак (" *"), который соответствует нулевому (0) или большему количеству символов во входящем URL-адресе, и ключевое слово "wildcard" со значением индекса, чтобы копировать входящие части URL-адреса в перенаправленный URL-адрес.

В таблице ниже приведены некоторые примеры упрощенного применения символов шаблона.

URL-адрес источника	URL-адрес целевого местоположения
/old/*	/new/<\$wildcard(1)\$>
/dispatch.asp?page=*&facet=Lang*	/page<\$wildcard(1)\$>/<\$wildcard(2)\$>

Упрощенное применение символов шаблона дает больше возможностей для сопоставления URL-адресов, чем простое сопоставление строк, и при этом не требуется обработка неограниченных регулярных выражений. Синтаксис прост, а сопоставление с шаблоном можно применять к URL-адресам, чтобы обрабатывать большое количество URL-адресов одним шаблоном.

Поскольку правила с подстановочными знаками основаны на регулярных выражениях, они обрабатываются немного медленнее, чем простое сопоставление строк. Большое количество правил может привести к снижению производительности в целом.

Добавление перенаправлений сайта

При изменении URL-адреса сайта для исходного URL-адреса устанавливается перенаправление на конечный URL-адрес. Это позволяет сохранить пользовательские закладки и сортировку поисковых систем.

Могут быть использованы два типа перенаправлений:

- Постоянное перенаправление, которое использует код состояния HTTP 301
- Временное перенаправление, которое использует код состояния HTTP 302

Чтобы открыть файл `redirect.json`, выполните следующие действия.

1. Откройте сайт для редактирования.
2. Нажмите  на боковой панели, а затем нажмите  **Перенаправления**.
3. Нажмите **Выбрать файл для загрузки**, перейдите к файлу и выберите его, а затем нажмите **ОК**.
4. При публикации обновления также будут опубликованы и применены внесенные изменения.

Задание правил перенаправлений в файле JSON

Можно указать правила перенаправления URL-адресов в файле JSON.

Используйте следующий формат в файле JSON, чтобы указать правила перенаправления URL-адресов.

```
{
  "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
    }
  ]
}
```

Внешняя содержащая структура в файле JSON представляет собой массив. Этот массив содержит экземпляры правил.

Сначала оцениваются правила "string", а затем правила "wildcard", по порядку. После обнаружения соответствия одному из правил, оценка последующих правил отменяется и создается соответствующее перенаправление.

Каждое правило имеет следующие свойства:

- Свойство "comment" — дополнительная строка, которая не влияет на оценку правил. Оно содержит примечания или комментарии.
- Свойство "expression" — это обязательная строка, которая совпадает с входящим URL-адресом, связанным с сайтом. В правиле с подстановочными знаками символ звездочки (*) соответствует нулевому или большему количеству символов.

- Свойство "location" — обязательная строка, указывающая местоположение или место назначения перенаправления. Перенаправление может быть задано полным или относительным URL-адресом.
- Свойство "code" — необязательное целое число, которое предоставляет код ответа HTTP, используемый при выдаче перенаправления. Данное значение должно быть одним из следующих целых чисел:
 - 301: указывает на окончательное перемещение ресурса. Это значение используется по умолчанию, если свойство "code" опущено.
 - 302: указывает на временное перемещение ресурса.
- Свойство "type" — это дополнительная строка, указывающая тип правила перенаправления. Данное значение должно быть одной из следующих строк:
 - "string" указывает более быстрое правило, выражение которого точно соответствует всему введенному URL-адресу.
 - "wildcard" указывает правило с подстановочными знаками, которое может соответствовать нескольким URL-адресам. Это значение по умолчанию, если данное свойство опущено.

Маркеры местоположения

Маркеры местоположения можно использовать для создания местоположения перенаправления. Каждый из следующих маркеров местоположения может помочь указать перенаправление:

- `<$urlPath$>`: часть пути соответствующего URL-адреса.
- `<$urlQueryString$>`: вся строка запроса URL-адреса из соответствующего URL-адреса.
- `<$urlQueryStringExcept(name1,name2)$>`: вся строка запроса URL из соответствующего URL-адреса минус именованные параметры.
- `<$wildcard(N)$>`: начинающийся с единицы индекс соответствующего подстановочного знака в соответствующем URL-адресе. (Это аналогично `\1..\9` в регулярных выражениях.)
- `<$name$>`: значение параметра именованного параметра в строке запроса. Например, если на входе указана строка запроса `msmith: ?page=42`, можно использовать `<$page$>` в заданном местоположении, чтобы поместить туда "42".

Ограничения

Приведенные ниже ограничения применяются ко всему файлу `redirects.json` и к содержащимся в нем правилам.

- Максимальный общий размер файла в Oracle Content Management составляет 250 КБ.
- Максимальное число правил в файле `rewards.json`: 1000.
- Максимальная длина "expression" для правила: 1000 символов.
- Максимальная длина "location" для правила: 2000 символов.
- Максимальное число маркеров "*" в выражении правила с подстановочными знаками: 10.

Пример соответствия строк

Правило:

```
{
  "type": "string",
  "expression": "/old/page.jsp?id=material&type=glass",
  "location": "/new/<$id$>.htm"
}
```

Следующий URL-адрес соответствует правилу:

`/old/page.jsp?id=material&type=glass`

- В результате было бы создано следующее местоположение: `/new/material.htm`
- Соответствует весь URL-адрес, включая строку запроса.
- Несмотря на то, что в местоположении используется выражение `<id>`, для этого примера оно не требуется, так как может соответствовать только одна возможная строка запроса. Местоположение может быть записано как `/new/material.htm`.

Следующий URL-адрес не соответствует правилу:

- `/old/page.jsp`
(Выражение правила содержит строку запроса, которая должна соответствовать.)
- `/old/page.jsp?id=material&type=glass&index=2`
(Дополнительное выражение `&index=2` в URL-адресе кандидата не полностью соответствует выражению правила.)
- `/old/page.jsp?type=glass&id=material`
(Порядок параметров строки запроса должен соответствовать правилу "string".)

Пример соответствия при использовании подстановочных знаков

Правило:

```
{
  "type": "wildcard",
  "expression": "/old/*/pages/*?id=*&item=sheet-*",
  "location": "/new/<$id$>/<$wildcard(4)$>.html"
}
```

Следующие URL-адреса соответствуют правилу:

- `/old/phones/android/pages/info.asp?id=XT1045&item=sheet-specs`
 - В результате было бы найдено следующее местоположение: `/new/XT1045/specs.html`
 - Часть пути URL-адреса соответствует, поэтому строка запроса также проверяется на соответствие условиям.
 - Параметры в данном примере соответствуют порядку параметров в выражении правила, но это не обязательно.

- /old/phones/android/pages/info.asp?item=sheet-specs&id=XT1045
 - В результате было бы найдено следующее местоположение: /new/XT1045/specs.html
 - Часть пути URL-адреса соответствует выражению правила до знака вопроса (?), поэтому параметры также проверяются на соответствие.
 - Несмотря на то, что параметры перечислены в выражении правила в другом порядке, они сопоставляются по отдельности.
- /old/phones/android/pages/info.asp?id=XT1045&item=sheet-specs&unrelated=thing
 - В результате было бы найдено следующее местоположение: /new/XT1045/specs.html
 - Часть пути URL-адреса соответствует, поэтому строка запроса также проверяется на соответствие условиям.
 - URL-адрес кандидата имеет дополнительный параметр &unrelated=thing, но поскольку именованные параметры запроса в выражении правила соответствуют, правило считается соответствующим.
 - Параметр unrelated был бы доступен в местоположении как маркер, так как выражение <\$unrelated\$> имело бы значение thing, даже если оно не вносит вклад в соответствие правилу.

Следующие URL-адреса не соответствуют:

- /old/pages/info.jsp
(Часть пути URL-адреса не соответствует части пути выражения правила.)
- /old/phones/android/pages/info.asp
(Часть пути URL-адреса соответствует части пути выражения правила, а параметры запроса в выражении правила нет.)
- /old/phones/android/pages/info.asp?id=cellular
(Часть пути URL-адреса соответствует части пути выражения правила, но не все параметры запроса в выражении правила соответствуют.)

Определение массива маркеров

Можно также создать массив определений маркеров в файле `redirects.json`, что поможет при настройке перенаправлений, поддерживающих несколько неофициальных URL-адресов. Это позволяет перенаправлять адрес в соответствии с характеристиками входящего URL-адреса.

Используйте приведенный ниже формат в файле `redirects.json`, чтобы определить маркеры для применения в URL-адресах правил перенаправления.

```
{
  "tokenDefinitions":
  [
    {
      "token": "sitePrefix",
      "type": "hostmatch",
      "expresion": "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"
    },
  ],
}

```

Определения меток имеют указанные ниже свойства.

- "token": имя маркера, который необходимо определить.
- "type": один из указанных ниже вариантов.
 - "hostmatch" для соответствия значению хоста входящего URL-адреса.
 - "pathmatch" для соответствия значению пути входящего URL-адреса.
 - "querymatch" для соответствия значению запроса входящего URL-адреса.
- "expression": выражение, которое должно использоваться для сопоставления. Поддерживаются подстановочные знаки.
- "value": значение, которое должно использоваться для маркера.
- "flags": по умолчанию при сопоставлении выражений учитывается регистр, если только для значения flags не задано значение caseinsensitive

При вычислении значения маркера перечисление содержимого массива tokenDefinitions будет выполняться в указанном порядке. Будет использоваться первое определение сопоставления. Если определения маркеров не соответствуют данному маркеру, вместо него будет использоваться пустая строка. По соображениям целесообразности и эффективности работы наиболее часто используемые маркеры должны располагаться в верхней части списка tokenDefinitions.

К tokenDefinitions применяются указанные ниже ограничения.

- Можно создать до 250 определений маркеров.
- Имя token должно содержать менее 100 символов.
- expression может содержать до 10 подстановочных знаков.
- expression должно содержать менее 1000 символов.
- value должно содержать менее 1000 символов.

Пример

Например, возможно, у вас есть следующий файл `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"
    },
  ],
}
```

В этом случае свойство `location` правила имеет маркер `<$pathPrefix$>`. Маркер `pathPrefix` определяется в разделе `tokenDefinitions`. Если входящий URL-адрес совпадает с `"vanity.com"`, то для `pathPrefix` будет задано значение `/fashion`. Это будет использоваться в ответе `location`, в результате чего мы получим `/fashion/about/new-privacy-policy.html`.

Предположим, что первый URL-адрес персонализированного домена — `http://example.com/legacy-privacy-policy.html`. Это будет соответствовать первому и единственному правилу перенаправления.

Объявленное значение `location` для этого правила — `<$pathPrefix$>/about/new-privacy-policy.html`. В этой ситуации необходимо оценить маркер `<$pathPrefix$>`. Для этого содержимое массива `tokenDefinitions` перечисляется, чтобы найти совпадение.

Учитывается первое определение маркера. Его `token` является предпочтительным, поэтому он будет оцениваться в дальнейшем. Выражение `vanity.com` не соответствует входящему URL-адресу `example.com`, поэтому это определение не удовлетворяет требованиям, и перечисление продолжается.

На этом этапе определения маркеров больше не существуют, поэтому для значения маркера `<$pathPrefix$>` используется пустая строка. Окончательное местоположение для этого перенаправления: `/about/new-privacy-policy.html`.

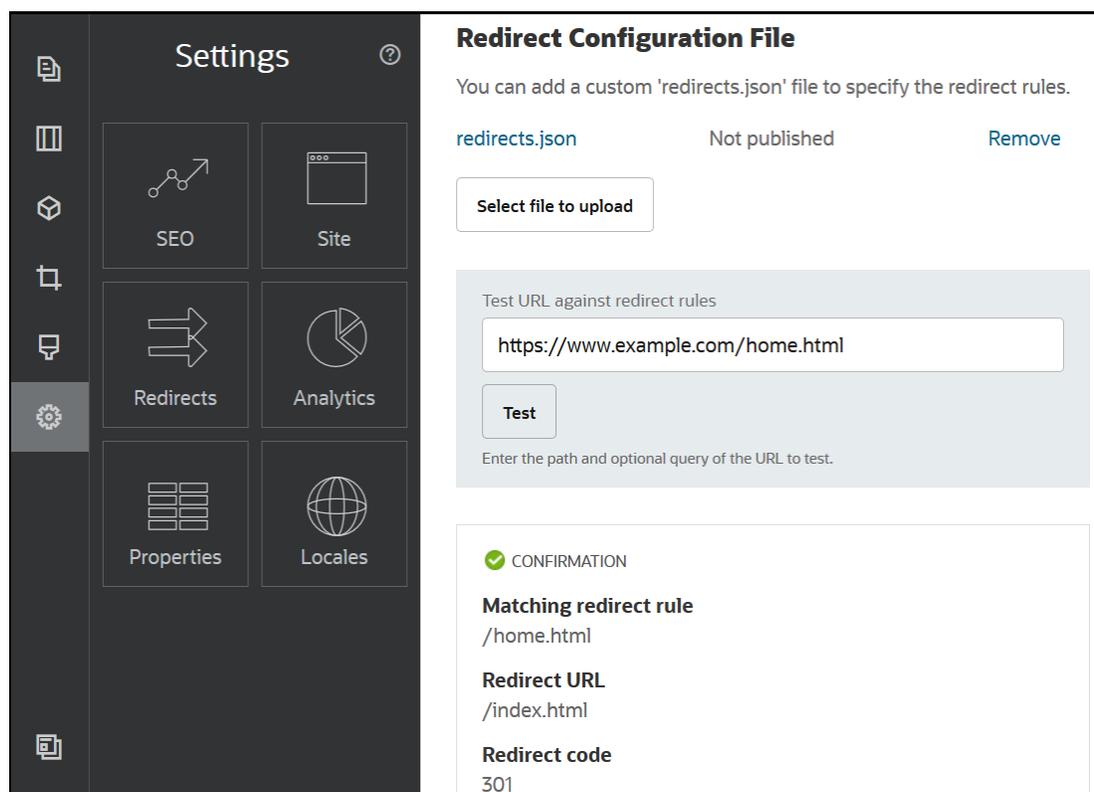
Предположим, что второй URL-адрес персонализированного домена — `http://vanity.com/legacy-privacy-policy.html`. Как и в случае с первым URL-адресом, объявленное значение `location` для этого правила — `<$pathPrefix$>/about/new-privacy-policy.html`. В этой ситуации необходимо оценить маркер `<$pathPrefix$>`. Для этого содержимое массива `tokenDefinitions` перечисляется, чтобы найти совпадение.

Учитывается первое определение маркера. Как и раньше, его `token` является предпочтительным, поэтому он будет оцениваться в дальнейшем. Выражение `vanity.com` соответствует входящему URL-адресу `vanity.com`, поэтому это определение удовлетворяет требованиям, а значение `/fashion` используется в качестве значения маркера.

Поскольку для маркера было найдено совпадение, перечисление содержимого массива определений маркеров прекращается и окончательное местоположение вычисляется как `/fashion/about/new-privacy-policy.html`.

Тестирование перенаправлений сайта

При редактировании сайта можно повторно протестировать его перенаправления, открыв панель **Настройки** и нажав **Перенаправления**. Введите URL-адрес для тестирования и нажмите **Тест**.



Загрузка файла правил перенаправления на сайт

Правила перенаправления можно загрузить на сайт в Oracle Content Management.

Чтобы загрузить файл `redirect.json` на сайт, выполните указанные ниже действия.

1. Откройте сайт для редактирования.
2. Нажмите  на боковой панели, а затем – .
3. Нажмите **Выбрать файл для загрузки**, перейдите к файлу и выберите его, а затем нажмите **ОК**.

4. При публикации обновления также будут опубликованы и применены внесенные изменения.

Сопоставление URL-адреса сайта

После создания и публикации сайта с помощью Oracle Content Management можно настроить службу доменных имен (DNS) так, чтобы сайт был доступен по зарегистрированному имени домена, например, `www.mysite.com`.

Служба доменных имен (DNS) указывает, где пользователи могут найти ваши веб-страницы, сопоставляя доменное имя местонахождению или каноническому имени (CNAME) вашего сайта.

Чтобы сопоставить доменное имя, потребуется следующее:

- URL-адрес вашего экземпляра Oracle Content Management. Обычно он имеет следующий вид:
`service-tenant.documents.datacenter.oraclecloud.com`
- Доменное имя, зарегистрированное регистратором доменных имен. Например, `www.example.com`. Также может быть поддоменом, например `www.example.com/subdomain`.
- Учетная запись у поставщика сети доставки контента (CDN). Oracle Content Management Обеспечивает интеграцию с Akamai. Обратитесь в службу поддержки Oracle, чтобы настроить Akamai для своего экземпляра.

Чтобы использовать собственную сеть доставки контента (CDN), а не Oracle Content Management, выполните указанные ниже действия.

У разных поставщиков услуг сервиса доменных имен разные веб-интерфейсы и разные процедуры обновления записи CNAME. В описанной ниже процедуре описываются необходимые сведения и общие инструкции.

Чтобы сопоставить URL-адрес сайта доменному имени, выполните указанные ниже действия.

1. Запросите сертификат SSL для домена у поставщика сети доставки контента. Например, `https://www.example.com`.
2. Настройте сеть доставки контента таким образом, чтобы:
 - a. Сеть доставки контента принимала все входящие запросы к домену и перенаправляла их с помощью протокола HTTPS.
 - b. Исходная точка указывает на домен из Oracle Content Management:
`service-tenant.documents.datacenter.oraclecloud.com`
3. Измените файл зоны сервера DNS, чтобы сопоставить имя домена с пограничным сервером, предоставленным поставщиком сети доставки контента:
`domain CNAME CDN Server`
4. Дождитесь распространения данного обновления. В зависимости от вашей DNS-службы это может занять от 2 до 48 часов.

После распространения данного изменения сайт доступен по доменному имени.
Пример:

`https://www.mysite.com/site_name`

По умолчанию конечная точка API-интерфейса REST Oracle Cloud для управления содержимым доступна при использовании стандартного URL-адреса, указанного для сайта. В частности, компоненты списка папок и файлов используют API REST для выполнения операций с папками и файлами. При использовании настраиваемого URL-адреса убедитесь, что с вашим доменным именем у вас есть доступ к конечной точке.
Пример:

`https://www.mysite.com/documents`

Повышение производительности сайта

Производительность доставки и рендеринга контента в браузере можно повысить, используя кэш браузера. Указанная выше визуализация без прокрутки (ATF) также позволяет улучшить визуализацию сайта.

- [Использование кэширования для повышения производительности](#)
- [Визуализация без прокрутки \(ATF\)](#)

Использование кэширования для повышения производительности

При доставке элементов контента, цифровых активов и сайтов следует использовать все преимущества кэша браузера посетителя для повышения производительности доставки и рендеринга контента в браузере.

Сайты, темы, элементы контента и цифровые активы кэшируются в кэше браузера посетителя на некоторое время. После обновления сайта, темы, элемента контента или цифрового актива ключ сброса кэша в URL-адресе изменяется, чтобы браузер мог извлечь другой URL-адрес и получить новый элемент.

Ключ кэша помогает управлять использованием кэша браузера путем формированию ссылок только на текущие ресурсы. Хотя ключ кэша включен в URL-адрес, он представляет собой логический элемент, а не физическое местоположение (папку), как это часто бывает. Изменение ключа кэша не указывает на необходимость поиска ресурса в другом физическом местоположении; оно просто уведомляет сервер о необходимости извлечения текущей версии ресурса.

Ресурсы могут быть статическими, такими как CSS, JS и файлы изображений, или динамическими, такими как данные страниц, сайта и элементов контента. При построении веб-сайта используются пять категорий ресурсов:

- Ресурсы продуктов — это ресурсы, входящие в состав продукта, которые обновляются при выпуске или исправлении новой версии продукта.
- Ресурсы сайта — это ресурсы, которые входят в состав сайта, такие как файл `structure.json`, данные страницы и изображения. Они обновляются при публикации сайта. Контроллер описан ниже.
- Ресурсы тем — это ресурсы, входящие в состав тем, такие как макеты, CSS и изображения. Они обновляются при публикации темы.
- Ресурсы компонентов — это ресурсы, входят в состав пользовательских компонентов. К ним относятся файлы HTML, JS и CSS, а также файлы изображений, из которых состоит компонент. Они обновляются при публикации компонента. Если один компонент изменяется и повторно публикуется, то ключ кэша изменяется для всех компонентов, так как это единственный ключ для всех компонентов.

- Ресурсы SaaS — это ресурсы, обслуживающие элементы контента и цифровые элементы. Они обновляются при публикации или повторной публикации элементов контента или при изменении целевого объекта коллекции.

В последующих разделах описывается кэширование для среды выполнения Oracle Content Management и Site Builder.

- [Кэширование времени выполнения](#)
- [Кэширование в Site Builder](#)

Кэширование времени выполнения

Для времени выполнения заголовков Oracle Content Management Cache-Control задается на 15 дней. В URL-адрес всех ресурсов добавляется ключ кэша.

В случае совпадения URL-адреса браузер обслуживает ресурс из локального кэша, если это доступно. При обновлении ресурса ключ кэша обновляется в URL-адресе, что заставляет браузер отправить серверу новый запрос и обновить локальный кэш.

Контроллер с ключами кэша также кэшируется на 1 минуту. Поэтому обновленные ключи кэша не отображаются в течение 1 минуты.

Во время выполнения сервер возвращает `controller.html` с последними ключами кэша для ресурсов продукта, сайта, темы, компонентов и SaaS. В файл `controller.html` добавляется сценарий с ключами; например:

```
<script type="text/javascript">
  var SCSCacheKeys = {
    product: '123',
    site: '456',
    theme: '789',
    component: '012',
    caas: '345'

  };
</script>
```

Эти ключи используются сценарием `controller.js` для построения URL-адресов, подобных тем, что описаны в следующей таблице.

Тип ресурсов	Примеры
Ресурсы продуктов	<code>/sitePrefix/productCacheKey/_sitesclouddelivery/...</code> <code>/mySite/_cache_947d/_sitesclouddelivery/</code>
Ресурсы тем	<code>/sitePrefix/themeCacheKey/_themesdelivery/themeName/...</code>

Тип ресурсов	Примеры
Ресурсы компонентов	<code>/sitePrefix/compCacheKey/_compdelivery/compName/...</code>
Ресурсы узла	<code>/sitePrefix/siteCacheKey/content/...</code> <code>/sitePrefix/siteCacheKey/structure.json</code> <code>/sitePrefix/siteCacheKey/pages/100.json</code>
Ресурсы CAAS	<code>RegularCaaSUrl?cacheKey=caasCacheKey</code>

Вставляя ключ кэша в URL-адреса, подобные этому, Oracle Content Management может заставить браузер загружать обновленные ресурсы, фактически изменив URL-адрес, чтобы браузер считал, что это новый ресурс.

 **Примечание.**

Для безопасных сайтов кэшируются только ресурсы продукта, темы и компонента, а не контент сайта или CaaS.

Кэширование в Site Builder

В Site Builder статические ресурсы кэшируются на 15 дней.

При использовании Site Builder кэшируются ресурсы продукта, темы и компонентов. (Это не происходит для ресурсов сайтов и CaaS.) При запуске или обновлении Site Builder ключи кэша темы и компонентов создаются заново.

Если внести изменения в тему или компонент и требуется, чтобы изменения отображались в Site Builder, необходимо обновить Site Builder (F5).

Визуализация без прокрутки (ATF)

Визуализация ATF создает впечатление, что веб-сайт загружается быстрее, чем на самом деле. Цель заключается в том, чтобы сначала отобразить все видимые части страницы, а затем, перед прокруткой вниз, визуализировать остальные части страницы, которые изначально не видны.

Ячейка может иметь назначение "визуализация без прокрутки", при котором на вкладке отображается значок.

Для рендеринга ячейки таким новым образом она должна быть помечена как `scs-atf` следующим образом:

```
<div class="scs-slot scs-atf" id="headline"></div>
```

Компонент должен уведомлять рендерер о завершении рендеринга. Готовые компоненты делают это по умолчанию. Пользовательский компонент может выполнять дополнительные вызовы и должен выполнять следующие действия:

1. Уведомить рендерер о том, что необходимо дождаться завершения рендеринга.
2. Уведомить рендерер о завершении.

Для #1, напротив файла `appinfo.json` пользовательского компонента, добавьте следующее свойство:

```
"initialData": {  
  . . .  
  "customRenderComplete": true,  
  . . .  
}
```

Для #2, в файле `render.js` компонента, убедитесь, что рендерер уведомлен о завершении, путем вызова следующего метода:

```
SitesSDK.setProperty('renderComplete', true);
```

Если не все компоненты в ячейке ATF своевременно сообщают о завершении обработки, рендерер ожидает 2 секунды, прежде чем продолжить обработку остальной части страниц. Если этого времени недостаточно, его можно продлить, объявив в шаблоне страницы следующую глобальную переменную:

```
var SCSAtfPassTimeout = 3000;
```

Примечание.

Время указывается в миллисекундах, поэтому в данном примере задано время ожидания 3 секунды.

Диагностические данные для процесса ATF предоставляются API-интерфейсом. Можно вызвать в консоли отладки следующий метод или, если это необходимо, получить к нему доступ со страницы:

```
SCSRenderAPI.getRenderMetrics();
```

Пример:

```
{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: 324.23500000000007}

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: 330.09000000000003}

23. 22:{atf: false, componentId: "fd603b96-2beb-4e87-
a54f-12d0e264cd0a", time: 331.38500000000005}

24. length:23

25. __proto__:Array(0)

5.   componentCount:23

6.   currentTime:16243.400000000001

7.   mainPassEndTime:331.38500000000005

8.   mainPassStartTime:316.475

9.   renderStartTime:264.36

10. __proto__:Object
```

Часть IV

Разработка для сайтов

В этой части подробно описывается, как начать разработку сайта, и разъясняются значение и процедуры разработки шаблонов, тем, стилей, компонентов и других полезных элементов. Здесь содержатся указанные ниже главы.

- [Настройка проектов и стилей](#)
- [Общие сведения об использовании фона](#)
- [Задание триггеров и действий](#)
- [Разработка шаблонов](#)
- [Разработка тем](#)
- [Разработка макетов](#)
- [Разработка компонентов](#)

Настройка проектов и стилей

У каждой темы для Oracle Content Management должен быть проект, который определяет внешний вид страниц и настройки стиля компонентов, используемых на сайте по умолчанию для темы.

- [О проектах](#)
- [Проектные файлы](#)
- [Настройка стилей списка диалогов](#)
- [Настройка стилей списков папок и файлов](#)
- [Настройка значков социальной панели](#)
- [Настройка новых расширений стилей интервью для Oracle Intelligent Advisor](#)

О проектах

У каждой темы для Oracle Content Management должен быть дизайн, который определяет внешний вид страниц и настройки стиля компонентов, используемых в теме.

При создании сайта пользователи должны выбрать шаблон, который включает тему по умолчанию. Можно создать или выбрать другую тему для сайта.

При создании темы, а также при проектировании макетов страниц необходимо указать доступные стили, которые будут отображаться на панели "Настройки" для каждого типа компонента, доступного пользователям (Абзац, Заголовок, Изображение и т. д.). В проекте может быть несколько стилей, чтобы задать настройки по умолчанию для различных компонентов. Можно выбрать шрифт текста и размер шрифта, кадрирование изображений и т. д., а затем сохранить их как именованный проект. Элементы проекта влияют на внешний вид макетов страниц при визуализации.

См. раздел "Работа со страницами сайта" в документе *Построение сайтов с помощью Oracle Content Management*.

Проектные файлы

Файлами дизайна по умолчанию для темы считаются следующие два файла: `design.json` и `design.css`.

- `design.json` указывает стили для компонентов
- `design.css` предоставляет определения значений классов (например, цвет и шрифт)

Эти файлы находятся в каталоге `/designs/default/` структуры темы.

Файл design.json

Файл design.json имеет следующую структуру:

```
{
  "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": []
    }
  }
}
```

Каждая запись "styles":[] может содержать список стилей для конкретного компонента. Например, компонент заголовка предоставляет следующие стили по умолчанию:

```
"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"  
    }  
]
```

Примечание.

Если для компонента создаются пользовательские стили и они сопоставляются со стилями в `design.json`, не требуется использовать полное имя, как то, что указано в системе. Просто укажите строку, которую планируется использовать. Например, вместо `"COMP_STYLE_BOX"` для имени просто используйте `"Box"`. Это означает, что в списке на вкладке **Стили** панели настроек этого компонента появится имя `"Box"`, а не `"COMP_STYLE_BOX"`.

Значения имен сопоставляются с фактическими словами для отображения в пользовательском интерфейсе, например следующим образом:

```
"COMP_STYLE_FLAT": "Flat",  
"COMP_STYLE_HIGHLIGHT": "Highlight",  
"COMP_STYLE_DIVIDER": "Divider",
```

Файл `design.css`

Файл `design.css` предоставляет определения значений классов. Вот несколько примеров.

```
.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;
```

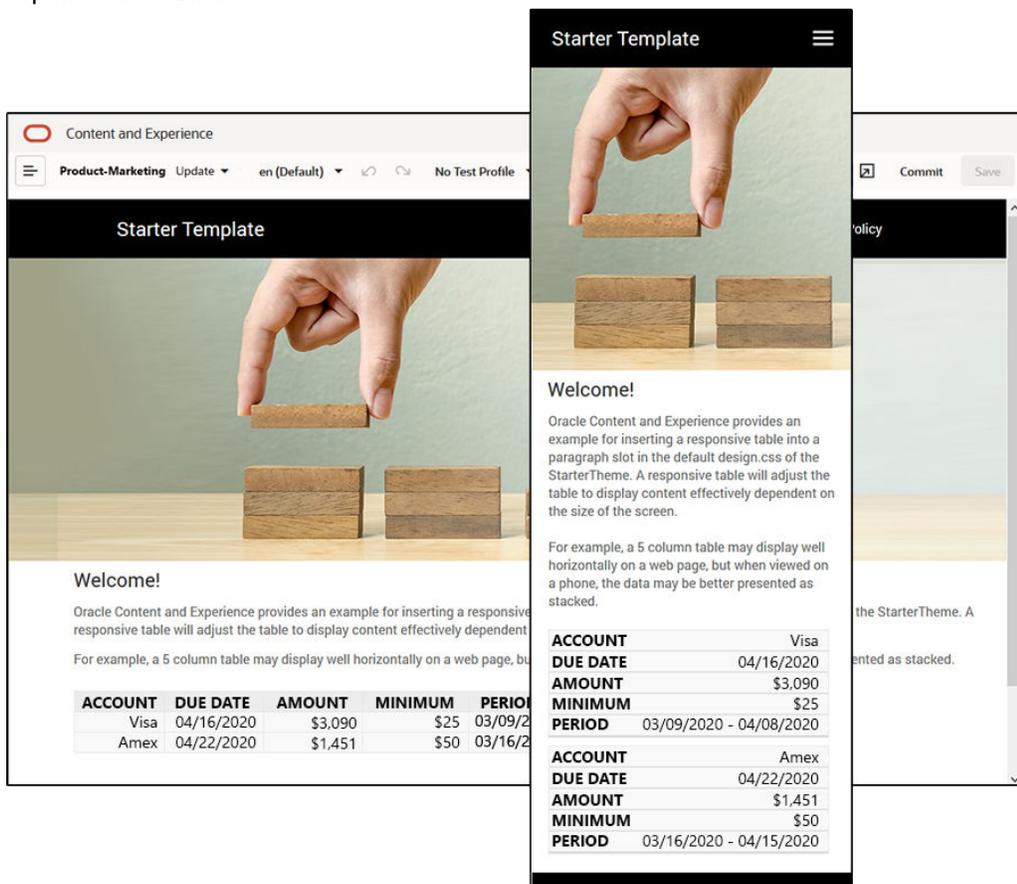
```
box-shadow: inset 0 1px 0 #f7f8f9;
text-shadow: 0 1px 0 #f7f8f9; }
```

```
.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; }
```

Дизайн адаптивной таблицы

Oracle Content Management предоставляет пример CSS адаптивной таблицы в компоненте абзаца, который позволяет реализовать многоярусное отображение данных строк на мобильных устройствах.

Адаптивная таблица настраивает таблицу для эффективного отображения контента в зависимости от размера экрана. Например, на веб-странице таблица из 5 столбцов хорошо отображается горизонтально, но при просмотре на телефоне данные могут быть лучше представлены в виде стека. Обратите внимание, что для правильного поведения адаптивным таблицам требуется строка заголовка.



В следующем созданном HTML-коде обратите внимание на то, что в каждую ячейку таблицы добавлен атрибут `data-label` со значениями, совпадающими с текстом заголовка столбца.

```
<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>
```

Если этот атрибут находится в каждой ячейке, при применении приведенных ниже правил CSS теги TD накладываются друг на друга.

```
.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;
}
```

Обратите внимание, что приведенное ниже правило для мультимедиа запрашивает размер экрана и вступает в силу, только если размер экрана меньше 767 пикселей:

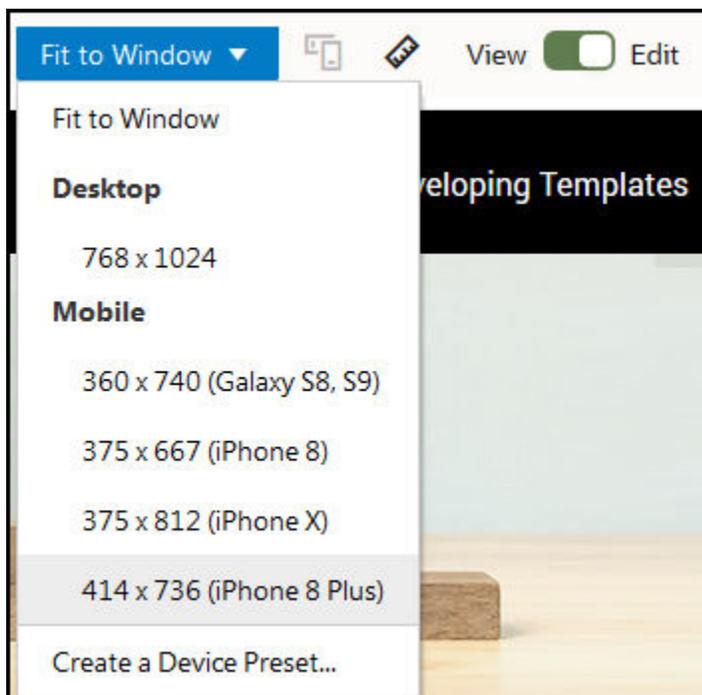
```
@media screen and (max-width: 767px) {
```

Пример кода для использования CSS, чтобы разрешить вставленным таблицам быть адаптивными, находится в файле `design.css` предоставленной темы `StarterTheme`. При построении сайта на основе `StarterTheme` таблицы, вставленные в ячейку абзаца, по умолчанию будут адаптивными. Чтобы вставить таблицу, выполните указанные ниже действия.

1. В разделе Oracle Content Management откройте сайт и переключитесь в режим **Настройки**.
2. Создайте новое обновление или выберите существующее обновление для изменения.

3. Перетащите новый компонент абзаца на страницу и щелкните там, где требуется вставить таблицу, или в существующем абзаце, где требуется вставить таблицу.
4. На панели инструментов форматированного текста нажмите  и задайте свойства таблицы. Убедитесь, что первая строка выбрана в качестве заголовка, и отрегулируйте ширину таблицы для эффективной работы на самом маленьком экране, который предполагается использовать. Например, если ожидается, что сайт будет отображаться на телефоне, ширина по умолчанию в 767 пикселей будет слишком большой, чтобы полностью отобразиться на экране телефона, даже в многоярусном виде. Можно задать меньшую ширину таблицы, например 300 пикселей, или установить ширину на 100 %.

По завершении переключитесь обратно в режим **просмотра** и выберите вариант просмотра с шириной экрана менее 767 пикселей для предварительного просмотра результатов. Режим просмотра необходим, так как таблица не ведет себя адаптивно в режиме редактирования.



Если при построении сайта на основе другой темы планируется использовать адаптивные таблицы, код из файла `design.css` StarterTheme необходимо скопировать в файл `design.css` используемой темы.

1. Чтобы скопировать код из файла `design.css` темы StarterTheme, на боковой панели навигации Oracle Content Management нажмите **Разработчик**.
2. Нажмите **Просмотреть все темы**.
3. Выберите **StarterTheme** и нажмите **Открыть**.
4. Нажмите имена проектов, чтобы открыть папку, а затем нажмите **По умолчанию**.
5. Выберите файл `design.css` в теме StarterTheme и нажмите **Выгрузить**.

- Откройте файл в текстовом редакторе и найдите раздел файла, который начинается с комментария *An example CSS of how to render a table responsively* (Пример использования CSS для адаптивного рендеринга таблицы).
- Выделите код до следующего комментария и скопируйте его.

```
/**
 * 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. Повторите эти действия, чтобы выгрузить файл design.css шаблона, который необходимо изменить. Откройте шаблон и вставьте скопированный код в файл.
9. Сохраните изменения и загрузите измененный файл design.css в изменяемую тему как новую версию.

Настройка стилей списка диалогов

Стиль компонента "Список диалогов" можно настроить, добавив селекторы в файл design.css.

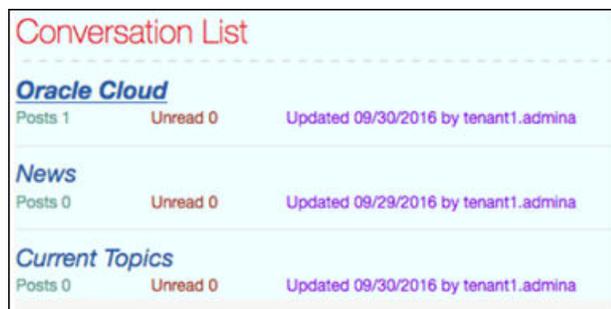
Используйте эти селекторы CSS для настройки стиля компонента "Список диалогов".

Имя селектора	Описание
.scs-convo-list-cust .scs-convo-list-container	Наиболее удаленное значение DIV компонента
.scs-convo-list-cust .scs-convo-list-title	Заголовок диалога в списке, когда он выбран
.scs-convo-list-cust .scs-convo-list-line-separator	Разделитель между заголовком списка и списком
.scs-convo-list-cust .scs-convo-list-convo-title	Заголовок диалога в списке
.scs-convo-list-cust .scs-convo-list-convo-line-separator	Разделитель между диалогами
.scs-convo-list-cust .scs-convo-list-active	Заголовок диалога в списке, когда он выбран
.scs-convo-list-cust .scs-convo-list-convo-posts	Количество сообщений в диалоге
.scs-convo-list-cust .scs-convo-list-convo-unread	Количество непрочитанных сообщений в диалоге
.scs-convo-list-cust .scs-convo-list-convo-updated	Дата последнего обновления диалога
.scs-convo-list-cust .scs-convo-list-no-convo-msg	Сообщение, когда список пуст
.scs-convo-list-cust .scs-convo-list-no-auth-msg	Сообщение, когда компонент "Список диалогов" визуализируется на общедоступном сайте без авторизации пользователя

См. раздел "Использование стилей и форматирование" в документе *Построение сайтов с помощью Oracle Content Management*.

Пример

В этом примере иллюстрируется использование пользовательских таблиц CSS для изменения шрифта, стиля и цвета в компоненте "Список диалогов".



Следующий код показывает пользовательские таблицы CSS, используемые для создания примера:

```
.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;
    }
    
```

Настройка стилей списков папок и файлов

Стили компонентов "Список папок" и "Список файлов" можно настроить, добавив селекторы в файл design.css.

Компонент "Список папок" используется для отображения списка папок в указанной папке учетной записи Oracle Content Management. Список папок автоматически обменивается данными с компонентом "Список файлов" и диспетчером документов на странице, чтобы отображать файлы в папке, выбранной в списке папок.

Компонент "Список файлов" можно использовать для просмотра файлов из указанной папки в учетной записи Oracle Content Management. Список файлов автоматически обменивается данными с компонентом "Список папок" на странице. В результате отображаются файлы в папке, выбранной в списке папок.

Селекторы CSS списка папок

Используйте эти селекторы CSS для настройки стиля компонента "Список папок".

Имя селектора	Описание
.scs-folder-list-cust .scs-folder-list-container	Наиболее удаленное значение DIV компонента
.scs-folder-list-cust .scs-folder-list-folder-title	Имя папки
.scs-folder-list-cust .scs-folder-list-line-separator	Разделитель между именем папки и списком подпапок
.scs-folder-list-cust .scs-folder-list-sub-folder-title	Имя подпапки
.scs-folder-list-cust .scs-folder-list-sub-folder-title-active	Имя подпапки, когда она выбрана
.scs-folder-list-cust .scs-folder-list-no-folder-msg	Сообщение, если нет подпапок для отображения

Селекторы CSS списка файлов

Используйте эти селекторы CSS для настройки стиля компонента "Список файлов".

Имя селектора	Описание
.scs-file-list-cust .scs-file-list-container	Наиболее удаленное значение DIV компонента
.scs-file-list-cust .scs-file-list-folder-title	Имя папки
.scs-file-list-cust .scs-file-list-line-separator	Разделитель между именем папки и списком файлов
.scs-file-list-cust .scs-file-list-row	Строка с информацией для файла
.scs-file-list-cust .scs-file-list-left-col	Эскиз файла, расположенный в левой части компонента
.scs-file-list-cust .scs-file-list-mid-col	Средний раздел компонента, который содержит имя, описание, последнее изменение и размер файла
.scs-file-list-cust .scs-file-list-file-title	Имя файла находится в средней части приложения
.scs-file-list-cust .scs-file-list-file-desc	Описание файла находится в средней части приложения
.scs-file-list-cust .scs-file-list-file-lastModified	Последнее изменение файла
.scs-file-list-cust .scs-file-list-file-size	Размер файла с вертикальным разделителем, начиная с последнего изменения
.scs-file-list-cust .scs-file-list-file-size-no-sep	Размер файла без вертикального разделителя (последнее изменение не показано)
.scs-file-list-cust .scs-file-list-right-col	Правый раздел приложения
.scs-file-list-cust .scs-file-list-file-download-icon	Значок загрузки, расположенный в правой части приложения
.scs-file-list-cust .scs-file-list-no-file-msg	Сообщение, если нет файлов для отображения

См. разделы "Списки файлов" и "Список папок" в документе *Построение сайтов с помощью Oracle Content Management*.

Настройка значков социальной панели

Можно создать пользовательские значки социальных сетей для использования на социальной панели на сайте по умолчанию темы.

Значки социальных сетей, которые отображаются на социальной панели сайта, определяются его темой. При изменении темы сайта значки социальных сетей изменяются вместе с темой. В темы Oracle Content Management включены общие значки для социальных сетей Facebook, Twitter, LinkedIn, Google+ и YouTube.

Пользовательские значки социальных сетей можно добавить в компонент значков социальных сетей, изменив файлы `design.json` и `design.css`.

Файл design.json

В файле design.json можно указать новые значки, используя имя и структуру классов, как показано в этом примере кода:

```
"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"
      }
    ]
  }
}
```

Файл design.css

В файле design.css можно добавить новые значки, используя имя и спецификацию url, как показано в этом примере кода:

```
.scs-facebook-icon {
  background-image: url("facebook.png"); }
.scs-twitter-icon {
  background-image: url("twitter.png"); }
.scs-linkedin-icon {
  background-image: url("linkedin.png"); }
```

Настройка новых расширений стилей интервью для Oracle Intelligent Advisor

Можно создать стиль для интервью Oracle Intelligent Advisor (OIA) (ранее Oracle Policy Automation), чтобы унифицировать внешний вид в соответствии с корпоративными требованиями.

В файле design.css можно настроить следующие селекторы классов CSS. Все эти селекторы классов имеют префикс "scs-opaintorse-".

Каждый класс определяет все стили для указанного компонента. Класс реализует полный контроль, а существующий стиль OIA не будет использоваться.

Селектор	Применяется к	Описание
interview	interview	Область интервью включает в себя весь контент интервью, включая заголовок, нижний колонтитул и область навигации.
interviewContent	контент интервью	Область контента интервью включает заголовок экрана и элементы управления, но не включает верхний и нижний колонтитулы и область навигации.
screenTitleBlock	блок заголовка экрана	Блок заголовка экрана представляет собой область, которая включает заголовок экрана, а также любые другие виджеты, содержащиеся в этой строке, такие как раскрывающийся список экрана и/или кнопки "Далее" и "Назад".
screenTitle	заголовок экрана	Область заголовка экрана — это просто область, содержащая заголовок экрана.
nextButton	кнопка "Далее"	Кнопка "Далее".
backButton	кнопка "Назад"	Кнопка "Назад".
restartButton	кнопка перезапуска	Кнопка перезапуска.
exitButton	кнопка выхода	Кнопка выхода.
header	header	Область верхнего колонтитула.
footer	footer	Область нижнего колонтитула.
question	текст вопроса	Стиль текста вопроса.
control	контейнер для элементов управления	Стиль элемента, содержащего элементы управления.
label	управление надписями	Создание стилей для элементов управления надписями.
controlError	все элементы управления	Стили для текстового контейнера ошибок.
controlErrorText	все элементы управления	Стиль для тега span в тексте ошибки.
textInput	текстовое поле с одной строкой, пароль и маскированный контент	Стили элементов управления вводом текста.
textAreaInput	многострочное текстовое поле	Стили для элементов управления многострочным вводом текста.
calendarInput	календарь	Стиль элементов управления вводом календарных данных. Поддерживает поле iconColor, которое позволяет изменить цвет значка календаря, и поле keepIcon, в котором указывается, должен ли отображаться значок календаря.
dropDownInput	раскрывающийся список	Стили для элементов управления вводом раскрывающегося списка.
filterDropDownInput	отфильтрованный раскрывающийся список	Стили для элементов управления вводом отфильтрованного раскрывающегося списка. Поддерживает поле iconColor, позволяющее изменить цвет стрелки раскрывающегося списка.

Селектор	Применяется к	Описание
listInput	фиксированный список	Стили для элементов управления вводом фиксированного списка.
radioInput	переключатели	Стили элементов управления вводом переключателей. При создании стилей для них могут быть использованы параметры <code>borderColor</code> и <code>fillColor</code> . Свойство <code>iconType</code> может изменить тип используемого значка. В настоящее время "tick" и "fill" — единственные поддерживаемые альтернативные параметры.
checkboxInput	флажок	Стили элементов управления вводом для флажков. При создании стилей для них могут быть использованы параметры <code>borderColor</code> и <code>fillColor</code> . Свойство <code>iconType</code> может изменить тип используемого значка. В настоящее время "square" и "fill" — единственные поддерживаемые альтернативные параметры.
autocompleteInput	настраиваемый поиск	Стили поля <code>autocomplete</code> при использовании расширения <code>customSearch</code> .
captchaInput	поле ввода CAPTCHA	Стили поля ввода, в котором пользователь вводит CAPTCHA.
signatureInput	элемент управления подписью	Стили для элементов управления подписью. Поддерживает дополнительное поле <code>inkColor</code> , которое позволяет изменять рукописные данные для подписи.
explanationHeader	элемент управления пояснением	Стили расширяемого заголовка верхнего уровня для элементов управления пояснением.
explanationText	элемент управления пояснением	Стиль расширенного текста элементов управления пояснением.
signatureClearButton	элемент управления подписью	Стили кнопки очистки на элементе управления подписью.
uploadAddButton	элемент управления загрузкой	Стили кнопки "Добавить" на элементе управления загрузкой.
entityRemoveButton	элемент управления коллекцией объектов	Элемент управления коллекцией объектов.

Примеры с расширениями стиля, определенными в `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;
}
```

Общие сведения об использовании фона

Цвет и изображение фона можно указать как для страниц сайта, так и для отдельных ячеек страницы. Различные виды фона для страниц, ячеек и компонентов накладываются друг на друга. Например, если указать фон для ячейки, он накладывается на фон, заданный для страницы.

- [О фонах и темах](#)
- [Порядок реализации фона](#)
- [Где хранятся настройки](#)

Общие сведения о фоне и темах

Фон в основном предназначен для использования на страницах и ячейках сайта. Набор фоновых изображений для страниц и ячеек должен быть настроен на уровне темы пользователями, а не разработчиками.

В ячейках тем можно настроить фоновые эффекты, однако такие эффекты могут переопределять фоновые эффекты, заданные для страниц при редактировании сайта.

Фон не изменяет темы или их составляющие файлы. Тема не может быть изменена настройками фона для страницы, а другая страница сайта, основанная на том же макете, не наследует никаких настроек фона исходной страницы. Для добавления или изменения фона темы требуется обновить тему.

Пользователи могут настроить разный фон на каждой странице сайта, а также на разных страницах, использующих один и тот же макет. Это невозможно в рамках темы. Хотя тема может указывать фон страницы, настройки фона пользователя могут переопределить эту спецификацию. (Настройки фона переопределяют лишь настройки определенной страницы; они никоим образом не меняют саму тему.)

Избегайте использования стилей фонов, переопределяющих тему, которые задаются пользователями на страницах и ячейках сайта. Это может происходить несколькими способами:

- Настройки фона страницы на сайте могут быть переопределены темой при использовании атрибутов "style" на основе элементов в <body> и ячейках.
- Настройки фона страницы на сайте могут быть переопределены темой, если стили фона темы помечены как "important" в файле каскадной таблицы стилей (CSS) темы.

См. раздел "Изменение фона или темы" в документе *Построение сайтов с помощью Oracle Content Management*.

Порядок реализации фона

Для реализации фона страниц и ячеек Oracle Content Management динамически создает таблицу стилей CSS в теге <head> страницы.

Селектор стилей — это селектор на основе тегов (`body`) для настроек фона страницы. Для настроек фона ячеек и селектора на основе идентификатора используется идентификатор ячейки.

Например, настройка цвета фона для страницы может привести к следующей разметке CSS в теге `<head>` страницы:

```
body
{
    background-color: #fa7c9d;
}
```

Аналогично, настройка изображения фона в ячейке может привести к следующей разметке CSS в теге `<head>` страницы:

```
#PageFooter
{
    background-image: url("footer_image.png");
}
```

Эта реализация означает, что стили, прямо указанные в атрибуте `"style"` тега `body` или в элементе ячейки, могут переопределять настройки, заданные в таблице стилей в коде `<head>`.

! Важно.

Разработчики тем следует соблюдать осторожность, чтобы не переопределить настройки фона с помощью стилей на основе элементов.

См. раздел "Изменение фона или темы" в документе *Построение сайтов с помощью Oracle Content Management*.

Где хранятся настройки

При рендеринге страниц настройки фона динамически записываются в теги `"style"` в коде `<head>` страницы.

Настройки фона сохраняются в файлах модели страницы (например, `<pageid>.json`). В частности, настройки фона страницы хранятся в разделе `properties.styles`, а настройки фона ячейки — в разделе `slots[<slot_id>].styles`.

Настройки фона хранятся в файлах JSON страницы, в частности, в разделе `"styles"`, как показано в этой репрезентативной выборке.

```
{
  "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"
    }
}
}
}

```

Задание триггеров и действий

Связь между компонентами (включая компоненты, визуализированные во встроенных рамках) может быть настроена таким образом, что триггер внутри компонента вызывает действие с другим компонентом.

- [Использование триггеров и действий](#)
- [Задание триггеров](#)
- [Установка действий](#)

Использование триггеров и действий

Связь между компонентами (включая компоненты, визуализированные во встроенных рамках) может быть настроена таким образом, что **триггер** внутри компонента вызывает **действие** в другом компоненте.

Триггеры — часть межкомпонентного взаимодействия Oracle Content Management. Любой компонент может вызвать любое количество триггеров. Компонент может предоставить полезные данные для триггера, которые затем передаются любому действию, которое выполняется при срабатывании триггера. Для каждого триггера можно выбрать действия, которые будут выполняться. Компоненты, созданные для совместной работы, могут автоматически вызывать триггеры для выполнения действий в другом компоненте, без вмешательства пользователя.

Основной процесс включает в себя:

1. Регистрация триггеров
2. Вызов триггеров
3. Регистрация действий
4. Выполнение действий для проверки настройки

Например, компоненты "Кнопка" можно использовать для выполнения одного или нескольких действий, таких как отображение или скрытие компонентов страницы и отображение сообщений. Один компонент может содержать список местоположений офисов. При выборе местоположения в этом списке сведения о местоположении отображаются в другом компоненте.

Для настраиваемых компонентов триггеры и действия — часть регистрационных данных компонентов, а не реализации компонентов. В регистрационных данных имеется запись "triggers": [], и "actions": [], содержащая список триггеров и действий, поддерживаемых компонентом. Фактический синтаксис аналогичен синтаксису локальных и удаленных компонентов, отличается только местоположение и способ его извлечения.

См. раздел "Использование триггеров и действий" в документе *Построение сайтов с помощью Oracle Content Management*.

Задание триггеров

Компонент может включать триггеры, которые будут выполнять действия в других компонентах. Триггеры, которые должны вызываться компонентами, необходимо зарегистрировать.

Компонент предоставляет полезные данные для триггера, которые передаются любому действию, которое выполняется при срабатывании триггера. Для каждого триггера можно выбрать действия, которые будут выполняться. Компоненты, созданные для совместной работы, могут автоматически вызывать триггеры для выполнения действий без вмешательства пользователя.

Регистрация триггеров

Для пользовательского компонента триггеры регистрируются как часть данных регистрации компонента. Чтобы добавить триггер, обновите массив свойств "triggers" с каждым триггером, поддерживаемым компонентом. Кроме того, необходимо указать полезные данные, поддерживаемые триггером, чтобы можно было создать пользовательский интерфейс, позволяющий пользователям сопоставить значения в полезных данных со свойствами, поддерживаемыми действием.

1. Измените файл `appinfo.json` и просмотрите запись `"triggers":[]`:

```
"triggers": [{
  "triggerName": "helloWorldWhoAreYou",
  "triggerDescription": "Show Who I Am",
  "triggerPayload": [{
    "name": "whoAreYou",
    "displayName": "Who I Am"
  }]
}],
```

2. Синхронизируйте файл с сервером сайтов.

В этом примере определена запись триггера `triggername` ("helloWorldWhoAreYou"). Значение имени должно быть уникальным. Затем дается описание триггера ("Show Who I Am"), которое используется в диалоговом окне пользовательского интерфейса для отображения триггера. Наконец, для триггера определяются полезные данные с отдельными значениями. Пользователи смогут выбирать записи в этих полезных данных и сопоставлять их с полями в действии.

После регистрации триггера при переходе на вкладку **Ссылка** на панели "Настройки" компонента можно увидеть и выбрать этот триггер.

Вызов триггеров

В любой момент триггеры могут быть вызваны компонентом. Обычно триггер вызывается в результате взаимодействия с пользователем, например при нажатии кнопки или выборе строки в таблице. Компонент может вызвать триггер на основе любого критерия, например, при изменении данных из-за вызова API-интерфейса REST. При срабатывании триггера может быть выполнено любое количество действий.

Вот пример вызова триггера:

1. Измените файл `render.js` и добавьте в объект `viewModel` функцию JavaScript, которая будет обращаться к Sites SDK для вызова триггера.

```
self.raiseTrigger = function (triggerName) {
  SitesSDK.publish(SitesSDK.MESSAGE_TYPES.TRIGGER_ACTIONS, {
    'triggerName': 'helloWorldWhoAreYou',
    'triggerPayload': { "whoAreYou": "This is " + self.whoAreYou() + "!" }
  });
};
```

2. Добавьте в пользовательский интерфейс запись для вызова функции, запускающей триггер (`-edit template.html`), и кнопку перед `</div>`.

```
<button data-bind="click raiseTrigger">Who Am I?</button>
```

3. Синхронизируйте или загрузите файл `render.js` на сервер своего экземпляра Oracle Content Management.

В объекте `ViewModel` создана функция JavaScript, вызываемая при нажатии кнопки. Эта функция вызывает комплект Sites SDK, чтобы он инициировал все действия, определенные для триггера "helloWorldWhoAreYou". Она также передается посредством объекта `triggerPayload` с одним полем — "whoAreYou". Значения "helloWorldWhoAreYou" и "whoAreYou" соответствуют значениям, введенным при регистрации триггера на предыдущем шаге.

Примечание.

При выполнении действия предопределенного порядка нет. Несмотря на то, что каждое действие вызывается в том порядке, в котором оно указано в списке, не нужно ждать его завершения, чтобы вызвать следующее действие. Если действие выполняет асинхронный вызов, оно может не завершиться до выполнения следующего действия.

Установка действий

Можно задать компонент для использования регистрации действия, чтобы его можно было поместить на страницу, которая будет выполнять действия в пределах компонента.

Регистрация действий

Действия вызываются в компонентах при срабатывании триггеров. Компонент может зарегистрировать любое количество действий, а также определить полезные данные, поддерживаемые действием. Выбирая действия, пользователи могут ввести полезные данные, передаваемые действию.

Как и при регистрации триггеров, действия, поддерживаемые компонентом, можно зарегистрировать в данных регистрации `appinfo.json` для темы.

Ниже приведен пример регистрации действия.

1. Измените файл `appinfo.json` своего компонента и обновите запись `"actions": []`,

```
"actions": [{
  "actionName": "helloWorldChangeWhoIAm",
  "actionDescription": "Change Who I Am",
  "actionPayload": [{
    "name": "whoAreYou",
    "description": "Who are you?",
    "type": {
      "ojComponent": {
        "component": "ojInputText"
      }
    },
    "value": ""
  }]
}]
```

2. После регистрации действие отображается в диалоговом окне действия, которое вызывается, если нажать имя триггера на вкладке **Ссылка** панели "Настройка" компонента.

Выполнение действий

После регистрации действия компоненты можно перетащить на страницу, на которой действия выполняются в пределах компонента. Чтобы выполнить действие, компонент должен прослушивать сообщение `EXECUTE_ACTION`. Это сообщение также включает передаваемые действию полезные данные, из которых необходимо извлечь ожидаемые значения.

Например, чтобы прослушивать сообщение `EXECUTE_ACTION`, измените файл `render.js` и обновите объект `ViewModel` с учетом следующих записей:

```
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);
  }
};
```

В результате создается функция JavaScript для выполнения действия, а затем используется комплект Sites SDK для вызова функции при появлении сообщения `EXECUTE_ACTION`.

Это действие вызывается при появлении сообщения `EXECUTE_ACTION`, а компонент обрабатывает только те действия, для обработки которых он предназначен. Для этого необходимо проверить имя действия, чтобы убедиться, что его можно обработать.

Полезные данные для действия — это набор значений. В данном примере предполагается, что значение — это первая запись в массиве. Как правило, значения полезных данных должны быть указаны в массиве.

 **Примечание.**

Поскольку прослушиватель действий представляет собой обратный вызов, необходимо использовать замыкание JavaScript или соответствующим образом привязать функцию, чтобы обеспечить доступ к `viewModel` при выполнении функции.

Разработка шаблонов

Пакет шаблонов содержит версию сайта для разработки, тему с макетами страниц, стиль и систему навигации, а также связанные компоненты, используемые на сайте. Oracle Content Management предоставляет набор шаблонов, которые можно использовать для создания сайтов и в качестве отправной точки для создания пользовательских шаблонов.

- [О шаблонах](#)
- [Структура базового шаблона](#)
- [Создание шаблона](#)
- [Экспорт шаблона](#)
- [Импорт шаблона](#)
- [Работа с начальным шаблоном](#)
- [Создание шаблона сайта на основе шаблона Bootstrap или шаблона разработки веб-сайтов](#)
- [Разработка шаблонов в Developer Cloud Service](#)

О шаблонах

Шаблон содержит все необходимые элементы для создания веб-сайта, включая сайт с примерами страниц и контента, тему со стилями, элементы навигации, активы (например, изображения), а также связанные компоненты.

В Oracle Content Management предоставлен ряд шаблонов для использования при создании сайтов. Как правило, эти шаблоны устанавливаются администратором во время инициализации сервиса. См. раздел *Конфигурирование настроек сайтов* в документе *Администрирование Oracle Content Management*.

При создании нового сайта необходимо выбрать шаблон. Шаблоны объединяют темы с сайтами и компонентами для управления функциями или решениями, такими как партнерский портал или маркетинговая кампания.

Список готовых шаблонов см. в разделе [Общие сведения о шаблонах](#).

Хотя разработчики и пользователи могут создавать новые шаблоны, изменять и заменять существующие шаблоны, одна из основных задач разработчиков — разработка новых шаблонов. В основном, этот процесс состоит из следующих этапов.

1. Создайте новый шаблон, скопировав существующий шаблон, например начальный шаблон JET. Это также позволяет получить тему, связанную с шаблоном.

Пример:

```
cec create-template My_JET_Template -f JETStarterTemplate
```

2. Экспортируйте шаблон в файле .zip в свою среду разработки.

3. Откройте файлы в пакете шаблонов и внесите свои изменения.
4. Создайте переработанный пакет шаблонов в файле `.zip`.
5. Используйте интерфейс Oracle Content Management для импорта нового шаблона в свой экземпляр.
6. Предоставьте совместный доступ к шаблону, чтобы другие могли его использовать.

В качестве альтернативы с помощью интерфейса Oracle Content Management можно изменить шаблон путем добавления и изменения макетов страниц и активов, расширить структуру сайта, добавить компоненты на страницы сайта и добавить предварительно заданный контент, который должен отображаться на всех сайтах, использующих данную тему в этом шаблоне.

См. также [Управление шаблонами](#).

Структура базового шаблона

Базовая структура шаблона включает сайт (с активами, макетами, страницами и контентом), связанную тему и пользовательские компоненты.

При создании сайта необходимо выбрать шаблон, чтобы предоставить структуру сайта и исходный контент, тему со спецификациями дизайна и макета, а также пользовательские компоненты.

Шаблон имеет определенную структуру, как иллюстрируется в данном примере, показывающем основные папки и файлы.

```
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
```

Папки и файлы компонентов и тем описаны в других разделах данного руководства. См. [О разработке компонентов](#) и [Базовая структура темы](#).

Примечания.

- Как правило, тема используется совместно с шаблонами, если только не применяется начальный шаблон JET или обычный начальный шаблон, который использует копию темы. См. [Работа с начальным шаблоном](#).
- Тема больше не содержит сайт. Папки и файлы сайта находятся в папке /`template`.

Папка `template_name/template` содержит папки и файлы для сайта.

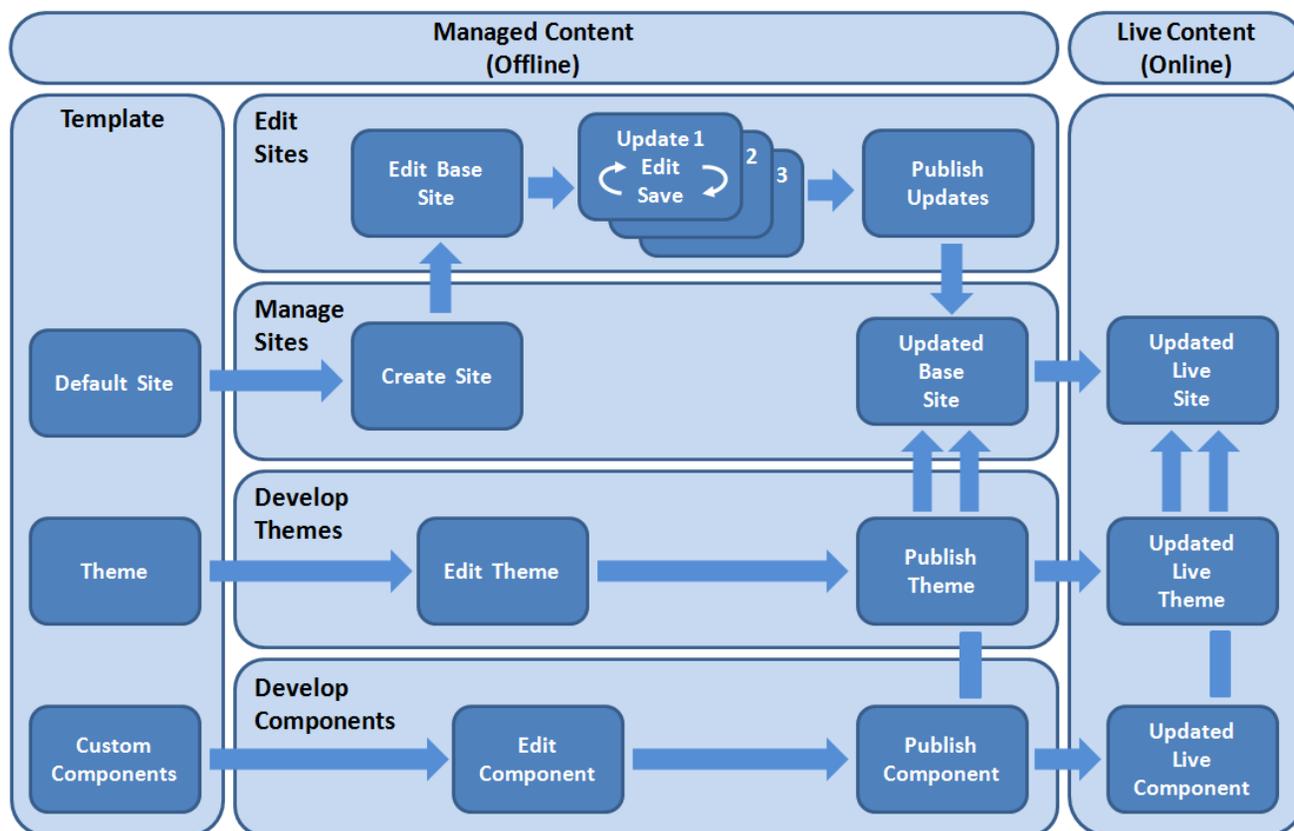
- `assets`: содержит изображения, отображаемые на странице сведений о шаблоне в пользовательском интерфейсе.
- `content`: содержит управляемый контент, используемый на сайте.
- `layouts`: в настоящее время не используется.
- `pages`: содержит все JSON-файлы страницы с данными. Используется формат `nnn.json`, где `nnn` — идентификатор страницы.
- `variants`: содержит сведения обо всех обновлениях сайта.
- `_folder.json`: содержит метаданные для шаблона, такие как имя автора сайта, имя сайта, GUID элемента, короткое и длинное описание сайта.
- `_folder_icon.png`: представляет сайт в пользовательском интерфейсе.

- `componentsused.json`: (Устарел.) Записывает пользовательские компоненты, используемые на сайте (если таковые имеются). Поддерживается только для обратной совместимости.
- `controller.html`: содержит код ключа, который отображает сайт в браузере. Если требуется внести изменения в этот файл, Oracle рекомендует сделать это с помощью настроек сайта в интерфейсе. Файл можно изменить в автономном режиме. См. [Настройка файла контроллера](#).
- `siteinfo.json`: определяет имя сайта и имя связанной темы, а также другие метаданные сайта. Не изменяйте этот файл.
- `structure.json`: определяет иерархию сайта для страниц (родительские и дочерние страницы). API для рендеринга можно использовать для рисования структуры дерева при настройке навигации по сайту. См. [Навигация на сайте](#) и [Обращение к API рендеринга](#).

Создание шаблона

Если вы хотите использовать сайт в качестве отправной точки для других сайтов, можно создать шаблон на основе этого сайта. Новый шаблон также можно создать, скопировав существующий шаблон и внося изменения в полученную копию.

При создании шаблона на основе существующего сайта копия сайта в новом шаблоне используется в качестве сайта по умолчанию. Шаблон содержит ссылки на используемую тему сайта и дополнительные компоненты, размещенные на страницах сайта. Тема и дополнительные компоненты не копируются в шаблон. Шаблон ссылается на них так же, как это происходит на сайте. Шаблон, созданный на основе сайта, отражает состояние этого сайта на момент создания шаблона. Изменения, внесенные на сайте после создания шаблона, не влияют на сайт, сохраненный в этом шаблоне.



При создании нового шаблона путем копирования существующего шаблона и переименования копии пользователь вносит изменения в копию. Обратите внимание, что при копировании шаблона, сведения об общем доступе к шаблону не копируются.

Не используйте для шаблонов, тем, компонентов, сайтов и страниц сайтов следующие имена: authsite, content, pages, scstemplate_*, _comps, _components, _compsdelivery, _idcservice, _sitescloud, _sitesclouddelivery, _themes, _themesdelivery. Следующие имена можно использовать только для страниц сайтов (не для шаблонов, тем, компонентов и сайтов): documents, sites.

Чтобы создать собственный пользовательский шаблон (с сайтом и темой), лучше всего использовать начальный шаблон JET или начальный шаблон, предоставленный Oracle Content Management, который содержит основные элементы сайта и связанной темы. Начальный шаблон содержит информацию и инструкции, записанные на страницах сайта, чтобы помочь узнать, как задать разметку и дизайн сайта и темы в пользовательском шаблоне.

См. [Управление шаблонами](#) и [Работа с начальным шаблоном](#).

Экспорт шаблона

Экспортированный шаблон можно изменить в автономном режиме, а затем импортировать как новый шаблон или заменить существующий. Можно также

экспортировать шаблон, переместить его в другой экземпляр Oracle Content Management и импортировать в этом экземпляре.

При экспорте шаблона фактически выполняется копирование шаблона в папку в Oracle Content Management в качестве одиночного файла .zip. Пакет шаблона можно выгрузить непосредственно из этой папки и распаковать для работы с отдельными файлами. После внесения всех необходимых изменений в файлы шаблона создайте zip-файл с пакетом шаблона, импортируйте его в Oracle Content Management, чтобы перезаписать исходный шаблон или создать новый.

 **Примечание.**

При экспорте шаблона сведения об общем доступе к шаблону не сохраняются.

Экспорт шаблона:

1. В боковом меню навигации Oracle Content Management нажмите **Разработчик**.
Отображается страница **Разработчик**.
2. Нажмите **Смотреть все шаблоны**.
Отображается список доступных шаблонов.
3. Выберите шаблон и выберите **Экспорт** в контекстном меню или нажмите  на панели действий.
4. Перейдите в папку или создайте новую папку, нажав **Создать**, указав имя, добавив описание (необязательно) и нажав **Создать**.
Чтобы открыть папку, нажмите на значок папки или ее имя.
5. Выберите папку, установив рядом с ней флажок, и нажмите **ОК**.
В выбранной папке создается файл шаблона. Файл имеет расширение ZIP, а его имя соответствует имени шаблона.

Импорт шаблона

Экспортированный шаблон можно изменить в автономном режиме, а затем импортировать как новый шаблон или заменить существующий. Можно также экспортировать шаблон, переместить его в другой экземпляр Oracle Content Management и импортировать в этом экземпляре.

При экспорте шаблона фактически выполняется копирование шаблона в Oracle Content Management в качестве файла .zip. Пакет шаблона можно выгрузить непосредственно из этой папки и распаковать для работы с отдельными файлами. После внесения всех необходимых изменений в файлы шаблона создайте файл .zip, содержащий пакет шаблона, импортируйте его в Oracle Content Management и замените исходный шаблон или создайте новый.

Импорт пакета шаблона:

1. На боковой панели навигации Oracle Content Management нажмите **Разработчик**

- Отображается страница **Разработчик**.
- Нажмите **Смотреть все шаблоны**.
Отображается список доступных шаблонов.
 - Нажмите **Создать** и выберите **Импортировать пакет шаблона**.
 - Если пакет шаблона загружен, перейдите в соответствующую папку. Чтобы открыть папку, нажмите на значок папки или ее имя.
Если пакет шаблона еще не загружен:
 - Перейдите в папку, куда требуется загрузить пакет шаблона, или создайте папку с помощью кнопки **Создать**. Укажите имя папки, при необходимости добавьте описание и нажмите **Создать**.
 - Нажмите **Загрузить**.
 - Найдите и выберите пакет шаблона, затем нажмите **Открыть**.
На индикаторе выполнения отображается имя файла и статус загрузки.
 - Выберите пакет шаблона, установив флажок рядом с именем файла, и нажмите **ОК**.
При отсутствии конфликтов между элементами импортированного шаблона и существующими шаблонами, темами или пользовательскими компонентами создаются новые папки Oracle Content Management для шаблона, связанной темы и любых пользовательских компонентов.
 - Если имя или идентификатор шаблона, темы или дополнительного компонента уже существует, отображается запрос на устранение конфликтов.
В зависимости от типа конфликта предлагается создать новый шаблон, тему или дополнительный компонент либо заменить существующий шаблон, тему или компонент импортируемой версией.

Работа с начальным шаблоном

Для создания нового сайта с темой и пользовательскими компонентами используется копия начального шаблона, предоставляемого средой Oracle Content Management.

Начальный шаблон собирает все элементы, необходимые для создания пользовательского веб-сайта, в одном пакете: сайт по умолчанию, макет, навигация, образец контента, тема, связанные элементы контента и т. д. В отличие от других шаблонов, предоставляемых с Oracle Content Management, начальный шаблон предоставляет простую в использовании базовую инфраструктуру для создания нового сайта, включая новую тему.

Примечание.

Тема обычно совместно используется в шаблонах, но в начальном шаблоне используется копия темы.

Начальные шаблоны, StarterTemplate и JETStarterTemplate, предоставляются вместе с другими шаблонами в Oracle Content Management, когда администратор включает шаблоны во время установки и настройки сервиса. Начальный шаблон Oracle JavaScript Extension Toolkit (JET) включает в себя новейшие стили JET для шаблонов,

содержит некоторый контент страниц (например, компоненты JET) и начальные компоненты для создания шаблонов и сайтов на основе JET.

Основной процесс

Ниже приведен основной процесс работы с начальным шаблоном.

1. Создайте новый сайт и выберите начальный шаблон. Вместе с новым сайтом создается новая начальная тема.

Примечание.

Тщательно выберите название сайта. Имя сайта дублируется как имя новой темы, а тема будет видна пользователям после публикации сайта. После создания темы изменить ее имя невозможно.

Начальный сайт и тема содержат набор папок и файлов, необходимых для начала разработки сайта и темы.

2. Синхронизируйте новую тему со своим рабочим столом. Вы можете работать на рабочем столе, чтобы расширить и настроить тему с помощью макетов и статических активов. Поскольку это копия темы и используется имя, характерное для шаблона, в тему можно вносить изменения, не затрагивая исходную тему.
3. Откройте сайт в Site Builder. Site Builder можно использовать для добавления элементов в структуру сайта, добавления компонентов и взаимодействий на страницы и добавления контента, который станет частью сайта по умолчанию нового шаблона. Контент сайта, указанный в начальном шаблоне, можно повторно использовать или удалить.
4. Когда сайт и тема готовы, синхронизируйте изменения со своим экземпляром Oracle Content Management, а затем создайте новый шаблон на основе выбранного сайта.
5. Предоставьте совместный доступ к шаблону, чтобы другие могли его использовать.

Создание шаблона с помощью начального шаблона

Чтобы использовать начальный шаблон, выполните указанные ниже действия.

1. В боковом меню навигации Oracle Content Management щелкните **Сайты**.
Отображается список существующих сайтов.
2. Нажмите **Создать**.
3. В диалоговом окне "Создание сайта" выберите начальный шаблон для использования в качестве основы для своего сайта.
Новый сайт использует тему, предоставленную в начальном шаблоне, переименованном, чтобы соответствовать названию нового сайта.
4. В данном диалоговом окне введите имя сайта. Это имя используется в URL-адресе сайта. Можно употреблять буквы, цифры, символы подчеркивания () и дефисы (-). Введенный пробел автоматически заменяется символом подчеркивания.

Не используйте для шаблонов, тем, компонентов, сайтов и страниц сайтов следующие имена: authsite, content, pages, scstemplate_*, _comps, _components, _compsdelivery, _idcservice, _sitescloud, _sitesclouddelivery, _themes, _themesdelivery. Следующие имена можно использовать только для страниц сайтов (не для шаблонов, тем, компонентов и сайтов): documents, sites.

 **Примечание.**

URL-адрес сайта Oracle Content Management зависит от регистра букв. Регистр букв в запросе или строках фрагментов определяется разработчиками в своем коде.

```
https://host[:port]][/]path[?query][#fragment]
      \_____/      \_____/
      Location      Data
```

5. Дополнительно можно ввести описание сайта.
6. По завершении нажмите кнопку **Создать**.

В индикаторе выполнения отображается имя нового сайта и состояние создания. После создания сайта имя появляется в списке сайтов. Сначала сайт находится в автономном состоянии.

Чтобы быстро найти недавно созданные сайты в списке, отсортируйте его по дате **последнего обновления**. Только что созданный сайт появится наверху списка.

 **Примечание.**

Создателю сайта автоматически назначается роль менеджера для созданного сайта.

7. Используйте настольное приложение для синхронизации темы с рабочим столом. Теперь должны отображаться иерархии папок и файлы для темы.
8. Начальная тема содержит минимальный набор папок и файлов, например:

```
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
```

Файл `x-close.png` содержит значок закрытия по умолчанию для всплывающего окна с сообщением о согласии на использование файлов cookie.

Папка `/layouts` содержит файл начального макета страницы (`index.html`) со следующим контентом:

- Набор тегов HTML, которые позволяют использовать файл в качестве макета страницы.
- Одна ячейка, содержащая предварительно заданный текст с инструкциями, такими как синхронизация темы с рабочим столом, инструкции по добавлению нового макета страницы, инструкции по добавлению компонентов в макет страницы и инструкции по созданию иерархии сайта с помощью нового макета страницы.
- Простой файл навигации JavaScript, в котором приведен пример использования функций и объектов JavaScript API-интерфейса рендерера. Для перемещения по иерархии сайта и создания необходимой разметки HTML для навигации по сайту необходим API-интерфейс средства рендеринга.

См. также [Базовая структура темы](#).

9. Завершив изменение темы, синхронизируйте папки и файлы темы со своим экземпляром Oracle Content Management. Чтобы увидеть, как изменения выглядят и ведут себя на сайте, откройте сайт в Site Builder. Скорее всего, это итеративный процесс.
10. Чтобы просмотреть или изменить сайт, выберите его и нажмите **Открыть** в контекстном меню или нажмите  на панели действий.
11. Переключите режим редактирования на **Edit** (Редактировать), чтобы можно было вносить изменения в сайт. Можно изменить существующие страницы и добавить новые, используя макет страницы, доступный в начальных темах.

Примечание.

Если вы знакомы со структурой макета страницы и ее использованием, можете удалить ненужные разделы, предоставленные начальным шаблоном, и включить макет в разделах, которые требуется использовать для одного из новых макетов.

12. Завершив вносить изменения в сайт, **сохраните** его и нажмите **Опубликовать**, чтобы объединить обновление с базовым сайтом.

13. Выберите сайт и на его основе создайте шаблон. При этом извлекаются активы и тема для нового шаблона.
14. Предоставьте совместный доступ к шаблону участниками, вместе с которым вы хотите использовать шаблон.

Создание шаблона сайта на основе шаблона Bootstrap или шаблона разработки веб-сайтов

Открытая архитектура Oracle Content Management означает возможность использования работы, выполненной в других платформах создания кода, таких как Foundation или Bootstrap. С помощью нескольких изменений можно превратить шаблон Bootstrap в тему и сделать ее частью шаблона Oracle Content Management.

Основной процесс

Ниже приведен обзор шагов, подробно описанных в последующих разделах:

1. [Предварительные условия](#)
2. [Создание сайта](#)
3. [Синхронизация папки темы](#)
4. [Настройка основной темы](#)
5. [Обновление страниц сайта](#)
6. [Обновление навигации](#)
7. [Обновление макетов сайта](#)
8. [Публикация сайта](#)
9. [Создание нового шаблона](#)

Предварительные условия

- Установите и запустите настольное приложение Oracle Content Management для синхронизации папок и файлов на локальном компьютере.
- Выгрузите папки с темами шаблонов, файлы и контент Bootstrap на локальный компьютер и подготовьте их к использованию.

Создание сайта

Создание сайта на основе начального шаблона Oracle Content Management.

1. На боковой панели навигации нажмите **Сайты**.
Отображается список существующих сайтов.
2. Нажмите **Создать**.
3. В диалоговом окне "Создание сайта" выберите шаблон **Начальный шаблон JET** или **Начальный шаблон**, на основе которого будет создан сайт.
4. В данном диалоговом окне введите имя сайта. Это имя используется в URL-адресе сайта. Можно употреблять буквы, цифры, символы подчеркивания (`_`) и дефисы (`-`). Введенный пробел автоматически заменяется дефисом.

Не используйте для шаблонов, тем, компонентов, сайтов и страниц сайтов следующие имена: authsite, content, pages, scstemplate_*, _comps, _components, _compsdelivery, _idcservice, _sitescloud, _sitesclouddelivery, _themes, _themesdelivery. Следующие имена можно использовать только для страниц сайтов (не для шаблонов, тем, компонентов и сайтов): documents, sites.

Примечание.

URL-адрес сайта Oracle Content Management зависит от регистра букв. Регистр букв в запросе или строках фрагментов определяется разработчиками в своем коде.

```
https://host[:port]][/]path[?query][#fragment]
      \_____/      \_____/
      Location      Data
```

5. Дополнительно можно ввести описание сайта.

6. По завершении нажмите кнопку **Создать**.

В индикаторе выполнения отображается имя нового сайта и состояние создания. После создания сайта имя появляется в списке сайтов. Сначала сайт находится в автономном состоянии.

Чтобы быстро найти недавно созданные сайты в списке, отсортируйте его по дате **последнего обновления**. Только что созданный сайт появится наверху списка.

Синхронизация папки темы

При создании сайта на основе начального шаблона создается копия темы начального шаблона с именем сайта, за которым следует имя темы. Например, тема для My_New_Site имеет имя My_New_SiteTheme.

Используйте настольное приложение для синхронизации папки и файлов темы для сайта с локальным компьютером. См. раздел "Начало работы с синхронизацией" в документе *Совместная работа над документами с помощью Oracle Content Management*.

Теперь на локальном настольном ПК должна быть иерархия папок и файлов темы шаблона. Вот пример:

```
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

```

Настройка основной темы

1. Скопируйте эти файлы Bootstrap в синхронизированные папки темы на локальном настольном ПК, чтобы наложить на существующие файлы.
 - Файлы `html` перемещаются в папку `theme_name/layouts`
 - Файлы `css` перемещаются в папку `theme_name/assets/css`
 - Файлы `js` перемещаются в папку `theme_name/assets/js`
 - Файлы `image` перемещаются в папку `theme_name/assets/images`, которая может быть разбита на подпапки с изображениями фона, нижнего колонтитула, людей и т. д.
2. Измените файлы `html` в папке `layout`, чтобы обновить относительные пути и добавить необходимые элементы. Для типичной темы Bootstrap существует много относительных путей к папке `/assets`, поэтому необходимо изменить их так, чтобы они указывали на папку темы.

Исправьте пути для папок `css`, `js` и `images`, чтобы использовать следующие папки:

```

_scs_theme_root_/assets/css/
_scs_theme_root_/assets/js/
_scs_theme_root_/assets/images/

```

Примечание.

После завершения этого шага часть `_scs_theme_root` автоматически подстраивается под окружение, в котором используется тема.

3. К каждому макету Oracle Content Management предъявляются три требования:
 - a. Включите в тег `<head>` файла `html` следующие теги:
 - `<!--$SCS_RENDER_INFO-->`
 - `<!--$SCS_SITE_HEADER-->`
 - `<!--$SCS_PAGE_HEADER-->`

- b. Добавьте сценарий рендерера в конец файлов макета, непосредственно внутри тега `<body>`. Оба этих пути автоматически настраиваются в Site Builder и средах выполнения.

```
<script data-main="/_sitescloud/renderer/renderer.js" src="/_sitescloud/renderer/require.js"></script>
```

- c. Добавьте следующие теги после включения файла `renderer.js`:

- `<!--$SCS_PAGE_FOOTER-->`
- `<!--$SCS_SITE_FOOTER-->`

Убедитесь, что измененные файлы темы синхронизированы с сайтом в Oracle Content Management.

Обновление страниц сайта

При открытии сайта в Site Builder сначала отображаются страницы сайта по умолчанию. Можно удалить ненужные страницы и переключить макет на страницах, которые требуется сохранить, на один из новых макетов.

1. На боковой панели навигации нажмите **Сайты**.

Отображается список существующих сайтов.

2. Выберите сайт и выберите **Открыть** в контекстном меню или нажмите  на панели действий.

3. Введите имя для обновления и необязательное описание, затем нажмите **Создать**.

В имени обновления можно указывать буквы, цифры, символы подчеркивания (`_`) и дефисы (`-`). Введенный пробел автоматически заменяется дефисом.

Если для сайта уже есть обновления, выберите обновление из списка и щелкните .

4. Site Builder открывается в режиме предварительного просмотра. Чтобы внести изменения или воспользоваться средствами навигации на боковой панели, убедитесь, что переключатель "Редактировать"  переведен в положение **Редактировать**.
5. Чтобы отредактировать определенную страницу, выберите ее в дереве сайта на левой боковой панели или с помощью собственной системы навигации сайта.
6. Чтобы удалить ненужную страницу, выберите ее и нажмите .
7. Чтобы добавить новую страницу, нажмите **Добавить страницу**. Положение страницы в дереве сайта можно изменить путем перетаскивания.
8. Чтобы изменить макет, связанный со страницей, выберите страницу на дереве сайта и нажмите  для отображения настроек страницы.

Перейдите к полю **Макет страницы** и в меню выберите другой макет. Количество и тип макетов страницы зависит от темы, связанной с вашим сайтом.

9. Кнопка **Сохранить** позволяет сохранить изменения в текущем обновлении. При необходимости можно продолжить работу в текущем обновлении или создать новые обновления.

Обновление навигации

При просмотре навигации в предварительном просмотре сайта она не совпадает с текущей иерархией, так как эта навигация, жестко заданная в макете, взятом из темы Bootstrap.

Обновите навигацию, чтобы заменить жестко заданный код в теме Bootstrap кодом, динамически генерируемым на основе иерархии сайта.

Измените локальные синхронизированные копии файлов темы.

1. Удалите жестко заданный код навигации из макетов. Вот пример жестко заданной навигации, которую необходимо удалить из типичной темы Bootstrap. Это типичный раздел заголовка с логотипом, частями "Переключить навигацию" для меню "Гамбургер", если страница слишком узкая (адаптивная часть) и используется жестко заданная навигация по странице для других страниц.

```
<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. Напишите код JavaScript для просмотра информации о структуре сайта и создания кода навигации, а затем включите этот код JavaScript в макеты, например:

```

<script type="text/javascript" src="_scs_theme_root_/assets/js/
navbar.js"></script>

```

3. Измените точный вывод файла `topnav.js` в соответствии с разметкой, ожидаемой в вашей конкретной таблице CSS.

Обновление макетов сайта

На данном этапе сайт функционирует, но в макетах отсутствуют редактируемые области (ячейки).

1. Найдите или добавьте элемент DIV в компоновку и назначьте его как ячейку.

Ячейки — это элементы DIV в макете, для которых в атрибуте класса указано значение `"scs-slot"`. Каждая ячейка должна иметь уникальный атрибут `id`. Пример:

```

<div id="slot-content1" class="scs-slot scs-responsive"></div>

```

Чтобы ячейка автоматически подстраивалась под размер окна просмотра браузера, включите атрибут класса `"scs-responsive"`.

2. При необходимости повторите этот шаг, чтобы создать дополнительные ячейки в этом или других макетах.

Публикация сайта

После завершения и сохранения всех изменений на сайте необходимо **опубликовать** сайт.

При публикации сайта все изменения в текущем обновлении объединяются в базу, превращая то, что было в обновлении, в новый базовый сайт.

Создание нового шаблона

Теперь есть рабочий сайт со связанной темой, поэтому полученный результат можно внести в шаблон для предоставления совместного доступа другим пользователям, чтобы они могли создавать сайты на его основе.

1. В боковом меню навигации Oracle Content Management нажмите **Разработчик**.
2. Нажмите **Смотреть все шаблоны**.
3. Нажмите **Создать** и выберите **На основе существующего сайта**.
4. Выберите новый сайт, созданный с помощью начального шаблона и шаблона Bootstrap.
5. Введите имя нового шаблона и нажмите **Создать**.
6. Чтобы упаковать шаблон для использования с другими экземплярами Oracle Content Management, выберите шаблон и параметр меню **Экспорт**, чтобы создать файл .zip, который можно выгрузить.

Разработка шаблонов в Developer Cloud Service

Developer Cloud Service можно использовать в целях разработки шаблонов для Oracle Content Management.

Выполните указанные ниже действия, чтобы разработать шаблон в Developer Cloud Service, протестировать его локально, а затем экспортировать в Oracle Content Management.

1. [Настройка инструментария OCE на локальном компьютере.](#)
2. [Вход в консоль Developer Cloud Service для Oracle Content Management.](#)
3. [Создание шаблона в Developer Cloud Service.](#)
4. [Добавление инструментария OCE в код проекта в новой репозитории GIT.](#)
5. [Создание шаблона в Developer Cloud Service.](#)

Можно создать новый шаблон для разработки, [копирования существующего шаблона в Developer Cloud Service](#) или [импорт шаблона из Oracle Content Management](#).

6. [Тестирование шаблона в локальной тестовой оснастке.](#)
7. [Объединение изменений.](#)
8. [Экспорт шаблона из Developer Cloud Service в Oracle Content Management.](#)

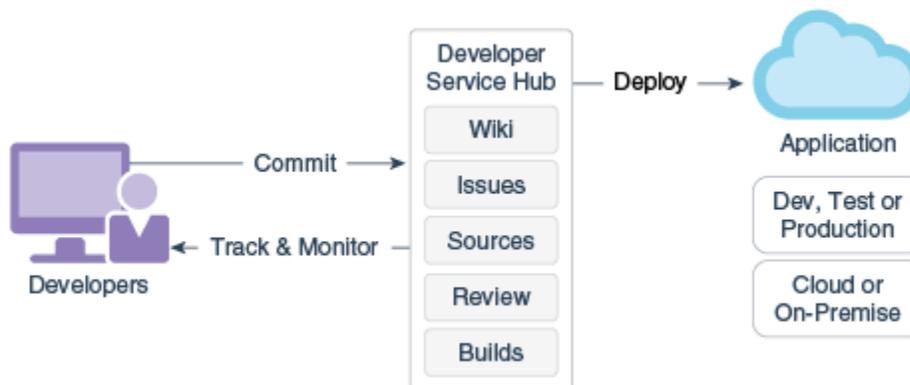
Вход в консоль Developer Cloud Service для Oracle Content Management

Начните разработку пользовательских компонентов для Oracle Content Management на консоли Developer Cloud Service.

Администратор сервисов Oracle Cloud может использовать инструмент My Service Administration для создания облачных сервисов и управления ими. Если вы являетесь

администратором экземпляров сервиса для Oracle Content Management и администратором сервиса для Standard Developer Service, вы можете настроить их и начать использовать.

1. Войдите в Oracle Cloud, используя информацию, предоставленную для вашей учетной записи.
2. Войдите в My Service Administration, чтобы создать экземпляр Oracle Content Management и сервис Standard Developer Service и управлять ими.



3. По запросу подтвердите свой адрес электронной почты Oracle Developer Cloud Service.
4. Настройте свой экземпляр Oracle Content Management, используя сведения о подписке для своего сервиса, и перейдите по URL-адресу Oracle Content Management для вашего экземпляра.
5. Перейдите по URL-адресу для Standard Developer Service.
6. Войдите в свою учетную запись Oracle Developer Cloud Service.

Перейдите по URL-адресу Developer Cloud Service и войдите в консоль.

Создание шаблона в Developer Cloud Service

Можно создать проект в Developer Cloud Service с помощью шаблона проекта "Content Experience Cloud" или проект с пустым репозиторием Git и импортировать инструментарий OCE из своего экземпляра Oracle Content Management.

- [Создайте проект Developer Cloud Service с помощью шаблона Oracle Content Management](#)
- [Создание проекта в Developer Cloud Service с помощью инструментария OCE из Oracle Content Management](#)
- [Добавление инструментария OCE в код проекта в новом репозитории GIT](#)

Создание шаблонов в Developer Cloud Service

Утилита командной строки `scs` позволяет создавать шаблоны Oracle Content Management на основе доступных исходных шаблонов.

Команда `cec create-template` используется для создания шаблона на основе одного из доступных исходных шаблонов. При вводе команды `cec create-template -h` в командной строке отображаются доступные исходные шаблоны.

Вот пример создания шаблона:

```
cec create-template CafeSupremoLite_yourname -f CafeSupremoLite
```

Windows: при создании шаблонов эта команда создает символьную ссылку для тем, которые должны визуализироваться во внешнем редакторе HTML с режимом WYSIWYG (например, `_scs_theme_root_`). Чтобы создать символьные ссылки в Windows, обычно необходимо выполнить эту утилиту командной строки с правами администратора. Если для редактирования темы не используется редактор с режимом WYSIWYG, не требуется выполнять утилиту с правами администратора, и можно проигнорировать ошибку создания символьной ссылки.

В предыдущем примере создается шаблон `CafeSupremoLite_yournameTheme`, а исходный код становится доступным в каталоге `cec-components/src/main/`. В таблице ниже показано расположение исходного кода после создания шаблона сайта в Developer Cloud Service.

Исходный код шаблона	Тема	Компоненты шаблона
<code>cec-components/src/main/templates</code>	<code>cec-components/src/main/themes</code>	<code>cec-components/src/main/components</code>

Локальная тестовая оснастка также показывает компоненты с возможностью их фильтрации по шаблону и типу.

Файлы тем и компонентов можно редактировать в любом текстовом редакторе или редакторе кода. См. [Тестирование с помощью локальной тестовой оснастки](#). После редактирования темы или компонента обновите контент окна браузера, чтобы увидеть изменения.

! Важно.

Исходный код шаблонов, тем и компонентов находится в папке `src/main/`. Не следует изменять файлы за пределами каталога `src/main`, так как они необходимы для работы локального сервера Oracle Content Management.

Копирование шаблона в Developer Cloud Service

Существующий шаблон Oracle Content Management можно скопировать в Developer Cloud Service.

Чтобы скопировать один из существующих шаблонов из `src/main/templates`, используйте команду `cec copy-template`. Если в шаблоне содержатся активы из других репозиторий, можно дополнительно указать сопоставление репозитория, в противном случае эти активы не будут копироваться.

В следующем примере шаблон `Temp1` копируется в новый шаблон с именем `Temp2`:

```
cec copy-template Temp1 -n Temp2
```

Импорт шаблона в Developer Cloud Service

Шаблоны можно импортировать из Oracle Content Management в Developer Cloud Service для дальнейшей разработки.

Если есть ZIP-файл шаблона с сервера Oracle Content Management, его можно импортировать в Developer Cloud Service для дальнейшей разработки, например для редактирования темы или компонентов. Используйте следующую команду:

```
cec import-template <location of the template zip file>
```

Укажите папку с zip-файлом в Oracle Content Management.

См. [О шаблонах](#) и [Экспорт шаблона](#).

Объединение изменений

После создания компонента, шаблона или макета контента или редактирования исходного кода на локальном компьютере необходимо объединить новые и измененные компоненты и шаблоны в хранилище Git проекта.

Чтобы объединить изменения в хранилище Git, в окне терминала введите по порядку следующие команды:

```
cd cec-components git pull
git add .
git status
git commit -a -m "Your comments" git pull
git push
```

Экспорт шаблона из Developer Cloud Service

ZIP-файл шаблона можно экспортировать из Developer Cloud Service и использовать для создания сайта в Oracle Content Management.

После завершения разработки шаблона его можно экспортировать, выполнив указанную ниже команду. В ответе команды указывается, где в Oracle Content Management создан ZIP-файл шаблона.

```
cec export-template CafeSupremoLite_yourname
```

См. [О шаблонах](#) и [Импорт шаблона](#).

21

Разработка тем

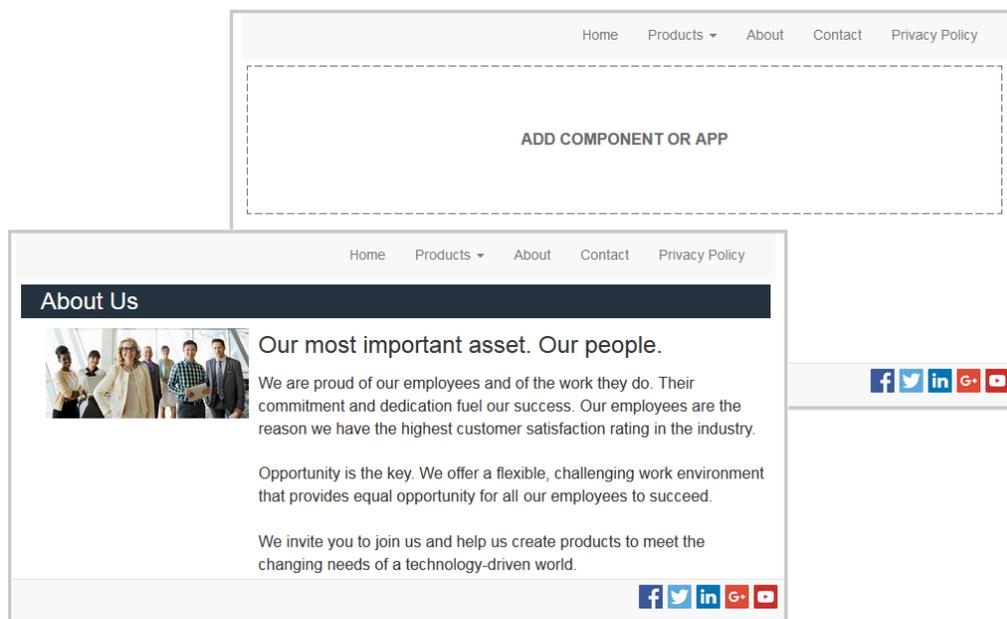
Тема определяет общий внешний вид (общий стиль) сайта, в том числе цветовую гамму, размеры и типы шрифтов, а также фон страниц. Темы обеспечивают визуальную согласованность страниц сайта. Можно создавать уникальные темы и вариации тем, указывая дизайн и образцы контента, которые затем можно использовать для создания сайтов, чтобы продвигать ваш бренд и ваше видение.

- [О темах](#)
- [Базовая структура темы](#)
- [Навигация на сайте](#)
- [Создание темы](#)
- [Связывание компонентов с темами](#)
- [API-интерфейс рендеринга сайтов](#)

О темах

Темы определяют общий вид сайта, включая его контент, внешний вид и поведение. Темы обеспечивают визуальную согласованность страниц сайта.

Разработка новой темы подразумевает указание макета, стиля, образца контента, навигации и всей основной информации, которая служит отправной точкой для нового сайта. Разработчики тем определяют внешний вид и поведение сайта. Тема должна разрабатываться с учетом способа ее использования; например, если ожидается, что большинство или все пользователи будут заходить на сайт с мобильных устройств. Проектирование пользовательских тем полезно, если есть пользователи, которые хотят создать много похожих сайтов. Тему можно создать на основе макетов страниц для общих шаблонов, которые можно использовать совместно в разных темах.



Тема содержит макеты страниц, которые используются для проектирования контента, внешнего вида и поведения сайтов. Измените дизайн и настройки и добавьте контент, чтобы создать сайт, представляющий ваши стиль, бренд и ваше видение.

Тема включает в себя следующие элементы:

- Активы для фоновых изображений или другого контента, входят в состав макетов страниц (изображения, файлы JavaScript и т. д.)
- Настройки стиля сайта (CSS)
- Несколько макетов страниц (HTML-файлы)
- Код для создания системы навигации по сайту (файлы JavaScript)
- Список основных стилей, которые можно использовать с компонентами (которые указаны в файлах `design.css` и `design.json`)

Тема также может включать *исходные данные*, которые используются для заполнения новой страницы, созданной на основе одного из шаблонов страниц. Например, пользователь создает новую страницу для раздела "Продукты" и выбирает макет страницы с названием `new_product.html`. Если тема содержит файл с именем `new_product-pageseed.json`, при первом создании новая страница заполняется содержимым исходного файла страницы. Как и в случае с образцом контента эти исходные данные могут быть изменены. Они предназначены только для обеспечения отправной точки создания страницы.

Можно создать тему, которая использует подмножество компонентов, предназначенных для работы с этой темой. Когда пользователь выбирает такую тему для своего сайта, он видит только компоненты, указанные для этой темы. См. [Связывание компонентов с темами](#).

Каждый сайт использует тему. При создании сайта на основе шаблона наследуется тема, связанная с шаблоном. Тему сайта можно изменить в любое время. Oracle Content Management содержит несколько шаблонов с темами, которые можно использовать для начала работы.

Если сайт использует новую неопубликованную тему, она автоматически публикуется вместе с сайтом при его первом переводе в онлайн-режим. Если вы внесли в тему изменения и хотите обновить онлайн-сайты, чтобы отобразить изменения, измененную тему необходимо опубликовать явным образом. Опубликовать тему могут только владельцы темы и пользователи с правами "Управление".

 **Примечание.**

Если опубликовать изменения в теме, оно отразится на всех использующих эту тему сайтах, которые находятся в интерактивном режиме. Например, если изменить указанный в теме шрифт по умолчанию и опубликовать тему, на всех использующих ее сайтах будет использоваться новый шрифт по умолчанию.

См. также [Управление темами](#).

Базовая структура темы

Базовая структура темы включает дизайн, навигацию и стили, указанные в папках, хранящихся в Oracle Content Management. Тема входит в состав шаблона сайта.

Когда пользователь выбирает шаблон для создания нового сайта, данные связанной темы загружаются автоматически. Если используется начальный шаблон JET или обычный начальный шаблон, тема автоматически копируется, а не создается ссылка на нее. Если разработчик использует начальный шаблон, ему требуется собственная копия темы.

Тема имеет определенную структуру папок и файлов, как иллюстрируется в данном примере, показывающем основные папки и файлы:

```
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
```

Некоторые папки содержат информацию определенного типа, включая следующие папки:

- `assets`: JavaScript, каскадная таблица стилей (CSS), изображения и другие файлы поддержки, на которые ссылаются макеты.

- `designs`: файлы `design.css` и `design.json`, которые используются для указания параметров стиля компонентов.
- `layouts`: HTML-файлы для шаблонов страниц, которые используются для отображения страниц сайта.

 **Примечание.**

В начале всех HTML-файлов должен быть элемент DOCTYPE, который выглядит следующим образом: `<!DOCTYPE html>`

- `publish`: после использования темы на опубликованном сайте перечисляется этот каталог с копиями файлов. Этот каталог отображается, если тема синхронизирована с помощью настольного приложения, однако он не отображается, если тема экспортирована как часть шаблона.
- `responsepages`: специальная страница для обработки ошибок (404).

 **Примечание.**

Если страница сайта помечена как страница ошибки, сообщение об ошибке 404 от темы игнорируется, а вместо него используется страница ошибки.

- `viewport.json`: задает настройки окна просмотра для темы.
- `_folder.json`: указывает имя и GUID темы. Пример:

```
{
  "themeName": "MarketingCampaignTheme",
  "itemGUID": "TB79D65F699B022AC4E11F4D4EE870070A1ADD86BBBB"
}
```

GUID создается средой Oracle Content Management при первом импорте или копировании темы. Имя темы назначается разработчиком темы при ее создании.

- `components.json`: (устарело) Запись пользовательских компонентов, используемых в теме. Поддерживается только для обратной совместимости.

Есть два ключевых файла, с которыми вы будете работать при создании новой темы. Эти файлы задают стили компонентов:

- `design.css`
- `design.json`

Рекомендуется поместить информацию о перемещении в один файл JavaScript, например в файл `nav.js`. Папка `/assets/js/` темы — хорошее место для такого файла.

Навигация на сайте

Определение иерархии сайта хранится в файле `structure.json`, связанном с сайтом. Иерархия загружается в память и становится доступной в контексте страницы как объект `SCS.structureMap`.

Site Builder считывает файл `structure.json`, чтобы нарисовать дерево сайтов в Site Builder. Файл `structure.json` содержит код страниц сайта. Пример:

```
"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": "",
  "linkTarget": "",
  "children": [],
  "overrideUrl": false
}
```

Страницы сайта должны содержать код JavaScript навигации, чтобы считывать эту структуру и составлять навигационные ссылки для сайта. Шаблоны, поставляемые с Oracle Content Management, включают примеры файлов JavaScript навигации, иллюстрирующие этот принцип работы.

Файл `topnav.js`, который используется в некоторых темах, предоставляемых Oracle Content Management, служит примером использования объекта `SCS.structureMap` вместе с вызовами API для рендеринга, такими как `SCSRenderAPI.getPageLinkData`, для перемещения по структуре сайта и составления HTML-разметки, необходимой для отображения меню навигации на странице. Ниже приведен код из примера файла `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();
            }
        }
    );
}
```

Вызовы API рендеринга можно использовать для создания навигационных ссылок, которые действуют, когда сайт находится в режимах редактирования и предварительного просмотра сайта, и на опубликованном сайте. См. [Обращение к API рендеринга](#).

Рекомендуется поместить информацию о перемещении в один файл JavaScript, например `topnav.js`. Такой файл JavaScript обычно хранится в папке `/assets/js/` темы, как показано в примерах тем, предоставляемых Oracle Content Management.

Создание темы

Новую тему можно создать, скопировав и изменив существующую тему. В новой теме также можно импортировать и использовать контент Bootstrap.

Примечание.

И при создании новой темы, и при обновлении существующей темы, всегда следует сделать копию темы и работать с копией. Протестируйте ее на примере сайта или скопируйте свой реальный сайт, чтобы убедиться, что он работает правильно. Обратите внимание, что изменения, внесенные в существующую тему, будут применены на всех сайтах, использующих эту тему, сразу после публикации измененной темы.

Копирование темы

1. На домашней странице нажмите **Разработчик**.
Отображается страница **Разработчик**.
2. Нажмите **Просмотреть все темы**.
Отображается список существующих тем. Можно управлять отображением тем, нажав на значок представления и выбрав нужный параметр в списке.
3. Выберите тему и выберите **Копировать** в контекстном меню или нажмите  на панели действий.
Копируются все папки и файлы темы, в том числе образцы страниц и контента.

Примечание.

При копировании темы ее совместно используемая информация не копируется.

4. Введите имя скопированной темы. Невозможно использовать имя другой темы.
В имени можно использовать буквы, цифры, символы подчеркивания (`_`) и дефисы (`-`). Введенный пробел автоматически заменяется дефисом.
Не используйте для шаблонов, тем, компонентов, сайтов и страниц сайтов следующие имена: `authsite`, `content`, `pages`, `scstemplate_*`, `_comps`, `_components`, `_compsdelivery`, `_idcservice`, `_sitescloud`, `_sitesclouddelivery`, `_themes`, `_themesdelivery`. Следующие имена можно использовать только для страниц сайтов (не для шаблонов, тем, компонентов и сайтов): `documents`, `sites`.
5. Дополнительно можно ввести описание темы.
6. Нажмите **Копировать**.
В индикаторе выполнения отображается имя новой темы и статус копирования. После копирования темы ее имя появляется в списке тем. Папки и файлы, составляющие тему, можно просмотреть, щелкнув имя темы в списке тем.
7. Используйте настольное приложение Oracle Content Management, чтобы синхронизировать папки и файлы темы с локальной системой. Это позволяет просматривать локальные папки и работать непосредственно с файлами. Изменения, внесенные в тему, синхронизируются автоматически. Изменения можно внести с помощью предпочтительных инструментов редактирования HTML, кода или текста.

Указание возможности использования пользовательских стилей на сайте

Администратор тем может указать стили, которые поставляются с темой, или настроить стили.

Настройка в файле `components.json` темы указывает возможность использования пользовательских стилей в средстве Site Builder. Это необходимо для контроля

того, что соавтор сайта сохраняет в стиле сайта при его создании и не использует, например другие шрифты и цвета.

Чтобы запретить использование пользовательских стилей на сайте, выполните указанные ниже действия.

1. Скройте параметр **Настроить** на панели **Настройки** для компонентов. В файле `components.json` для темы вместе с другими определениями компонентов добавьте следующий объект:

```
[
  {
    "showStyleClassOnly": "true"
  }
]
```

2. Настройте группы и кнопки панелей инструментов в SKEditor для удаления пользовательского стиля: В файле `components.json` для темы можно указать свойства `toolbarGroups` и `removeButtons` для SKEditor, чтобы настроить группы и кнопки в группах для компонентов `Title` и `Paragraph`.

Например, добавьте следующие объекты вместе с другими определениями компонентов, чтобы удалить стили, шрифт и цвета из группы `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. Запретите вставку стилей в SKEditor.

Вставка осуществляется в соответствии с настроенными стилями и кнопками. В предыдущем примере из вставленных данных удаляется свойство `Font`.

Запрет использования пользовательских стилей проявляется следующим образом:

- Во всех встроенных компонентах на вкладке **Стиль** скрыт пункт "Настроить".
- В SKEditor кнопки панели инструментов для настройки стилей скрыты, а спецификация семейства шрифтов, цвет шрифта и т. д. переопределены.
- Кроме того, редактор SKEditor не должен позволять задавать стиль со вставленным текстом. Допускается использование полужирного шрифта или курсива.

Скрытие компонентов и макетов разделов для темы

Изменив файл `component.json` темы, можно скрыть компоненты и макеты разделов, чтобы предотвратить их использование разработчиком сайта. Возможно, вы захотите сделать это, чтобы обеспечить общий вид. См. [Скрытие компонентов и макетов разделов для темы](#).

Проверка темы с помощью сайта

После завершения редактирования темы необходимо убедиться, что все элементы работают с сайтом согласно плану.

1. Убедитесь, что скопированные папки и файлы синхронизированы с настольным приложением Oracle Content Management, а все изменения сохранены.
2. Откройте сайт (это может быть тестовый или существующий автономный сайт) в Site Builder и переключите сайт на использование новой темы.
3. Протестируйте сайт с этой темой, добавляя страницы и используя настройки, просматривая изображения, проверяя навигацию и любые другие элементы, которые были изменены в новой теме.
4. Убедитесь, что все изображения отображаются правильно, включая все файлы и ссылки.
5. Опубликуйте тему, чтобы сайт реализовал изменения темы, затем зарегистрируйте ее в среде выполнения.

См. [Публикация тем](#).

Использование темы Bootstrap

Сходства между темами Oracle Content Management и темами Bootstrap позволяют преобразовать существующие страницы и контент с темой Bootstrap для использования в теме Oracle Content Management.

Bootstrap — это бесплатный набор инструментов с открытым исходным кодом для создания веб-сайтов и веб-приложений. Он содержит шаблоны проектирования на основе HTML и CSS для компонентов интерфейса и расширений JavaScript.

Выполняются те же инструкции по созданию темы, но приходится работать с кодом (с помощью любого выбранного редактора), чтобы импортировать и редактировать любые нужные страницы или контент из темы Bootstrap.

Маловероятно применение всех возможностей темы Bootstrap ко всем страницам и контенту, где она используется. Как правило, для использования в теме Oracle Content Management выбираются некоторые части темы, возможно, только некоторые шаблоны страниц или лишь некоторые их части. Например, тема Bootstrap содержит несколько блоков внутри шаблонов, включая верхний колонтитул, блок навигации, блок тела и блок нижнего колонтитула. Так как темы Bootstrap поставляются как набор файлов, которые можно просматривать непосредственно в файловой системе (без веб-сервера), они содержат много дублирующихся фрагментов на каждой странице (это обусловлено отсутствием механизма сборки страниц). При использовании этой информации в теме Oracle Content Management необходимо взять эти части и добавить в многократно используемые шаблоны страниц Oracle Content Management, а затем использовать Oracle Content Management для динамической сборки нескольких страниц на основе этих шаблонов.

Один из примеров добавления информации в многократно используемые шаблоны — раздел навигации. Система навигации в типичной теме Bootstrap дублируется на всех страницах, но при перемещении в тему Oracle Content Management необходимо использовать код JavaScript, который динамически перемещается по иерархии сайта и создает структуры навигации. Затем этот сценарий просто включается во все страницы и все они получают систему навигации, а эта система навигация адаптируется при добавлении страниц на сайт или удалении из него.

См. [Создание шаблона сайта на основе шаблона Bootstrap или шаблона разработки веб-сайтов](#).

Скрытие компонентов и макетов разделов для темы

По умолчанию все компоненты и схемы разделов доступны лицам, разрабатывающим сайт. Иногда требуется скрыть компонент или макет раздела, чтобы он был недоступен при использовании определенной темы. Например, если требуется обеспечить единообразие внешнего вида, при котором не используются компоненты кнопок или макеты разделов со слайдами, их можно скрыть в теме, изменив файл `components.json` темы.

Чтобы скрыть компоненты и макеты разделов в теме, выполните указанные ниже действия.

1. В Oracle Content Management нажмите **Разработчик** в левом меню навигации.

2. На странице "Разработчик" нажмите **Просмотреть все темы**.
3. На странице "Темы" выберите тему для изменения и нажмите **Открыть** в контекстном меню или нажмите  на панели действий.
4. Выберите файл components.json и нажмите **Выгрузить**.
5. Откройте файл components.json в текстовом редакторе.
6. Добавьте объекты, чтобы указать тип и идентификатор контента, и задайте свойству hidden значение true.
Например, ниже приведен пример записи, которая скрывает компонент кнопки:

```
"type": "scs-button",  
"id": "scs-button",  
"hidden": true
```

Аналогично, ниже приведен пример записи, которая скрывает макет раздела со слайдами:

```
"type": "scs-sectionlayout",  
"id": "scs-sl-slider",  
"hidden": true
```

Полная запись в файле components.json, которая скрывает компонент кнопки и макет раздела со слайдами, выглядит следующим образом:

```
[  
  {  
    "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. После внесения необходимых изменений загрузите файл `components.json` в тему как новую версию.

Скрытие параметров компонента выравнивания, ширины или интервала для темы

По умолчанию большинство компонентов позволяет создавать сайты, указывая параметры выравнивания, ширины и интервала в диалоговом окне настроек компонента. Тем не менее, разработчик может захотеть скрыть эти параметры в пользовательском компоненте, чтобы соавторы не могли настроить рендеринг контента не согласованным с темой сайта способом. Например, чтобы обеспечить единообразие внешнего вида, при котором кнопки располагаются по центру с интервалом 30 пикселей, можно скрыть соответствующие параметры, чтобы их можно было изменить в настройках, редактируя файл `components.json` темы.

Иногда в пользовательском компоненте может потребоваться переопределить исходные значения по умолчанию. Это также можно сделать, редактируя файл `components.json`. Любые изменения исходных значений будут применены только к компонентам, добавленным на страницу после загрузки отредактированного файла `components.json` в качестве новой версии.

Чтобы для компонента скрыть параметры выравнивания, ширины или интервала в теме, выполните указанные ниже действия.

1. В Oracle Content Management нажмите **Разработчик** в левом меню навигации.
2. На странице "Разработчик" нажмите **Просмотреть все темы**.
3. На странице "Темы" выберите тему для изменения и нажмите **Открыть** в контекстном меню или нажмите  на панели действий.
4. Выберите файл `components.json` и нажмите **Выгрузить**.
5. Откройте файл `components.json` в текстовом редакторе.
6. Добавьте объекты, чтобы указать тип и идентификатор компонента, и задайте свойства, которые требуется скрыть, значение `true`. Например, следующая запись скрывает свойства выравнивания, ширины и интервала компонента "Кнопка" в диалоговом окне настроек:

```

"type": "scs-button",
"id": "scs-button",
"hideAlignmentAndWidth": true,
"hideSpacing": true,

```

Если требуется переопределить свойства выравнивания и интервала по умолчанию для компонента "Кнопка", укажите начальные значения, как показано в примере ниже:

```
"initialData": {
  "alignment": "center",
  "marginTop": 30,
  "marginBottom": 30
```

Полная запись в файле components.json с указанием значений по умолчанию и скрываем параметров выравнивания, ширины и интервала в диалоговом окне настроек компонента "Кнопка", используемого в теме, выглядит следующим образом:

```
[
  {
    "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
      }
    ]
  }
]
```

- После внесения необходимых изменений загрузите файл components.json в тему как новую версию.

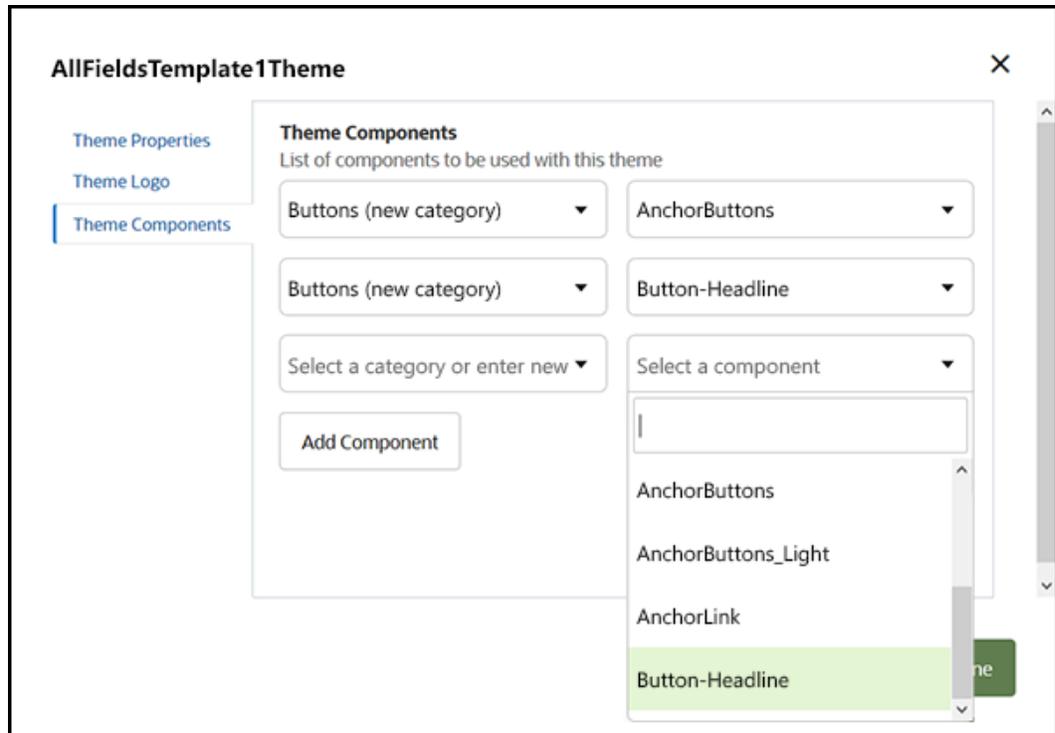
Связывание компонентов с темами

Компоненты, макеты разделов и группы компонентов можно связать с темой, чтобы использовать вместе с ней определенную подгруппу компонентов.

Как разработчик вы можете создать тему и компоненты для шаблона, который маркетологи будут использовать для создания сайтов, рекламирующих продукты вашей организации. Связывая компонент с темой, он делается доступным, когда пользователь выбирает **Компоненты темы** в Site Builder. На вкладке **Компоненты темы** пользователь увидит только компоненты, связанные с данной темой. Если компонент связан с темой, что этот компонент экспортируется вместе с шаблоном сайта, даже если он не используется на сайте.

Чтобы связать компонент с темой, выполните указанные ниже действия.

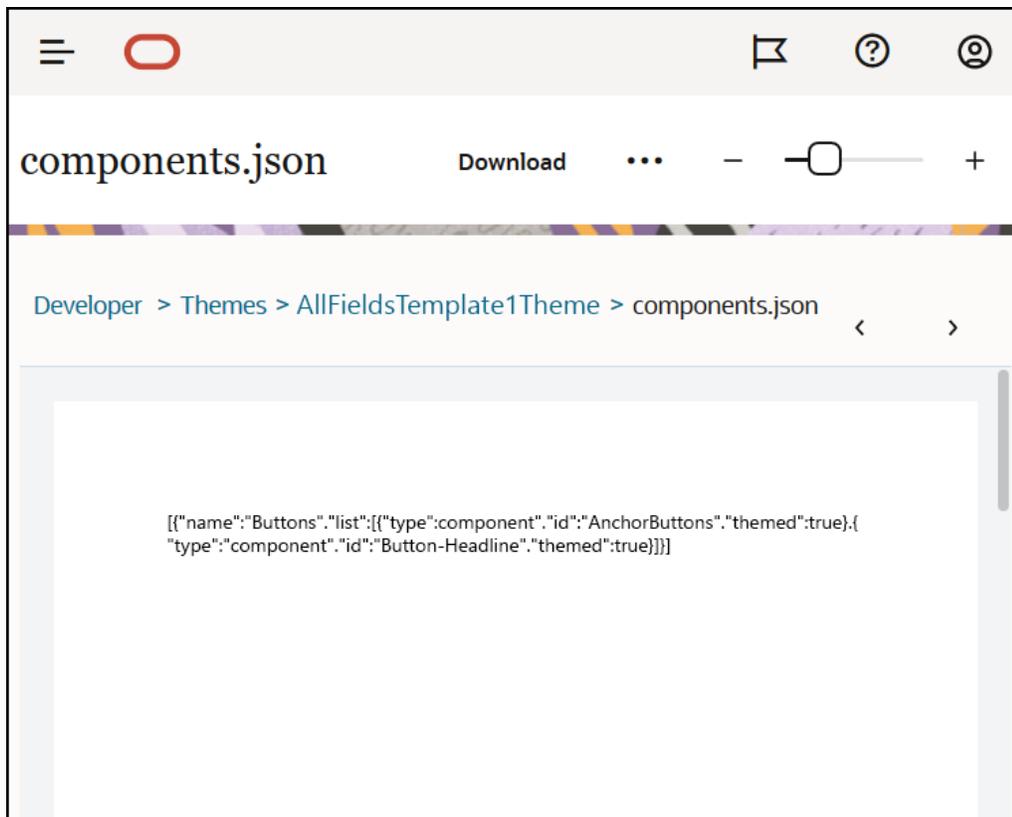
1. В Oracle Content Management нажмите **Разработчик** в левом меню навигации.
2. На странице **Разработчик** нажмите **Просмотреть все темы**.
3. На странице "Темы" выберите тему и нажмите **Свойства** в контекстном меню или на панели действий.
4. Нажмите **Компоненты темы**, чтобы открыть вкладку **Компоненты темы**.
5. Выберите один или несколько компонентов, которые необходимо связать с темой, и распределите компоненты по категориям:
 - a. В раскрывающемся списке **Выберите компонент** выберите пользовательский компонент.
 - b. Для первого выбранного компонента введите имя категории, чтобы создать категорию компонента.
 - c. Для каждого выбранного дополнительного компонента выберите существующую категорию или создайте новую.
 - d. Нажмите **Добавить компонент**.



6. Нажмите **Сохранить**, чтобы связать выбранные компоненты с темой.
7. На странице **Темы** выберите тему и нажмите "Свойства", чтобы снова открыть вкладку **Компоненты темы**.
8. Убедитесь, что выбранные компоненты сохранены.

Чтобы удалить связанный компонент из темы, нажмите **X** рядом с именем компонента на вкладке **Компоненты темы**, а затем нажмите **Да** в диалоговом окне **Удаление компонента**.

На странице **Сведения** шаблона содержится список компонентов, связанных с темой, которую использует данный шаблон. Ссылки на компоненты, связанные с темой, хранятся в файле `components.json` темы.



После настройки связи с темой в Oracle Content Management можно развернуть шаблон и поделиться им с отделом маркетинга. Когда маркетологи выбирают такую тему для своего сайта, они видят только те компоненты, которые были указаны для темы на вкладке **Компоненты тем** в Site Builder. Списки компонентов **Пользовательские** и **Все** также включают компоненты, связанные с темами.

Свяжите компонент с темой в инструментарии OCE.

Чтобы связать компонент с темой в инструментарии OCE, можно использовать приведенную ниже команду `sec`. Компонент отображается на вкладке **Компоненты темы** в Site Builder, а также в списках компонентов **Пользовательские** и **Все**.

```
sec add-component-to-theme <component>
```

Следующая команда `sec` удаляет связь компонента с темой.

```
sec remove-component-from-theme <component>
```

Информацию об инструментарии OCE см. в разделе **Разработка с помощью инструментария OCE**.

API-интерфейс рендеринга сайтов

[API-интерфейс рендеринга сайтов](#) для Oracle Content Management (SCSRenderAPI) — это глобальный для окна объект, присутствующий на всех веб-страницах Oracle Content Management. В первую очередь он отвечает за рендеринг ячеек и компонентов страницы. Также этот объект предоставляет интерфейс для кода JavaScript, присутствующего в макетах темы. Если вы работаете с темами или компонентами как разработчик, вы можете найти эти события и функции. SCSRenderAPI **во время выполнения** обеспечивает рендеринг в режимах обычного и предварительного просмотра, а SCSRenderAPI **во время проектирования** обеспечивает рендеринг в режимах навигации, редактирования и примечаний.

Во время выполнения [API-интерфейс рендеринга сайтов](#) имеет три основные цели:

- заполнить ячейки на странице компонентами и контентом;
- удовлетворить информационные запросы, отправленные кодом JavaScript, который находится в макете страницы;
- вызвать события в течение жизненного цикла процесса рендеринга страницы.

Во время проектирования [API-интерфейс рендеринга сайтов](#) имеет четыре основные цели:

- заполнить ячейки на странице компонентами и контентом;
- удовлетворить информационные запросы, отправленные кодом JavaScript, который находится в макете страницы;
- вызвать события в течение жизненного цикла процесса рендеринга страницы и редактирования;
- взаимодействовать с приложением Site Builder, чтобы разрешить редактирование страниц и создание аннотаций.

[API-интерфейс рендеринга сайтов](#) загружается в веб-страницах Oracle Content Management, помещая в макеты тем следующий тег сценария:

```
<script data-main="/_sitescloud/renderer/renderer.js" src="/_sitescloud/renderer/require.js"></script>
```

Обычно он располагается в нижней части макетов. (Обратите внимание, что URL-адреса в этом теге автоматически настраиваются для соответствующей среды: время проектирования или время выполнения.) Объект SCSRenderAPI загружается асинхронно; пользовательский код JavaScript может прослушивать доступность API рендеринга сайтов, обрабатывая событие `srenderstart`.

Разработка макетов

Макет определяет способ организации контента на странице и используется для создания HTML-кода страниц, используемых на веб-сайтах Oracle Content Management.

- [О макетах](#)
- [Поисковая оптимизация \(SEO\)](#)
- [Общие сведения о файле components.json и формате](#)
- [Настройка групп панелей инструментов в Site Builder](#)
- [Ограничение компонентов в ячейках](#)
- [Как сделать контент макета редактируемым](#)
- [Создание макета раздела](#)
- [Создание макета раздела с поддержкой медленной загрузки \(Lazy Load\)](#)
- [Разработка пользовательских макетов разделов с помощью API](#)
- [Разработка макетов контента](#)

О макетах

Макет определяет организацию контента на странице сайта. Разные макеты могут содержать разное количество именованных ячеек. Ячейка – это область, которая занимает всю ширину страницы и может вмещать содержимое одного или нескольких типов.

В каждой теме есть несколько макетов страниц. См. [О темах](#).

При добавлении страницы на сайт требуется выбрать ее макет. В каждом макете есть области на странице (известные как ячейки), в которые можно перетаскивать контент. Перемещаемое в ячейки содержимое полностью зависит от вас. Это может быть все что угодно - от заголовков, текста и разделителей до мультимедиа, галерей и социальных сетей.

Макет содержит допустимые HTML-конструкции, а также специальную разметку, понятную рендереру Oracle Content Management. Макет должен начинаться с инструкции DOCTYPE, чтобы настроить браузер на отображение страницы в режиме, соответствующем стандартам. Например: `<!DOCTYPE html>`. Эта инструкция требуется для определенных компонентов для достижения наилучших результатов.

В этом примере кода показан минимальный макет:

```
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         <!--$SCS_SITE_HEADER-->  
13         <!--$SCS_PAGE_HEADER-->  
14     </head>  
15     <body>  
16         <div id="topNavigation"></div>  
17         <div id="mainContentSlot" class="scs-slot scs-responsive"></  
div>  
18  
19         <script data-main="/_sitescloud/renderer/renderer.js"  
20             src="/_sitescloud/renderer/require.js"></script>  
21         <!--$SCS_SITE_FOOTER-->  
22     </body>  
23 </html>
```

При отображении страницы в браузере раскрываются различные маркеры.

- [!--\$SCS_THEME_NAME--]
Этот маркер разворачивается в имя темы, выбранной в данный момент для сайта. Использование этого маркера позволяет копировать тему, поскольку URL-адреса, использующие этот маркер, будут ссылаться на текущую тему.
- [!--\$SCS_DESIGN_NAME--]
Этот маркер разворачивается в имя дизайна, выбранного в данный момент для сайта. Это позволяет использовать макет для нескольких дизайнов в рамках темы.
- [!--\$SCS_RENDER_INFO--]
Этот маркер разворачивается в тег сценария, содержащий сведения об иерархии страниц и рендеринге компонентов для страницы. Его следует поместить в раздел <head> макета.
- [!--\$SCS_SITE_HEADER--]
Этот маркер разворачивается в значение верхнего колонтитула сайта, указанного в поле "Верхний колонтитул" в свойствах поисковой оптимизации (SEO). Здесь можно ввести общую для всего сайта разметку, которую требуется разместить на всех страницах. См. [Настройка свойств поисковых систем](#).
- [!--\$SCS_PAGE_HEADER--]
При этом расширяется значение верхнего колонтитула страницы, которое находится в поле "Заголовок страницы" в свойствах "Настройки страницы" в Site Builder. Здесь можно ввести специфическую для страницы разметку, которую требуется разместить для этой страницы.
- [!--\$SCS_SITE_FOOTER--]
Этот маркер разворачивается в значение нижнего колонтитула сайта, указанного в поле "Нижний колонтитул" в свойствах поисковой оптимизации

(SEO). Здесь можно ввести общую для всего сайта разметку, которую требуется разместить на всех страницах. См. [Настройка свойств поисковых систем](#).

Примечание.

В маркерах также могут использоваться префикс `<!--$` и суффикс `-->` в качестве разделителей вместо `[!--$` и `--]`.

Когда сайт находится в оперативном режиме, приведенные ниже маркеры в ссылке заменяются действительными значениями, которые учитывают контекст, в котором они используются. Это позволяет использовать такую ссылку и при редактировании сайта, и при его публикации в сети.

- `/_sitescloud/` заменяется на `/_sitesclouddelivery/`
- `/_themes/` заменяется на `/_themesdelivery/`

Ячейки — это элементы DIV в макете, для которых в атрибуте класса указано значение "scs-slot". Ячейка — это место, куда пользователи добавляют компоненты для заполнения сайта контентом. Несколько элементов DIV можно назначить в качестве ячеек, назначив атрибут класса "scs-slot". Каждая ячейка должна иметь уникальный атрибут id.

Примечание.

Ячейки не могут быть вложенными, но можно предложить, что их можно накладывать с помощью CSS. Для этого используйте группу компонентов или макет раздела. См. [Создание макета раздела](#).

Ячейки, для которых также задано значение атрибута класса "scs-responsive", обновляют свое содержимое по мере изменения разрешения окна просмотра браузера. Это позволяет ячейкам должным образом отображаться на больших настольных дисплеях или небольших мобильных устройствах.

Последний тег `<script>` (строка 19 в данном примере) загружает код рендеринга Oracle Content Management. Этот код отвечает за рисование компонентов на странице, а также предоставляет пользовательскому коду получить доступа к API для рендеринга. Без тега `<script>` страницы, основанные на данном макете, не могут быть изменены приложением Site Builder.

Поисковая оптимизация (SEO)

В Site Builder можно указать ключевые слова, чтобы помочь поисковым системам идентифицировать контент сайта.

Настройки SEO

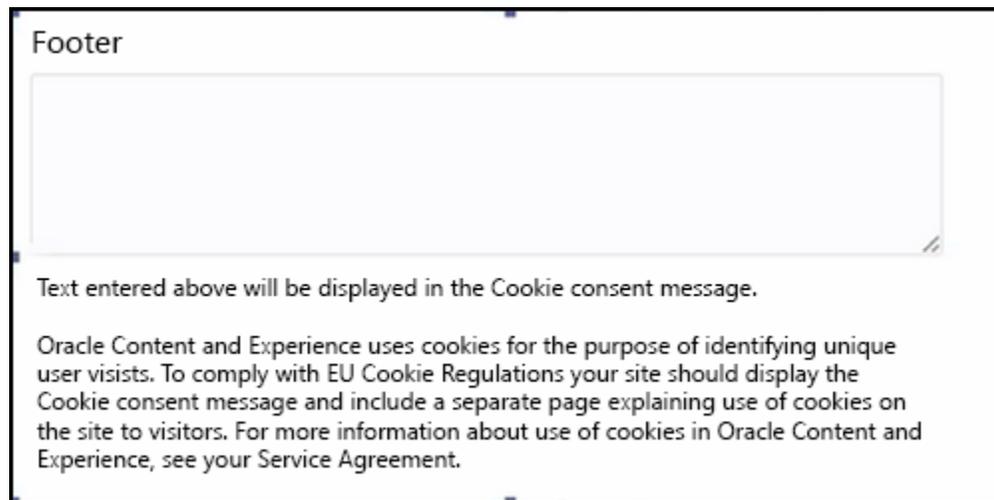
Настройки поисковой оптимизации (SEO) можно определить на уровне сайтов и на уровне страниц. Текст SEO будет включен во все готовые шаблоны, в нижнем колонтитуле.

См. [Настройка свойств поисковых систем](#).

Файлы cookie для посетителей сайта

Посетители сайта используют файлы cookie для SEO, по одному файлу cookie для каждого браузера и каждого сайта, для выставления счетов. Каждый файл cookie необходимо обновлять каждый час или через 24 часа.

Каждый сайт должен содержать всплывающее окно, которое уведомляет посетителей о файлах 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.

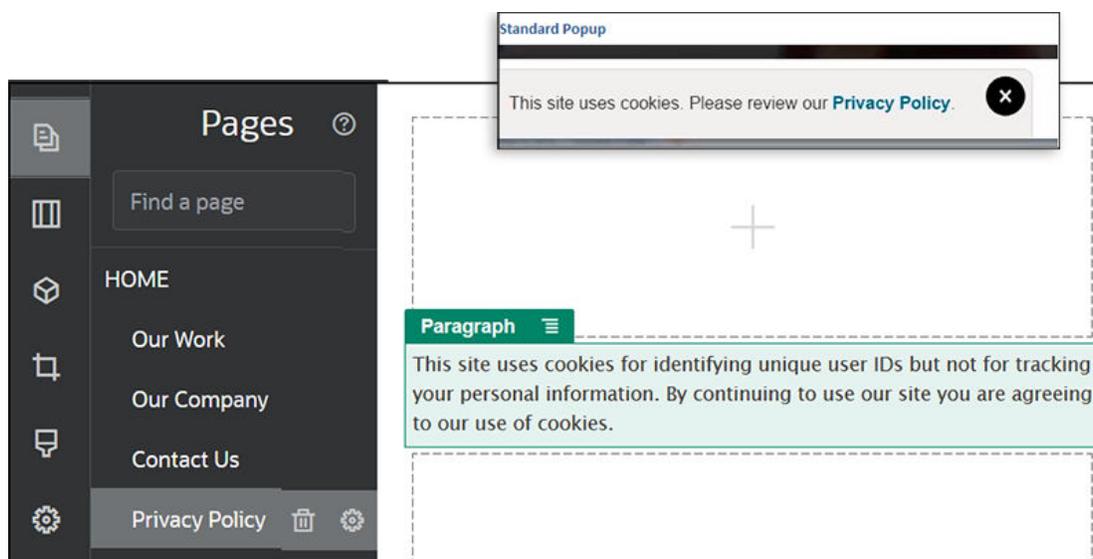
Всплывающее окно должно содержать следующую информацию:

- текст верхнего и нижнего колонтитулов;
- тег Div в нижнем колонтитуле с определенным идентификатором, из которого текст выбирается и привязывается к

```
<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>
```

Страница политики конфиденциальности

Текст будет содержать ссылку на страницу политики конфиденциальности с рекомендательным текстом.



Внутренний HTML-код для поисковой оптимизации

При сохранении данных компонента в Site Builder можно сохранить внутренний HTML-код, который создается при визуализации компонента во время выполнения. Этот внутренний HTML-код хранится в данных страницы, поэтому при рендеринге страницы эти данные можно вставить в страницу вместо компонента. Это может происходить в самом начале представления страницы, что дает механизму поиска больше шансов на успешное сканирование контента страницы.

Общие сведения о файле components.json и формате

Тема должна иметь файл components.json, расположенный в */ThemeName/components.json*, который указывает компоненты, используемые в теме.

Этот файл должен содержать допустимый JSON и, как минимум пустой массив JSON `[]`.

Синтаксис файла components.json содержит список всех локальных компонентов и полностью поддерживает категоризацию компонентов. (Удаленные компоненты зарегистрированы в каталоге компонентов.)

Независимо от того, какие компоненты добавлены в файл components.json уровня темы (включая "ни одного"), Oracle Content Management вносит набор компонентов по умолчанию, доступных пользователям. Этот набор по умолчанию определен в исходном коде. Ниже приведен список компонентов и (предварительно заданных) компонентов, визуализируемых во встроенных рамках. Кроме того, все удаленные компоненты, зарегистрированные на уровне сервиса и доступные пользователям в данном экземпляре, будут доступны в Site Builder.

В комплект поставки Oracle Content Management входят следующие локальные компоненты.

Имя	Тип	Идентификатор
Заголовок	scs-title	scs-title

Имя	Тип	Идентификатор
Абзац	scs-paragraph	scs-paragraph
Изображение	scs-image	scs-image
Галерея	scs-gallery	scs-gallery
Сетка галереи	scs-gallerygrid	scs-gallerygrid
Документ	scs-document	scs-document
Кнопка	scs-button	scs-button
Карта	scs-map	scs-map
Разделитель	scs-divider	scs-divider
Разделитель	scs-spacer	scs-spacer
YouTube	scs-youtube	scs-youtube
Социальная панель	scs-socialbar	scs-socialbar
Видео	scs-video	scs-video
Статья (пользовательский компонент)	scs-component	scs-comp-article
Заголовок (пользовательский компонент)	scs-component	scs-comp-headline
Изображение и текст (пользовательский компонент)	scs-component	scs-comp-image-text

Эти компоненты, визуализированные в встроенных рамках, входят в состав Oracle Content Management. Они не включают зарегистрированные удаленные компоненты.

Имя	Тип	Идентификатор
Диалог	scs-app	Диалог
Диспетчер документов	scs-app	Диспетчер документов
Список папок	scs-app	Список папок
Список файлов	scs-app	Список файлов
Кнопка Facebook "Нравится"	scs-app	Кнопка Facebook "Нравится"
Кнопка Facebook "Рекомендовать"	scs-app	Кнопка Facebook "Рекомендовать"
Кнопка Twitter "Подписаться"	scs-app	Кнопка Twitter "Подписаться"
Кнопка Twitter "Поделиться"	scs-app	Кнопка Twitter "Поделиться"

Общий формат

Общий формат файла components.json:

- В каждом компоненте указаны его свойства. Свойства "components" или "apps" верхнего уровня устарели.

- Каждый компонент имеет свойство "type". Компоненты могут иметь только определенные значения (все возможные значения перечислены в таблице компонентов по умолчанию).
- Каждый компонент имеет свойство "id", которое должно быть уникальным. Это свойство позволяет различать компоненты с одинаковым значением "type". Ранее у приложений было свойство "appName". Хотя свойство "appName" продолжает работать, если свойство "id" недоступно, свойство "appName" устарело.
- Каждый компонент имеет свойство "name", которое служит отображаемым именем в пользовательском интерфейсе. Если резервные значения не указаны, для компонентов данное значение — это имя соответствующего компонента по умолчанию, а для удаленных компонентов значение — идентификатор.

Ниже приведен пример файла components.json:

```
[
  {
    "name": "COMP_CONFIG_TEXT_CATEGORY_NAME",
    "list": [
      {
        "type": "scs-title",
        "id": "my-headline",
        "name": "My Headline",
        ...
      },
      {
        ...
      },
      ...
    ],
  },
  {
    "name": "My own category name",
    "list": [ ... ]
  }
]
```

Общая структура представляет собой массив JSON объектов категорий. Каждый объект категории имеет свойства "name" и "list". Свойство "name" может быть ключом, который сопоставляется локализованной строке. Если категорий по умолчанию недостаточно, можно указать собственное имя категории, которое не будет локализовано. В следующей таблице перечислены доступные категории по умолчанию и соответствующие ключи.

Клавиша	Имя категории (английский язык)
COMP_CONFIG_CONTENT_CATEGORY_NAME	Контент
COMP_CONFIG_CUSTOM_CATEGORY_NAME	Пользовательские
COMP_CONFIG_MEDIA_CATEGORY_NAME	Носитель
COMP_CONFIG_SOCIAL_CATEGORY_NAME	Соц. сеть
COMP_CONFIG_TEXT_CATEGORY_NAME	Текст

Свойство "list" в каждом объекте категории содержит массив объектов компонентов. Каждый компонент или объект должен иметь свойства "type" и "id". Другие свойства не обязательны.

- Свойство "type" должно соответствовать одному из типов компонентов по умолчанию. Если свойства "type" еще нет, компонент не отображается.
- Свойство "id" должно быть уникальным для всех компонентов. Если "id" уже существует, компонент не будет отображаться.
- Свойство "name" — это отображаемое имя компонента в пользовательском интерфейсе. Это заменяет предыдущее свойство "appName" для приложений (теперь удаленные компоненты).
- Все остальные свойства обрабатываются так же, как и в предыдущих версиях.

Добавление новых компонентов в components.json

Изменять компоненты по умолчанию запрещено. Однако на основе существующего компонента по умолчанию можно создать новый компонент. Например, новый компонент можно создать на основе компонента "scs-title", который задает текст по умолчанию. При добавлении нового компонента, как минимум, требуется указать свойства "type" и "id".

- Свойство "type" должно соответствовать одному из типов компонентов по умолчанию. Если свойства "type" еще нет, компонент не отображается.
- Свойство "id" должно быть уникальным для всех компонентов. Если "id" уже существует, компонент не будет отображаться.

Ниже приведен пример кода для добавления нового компонента "Заголовок". Этот компонент будет отображаться вместе с компонентом "Заголовок" по умолчанию.

```
[
  {
    "name": "COMP_CONFIG_TEXT_CATEGORY_NAME",
    "list": [
      {
        "type": "scs-title",
        "id": "my-headline"
      }
    ]
  }
]
```

Ниже приведен пример кода для добавления нового компонента "Заголовок" с отображаемым именем и текстом по умолчанию.

```
[
  {
    "name": "COMP_CONFIG_TEXT_CATEGORY_NAME",
    "list": [
      {
        "type": "scs-title",
        "id": "my-headline",

```

```
        "name": "My Headline",
        "initialData": {
            "userText": "This is a second title component"
        }
    }
]
]
```

Обратите внимание, что компонент "Заголовок" использует все свойства компонента "Заголовок" по умолчанию в качестве основы и применяет изменения на уровне темы для создания нового компонента.

Обратная совместимость

Файлы `components.json` в предыдущем формате по-прежнему могут быть прочитаны.

- Файлы со свойствами верхнего уровня `"components"` или `"apps"`.
- Если файл содержит свойство `"apps"`, пользовательские удаленные компоненты с этим свойством по-прежнему загружаются.
- Если файл содержит свойство верхнего уровня `"apps"`, предполагается, что все удаленные компоненты, перечисленные ниже, имеют тип `"scs-app"`.
- Если свойство `"appName"` присутствует, задайте свойству `"id"` значение `"appName"`. Отображаемое имя будет совпадать со значением `"name"`, если оно указано, или вернется к значению `"id"`.

Настройка групп панелей инструментов в Site Builder

Для пользовательских компонентов можно настроить копии групп готовых панелей инструментов, отображаемых на панели инструментов Site Builder.

Можно создать собственную версию готового компонента, которая имеет ограниченный набор и доступна в настраиваемом теге. Никакие настройки не влияют на готовые компоненты.

Можно создать собственную версию готового компонента, которая имеет ограниченный набор и доступна в настраиваемом теге, но это не повлияет на готовые компоненты.

Группы панелей инструментов то, что видно на панели инструментов, если нажать компонент Title (Заголовок) или Paragraph (Абзац) для его редактирования. Можно удалить элементы, поддерживаемых компонентами Title или Paragraph, и изменить их порядок, но дополнительные подключаемые модули не разрешены.

Для настройки группы панелей инструментов в Site Builder применяется тот же синтаксис, что CKEditor использует для своей конфигурации `toolbarGroups`. См. <https://ckeditor.com/latest/samples/toolbarconfigurator/index.html#basic>.

Компоненты Title и Paragraph поддерживают следующие группы:

- `"basicstyles"` — доступен жирный шрифт / курсив / подчеркивание <https://ckeditor.com/cke4/addon/basicstyles>
- `"styles"` — стили шрифта https://docs.ckeditor.com/ckeditor4/latest/guide/dev_styles.html

- "colors" — цвет текста и фона https://docs.ckeditor.com/ckeditor4/latest/guide/dev_colorbutton.html
- "undo" — отменить/повторить текущий экземпляр в CKEditor
- "links" — пользовательский дополнительный модуль диалога гиперссылки
- "paragraph" — нумерованный/маркированный список и поддержка отступов
 - "list"
 - "indent"
- "align" — выравнивание по левому, правому краю или по центру
- "cleanup" — удалить любые стили для выделенного текста

Компонент Paragraph также поддерживает подключаемые модули для вставки изображений и таблиц

- "insert"
 - "image"
 - "table"

Кроме того, можно использовать следующий разделитель строк:

- "/"

Примечание.

Если задать любое другое значение в конфигурации группы панелей инструментов, оно будет удалено до создания панели инструментов Site Builder. Вы не можете указать "extraPlugins". Поддерживается только конфигурация "имя"/"группы". Любые записи "элементов" будут игнорироваться.

Например, если необходимо запретить пользователям определять шрифты, цвета, стили или размеры, можно обновить конфигурацию панели инструментов указанным ниже образом. Для "id" необходимо указать пользовательские значения, отличные от значений, заданных в поле.

```
[{
  "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;"
    }
}]

```

Проверка

Замените файл `components.json` предыдущим кодом, а затем отредактируйте сайт (обновите окно браузера, если редактирование уже начато). На этом этапе при редактировании компонента Title (Заголовок) стили шрифта или цвета для выделения больше не отображаются. Компонент Paragraph (Абзац) по-прежнему их отображает, а список доступных размеров шрифтов ограничен значениями 16, 24 и 48.

Группы панелей инструментов по умолчанию

Группы панелей инструментов по умолчанию для заголовка и абзаца:

- Заголовок

```

[ {
    "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"]
    }
]

```

- Абзац

```

[ {
    "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"]
}
]

```

Ограничение компонентов в ячейках

Для любой ячейки компоновки можно указать определенные ограничения для компонентов, разрешенных в ячейке.

Если ограничить компоненты в ячейке, такие компоненты нельзя добавить или перетащить в ячейку, а при попытке перетаскивания в ячейку неразрешенного компонента отображается предупреждение.

Чтобы настроить это ограничение, измените макеты в теме (например, файл макета `themes\theme_name\layouts\oneslot.htm`) и добавьте пользовательские атрибуты данных в тег DIV ячейки.

Это формат пользовательских атрибутов. Основное отличие от предыдущих версий заключается в том, что пользователи должны указать только идентификатор `id` компонента. Ранее для ограничения компонента пользователям приходилось использовать подробный синтаксис "`<type> <id>`" (который все еще поддерживается).

```
data-allowed-items='["<id>:", "<type>", "<type>:<id>", ...]'
data-disallowed-items='["<id>:"<type>", "<type>:<id>", ...]
```

 **Примечание.**

В значениях параметров `data-allowed-items` и `data-disallowed-items` можно использовать двойные кавычки (") или одинарные кавычки ('). В следующем примере в значении параметра `data-allowed-items` используются одинарные кавычки вокруг массива JSON, а в значении параметра `data-disallowed-items` — двойные кавычки вокруг массива JSON:

```
<div id="slot101"
  class="scs-slot"
  data-allowed-items='["scs-app", "scs-title"]'
  data-disallowed-items="['File List', 'scs-map']">
</div>
```

В таблице ниже перечислены компоненты, предоставленные вместе с Oracle Content Management, и их соответствующие идентификаторы. Чтобы предотвратить конфликт имен, не добавляйте к идентификатору пользовательского компонента (локального или удаленного) префикс `scs-` либо используйте тип или идентификатор, указанный в этой таблице.

Имя	Тип	Идентификатор
Диспетчер документов	scs-app	Диспетчер документов
Кнопка Facebook "Нравится"	scs-app	Кнопка Facebook "Нравится"
Кнопка Facebook "Рекомендовать"	scs-app	Кнопка Facebook "Рекомендовать"
Список файлов	scs-app	Список файлов
Список папок	scs-app	Список папок
Кнопка Twitter "Подписаться"	scs-app	Кнопка Twitter "Подписаться"
Кнопка Twitter "Поделиться"	scs-app	Кнопка Twitter "Поделиться"
Кнопка	scs-button	scs-button
Статья (пользовательский компонент)	scs-component	scs-comp-article

Имя	Тип	Идентификатор
Заголовок (пользовательский компонент)	scs-component	scs-comp-headline
Изображение и текст (пользовательский компонент)	scs-component	scs-comp-image-text
Группа компонентов	scs-componentgroup	scs-componentgroup
Поиск в содержимом	scs-contentsearch	scs-contentsearch
Список содержимого	scs-contentlist	scs-contentlist
Заполнитель содержимого	scs-component	scs-contentplaceholder
Элемент контента	scs-component	scs-contentitem
Разделитель	scs-divider	scs-divider
Документ	scs-document	scs-document
Галерея	scs-gallery	scs-gallery
Сетка галереи	scs-gallerygrid	scs-gallerygrid
Изображение	scs-image	scs-image
Карта	scs-map	scs-map
Абзац	scs-paragraph	scs-paragraph
Социальная панель	scs-socialbar	scs-socialbar
Разделитель	scs-spacer	scs-spacer
Заголовок	scs-title	scs-title
YouTube	scs-youtube	scs-youtube

Пользователи могут создавать локальные или удаленные компоненты. Имя, указанное в этом примере (*My_Local_Component*), служит идентификатором, который можно использовать для ограничения этого компонента в ячейках.

Create Local Component ✕

Provide a name for your component

Use only letters, numbers, hyphens, and underscores in component names.

Provide a description for your component (optional)

Component Type

Cancel Create

Как сделать контент макета редактируемым

Определенный текст или контент изображения в макете можно настроить так, чтобы пользователи могли редактировать его, работая со страницами на основе данного макета.

Эту функциональную возможность можно использовать в любой теме, включая темы Bootstrap. Изменения автоматически учитывают стиль исходной страницы.

К следующим HTML-тегам в макете можно добавить простую разметку: `<p>`, с `<h1>` по `<h6>`, `<div>` и ``.

Варианты форматирования текста и изображений:

- **Text:** позволяет пользователям задавать значения Bold (Полужирный шрифт), Italic (Курсив), Underscore (Подчеркивание) и Link (Ссылка).
- **Image:** позволяет пользователям указать ссылку на файл изображения. Измените свойства заголовка (то, что видит пользователь при наведении курсора на изображение) и добавьте альтернативный текст для обеспечения доступности.

Изменение элемента текста или изображения, позволяющее его редактировать другими пользователями, требует двух действий:

1. Добавьте `scs-editable` в `class`.
2. Добавьте уникальный атрибут `id`.

После изменения макета данная функциональная возможность будет доступна для всех страниц на основе этого макета (даже для новых страниц).

При копировании и вставке страницы изменения копируются на новую страницу.

Изменение тега заголовка

Ниже приведен пример изменения тега заголовка в макете, позволяющего его редактировать другими пользователями.

1. Синхронизируйте файл макета с локальным настольным ПК или измените исходный HTML-файл для заголовка `<h1>`.
2. Добавьте `scs-editable` к `class`, а атрибут `id="test-heading"` в строку кода заголовка, чтобы к нему можно было прикрепить редактор. Пример:

```
<h1 class="brand-heading scs-editable" id="test-heading">Sample  
Heading Value</h1>
```

3. Сохраните файл.
4. Выполните синхронизацию с Oracle Content Management и перезагрузите браузер.

Когда пользователь наводит курсор на заголовок, вокруг этого заголовка должна появиться черная рамка, указывающая на то, что контент доступен для редактирования. Когда пользователь нажимает на заголовок, рамка становится зеленой, указывая на то, что теперь пользователь может редактировать контент заголовка.

5. Если тема не опубликована, сделайте это. Если она опубликована, то изменение появится при обновлении браузера.

После публикации темы пользователи сайта могут нажать заголовок и изменить его в Site Builder.

Для элемента `scs-editable` создается псевдокомпонент, чтобы его можно было изменить в Site Builder и сохранить вместе с данными страницы. Во время выполнения перед рендерингом страницы контроллер заменяет теги с атрибутом `scs-editable` значениями, заданными в Site Builder.

Изменение тега изображения

Процедура изменения тега `image` в макете, позволяющая его редактировать другим пользователям, аналогична процедуре для тега `text`.

1. Добавьте `scs-editable` в `class`.
2. Добавьте уникальный идентификатор `id` изображения.

Пользователь может нажать изображение, а затем изменить свойства, чтобы использовать другое изображение.

Ниже приведен пример кода для изображения, которое может редактироваться пользователями:

```

```

Создание макета раздела

Создайте макет раздела, чтобы разместить контент в ячейке на странице сайта.

Корпоративный пользователь может расположить элементы контента на сайте в соответствии с макетами разделов, которые вы предоставляете как разработчик. На основе макета по умолчанию можно создать новые макеты разделов.

Макет раздела можно экспортировать для внесения изменений в автономном режиме, а затем импортировать его в качестве нового компонента или макета либо заменить существующий компонент или макет. Экпортируйте макет раздела отдельно или в составе пакета шаблонов, включающего пользовательские компоненты и макеты.

Доступны следующие готовые макеты разделов:

- Горизонтально
- Два столбца
- Три столбца
- Вертикально
- С вкладками
- Ползунок

Их можно сразу использовать в Site Builder без необходимости создавать что-либо.

Файлы для этих макетов разделов содержат комментарии с более подробной информацией о структуре файлов макетов разделов. Чтобы просмотреть комментарии, на основе готового макета можно создать новый макет раздела, а затем экспортировать новый макет для редактирования, как описано в приведенной ниже процедуре.

Чтобы создать макет раздела, выполните указанные ниже действия.

1. На домашней странице нажмите **Разработчик**.
2. Нажмите **Смотреть все компоненты**.
3. В раскрывающемся меню **Создать** справа выберите **Создать макет раздела**.
4. В диалоговом окне **Создание макет раздела** укажите имя и описание своего компонента макета раздела.
5. Чтобы экспортировать макет раздела для редактирования, выберите его и в контекстном меню нажмите **Экспорт** или на панели действий нажмите .
 - a. Перейдите к существующей папке или нажмите **Создать**, чтобы создать новую папку и указать имя и, при необходимости, описание.
 - b. Установите флажок рядом с именем папки и нажмите **ОК**.
 - c. Нажмите на значок или имя папки, чтобы открыть ее.

В выбранной папке создается файл пакета макета с заданным именем макета раздела и расширением имени `.zip`. Выгрузите файл в среду разработки для редактирования указанных файлов.

Информацию об API-интерфейсе макета раздела можно найти в разделе [Разработка пользовательских макетов разделов с помощью API](#).

6. Импортируйте измененные файлы в качестве нового макета раздела или для замены существующего макета раздела.
 - a. На домашней странице перейдите на вкладку **Контент**, а затем нажмите **Консоль**.
 - b. Загрузите измененный макет раздела в папку, в файл с расширением имени **.zip**, который содержит экспортированные папки и файлы с теми же именами.
 - c. На странице **Разработчик** нажмите **Просмотреть все компоненты**.
 - d. В меню **Создать** выберите **Импорт компонента**.
 - e. Установите флажок рядом с загруженным ZIP-файлом, содержащим измененный макет раздела, и нажмите **ОК**.

Измененный макет раздела импортируется в выбранную папку.

Можно также экспортировать раздел макета, чтобы скопировать или переместить его в другой экземпляр Oracle Content Management и импортировать в этот экземпляр.

Создание макета раздела с поддержкой медленной загрузки (Lazy Load)

По мере запроса список контента может вызвать макет раздела с дополнительными компонентами.

При визуализации списка контента можно выбрать макет раздела для рендеринга всех возвращаемых элементов контента. Это позволяет создавать различные макеты элементов контента, например таблицы, ползунки или компоновки из восьми столбцов. Эти пользовательские макеты разделов также могут участвовать в расширенных функциях разбивки на страницы.

Списки контента поддерживают следующую разбивку на страницы:

- **Разбивка на страницы**
- **Загрузка при прокрутке**
- **Загрузка при нажатии**

Для стандартной функции разбивки на страницы от макета раздела не требуется никаких действий. Когда пользователь нажимает следующую страницу, она повторно визуализируется со следующим набором элементов. Тем не менее, для действий **Загрузка при прокрутке** и **Загрузка при нажатии** вместо повторной визуализации макета раздела к нему добавляются дополнительные компоненты. Это используется в основном для модели бесконечной прокрутки, в которой сначала загружаются первые *n* элементов, а по мере прокрутки страницы пользователем извлекается и визуализируется следующий набор элементов. Для поддержки действий **Загрузка при прокрутке** и **Загрузка при нажатии**, для пользовательского макета разделов необходимо создать следующие файлы:

1. `render.js`: внедрение `addComponent()` API. Вызывается для каждого нового компонента, добавляемого в макет раздела.

```
// 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`: включите следующее, чтобы в списке контента учитывалось то, что макет раздела поддерживает `addComponent()` api.

```

"contentListData": {
  "addComponent": true
},

```

После обновления `appinfo.json`, когда пользователь выбирает этот макет раздела на панели настроек и переходит на экран разбивки на страницы, отображаются команды `Load on click` и `Load on scroll`.

Разработка пользовательских макетов разделов с помощью API

Пользовательские макеты разделов можно создавать в Oracle Content Management с помощью API-интерфейса для макетов разделов, который включает API-интерфейсы для рендеринга и редактирования.

Начальные файлы можно посмотреть в разделе [Создание макета раздела](#). Начальные файлы для макетов разделов содержат комментарии с подробными сведениями о структуре файлов макетов разделов.

API-интерфейсы рендеринга

API-интерфейсы рендеринга, загруженные из модуля `render.js`, используются в Site Builder и во время выполнения.

API-интерфейс рендеринга	Описание	Входные параметры	Возвращаемый результат
(Constructor)	Инициализирует модуль рендеринга макета раздела.	Объект JavaScript, содержащий следующие свойства: <ul style="list-style-type: none"> • sectionlayoutData (объект): данные макета раздела, найденные в модели страницы. • componentId (строка): значение <code>componentId</code> макета раздела, обычно GUID. • renderMode (строка, необязательно): режим рендеринга для операции рендеринга. • customSettingsData (объект): копия <code>customSettingsData</code>, найденная в <code>sectionLayoutData</code>. 	API-интерфейсы рендеринга макетов разделов инициализированы.

API-интерфейс рендеринга	Описание	Входные параметры	Возвращаемый результат
render	Порождает DOM-элементы, подходящие для макета раздела, на страницу, включая теги DIV контейнеров для дочерних компонентов.	container (элемент): DOM-элемент, в который должен превратиться макет раздела в результате рендеринга.	Этот метод возвращает результат, а затем осуществляется рендеринг дочерних компонентов. Дочерние компоненты можно выявить по дочерним элементам <code>div[id]</code>
addComponent	Используется с компонентами списка контента для динамического добавления дочерних компонентов в макет раздела. Это необязательная функция.	container (элемент): DOM-элемент, в который должен превратиться новый компонент в результате рендеринга. componentId (строка): идентификатор нового компонента, добавляемого в макет раздела.	Этот метод возвращает результат, а затем осуществляется рендеринг элемента, идентификатор которого совпадает с введенным идентификатором <code>componentId</code> .

API-интерфейсы редактирования

Модуль `edit.js` загружается, если свойству `hasEditHandlers` задано значение `true` в файле `appinfo.json`, связанном с макетом раздела.

API-интерфейсы редактирования используются в Site Builder.

Все функции этого модуля, за исключением функции `Constructor`, являются дополнительными.

API-интерфейс редактирования	Описание	Входные параметры	Возвращаемый результат
(Constructor)	Инициализирует модуль редактирования макетов разделов.	Объект JavaScript, содержащий следующее свойство: <ul style="list-style-type: none"> componentId (строка): значение <code>componentId</code> макета раздела, обычно GUID. 	API-интерфейсы редактирования макетов разделов инициализированы.

API-интерфейс редактирования	Описание	Входные параметры	Возвращаемый результат
getCapabilities	Возвращает объект, описывающий возможности редактирования макета раздела.	<p>Объект JavaScript, описывающий возможности редактирования макета раздела.</p> <p>После ввода этой функции предоставляются возможности по умолчанию. Функция может нужным образом изменить объект Capabilities.</p> <p>Объект Capabilities может включать следующие возможности:</p> <ul style="list-style-type: none"> • title (строка): заголовок макета раздела, который отображается для пользователя. • settingsTitle (строка): заголовок, который отображается в диалоговом окне панели настроек. • hasSettings (логическое значение): указывает, поддерживается ли панель настроек макетом раздела. • allowMove (логическое значение): указывает, позволяет ли макет раздела перемещать дочерние элементы. • allowDelete (логическое значение): указывает, позволяет ли макет раздела удалять дочерние элементы. • isHidden (логическое значение): указывает, является ли макет раздела скрытым в результате пользовательских настроек. • dropTarget (логическое значение): указывает, является ли макет раздела целевым объектом для операций перетаскивания. • customMenuOptions (массив): пользовательские пункты меню для добавления элементов в контекстное меню макета раздела. Каждый объект menu-option имеет следующие свойства: 	(Объект): возможности для макета раздела.

API-интерфейс редактирования	Описание	Входные параметры	Возвращаемый результат
getCaptionContent	Возвращает отображаемое имя макета раздела, которое будет отображаться в элементах пользовательского интерфейса.	Нет.	(Строка): Отображаемое имя макета раздела.

API-интерфейс редактирования	Описание	Входные параметры	Возвращаемый результат
filterCapabilities	Позволяет макету раздела изменить объект Capabilities (Возможности) до того, как элементы меню будут показаны пользователю. Этот API-интерфейс можно использовать для настройки или удаления пунктов меню. (См. также getCapabilities .)	Объект JavaScript, описывающий возможности редактирования макета раздела. Во время ввода этой функции предоставляются возможности по умолчанию.	(Объект) Возможности для макета раздела.
onDragOver	Вызывается во время перетаскивания, чтобы указать, можно ли перетащить выбранный элемент на макет раздела.	eventObject (объект события): объект события, содержащий информацию о событии перетаскивания. dataTransfer (объект DataTransfer): объект DataTransfer с информацией об элементе, перетаскиваемом на макет раздела.	(Логическое значение) Значение, указывающее, может ли перетаскиваемый элемент быть принят макетом раздела. Возвращает значение <i>true</i> , если макет раздела может принять перетаскиваемый элемент, в противном случае возвращается значение <i>false</i> .

API-интерфейс редактирования	Описание	Входные параметры	Возвращаемый результат
onDrop	Вызывается при завершении перетаскивания, чтобы указать, что перетаскиваемый элемент следует поместить в макет раздела.	eventObject (объект события): объект события, содержащий информацию о событии завершения перетаскивания. dataTransfer (объект DataTransfer): объект DataTransfer с информацией об элементе, который перетаскивается в макет раздела.	(Логическое значение) Значение, указывающее, было ли завершение перетаскивания обработано макетом раздела. Если возвращается значения <i>true</i> , осуществляется обход логики по умолчанию.
onAddComponent	Уведомляет макет раздела о том, что в результате перетаскивания в макет раздела добавлен элемент.	eventObject (объект события): объект события, содержащий информацию о событии перетаскивания. dataTransfer (объект DataTransfer): объект DataTransfer с информацией об элементе, который перетаскивается в макет раздела. componentId (строка): значение <code>componentId</code> недавно добавленного элемента.	Уведомление макета раздела.
onMoveComponent	Уведомляет макет раздела о том, что в результате перетаскивания в макет раздела перемещен элемент.	eventObject (объект события): объект события, содержащий информацию о событии перетаскивания. dataTransfer (объект DataTransfer): объект DataTransfer с информацией об элементе, который перетаскивается в макет раздела. componentId (строка): значение <code>componentId</code> перемещенного элемента.	Уведомление макета раздела.
getSettingsData	Позволяет макету раздела изменить данные настроек перед отображением панели настроек.	settingsData (объект): данные настроек по умолчанию, вычисленные для макета раздела	(Объект) Данные настроек для макета раздела.

API-интерфейс редактирования	Описание	Входные параметры	Возвращаемый результат
updateSettings	<p>Позволяет макету раздела изменять настройки после закрытия панели настроек. Этот API-интерфейс вызывается непосредственно перед сохранением настроек в модели страницы.</p>	<p>parameters (объект): объект необработанных параметров, возвращаемый с панели настроек.</p> <p>sectionLayoutData (объект): данные макета раздела, которые будут сохранены. Данные по умолчанию будут созданы на основе parameters (Object) и переданы функции в этом параметре.</p>	(Объект) Данные макета раздела для хранения в модели страницы.
dispose	<p>Позволяет модулю редактирования освободить память, отключать события и освободить ресурсы, связанные с обработчиками и редактированием.</p> <p>Этот API-интерфейс вызывается, когда необходимо полностью перерисовать макет раздела, как в случае операции отмены/повтора.</p>	Нет.	Перерисовка макета раздела включена.

Разработка макетов контента

Макеты контента помогают пользователям просматривать данные в элементах контента с помощью списка контента или компонентов заполнителей контента, используемых на страницах сайтов. Можно создать несколько макетов контента для типа контента, чтобы создать различные виды или представить различные части элемента контента.

Например, для типа контента Blog-Post могут потребоваться различные макеты контента в зависимости от того, как и где будет использоваться контент Blog-Post. На домашней странице сайта может отображаться список элементов Blog-Post, но при нажатии на записи в блоге на главной странице на странице **Сведения** могут отображаться сведения об этой записи в блоге.

На домашней странице имеется список контента, настроенный на отображение элементов типа контента Blog-Post с использованием макета контента Blog-Post-Summary в виде представления элементов.

На странице "Сведения" используется макет контента Blog-Post-Header в качестве заместителя контента для отображения изображения верхнего колонтитула и заголовка. Макет раздела с двумя столбцами включает два заместителя контента шириной 70 и 30 процентов с использованием макетов контента Blog-Post-Content и Blog-Post-Author. Как можно видеть, для визуализации одного типа контента используются четыре различных макета контента.

Макет контента можно создать одним из двух способов:

- В Oracle Content Management выберите **Разработчик > Смотреть все компоненты > Создать > Создать макет контента**.
- В проекте инструментария OCE используйте команду `oec create-contentlayout`.

Oracle Content Management создает макет контента по умолчанию для типа контента. Чтобы изменить макет контента по умолчанию, можно изменить следующие файлы:

- `assets/layout.html`
Измените этот файл, чтобы изменить представление HTML.
- `assets/design.css`
Измените этот файл, чтобы стилизовать макет контента.
- `assets/render.js`
Измените этот файл, чтобы изменить данные, используемые в `layout.html`, или добавить динамическое поведение к макету контента.

Компоненты макета контента визуализируют элемент контента с сервера Oracle Content Management. Большая часть активов хранится на сервере Oracle Content Management. Иногда для создания стилей может потребоваться использовать статический актив, который доступен локально в самом макете контента, например фоновое изображение. Например, в приведенном ниже макете контента абсолютный URL-адрес файла `images/background.jpg` может быть создан в `render.js` и использован в `layout.html`.

Самый простой способ создания абсолютного URL-адреса — использовать встроенный файл `require.js`. Определите "require" в зависимый ресурс и используйте `require.toURL()` для создания URL-адреса, как показано в следующем коде:

```
define([
  'require',
  'jquery',
  'mustache',
  'text!./layout.html',
  'css!./design.css'
```

```
], function (require, $, Mustache, templateHtml, css) {  
    'use strict';  
  
    var imageURL = req.imageUrl('./images/background.jpg');
```

В последующих разделах описывается разработка макетов контента:

- [Создание макетов контента с помощью Oracle Content Management](#)
- [Передача представления макета макету контента](#)
- [Создание URL-адреса страницы сведений о сайте с помощью API](#)
- [Локальная разработка макетов контента с помощью Developer Cloud Service](#)
- [Разворачивание макросов в запросах списков содержимого](#)
- [Разработка надежных макетов контента](#)
- [Создание примера шаблона блога](#)
- [Добавление сопоставлений макетов контента к шаблонам](#)
- [Тестирование макетов контента с помощью локальной тестовой оснастки](#)
- [Импорт шаблонов с макетами контента в Oracle Content Management](#)

См. также

Информацию об управлении макетами контента см. в разделе [Управление пользовательскими компонентами и макетами](#).

Сведения об использовании цифровых активов и других элементов контента на сайте см. в разделах "Использование активов" и "Управление цифровыми активами" в документе *Управление активами с помощью Oracle Content Management*.

Создание макетов контента с помощью Oracle Content Management

Создайте макет контента для размещения полей в элементе контента при отображении на странице сайта. При добавлении на страницу элемент контента будет использовать выбранный макет контента.

Корпоративный пользователь может создавать и использовать элементы контента на основе типов и макетов контента, которые вы предоставили как разработчик. На основе макета по умолчанию можно создавать новые макеты контента. Так как с элементом контента связано несколько макетов контента, разработчики сайтов могут отображать элементы контента в различном контексте без изменения контента.

При использовании в компоненте списка контента макет контента однократно повторяется для каждого элемента контента. Затем компоновки контента упорядочиваются по макету раздела.

Макет контента можно экспортировать, чтобы изменить в автономном режиме, а затем импортировать для замены существующего макета контента.

Чтобы создать макет контента, выполните указанные ниже действия.

1. На домашней странице Oracle Content Management нажмите **Разработчик** на боковой панели навигации.

Отображается страница **Разработчик**.

2. Нажмите **Смотреть все компоненты**.
3. В раскрывающемся меню **Создать** справа выберите **Создать макет контента**.
4. В диалоговом окне **Создание макета контента** выберите типы контента, которые будут использовать макет, выберите поля для отображения, и включите параметр **Добавить поддержку пользовательских настроек при использовании на сайтах**, если требуется, чтобы создатели сайтов могли добавлять пользовательские настройки и стили при добавлении компонентов "элемент контента" и "список контента" на страницу.

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. Укажите имя и описание компонента макета контента и нажмите **Создать**. Этот макет контента добавляется к компонентам.

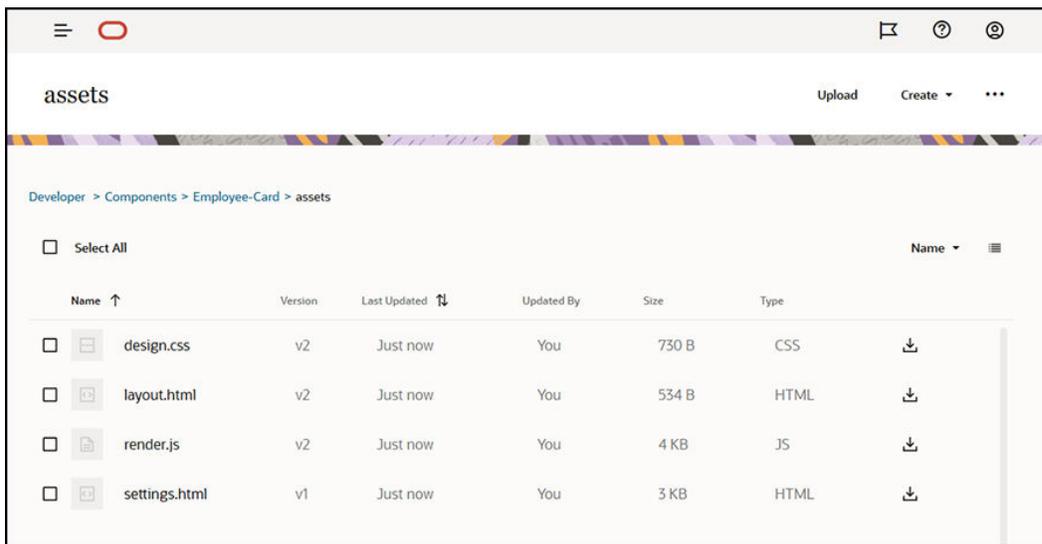
 **Примечание.**

В заголовках макета контента допустимы только буквы, цифры, дефисы и символы подчеркивания.

6. Чтобы экспортировать макет контента для редактирования, выберите его и в контекстном меню нажмите **Экспорт** или на панели действий нажмите .

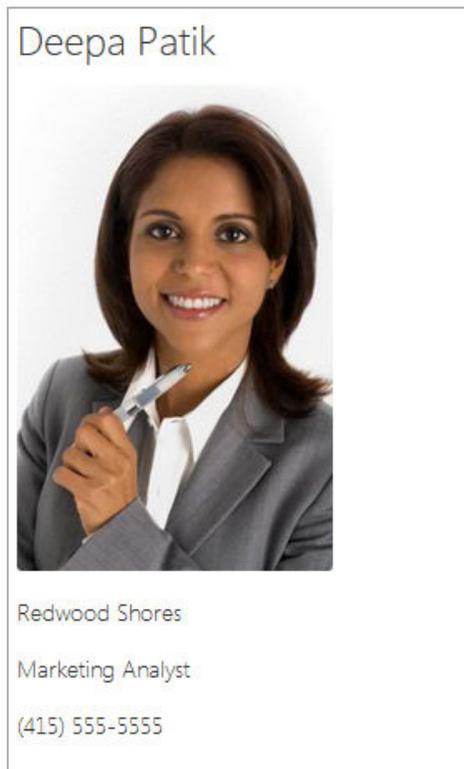
- a. Перейдите к существующей папке или выберите **Создать**, а затем **Папка**, чтобы создать новую папку, и введите имя и (при необходимости) описание.
- b. Установите флажок рядом с именем папки и нажмите **ОК**.
- c. Нажмите на значок или имя папки, чтобы открыть ее.

В выбранной папке создается файл пакета макета с заданным именем макета контента и расширением имени `.zip`. Выгрузите файл в среду разработки для редактирования указанных файлов.

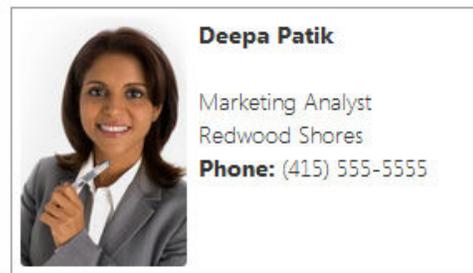


Эти файлы управляют макетом полей в элементах контента, использующих макет контента карты сотрудника. Если включена функция **Добавить поддержку пользовательских настроек при использовании на сайтах**, то создается также дополнительный файл с именем `settings.html`, который обеспечивает визуализацию по умолчанию одного элемента контента для его отображения.

Default Layout



Card Layout



7. Измените файлы `design.css`, `layout.html` и `render.js`, чтобы получить нужный макет контента.

Например, в следующих файлах укажите макет контента карты сотрудника:

- a. Измените файл `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. Измените файл `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>
```

с. Измените файл `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;
});
```

Подробнее о редактировании файла `render.js` и других файлов см. в разделе [Разработка компонентов](#).

8. Импортируйте измененные файлы, чтобы заменить существующий макет контента.

- a. На домашней странице нажмите **Документы**.
- b. Измененный макет контента в файле с расширением имени **.zip**, который содержит экспортированные папки и файлов с теми же именами, загрузите в папку.

Если требуется импортировать его как новый макет контента, необходимо в файле `_folder.json` изменить GUID макета контента.

- c. На домашней странице нажмите **Разработчик**.
Отображается страница **Разработчик**.
- d. Нажмите **Смотреть все компоненты**.
- e. В меню **Создать** выберите **Импорт компонента**.
- f. Установите флажок рядом с загруженным ZIP-файлом, содержащим измененный компонент, и нажмите **ОК**.

Измененная компоновка контента импортируется в **компоненты**.

Макет контента также можно экспортировать, чтобы скопировать или переместить его в другой экземпляр Oracle Content Management и импортировать в этот экземпляр.

Передача представления макета макету контента

При разработке макета контента можно получить свойства базового компонента, если данный макет контента используется на сайтах.

Для этого конкретного сценария использования можно получить представление макета контента, выбранное для типа контента в списке контента. Затем можно изменить способ визуализации компонента в зависимости от категории. Без доступа к этому свойству потребовалось бы создать два фактически одинаковых макета контента.

Свойство `contentLayoutCategory` доступно для макетов контента, визуализированных как для элемента контента, так и для компонентов списка контента. Доступ к этому свойству можно получить с помощью Sites SDK следующим образом.

```
scsData.SitesSDK.getProperty('contentLayoutCategory', function  
(layoutCategory)  
    { console.log(layoutCategory); });
```

См. Комплекты Oracle Content Management SDK.

Это свойство доступно, только когда макет контента визуализируется из элемента контента или списка контента, которые находятся на сайте Oracle Content Management. Это недоступно, если макет контента визуализирован из стороннего приложения.

Создание URL-адреса страницы сведений о сайте с помощью API

При визуализации списка элементов контента в макете контента из собственного запроса можно создать ссылку на страницу сведений для элемента контента. Для создания URL-адреса страницы **Сведения о сайте** можно использовать API-интерфейс `SCSRenderAPI.getPageLinkData` комплекта Sites SDK.

Свойство `detailPageId` — это идентификатор страницы сведений, выбранной в элементе контента или списке контента. Если значение не задано, возвращаемое значение соответствует первой странице в `SiteStructureMap` с заданным свойством `isDetailPage`.

Для доступа к этому свойству можно использовать Sites SDK. Это доступно, только если такой макет контента используется для сайта Oracle Content Management. Доступ к нему можно получить только посредством параметра `scsData`, который передается как один из аргументов при создании макета контента. Пример:

```
scsData.SitesSDK.getProperty('detailPageId', function (detailPageId)  
{ console.log(detailPageId); });
```

После получения идентификатора `detailPageId` его можно использовать для создания ссылки на страницу сведений.

`SCSRenderAPI` имеет функцию `getPageLinkData()`, которая принимает `pageId` и дополнительные параметры и создает требуемый URL-адрес страницы, передавая указанные параметры. Подпись для этой функции:

```
SCSRenderAPI.getPageLinkData(pageId,  
    options);
```

Она содержит следующие параметры:

- **pageID:** такой же, как идентификатор `detailPageId`, возвращаемый из свойства `detailPageId` Sites SDK.
- **параметры:**
 - `contentType`

- contentId
- contentName

Возвращаемое значение — это объект со следующими свойствами:

- hideInNavigation
- href
- href

В следующем примере все это сводится воедино:

```
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);
});
```

Это будет выглядеть следующим образом: `"/sites/{site}/{detailPageName}/
{contentType}/{contentId}/{contentSlug}"`

Если `pageId` не соответствует странице сведений, значения контента не добавляются в URL-адрес.

См. Комплекты Oracle Content Management SDK.

Локальная разработка макетов контента с помощью Developer Cloud Service

Developer Cloud Service можно использовать вместе с инструментарием OCE для создания, редактирования, настройки и тестирования макетов контента локально для Oracle Content Management.

Выполните указанные ниже действия, чтобы подготовиться к разработке макетов контента с помощью Developer Cloud Service.

1. [Настройка инструментария OCE на локальном компьютере.](#)
2. [Вход в консоль Developer Cloud Service для Oracle Content Management](#)
3. [Создание шаблона в Developer Cloud Service.](#)
4. [Добавление инструментария OCE в код проекта в новой репозитории GIT.](#)
5. [Создать макет контента](#)

В следующих темах описывается, как использовать инструментарий OCE для локальной разработки макета контента с помощью Developer Cloud Service:

- [Создание макета контента с помощью Developer Cloud Service](#)
- [Определение модуля RequireJS](#)

- [Настройка параметра функции Constructor](#)
- [Рендеринг макета контента](#)
- [Редактирование макета контента в шаблоне Mustache](#)
- [Добавление динамического преобразования DOM](#)
- [Определение стилей в файле design.css](#)
- [Получение ссылочных позиций](#)
- [Получение URL-адреса медиаданных](#)
- [Вызов триггеров](#)
- [Переход на страницу поиска с помощью запроса поиска](#)
- [Развертывание макрокоманд и рендеринг RTF](#)
- [Ссылка на страницу сведений](#)

После разработки макета контента с помощью шаблона Developer Cloud Service [объедините изменения](#) с репозиторием проекта Git.

Создание макета контента с помощью Developer Cloud Service

Для типа контента в проекте Developer Cloud Service можно создать макет контента с помощью команды `cec create-contentlayout`.

Чтобы создать макет контента в проекте Developer Cloud Service, выполните указанные ниже действия.

1. В окне терминала перейдите к каталогу `cec-components`.
2. Введите команду `cec create-contentlayout`, чтобы просмотреть параметры и примеры данной команды:

```
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. Введите следующую команду, чтобы узнать, какие типы контента доступны на сервере:

```
cec list-server-content-types
```

4. Как показано в разделе "Использование", можно создать макет контента для типа контента на сервере Oracle Content Management или для типа контента в шаблонах в каталоге `cec-components/src/main/templates`. Например, следующая команда создает макет контента для типа на сервере:

```
cec create-contentlayout Blog-Post-Overview-Layout -c Blog-Post -r
```

Макет контента можно изменить, редактируя следующие файлы:

- `assets/layout.html`

Этот файл указывает представление HTML. См. [Редактирование макета контента в шаблоне Mustache](#).

- `assets/design.css`

Этот файл указывает стиль макета контента. См. [Определение стилей в файле design.css](#).

- `assets/render.js`

Этот файл указывает данные, используемые в `layout.html`, и позволяет добавлять динамическое поведение к макету контента. См. [Определение модуля RequireJS](#).

Если был создан макет, поддерживающий пользовательские настройки (с помощью параметра `-a` при создании макета), также создается дополнительный файл с именем `settings.html`, который обеспечивает визуализацию по умолчанию одного элемента контента для отображения.

Определение модуля RequireJS

Модуль RequireJS можно определить в файле `render.js`. Сайты загружают зависимые ресурсы, такие как JQuery, Mustache, а также подключаемые модули RequireJS Text и RequireJS CSS.

```
define([
  'jquery',
  'mustache',
  'text!./layout.html',
```

```
    'css!./design.css'  
  ], function ($, Mustache, templateHtml, css) {
```

Для рендеринга макета можно использовать систему шаблонов Mustache.

Файл `assets/render.js` для макета контента имеет следующие свойства:

- Это должен быть модуль RequireJS
- Он должен возвращать JavaScript-функцию `Constructor`. Сайты вызывают функцию `Constructor`, передавая объект параметра. Объект параметра содержит данные элемента контента и API-интерфейсы, необходимые для рендеринга макета.
- Эта функция `Constructor` должна иметь метод `render(parentObj)` для обработки рендеринга макета контента. Он должен добавить объект макета контента `DOM` в объект `parentObj`, который передается методу `render()`.
- Модуль `RequireJS` может использовать зависимые ресурсы, такие как `jQuery`, `Mustache`, а также подключаемые модули `RequireJS Text` и `RequireJS CSS`. Эти зависимые ресурсы будут загружены сайтами. Также можно использовать другие библиотеки.

Настройка параметра функции `Constructor`

Когда сайты создают новый экземпляр функции конструктора, он передает параметр, который содержит `contentItemData`, `scsData` и `contentClient`, чтобы способствовать разработке макета контента.

Ниже приведен пример кода для функции конструктора:

```
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;
```

Параметр функции конструктора включает следующие объекты:

- **params.contentItemData:** содержит элемент контента, включая его имя, описание, идентификатор и данные. Например, поле "blogpost_title" в элементе контента можно открыть с помощью выражения `params.contentItemData.data['blogpost_title']`.
- **params.scsData:** этот объект передает информацию, когда конструктор вызывается из сайтов. Этот объект не существует для макетов контента, визуализированных в сторонних приложениях. Этот объект содержит объект Sites SDK, метод `contentTriggerFunction` для вызова триггера и ссылки на страницу **Сведения**.
- **params.contentClient:** это объект `contentClient`, созданный на основе Content SDK и используемый для вызова макета контента. Таким образом, он настраивается с помощью соответствующих параметров для сервера контента. Если необходимо выполнить дополнительные вызовы сервера контента, можно использовать этот объект `contentClient`, а не создавать собственный объект. Этот объект содержит клиентские API-интерфейсы для контента. API-интерфейсы доступны для создания запросов, поиска и получения элементов контента и их типов контента. Также доступны другие вспомогательные API-интерфейсы, например `expandMacros()`, для развертывания макрокоманд, используемых в RTF.

Рендеринг макета контента

Метод `render(params)` для `ContentLayout` визуализирует макет контента из шаблона. По умолчанию для макетов контента используется шаблон `Mustache`, но можно использовать любую нужную технологию шаблонов.

Метод `render(params)` для `ContentLayout` может использовать для рендеринга шаблона с данными следующий код:

```
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);
}
```

Необходимые данные можно добавить в объект `content`, созданный на основе `params.contentItemData`. Oracle рекомендует объединить в этот объект свойства из `params.scsData`, чтобы шаблон также их использовать. Визуализированный шаблон следует добавить к родительскому объекту, переданному методу `render()`.

Редактирование макета контента в шаблоне `Mustache`

Шаблон `Mustache` по умолчанию можно изменить в файле `assets/layout.html`, который содержит макет контента по умолчанию.

Простой шаблон для визуализации заголовка блога:

```
<h1>
  {{blogTitle}}
</h1>
```

Добавление динамического преобразования DOM

После вызова `Mustache.render()` и добавления шаблона к родительскому объекту в файл `render.js` можно добавить динамическое преобразование DOM.

Например, можно прикрепить прослушиватель, динамически изменяя следующий стиль:

```
// Dynamic DOM Manipulation can be done here
$('h1').click(function (event) {
  alert('Title is : ' + $(this).text());
});
```

Определение стилей в файле design.css

В файле `design.css` можно определить любой стиль, используемый в шаблоне макета контента.

Файл `design.css` загружается в определение модуля с помощью подключаемого модуля `RequireJS CSS`.

Получение ссылочных позиций

Для типа контента с полем ссылочных данных, которое ссылается на другой тип контента, можно получить ссылочную позицию.

Например, поле `Author` в типе контента `Blog-Post` представляет собой ссылку на тип контента "Автор". В макете контента для типа `Blog-Post` команда `contentClient.getItems()` связывает сведения о ссылочной позиции "Автор" с текущим элементом типа `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
});
```

Получение URL-адрес медиаданных

Команду `contentClient.getRenditionURL()` можно использовать для получения стандартного представления цифрового актива, такого как изображение.

```
blogHeaderImage = contentClient.getRenditionURL({
    'id': data['blog-post_header_image'].id
}),
```

Если нужны другие представления, такие как `thumbnail`, цифровой актив можно получить с помощью команды `contentClient.getItems()` и сослаться на объекты `item.data.revisions.default` и `item.data.revisions.thumbnail`.

Вызов триггеров

Выражение `scsData.contentTriggerFunction(pipad)` можно использовать для вызова триггера из макета контента.

Ниже приведен пример вызова триггера из макета контента "Автор" при нажатии имени автора:

```
{{#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}}
```

Полезные данные — это поисковый запрос для выбранного автора, который может прослушиваться другими элементами списка контента на странице.

Переход на страницу поиска с помощью запроса поиска

Наиболее распространенный сценарий использования — переход на страницу поиска с помощью динамического запроса поиска при нажатии ссылки в макете контента.

Например, предположим, что требуется перейти на страницу поиска с именем "Авторы", нажав на ссылку "Дополнительные статьи этого автора" в макете контента и передавая полезные данные поиска. Для этого используется приведенный ниже код. Обратите внимание, что глобальные объекты `SCS` и `SCSRenderAPI` доступны для использования в макете контента при выполнении на странице сайтов.

```
$('.more-from-author').click($.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));

```

Если ожидается многократное использование одного и того же контента на одной странице, уникальный идентификатор лучше использовать в селекторе CSS, а не в селекторе классов, например `$('.more-from-author').click(...)`.

Пример:

```

template.html
    <div id="{{navigateId}}">...</div>

render.js
    content.navigateId = this.scsData.id + 'detailTrigger';
    $('#' + navigateId).click(...)

```

Развертывание макрокоманд и рендеринг RTF

RTF в элементе контента позволяет встроить цифровое изображение.

Для правильного рендеринга такого RTF в макете контента используйте API-интерфейс `contentClient.expandMacros()`. Это позволяет разрешить все ссылки на цифровые активы внутри RTF.

```

data["starter-blog-post_content"] =
    contentClient.expandMacros(data["starter-blog-post_content"]);

```

Если для рендеринга используется Mustache, для рендеринга значения RTF следует использовать скобки `{{{ }}}` , так как RTF содержит HTML-код. Когда вокруг переменной используются скобки `{{{ }}}` , Mustache не пропускает HTML-код.

Ссылка на страницу сведений

Ссылка на страницу сведений доступна посредством выражения `scsData.detailPageLink`.

Например, если требуется перейти на страницу сведений, чтобы отобразить сведения о блоге при нажатии на его заголовок, вы можете использовать ссылку на страницу сведений следующим образом:

```
<a href="{scsData.detailPageLink}">
  <h2 class="post-title">
    {{blogTitle}}
  </h2>
</a>
```

Разворачивание макросов в запросах списков содержимого

В запросе списка контента можно определить значения свойств, которые вычисляются при обработке страницы для отображения недавно обновленного контента.

Большинство свойств компонентов в пределах сайтов статично. Пользователи выбирают или вводят фиксированную строку или значение для одного из свойств компонента, и оно не зависит от времени или места обработки страницы. Однако можно определить значения свойств, которые вычисляются при обработке страницы. Это полезно для отображения контента, который недавно обновлялся в результате выполнения запросов контента. Пользователи могут вводить даты, например "за последние 3 дня".

Для нескольких свойств можно вставить расширение Mustache JS. Значения, на которые ссылаются эти строки, выводятся из модели, которая выполняется при обработке страницы. Готовая модель обрабатывает даты, отформатированные для вызовов API-интерфейса Content REST. Эту модель можно расширить, добавив дополнительные значения в соответствии с требованиями пользователей.

Пример строки, которую можно ввести для свойства:

```
Content List component:
  Additional Query String property:
    updatedDate gt "{{#content.date}}today - 3 days{{/
content.date}}"
```

Эта запись Mustache для даты будет оцениваться во время выполнения, чтобы возвращаемое значение изменялось в зависимости от времени выполнения (то есть, разворачивается до `updatedDate gt "2220181002060000000"`). Таким образом пользователь может создать любую сложную строку даты, не вводя заранее определенное значение.

Поддерживаемые свойства компонентов

Следующие свойства поддерживают синтаксис шаблона Mustache JS:

- Список содержимого
 - Дополнительная строка запроса

- Например: `updatedDate gt "{{#content.date}}today - 3 days{{/content.date}}"`
- Заголовок, абзац, текст
 - RTF, введенный с помощью CKEditor
 - Например: `"Content REST API format for date: {{#content.date}}now{{/content.date}}"`

Примечание.

Без пользовательской модели для шаблона Mustache расширение из раздела "Заголовок, абзац, текст" не очень полезно. Тем не менее, очень полезно проверить данные, вводимые в строке дополнительного запроса, так как они оцениваются при переключении между редактированием и просмотром и сразу же становятся видимыми.

Поддерживаемый синтаксис компонентов

Объект `content.date` поддерживается без предварительной настройки. Для этого требуются два основных параметра: `today` и `now`.

Значение `today` принимает текущее время браузера, преобразует его в полночь сегодня, а затем преобразует полученное значение во время UTC.

- `{{#content.date}}today{{/content.date}}` распространяется до значения браузера в полночь сегодня, преобразованного в значение UTC в формате даты API-интерфейса Content REST. Пример:

```
2220181008065959999
```

- Затем его можно дополнить:

```
today +/- [day | week | month | year]
```

- Значение `today` также ведет себя по-разному при добавлении к нему или вычитании из него. При вычитании из него используется утреннее время. При добавлении к нему используется время в полночь. Пример:

- `{{#content.date}}today - 1 day{{/content.date}}` распространяется до вчерашнего дня в начале суток.

- `{{#content.date}}today + 2 days{{/content.date}}` распространяется до послезавтра в полночь.

Значение `now` принимает текущее время браузера и преобразует его во время UTC без каких-либо изменений.

- `{{#content.date}}now{{/content.date}}` распространяется до текущего времени браузера, преобразованного в значение UTC в формате даты API-интерфейса Content REST.

- `now` можно дополнять параметром `hour`. Итак, получаем следующее:

```
now +/- [hour | day | week | month | year]
```

- Пример:
 - `{{#content.date}}now + 2 hours{{/content.date}}` — значение "через два часа от текущего момента", преобразованное во время UTC в формате даты API-интерфейса Content REST
 - `{{#content.date}}now - 1 day{{/content.date}}` - значение "вчера в это время браузера", преобразованное во время UTC в формате даты API-интерфейса Content REST

Использование поддерживаемого синтаксиса компонентов

Предположим, что требуется вернуть все за последние 3 недели. Чтобы использовать расширение макроса в дополнительной строке запроса, необходимо ввести следующее:

```
updatedAt gt "{{#code.date}}today - 3 weeks{{/code.date}}"
```

Возвращается только дата, поэтому для работы с вызовом API-интерфейса Content REST при построении строки запроса добавляются кавычки, как при вводе статического значения.

MustacheJS

Синтаксис описывается на страницах с шаблонами Mustache JS по адресу mustache.github.io/mustache.5.html.

Было внесено одно изменение в экземпляр Mustache, который выполняется при расширении строк. Mustache предоставляет расширение `text`, использующее формат `{{ }}`, и расширение `html`, формат `{{{ }}`. Разница между этими двумя выражениями заключается в том, что расширение "text" кодирует строку в формате HTML; то есть, если значение развернуто до `a < b`, результатом будет `a < b`. Это не те строки, которые хотелось бы построить для URL-адресов. Вы могли бы посоветовать пользователю использовать расширение HTML, но это ведет лишь к непроизводительным издержкам и появлению дополнительных проблем, таких как объяснение того, почему необходимо использовать формат `{{{ }}`.

Чтобы избежать этого, модуль Mustache был настроен таким образом, чтобы не избегать значений при использовании формата `{{ }}`. Это означает, что выражения `{{ }}` и `{{{ }}` ведут себя одинаково. При таком подходе при необходимости кодирование результатов остается пользователям в качестве упражнения.

Модель ООТВ Mustache

Для расширения Mustache требуется применить модель к шаблону. В предыдущем примере выражение `{{#content.date}}` уже задано, а разработчик может добавить новое свойство, например `{{person}}`. Если пользователь вводит в шаблоне Mustache значение, которого нет в модели, в результате получается пустая строка. Таким образом, в случае выражения `Hello {{person}}` оно разворачивается просто до `Hello`, если разработчик не добавит в модель объект `person`.

Используемый объект модели представляет собой глобальный объект, вызываемый `SCSMacros`. Разработчик может добавлять в этот объект любые дополнительные записи. При оценке шаблона объект передается модулю Mustache.

Готовый объект модели в настоящее время поддерживает только объект `content.date`:

```
{
  content: {
    date: <lambda implementation>
  }
}
```

Пользовательская модель Mustache

Поддерживаемые объекты могут быть улучшены разработчиками в соответствии со своими требованиями. Таким образом, они могут ввести объект `lastTwoDays` и упростить расширение просто до выражения `{{lastTwoDays}}`.

Чтобы расширить модель для поддержки объекта, аналогичного `Hello {{person}}`, в предыдущем примере потребовалось бы добавить в `SCSMacros` объект `person`. Это необходимо сделать до обработки страницы. Это можно сделать в макете страницы, добавив тег сценария в его начало. Пример:

```
<script type="text/javascript">
window.SCSMacros = window.SCSMacros || {}; // define/get the SCSMacros
object
window.SCSMacros.person = "World";
</script>
```

После внесения этого изменения шаблон `Hello {{person}}` разворачивается до `Hello World`.

Если требуется передать значения данному объекту (например, `Hello {{#person}}personId{/person}}`), необходимо реализовать `mustache lambda`, а также перенести и развернуть значение в данной реализации.

Пример:

```
<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>
```

После внесения этого изменения шаблон `Hello {{#person}}111{/person}}` был бы развернут до `Hello Small`, а шаблон `Hello {{#person}}222{/person}}` — до `Hello Big`.

Примечание.

Расширение Mustache выполняется синхронно. Если требуется извлечь асинхронные значения, их необходимо разрешить в модели перед попыткой выполнения расширения mustache. В настоящее время это не поддерживается, хотя возможны специальные варианты реализации.

Разработка надежных макетов контента

Макеты контента должны быть надежными для данных ответа трех типов, получаемых от вызовов REST контента:

- Элемент контента: если указан параметр `expand=all`, разворачиваются ссылки и большие текстовые поля.
- Элемент контента: без параметра `expand=all`, ссылки не разворачиваются, но имеются большие текстовые поля.
- Запросы контента не будут содержать развернутые ссылки или большие текстовые поля.

Для повышения производительности вызовов REST контента, который извлекает данные активов, начиная с выпуска Oracle Content Management 19.2.3 больше не включает параметр `expand=all`. Параметр `expand` сообщает Oracle Content Management о необходимости детализации и извлечения всех связанных элементов, а также текущего элемента в ответе. Пользовательские макеты контента, в которых используются данные, извлеченные с помощью параметра `expand`, необходимо обновить, чтобы учесть случаи, когда извлеченные данные не содержат значения полей, на которые указывают ссылки.

Визуализация элементов контента

Макеты контента используются для визуализации элементов контента. Они получают данные элементов контента, преобразуют их в HTML-код и вставляют в страницу.

По умолчанию макеты контента используют шаблон Mustache для визуализации элементов контента, хотя они могут быть реализованы с помощью любой технологии JavaScript. При визуализации шаблона Mustache предполагается, что данные имеют определенный формат. В файле `render.js` макета контента необходимо убедиться, что передаваемая шаблону модель соответствует этому формату.

Макеты контента визуализируются в нескольких сценариях использования:

- При использовании в пользовательском интерфейсе управления активами Oracle Content Management данные могут находиться в состоянии "редактируются", чтобы пользователи могли предварительно просмотреть изменения, прежде чем их сохранять.
- При использовании на сайте Oracle Content Management в списке контента или элементе контента к данным добавляется дополнительная информация о сайте, на котором они используются.
- При использовании посредством вызова `contentClient.renderLayout()` комплекта Content SDK, когда пользователь Content SDK передает нужные данные непосредственно макету контента.

В целях повышения производительности существует общий компромисс между созданием одного запроса, который может вернуть все необходимые данные, или нескольких запросов, чтобы структура визуализировалась как можно быстрее, по быстрому исходному запросу. Области заполняются последовательно посредством последующих запросов. Выбор модели зависит от данных и сценариев использования.

Кроме того, формат данных, передаваемых макету контента, зависит от способа извлечения данных. Например, если используется Content REST с параметром `expand`, также могут быть возвращены ссылки полей на другие элементы контента, как по отдельности, так и в группе. Если элементы контента, на которые указывают ссылки, не включены, необходимо выполнить дополнительные вызовы REST.

Для обработки всех случаев разработчик макета контента должен гибко подходить к формату полученных данных. Кроме того, при необходимости разработчик может извлечь дополнительные данные и преобразовать их в формат, ожидаемый при визуализации шаблона.

Стандартизация структуры данных для макета контента

Разработчику макетов контента необходимо стандартизировать структуру данных, получаемых макетом контента.

Если присутствуют все данные, макет контента может просто визуализировать компонент. Если присутствуют не все данные, может потребоваться создать дополнительные запросы для макета контента. Во всех случаях макет контента никогда не должен принимать определенный формат данных, а вместо этого преобразовывать данные в формат, который будет визуализироваться.

Необходимо убедиться в наличии всех ожидаемых данных. Если данных нет, необходимо выполнить дополнительные запросы. Следующие поля могут отсутствовать в данных:

- Запись `"fields"` для ссылочных полей
- Большие текстовые поля

Поскольку макеты контента предназначены для определенных типов контента, разработчику макетов контента известен список необходимых полей. Для каждого из этих полей данные необходимо извлечь, чтобы можно было визуализировать макет контента. Существует два варианта: извлечь отсутствующие данные, а затем выполнить рендеринг с полными данными или немедленно выполнить визуализацию, а затем извлечь отсутствующие данные для заполнения пустых полей.

Вариант 1. Извлечь отсутствующие данные и затем выполнить рендеринг с полными данными

Создайте ожидаемый объект, чтобы извлечь необходимые данные и продолжить рендеринг при возвращении всех ожидаемых объектов.

Например, у нас есть следующие типы контента с соответствующими полями:

- `starter-blog-author`
 - поля
 - * `starter-blog-author_name` — текстовое поле

- * starter-blog-author_bio — текстовое поле
- starter-blog-post
 - поля
 - * starter-blog-post_title — текстовое поле
 - * starter-blog-post_content — большое текстовое поле
 - * starter-blog-post_author указывает на элемент starter-blog-author

Макет контента имеет следующий шаблон для рендеринга этих ожидаемых значений полей:

```

{{#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}}

```

Макет контента можно вызвать с помощью данных из следующих запросов:

- Запрос элемента с параметром expand" — все данные предоставлены
 - /content/published/api/v1.1/items/{id}?expand=fields.starter-blog-post_author&channelToken=8dd714be0096ffaf0f7eb08f4ce5630f
 - Это формат данных, необходимый для успешного заполнения всех значений в шаблоне. Если используется любой другой запрос, для извлечения данных и их преобразования в этот формат требуется дополнительная работа.
 - "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": "..."
  }
}
          
```

- Запрос элемента без параметра "expand" — отсутствуют ссылочные поля элементов **"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"
  }
}
```

- Запрос SCIM — отсутствует большое текстовое поле **"starter-blog-post_content"**, отсутствуют ссылочные поля элементов **"starter-blog-post_author.fields"**:

```
- /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"
  }
}
```

Чтобы обеспечить согласованный рендеринг с любым из этих запросов, сценарий `render.js` из макета контента должен проверить, что все ссылочные поля развернуты и большие текстовые поля присутствуют.

Если это не так, необходимо запросить их обратно, исправить данные и затем выполнить рендеринг с полными данными.

Пример функции `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 < referredFields.length; i++) {  
          if(referencedItem.id === fieldsData[referredFields[i]].id)  
{  
            // copy across the fields values  
            fieldsData[referredFields[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);
}
});
}
```

Вариант 2. Немедленно выполнить рендеринг, а затем извлечь недостающие данные для заполнения бланков

Производительность можно повысить, отделяя элементы, которые могут отсутствовать, и выполняя их рендеринг при втором проходе. Для этого потребуется два шаблона Mustache: первый выполняет первоначальный рендеринг, оставляя "пустоты", которые заполняются во время второго рендеринга после получения всех данных.

Для этого необходимо настроить шаблон Mustache для поддержки нескольких проходов, используя отдельные шаблоны для "пустот", или используя макрокоманду шаблона возврата модели, а не фактических значений. В любом случае необходимо "скрыть" эти пустоты, пока данные не будут извлечены, а затем заполнить их и отобразить соответствующую анимацию пользовательского интерфейса, чтобы избежать слишком сильного "дерганья" страницы.

Создание примера шаблона блога

Образец BlogTemplate демонстрирует возможности компоновки контента.

Вы можете создать шаблон в Developer Cloud Service, просмотреть макеты контента в шаблоне и протестировать возможности поиска:

1. Создайте шаблон типа BlogTemplate:

```
oac create-template MyBlogTemplate -f BlogTemplate
```

Новый шаблон MyBlogTemplate будет создан путем копирования BlogTemplate.

2. Откройте <http://localhost:8085/>, а затем последовательно нажмите **Шаблоны** и **MyBlogTemplate**.
3. Отображается список записей в блоге. Нажмите одну из них. Открывается страница "Сведения", на которой для рендеринга используются три различных макета контента.
4. Нажмите **Другие статьи этого автора**. Выполняется переход на страницу поиска.
5. Нажмите имя автора. Этот макет контента вызывает триггер и отображает статьи на правой стороне.

6. Снова перейдите на домашнюю страницу и протестируйте поиск.

Добавление сопоставлений макетов контента к шаблонам

После создания макетов контента можно добавить к локальным шаблонам в проекте Developer Cloud Service, добавив сопоставление макета контента.

Используйте команду `cec add-contentlayout-mapping`, чтобы добавить сопоставление макета контента для шаблона. Требуется тип контента, на котором основан макет контента (`-c`) и шаблон, для которого предназначено сопоставление (`-t`). Пример:

```
cec add-contentlayout-mapping Blog-Post-Detail-Layout -c Blog-Post -t
BlogTemplate
```

Сопоставление макета контента по умолчанию — это стиль по умолчанию для рабочего стола. Можно указать параметр `-s <layoutstyle>`, чтобы по имени указать другой стиль макета, например "Обзор" или "Сведения":

```
cec add-contentlayout-mapping Blog-Post-Detail-Layout -c Blog-Post -t
BlogTemplate -s Details
```

Также с помощью параметра `-m` можно задать сопоставление для мобильного устройства:

```
cec add-contentlayout-mapping Blog-Post-Detail-Layout -c Blog-Post -t
BlogTemplate -m
```

Тестирование макетов контента с помощью локальной тестовой оснастки

После добавления типов контента и сопоставлений макетов контента в шаблон макеты контента можно протестировать в локальной тестовой оснастке.

См. [Тестирование с использованием локальной тестовой оснастки](#).

Тестирование с помощью локальной тестовой оснастки

Запустите пользовательские компоненты, шаблоны и макеты контента в локальной тестовой оснастке, прежде чем импортировать их в Oracle Content Management.

Чтобы запустить локальную тестовую оснастку, выполните указанные ниже действия.

1. В окне терминала введите команду `cd cec`.
2. Введите команду `cec development &` или `cec deploy --server <server-name> &`
3. Откройте браузер по адресу `http://localhost:8085`, чтобы просмотреть компоненты, шаблоны и макеты контента, выполняемые в локальной тестовой оснастке.
4. Компоненты, шаблоны, темы и т. д. можно найти в следующих каталогах:
 - `cec/src/main/components`
 - `cec/src/main/templates`

- `cec/src/main/themes`

Импорт шаблонов с макетами контента в Oracle Content Management

После разработки и тестирования макетов контента из проекта Developer Cloud Service можно экспортировать шаблон с макетами контента, а затем импортировать шаблон в Oracle Content Management.

См. [Экспорт шаблона](#).

Разработка компонентов

Можно разрабатывать компоненты для использования на сайтах Oracle Content Management.

- [О компонентах](#)
- [О разработке компонентов](#)
- [Создание компонента](#)
- [Разработка пользовательских компонентов с помощью Developer Cloud Service](#)
- [Разработка переводимых компонентов для многоязычных сайтов](#)
- [Построение компонента H1 с помощью панели настроек](#)
- [Создание простого компонента с помощью HTML-шаблонов](#)
- [Сравнение локальных компонентов с удаленными](#)
- [Настройки рендеринга компонентов](#)
- [Реализация локального компонента](#)
- [Классы стилей для компонентов](#)
- [Порядок создания стилей встроенных компонентов](#)
- [Задание свойств компонентов](#)
- [Компоненты с рендерингом в задаваемых в строках фреймах](#)
- [Об идентификаторе экземпляра и структуре компонентов с рендерингом в задаваемых в строках фреймах](#)
- [Безопасность удаленных компонентов](#)
- [Регистрация удаленного компонента](#)
- [Удаление компонента](#)
- [Sites SDK](#)

О компонентах

Компонент — это контент определенного типа, который можно добавить на страницу сайта. Компоненты Oracle Content Management включают такие элементы, как абзац, заголовок, изображение, разделитель и т. д.

Oracle Content Management поддерживает компоненты следующих типов:

- Локальный компонент: файлы хранятся в Oracle Content Management
- удаленный компонент: файлы хранятся на удаленном сервере.

Можно настроить визуализацию локальных компонентов непосредственно на странице или во встроенной рамке на странице. Удаленные компоненты всегда визуализируются во встроенной рамке.

Oracle Content Management предоставляет набор компонентов по умолчанию для каждого шаблона (который также включает темы и сайты). Можно создавать новые компоненты и использовать эти компоненты по умолчанию в своих компонентах. После добавления компонента на сайт можно изменить настройки его свойств в соответствии со своими требованиями, указав контент страницы, шрифты и размеры шрифта, кадрирование и размещение изображений, а также другие стили. Доступные для редактирования настройки зависят от типа компонента.

Эти компоненты входят в состав Oracle Content Management.

Имя	Тип	Идентификатор
Заголовок	scs-title	scs-title
Абзац	scs-paragraph	scs-paragraph
Текст	scs-title	scs-text
Изображение	scs-image	scs-image
Галерея	scs-gallery	scs-gallery
Сетка галереи	scs-gallerygrid	scs-gallerygrid
Документ	scs-document	scs-document
Кнопка	scs-button	scs-button
Карта	scs-map	scs-map
Разделитель	scs-divider	scs-divider
Разделитель	scs-spacer	scs-spacer
YouTube	scs-youtube	scs-youtube
Социальная панель	scs-socialbar	scs-socialbar
Статья (пользовательский компонент)	scs-component	scs-comp-article
Заголовок (пользовательский компонент)	scs-component	scs-comp-headline
Список папок	scs-app	Список папок
Список файлов	scs-app	Список файлов
Диспетчер документов	scs-app	Диспетчер документов
Кнопка Facebook "Нравится"	scs-app	Кнопка Facebook "Нравится"
Кнопка Twitter "Подписаться"	scs-app	Кнопка Twitter "Подписаться"
Кнопка Twitter "Поделиться"	scs-app	Кнопка Twitter "Поделиться"
Кнопка Facebook "Рекомендовать"	scs-app	Кнопка Facebook "Рекомендовать"
Изображение и текст (пользовательский компонент)	scs-component	scs-comp-image-text

См. раздел "Организация содержимого страницы" в документе *Построение сайтов с помощью Oracle Content Management*.

О разработке компонентов

Разработка собственного пользовательского компонента позволяет разрабатывать составные элементы для встраивания в страницу сайта с использованием любой доступной технологии страницы. Это позволяет эффективно расширить список компонентов, поставляемых с Oracle Content Management.

Записи для всех зарегистрированных компонентов хранятся в каталоге компонентов, который представляет собой папку в Oracle Content Management.

Не используйте для шаблонов, тем, компонентов, сайтов и страниц сайтов следующие имена: authsite, content, pages, scstemplate_*, _comps, _components, _compsdelivery, _idcservice, _sitescloud, _sitesclouddelivery, _themes, _themesdelivery. Следующие имена можно использовать только для страниц сайтов (не для шаблонов, тем, компонентов и сайтов): documents, sites.

Типы компонентов

Каталог компонентов поддерживает компоненты следующих типов:

- Локальный компонент
- Локальный компонент, визуализированный во встроенной рамке
- Удаленный компонент

Тип компонента сохраняется как атрибут расширения "xScsAppType" папки компонента. Далее следуют допустимые значения.

Тип	Описание
Локальный компонент	Ожидается, что все зависимые ресурсы локальны.
Локальный компонент, визуализированный во встроенной рамке	Компонент обслуживается из того же домена, что и Oracle Content Management.
Удаленный компонент	Компонент обслуживается из удаленного местоположения.

Структура файлов компонентов

У каждого типа компонентов свой набор файлов, созданных в каталоге компонентов, в зависимости от способа их реализации.

Локальный компонент:

```
/Components/component-name
  appinfo.json
  _folder_icon.jpg
  assets
    settings.html
    render.js
```

Локальный компонент с использованием встроенной рамки:

```

/Components/component-name
  appinfo.json
  _folder_icon.jpg
  assets
    settings.html
    render.js
  js
    sites.min.js
    knockout.min.js
    jquery.min.js

```

Удаленный компонент:

```

/Components/component-name
  appinfo.json
  _folder_icon.jpg
  keys.json

```

Метаданные папки

Данные регистрации компонентов хранятся в метаданных папки. Указанные ниже свойства используются для уникальной идентификации компонента и его типа.

Свойство	Описание
имя приложения	Имя папки, которая обозначает имя компонента.
описание приложения	Описание папки.
guid приложения	Каждый компонент связан с GUID и хранится как атрибут расширения <code>xScsItemGUID</code> . GUID создается сервером при создании компонента.
тип приложения	Свойство определяет тип компонента. Оно сохраняется как атрибут расширения <code>xScsAppType</code> .

 **Примечание.**

Свойство `iconURL`, которое хранится как атрибут расширения `xScsAppIconUrl`, устарело.

Файл appinfo.json

Файл регистрации `appinfo.json` для каждого типа компонента содержит только данные, которые не доступны в метаданных папки. Свойства компонента, определенные в метаданных папки, не дублируются в файл `appinfo.json`.

Локальный компонент:

```
{
  "settingsData":{
    "settingsHeight":80,
    "settingsRenderOption"; "dialog",
    "settingsWidth":300,
    "componentLayouts":[],
    "triggers":[],
    "actions":[]
  },
  "initialData":{
    "customSettingsData":[]
  }
}
```

Локальный компонент, визуализированный во встроенной рамке:

```
{
  "endpoints": {
    "settings": {
      "height": "300",
      "width": "400"
    }
  }
  "initialData": {
    "customSettingsData": {}
  }
}
```

Удаленный компонент:

```
{
  "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": {}
  }
}
```

Нет зарегистрированных значений "url" для локальных компонентов. Они используют определенные файлы, содержимое которых можно изменять, но нельзя изменить местоположение или имя.

- Локальные компоненты используют файлы `assets/render.js` и `assets/settings.html`.
- Локальные компоненты, визуализированные во встроенной рамке, используют файлы `assets/render.html` и `assets/settings.html`.
- Удаленные компоненты используют указанные значения `"url"`.

Создание компонента

Можно создать пользовательские компоненты для использования в Oracle Content Management.

Oracle Content Management предоставляет множество предварительно определенных компонентов для использования при создании веб-сайта, а разработчик может создавать пользовательские компоненты с доступом к тем же функциям и возможностям. Компоненты этих типов включают локальные компоненты, компоненты с локальным рендерингом во встроенной рамке и компоненты с удаленным рендерингом во встроенной рамке.

Локальный компонент

При создании локального компонента предоставляется полнофункциональный образец компонента, который используется в качестве основы для создания собственного компонента. Выберите **По умолчанию**, чтобы создать версию Knockout компонента, или выберите **Шаблон**, чтобы создать версию на основе Mustache. См. раздел "Создание локальных компонентов или макетов" в документе *Построение сайтов с помощью Oracle Content Management*.

Локальный компонент со встроенной рамкой

При создании локального компонента со встроенной рамкой предоставляется полнофункциональный образец компонента, который используется в качестве основы для создания собственного компонента. Следует использовать те же инструкции, что и при создании локального компонента. Также выберите параметр **Обрабатывается в песочнице**, чтобы создать версию компонента со встроенной рамкой, которая сохраняется локально. См. [Компоненты с рендерингом в задаваемых в строках фреймах](#).

Удаленный компонент

При создании удаленного компонента, который использует встроенную рамку, выберите параметр **Обрабатывается в песочнице**.

Скопируйте созданные файлы на свой удаленный сервер и зарегистрируйте удаленный компонент. Протестируйте компонент, прежде чем включить его в опубликованный сайт.

См. разделы [Компоненты с рендерингом в задаваемых в строках фреймах](#) и [Регистрация удаленного компонента](#).

Разработка пользовательских компонентов с помощью Developer Cloud Service

Oracle Developer Cloud Service помогает разрабатывать шаблоны, темы и пользовательские компоненты для Oracle Content Management.

Интеграция Developer Cloud Service с Oracle Content Management предоставляет шаблон с инструментами для разработки шаблонов и компонентов. Кроме того, в нем представлены примеры модульных тестов, с которых можно начать. Интеграция включает репозиторий Git и инструменты, которые помогают разрабатывать шаблоны и компоненты, а также локальную тестовую оснастку для быстрой итеративной разработки шаблонов, тем и пользовательских компонентов.

Developer Cloud Service помогает выполнять следующие задачи:

- Настройка локальной среды разработки для использования экземпляра Oracle Content Management в целях локальной разработки и тестирования шаблонов, тем и компонентов
- Создание шаблонов и компонентов на основе примеров или начальных шаблонов, их выполнение в тестовой оснастке, их изучение и разработка шаблонов, тем и компонентов в среде Developer Cloud Service
- Импорт созданных шаблонов или компонентов из Oracle Content Management в среду Developer Cloud Service для управления источниками и дальнейшей разработки
- Экспорт шаблонов или компонентов из среды Developer Cloud Service для импорта в Oracle Content Management в целях использования на веб-сайтах
- Копирование существующего компонента
- Создание модульных тестов
- Оптимизация компонентов
- Развертывание компонентов в Oracle Content Management

В последующих разделах описывается, как использовать интеграцию Developer Cloud Service в целях разработки для Oracle Content Management.

- [Разработка пользовательских компонентов для Oracle Content Management](#)
- [Оптимизация компонентов \(уменьшение\) для повышения производительности](#)
- [Выполнение заданий непрерывной интеграции](#)

Разработка пользовательских компонентов для Oracle Content Management

Developer Cloud Service и локальный компьютер используются в целях разработки пользовательских компонентов для Oracle Content Management.

В последующих разделах описывается процедура разработки и тестирования компонентов Oracle Content Management с помощью Developer Cloud Service.

1. [Настройка инструментария OCE на локальном компьютере.](#)

2. [Вход в консоль Developer Cloud Service для Oracle Content Management.](#)
3. [Создание шаблона в Developer Cloud Service.](#)
4. [Добавление инструментария OCE в код проекта в новом репозитории GIT.](#)
5. [Разработка пользовательского компонента](#)
6. [Тестирование с помощью локальной тестовой оснастки](#)
7. [Запись и выполнение модульных тестов](#)

Разработка пользовательского компонента

Утилита командной строки `cec` используется для создания нового компонента, разработки компонента локально и последующего экспорта компонента в Oracle Content Management.

Создание компонента

Для локального создания и разработки пользовательского компонента используйте следующую команду:

```
cec create-component <component-name> -f <source>
```

Выберите одно из следующих значений параметра `source`:

- 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

Пример:

```
cec create-component MyLocalComponent1 -f local
```

Компонент создается в репозитории Git в каталоге `cec-components/src/main/components`.

Каталог `src/main/components` содержит предварительно заданный компонент `Sample-To-Do`. Все создаваемые компоненты сохраняются в этом каталоге.

Копирование компонента

Компонент в Developer Cloud Service можно скопировать с помощью команды `cec copy-component`:

```
cec copy-component <source> [<destination>]
```

Эта команда копирует существующий компонент с `<source>` в `<destination>`.

Экспортируйте или разверните компонент в Oracle Content Management

После разработки и тестирования на локальном сервере компонента можно экспортировать с помощью следующей команды. При этом создается ZIP-файл компонента. Вы можете вручную импортировать этот ZIP-файл компонента в Oracle Content Management.

```
cec export-component <component name>
```

Кроме того, компонент можно развернуть непосредственно в Oracle Content Management из Developer Cloud Service с помощью следующей команды:

```
cec deploy <component name>
```

Команда `cec deployAll` разворачивает все компоненты в `src/main/components`.

Импорт компонентов в Developer Cloud Service

Если имеется ZIP-файл компонента с сервера Oracle Content Management, его можно импортировать в Developer Cloud Service для дальнейшей разработки. Используйте следующую команду:

```
cec import-component <location of the component zip file>
```

Важно.

Исходный код для компонентов находится в разделе `src/main/components`. Не следует изменять файлы за пределами каталога `src/main/components`, так как они необходимы для работы локального сервера Oracle Content Management.

Запись и выполнение модульных тестов

Начните создание модульных тестов с примера модульного теста и выполните их для своих пользовательских компонентов Oracle Content Management.

Начало работы с примером модульного теста

Примеры Developer Cloud Service для Oracle Content Management содержат следующие файлы, которые полезны при создании модульных тестов:

- **src/test/unit:** содержит модульный тест для компонента `Sample-To-Do`. Служит примером для создания модульных тестов для кода JavaScript компонента, который включает модули RequireJS. Для тестирования модулей JavaScript используются платформы Mocha и Chai.
- **Index.html:** выполняет модульный тест. Он загружает Mocha, Chai и основной файл теста `test-main.js`.
- **Test-main.js:** загружает модуль модульного теста, `Sample-To-Do-Test`, и выполняет модульные тесты Mocha.
- **Sample-To-Do-Test.js:**
 - Определяет фактические тесты. Он загружает исходный код компонента, `components/Sample-To-Do/assets/render.js`, с помощью RequireJS.
 - Тестирует методы `add()`, `delete()`, `title()` и `placeholder()`.

Создание модульных тестов для своих компонентов

Чтобы создать модульный тест для пользовательского компонента, выполните указанные ниже действия.

1. Создайте модульный тест, подобный сценарию `Sample-To-Do-Test.js`.
2. Загрузите созданный тест в `test-main.js`.

Локальное выполнение модульного теста

Чтобы выполнить модульный тест локально, выполните указанные ниже действия.

1. Клонировать репозиторий Git локально.
2. В окне терминала на локальном компьютере введите `npm install`.
3. Введите `npm start`.
4. Введите `npm test` или откройте в браузере страницу с адресом `http://localhost:8085/unit/`.

Оптимизация компонентов (уменьшение) для повышения производительности

Повысьте производительность компонентов и уменьшите размер выгружаемых файлов, оптимизировав код JavaScript с помощью Developer Cloud Service.

Оптимизация компонентов

Пример оптимизации исходного кода компонента (дополнительно) представлен для компонента `Sample-Text-With-Image`. Для уменьшения и объединения исходного кода можно использовать оптимизатор `RequireJS`. При уменьшении компонента удаляются ненужные пробелы в коде JavaScript, что повышает производительность и уменьшает размер выгружаемых файлов. Оптимизация включает в себя следующие действия:

- Уменьшение кода JavaScript.
- Сжатие CSS.
- Объединение кода JavaScript, HTML и CSS в один файл `render.js`.

Эта оптимизация сокращает размер выгружаемых файлов и повышает производительность.

Включение оптимизации для компонента

В компоненте `Sample-Text-With-Image` оптимизация уменьшает и объединяет все файлы, отмеченные как зависимые ресурсы в файле `render.js` (`template.html`, `data-defaults.js` и `design.css`) в файл `render.js`. Другие компоненты, которые уже поддерживают оптимизацию:

- `Sample-Folder-List`
- `Sample-File-List`
- `Sample-Documents-Manager`
- `Sample-Process-Start-Form`
- `Sample-Process-Task-List`
- `Sample-Process-Task-Details`

Чтобы включить оптимизацию для локального компонента, выполните указанные ниже действия (обязательные).

1. Скопируйте файл `gulpfile.js` из компонента `Sample-Text-With-Image` в свой компонент.
2. Исправьте имена модулей в файле `gulpfile.js` для зависимых ресурсов, указанных в файле `render.js`. См. комментарии в файле `gulpfile.js`.

Примечание.

Если есть компонент с подпапкой `css` папки `assets`, переименуйте папку `css` в `styles`, чтобы избежать ошибок во время оптимизации.

Выполнение заданий непрерывной интеграции

Задачи непрерывной интеграции можно выполнять, чтобы поддерживать сайт в актуальном состоянии.

См. wiki-страницу "Начало работы с разработкой пользовательских компонентов СЕС" в Developer Cloud Service. Для непрерывной интеграции можно настроить задание построения.

Разработка переводимых компонентов для многоязычных сайтов

Разработчики пользовательских компонентов могут указать, какие строки в пользовательском компоненте должны принимать участие в переводе страниц многоязычных сайтов (MLS).

Для этого необходимо ввести свойство `nls` на верхнем уровне при сохранении данных в объект `customSettingsData`.

Пример:

```
SitesSDK.setProperty('customSettingsData', {
  'nls': {
    linkText: 'More...'
  }
});
```

При создании задания перевода Oracle Content Management проверяет свойства верхнего уровня объекта `customSettingsData` и экспортирует весь объект `nls` для каждого экземпляра пользовательского компонента на странице. Переводчики переводят эти значения, которые затем можно импортировать обратно на сайт.

После импорта перевода сайта правильная версия объекта `nls` возвращается в объект `customSettingsData` для переведенной локали.

Например, если перевести сайт на французский язык, а затем отобразить страницу на французском языке, значение объекта `nls` в объекте `customSettingsData`, переданном пользовательскому компоненту, обновляется до следующего выражения:

```
{
  'nls': {
    'linkText': 'Plus...'
  }
},
```

Формат объекта `nls` в `customSettingsData` должен быть ограничен парами имя/значение. Это помогает при переводе и обеспечивает правильное применение переведенных значений к базовым значениям на сайте при рендеринге страницы в переведенной локали.

Построение компонента H1 с помощью панели настроек

Можно создать минимальный компонент Oracle Content Management с простыми HTML-шаблоном и CSS. Этот компонент H1 имеет простую панель настроек и запись темы в `design.json`, чтобы позволить другим пользователям Oracle Content Management выбирать из трех встроенных стилей при использовании компонента в редакторе.

При создании нового компонента предоставляется набор предварительно заданных файлов, которые уже готовы к работе. Предварительно заданные файлы охватывают большую часть функциональных возможностей компонента в продукте. Можно изменить предварительно заданный код, чтобы создать собственный компонент, которому для получения конечного результата требуется лишь небольшое подмножество предварительно заданного кода.

Компонент H1 можно создать с помощью панели настроек за пять шагов:

1. [Создание нового локального компонента](#)
2. [Построение базового компонента H1](#)
3. [Добавление CSS для компонента](#)
4. [Добавление панели настроек для изменения текста заголовка](#)
5. [Обновление темы для выбора стиля компонента H1 другими пользователями](#)

Создание нового локального компонента

С помощью Oracle Content Management создайте локальный компонент, который можно сразу поместить на страницу. Это отправная точка создания любого нового компонента.

Чтобы создать новый локальный компонент, выполните указанные ниже действия.

1. На боковой панели навигации нажмите **Разработчик**, затем нажмите **Смотреть все компоненты**.
2. Выберите **Создать > Создать локальный компонент**.
3. Введите имя, например `H1_Component`, и, если необходимо, описание.
4. Нажмите **Создать**, чтобы создать новый компонент.

После успешного создания компонент должен отображаться в списке компонентов на странице компонентов, а также в палитре **Добавить > Пользовательский компонент** для любого создаваемого сайта. Выполните указанные ниже действия для проверки создания компонента (контрольная точка 1).

1. Создайте новый сайт, используя любой предварительно заданный шаблон. Например, создайте сайт с именем `ComponentTest`, используя шаблон `StarterTemplate`.
2. Выберите пункт **Редактировать** и создайте обновление сайта, чтобы открыть его в редакторе.
3. Отредактируйте страницу на созданном сайте.
4. Нажмите кнопку **Добавить (+)** на левой панели и для списка пользовательских компонентов выберите **Пользовательский**.
5. На палитре пользовательских компонентов выберите элемент **H1_Component** и перетащите его на страницу.

Теперь вы должны видеть рендеринг по умолчанию созданного локального компонента.

6. Выберите контекстное меню для компонента.
7. В раскрывающемся меню выберите **Настройки**.

Настройки можно изменить, чтобы узнать, как изменится рендеринг предварительно заданных компонентов.

Можно изменить предварительно заданные файлы, чтобы создать новый пользовательский компонент.

Построение базового компонента H1

Для создания компонента H1 можно удалить большую часть содержимого предварительно заданных файлов. Этот компонент отображает текст заголовка, который вводится при создании параметра `viewModel`. Позднее можно предоставить настройки и стили для компонента.

Чтобы просмотреть структуру локального компонента, выполните указанные ниже действия:

1. С помощью настольного приложения Oracle Content Management для синхронизации найдите компонент и синхронизируйте его с файловой системой.
 - В последней версии настольного приложения для синхронизации выберите команду **Начать синхронизацию** или **Выбрать папки для синхронизации**.
 - Если настольного приложения для синхронизации нет, компонент можно выбрать на вкладке **Компоненты** в Oracle Content Management, а затем развернуть его, чтобы просмотреть файлы.
2. Если перечислить файлы, относящиеся к компоненту, можно увидеть следующие файлы:
 - Файлы компонента в папке `assets`:
 - `render.js`
 - `settings.html`
 - `appinfo.json`: файл JSON с описанием компонента.
См. [О разработке компонентов](#).
 - `folder_icon.jpg`: значок, отображаемый в каталоге компонентов.

Чтобы создать компонент H1, выполните указанные ниже действия.

1. Откройте файл `appinfo.json` и замените его содержимое следующими строками:

```
{
  "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. Откройте файл `render.js` в папке `assets` в предпочтительном текстовом редакторе.
3. Измените содержимое файла `render.js` на следующие строки:

```

/* 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. В папке `assets` создайте новый файл `render.html`, который будет простым HTML-шаблоном компонента.
5. Используйте в файле `render.html` следующее содержимое:

```
<h1 data-bind="text: headingText()">
</h1>
```

Папка `assets` компонента теперь содержит три файла.

- `render.html`
- `render.js`
- `settings.html`

Добавьте новый компонент H1 на свою страницу (контрольная точка 2).

Добавление CSS для компонента

Можно добавить таблицу CSS, которая предоставит стиль по умолчанию для компонента.

Чтобы добавить таблицу CSS, выполните указанные ниже действия.

1. Добавьте файл `design.css` в папку `assets` компонента со следующим контентом:

```
.h1-component-default-style .scs-component-content {
font-family: "Helvetica Neue", "Helvetica", "Arial", sans-serif;
font-size: 24px;
color:red;
font-weight: normal; }
```

2. Добавьте файл `appinfo.json`, чтобы объявить префикс класса стилей, который будет использоваться для задания стиля компонента. Если добавлен параметр `styleClassName` компонента `h1-component`, при перетаскивании компонента на страницу используется стиль по умолчанию `h1-component-default-style`. Новый контент файла `appinfo.json`:

```
{
  "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. Обновите `render.js`, чтобы загрузить файл CSS, заменив строку 2 (`define(['knockout', 'jquery', 'text!./render.html'], function(ko, $, template) {})` приведенной ниже строкой.

```

define(['knockout', 'jquery', 'text!./render.html', 'css!./design.css'], function(ko, $, template, css) {

```

Примечание.

Поскольку файл `appinfo.json` был изменен, необходимо обновить окно браузера и снова добавить компонент на страницу, чтобы получить изменения.

Проверьте, что компонент теперь принимает данные по умолчанию из файла `appinfo.json`.

Добавление панели настроек для изменения текста заголовка

Обновите файл `settings.html`, чтобы предоставить панель настроек, которую можно использовать для задания текста компонента H1.

Чтобы добавить панель настроек для изменения текста заголовка, выполните указанные ниже действия.

1. Обновите файл `settings.html`, чтобы получить следующий контент:

```

<!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>
```

2. Выберите компонент в Site Builder и нажмите **Настройки**.

Панель настроек имеет небольшой размер и предназначена для встраивания в верхнюю часть панели настроек компонента. Найдите раздел "Heading Text".

Чтобы убедиться в возможности изменения текста заголовка (контрольная точка 4), выполните указанные ниже действия.

При изменении текста заголовка на панели настроек компонент обновляется для отображения нового текста.

Обновление темы для выбора стиля компонента H1 другими пользователями

Стили для компонента с темой можно зарегистрировать, чтобы другие пользователи могли переключаться между стилями, которые вы предоставляете для своего компонента, на панели настроек или на вкладке **Стиль**.

Порядок обновления темы, чтобы другие пользователи могли выбрать стиль компонента:

1. Добавьте еще несколько стилей в файл `design.css` своего компонента. Перед именем каждого стиля указывайте зарегистрированный префикс `styleClassName` компонента, как определено в файле `appInfo.json`. Для этого компонента такой префикс имеет вид `h1-component`.

Добавлены еще два стиля: `h1-component-gothic-style` и `h1-component-courier-style`.

Новое содержимое файла `design.css`:

```
.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; }
```

2. Зарегистрируйте свои стили в файле `design.json` темы. Этот файл можно найти в теме, используемой сайтом. Перейдите к файлам тем в папке проектов, а затем к папке по умолчанию и добавьте в файл `design.json` раздел для своего компонента.

Приведенный ниже **полужирный** текст служит примером добавления:

```

"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": [

```

Добавленные имена в данном фрагменте файла `design.json` ("Plain", "Courier", "Gothic") появятся на панели настроек компонента на вкладке стиля, как показано ниже. В случае их выбора к компоненту применяются соответствующие стили ("*h1-component-default-style*", "*h1-component-courier-style*", "*h1-component-gothic-style*").

Создание простого компонента с помощью HTML-шаблонов

Можно создавать компоненты Oracle Content Management, которые используют HTML и CSS с шаблонами Mustache JS для рендеринга контента на основе данных, введенных на панели настроек.

Предоставляемая панель настроек позволяет редактировать данные, которые будут визуализироваться в шаблоне HTML.

В примерах для данной процедуры используются шаблоны Mustache JS.

С помощью шаблонов HTML можно создать простой компонент за два шага.

1. [Создание HTML-компонента](#)
2. [Добавление в HTML-компонент данных, предоставленных пользователем](#)

Создание HTML-компонента

В каталоге компонентов Oracle Content Management можно создать новый локальный компонент, а затем построить HTML-компонент, добавив и изменив файлы в папке активов.

Чтобы создать и построить HTML-компонент, выполните указанные ниже действия.

1. Создайте компонент и назовите его HTML Component.

См. [Создание нового локального компонента](#).

2. Добавьте файл `mustache.min.js` в папку `assets` в HTML-компоненте.
3. В папке `assets` создайте новый файл `render.html` с HTML-кодом для тела компонента. В этом примере создается файл `render.html` со следующим контентом:

```
<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. В папке `assets` создайте новый файл `design.css` с таблицами CSS для компонента. В этом примере добавьте в файл `design.css` следующие строки:

```
.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. Откройте файл `render.js` в папке `assets` и измените его содержимое на примененное ниже. Независимо HTML-кода и таблиц CSS, использованных в предыдущих действиях, следующий файл `render.js` преобразует HTML-код и таблицы CSS в страницу:

```

/* 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
    , $.proxy(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;
});

```

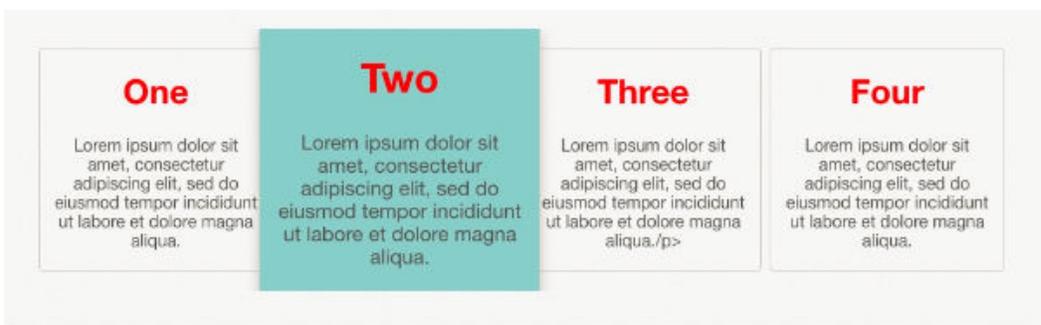
Чтобы убедиться в построении HTML-компонента (контрольная точка 1), выполните указанные ниже действия.

1. Убедитесь, что в папке `assets` в компоненте содержатся следующие пять файлов:
 - `design.css`
 - `mustache.min.js`
 - `render.html`
 - `render.js`
 - `settings.html`
2. Добавьте новый HTML-компонент на страницу своего тестового сайта. В редакторе в режимах редактирования и предварительного просмотра компонент должен отображаться на странице следующим образом:

Режим редактирования



Режим предварительного просмотра



Добавление в HTML-компонент данных, предоставленных пользователем

Можно добавить панель **Пользовательские настройки**, которая предоставляет поля для всех элементов данных, определенных в HTML-компоненте. Затем пользователь может ввести значения для экземпляра компонента на соответствующей странице панели **Пользовательские настройки**.

Чтобы добавить в HTML-компонент данные, предоставленные пользователем, выполните указанные ниже действия.

1. Обновите HTML-шаблон в файле `render.html`, чтобы включить все необходимые элементы данных. В следующем примере используется язык для создания шаблонов Mustache JS, поэтому элементы данных необходимо добавлять с помощью синтаксиса `{{ и }}`, например следующим образом:

```
<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>
  <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. Укажите значения по умолчанию для следующих полей данных, изменив содержимое файла `appInfo.json` в HTML-компоненте:

```
{
  "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. Добавьте панель **Пользовательские настройки**, которая будет искать значения данных в файле HTML-шаблона `render.html`, чтобы создать для пользователя настройки, которые можно изменить в редакторе. Измените содержимое файла `settings.html` на следующие строки:

```
<!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));
            }
        };

        // 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 and can start saving data
                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>

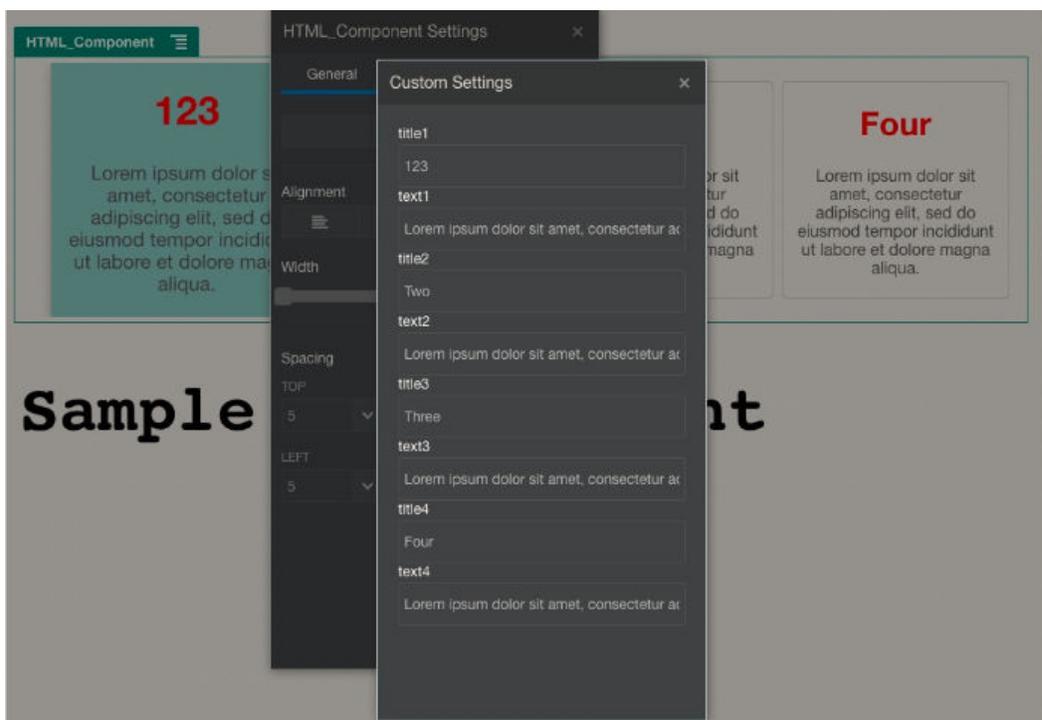
```

Примечание.

Поскольку файл `appinfo.json` был изменен, необходимо обновить окно браузера и снова добавить компонент на страницу, чтобы получить изменения.

Чтобы убедиться в том, что теперь компонент получит данные по умолчанию из файла `appinfo.json` и можно изменить значения на панели **Пользовательские настройки** (контрольная точка 2), выполните указанные ниже действия.

1. После изменения файла `appinfo.json` необходимо обновить окно браузера и снова добавить компонент на страницу, чтобы получить изменения.
2. Откройте панель **Пользовательские настройки** в HTML-компоненте и введите данные.



3. Убедитесь, что компонент Oracle Content Management использует код HTML и CSS с шаблонами Mustache JS для визуализации контента на основе данных, введенных на панели **Пользовательские настройки**.

Сравнение локальных компонентов с удаленными

Локальные и удаленные компоненты реализуются по-разному.

В таблице ниже перечислены различия в способах реализации компонентов. Это может помочь определить, что нужно использовать: локальный компонент или удаленный компонент, визуализированный во встроенной рамке.

Реализация локального компонента	Реализация удаленных компонентов
Интеграция сложного пользовательского интерфейса, ориентированного на контент, в Oracle Content Management	Интегрирует логику приложения в Oracle Content Management
Выполняет код JavaScript для рендеринга контента страницы	Для визуализации контента на странице используется HTML-тег <code><iframe></code>
Использует любой стек технологий JavaScript	Может использовать любую технологию, а не только JavaScript
Использует стек JavaScript в Oracle Content Management и может повторно использовать компоненты Oracle Content Management	Не интегрируется со стекком JavaScript в Oracle Content Management
Добавляет зависимости активов из документов /assets, хранящихся на сервере Oracle Content Management	Требуются URL-адреса HTTPS и соответствующие сертификаты для доступа ко всем активам
Размещается с шаблоном и публикуется независимо от темы Oracle Content Management	Для конечных точек URL-адреса требуется размещенный сервер среднего яруса
Поскольку код выполняется на странице, нарушение кода компонента может привести к разрыву страницы	Страница визуализируется независимо от контента встроенных рамок, поэтому страница не разрывается в случае сбоя данного компонента
Медленная визуализация компонента может замедлить визуализацию страницы	Страница загружается независимо от контента встроенных рамок, поэтому страница загружается максимально быстро, а затем загружается контент встроенных рамок

Примечание.

Хотя визуализация локальных и удаленных компонентов отличается, панель "Настройки" реализуется в Site Builder таким же образом. Панель "Настройки" для компонентов обоих типов визуализируется с помощью встроенных рамок и использует один и тот же комплект JavaScript SDK для взаимодействия между границам и между доменами.

Настройки рендеринга компонентов

Для рендеринга настроек компонента можно использовать URL-адрес настроек компонента и параметры рендеринга настроек компонента.

URL-адрес настроек компонента

URL-адрес настроек компонента визуализируется с помощью встроенной рамки и вызывается с параметрами, чтобы позволить перенести на страницу определенные настройки фактического компонента. URL-адрес настроек имеет следующий формат:

```
{Component Settings URL}?instance=<app-  
instance>&width=<width>&currCompId=<id of the app associated with the  
settings panel>&locale=<locale>
```

Имя	Тип	Описание
Component Settings URL	URL	URL-адрес настроек компонента
width	Номер	Ширина встроенной рамки настроек в пикселях
currCompId	Строка	Идентификатор текущего компонента, отредактированный на панели "Настройки"
locale	Строка	Текущий языковой стандарт сайта хоста (Site Builder). Формат: <language>_<dialect>. Пример: En_us.

Параметры рендеринга настроек локального компонента

Есть три варианта рендеринга встроенной рамки на панели "Настройки" для локального компонента в зависимости от размера и сложности встроенной рамки. Каждый вариант указан в свойстве `settingsRenderOption`.

Примечание.

Эти варианты доступны только для локальных компонентов. Панель "Настройки" удаленных компонентов всегда визуализируется в диалоговом окне.

- `inline`
 - Используйте этот вариант, только если пользователю требуется ввести несколько небольших свойств.
 - Встроенная рамка вставляется во вкладку **Общие** вместо обычной кнопки для перехода к панели "Настройки".
 - Обеспечивает наиболее интегрированное решение, требующее минимального количества нажатий кнопки мыши пользователем, но при этом оно имеет ограниченное пространство.

- `panel`
 - Используйте этот вариант, если есть более длинный список свойств, но они все равно могут быть разумно отображены в пределах 300 пикселей стандартной панели "Настройки".
 - Встроенная рамка сдвигается в область просмотра, и появляется кнопка "Назад" для возврата на вкладку "Общие".
 - Предоставляет интегрированное решение, позволяющее взаимодействовать со встроенной рамкой на странице.
- `dialog`
 - Режим по умолчанию для обработки более общих макетов настроек, требующих сложного пользовательского интерфейса.
 - Встроенная рамка отображается в модальном диалоговом окне на странице.

Реализация локального компонента

Экземпляр компонента создается объектом "фабрика компонентов"

Запись объекта "фабрика компонентов" в файле регистрации компонента: `initData.componentFactory`. Это ссылка на файл JavaScript, который RequireJS использует для загрузки фабрики компонентов. Он должен вернуть объект JavaScript, который реализует интерфейс `componentFactory`

Объект "фабрика компонентов" должен реализовать API-интерфейс `customComponentFactory.createComponent(args)`, который создает каждый экземпляр компонента.

`args` содержит следующие настройки:

- `SitesSDK`: пакет Sites SDK Oracle Content Management.
- `ID`: уникальный идентификатор (GUID) компонента, добавленного на страницу.
- `viewMode`: текущий режим рендеринга страницы. Режиму редактирования страницы соответствует значение "Edit". Предварительному просмотру страницы соответствует значение "Navigate" (Навигация). Во время выполнения, т. е. когда сайт публикуется, значение не определено. Можно предоставить различные варианты реализации, в зависимости от того, какие функциональные возможности должны быть доступны для каждого режима. Например, ссылки не должны быть активными, если страница находится в режиме редактирования.

Примечание.

Не обязательно использовать JQuery или Knockout для своего компонента, но если требуется использовать функции Oracle Content Management, такие как вложенные компоненты, необходимо использовать версию Knockout, предоставленную средой Oracle Content Management. Эта версия Knockout имеет расширенную регистрацию компонентов и обработчики, которые в противном случае были бы недоступны.

Для самого компонента комплект SDK передается при создании экземпляра компонента, чтобы компонент мог взаимодействовать с жизненным циклом страницы. Функции жизненного цикла страницы должны быть реализованы компонентом, и они вызываются средой Oracle Content Management для рендеринга компонента на странице.

Для реализации компонента предоставляются обязательные и дополнительные API-интерфейсы.

Обязательные API-интерфейсы

`customComponent.render(container)`: запрашивает вставку компонента в предоставленный элемент контейнера DOM.

- `container`: элемент контейнера DOM для HTML-кода пользовательского компонента.

Дополнительные API-интерфейсы

`customComponent.dispose()`: вызывается при удалении компонента со страницы. Позволяет компоненту удалить ненужные ресурсы.

Классы стилей для компонентов

Можно создать предварительно определенный список стилей, которые пользователи могут применить к компоненту.

Предварительное определение списка стилей для компонента аналогично определению классов стилей для компонентов, предоставляемых Oracle Content Management, посредством файлов проекта темы. Пользовательские классы стилей для компонента именовются в файле `appinfo.json`.

Дополнительные стили определяются в файлах `design.css` и `design.json`. Файл `json` обеспечивает сопоставление имени, которое будет отображаться в пользовательском интерфейсе, с фактическим базовым именем класса `css`, а файл `css` предоставляет сведения для каждого класса стилей.

Файл `design.json` имеет следующую структуру для компонентов:

```
{
  "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": []
}
}
}

```

Каждая запись "styles":[] может содержать список стилей для конкретного компонента. Запись "name" может быть ссылкой на встроенную локализованную строку или указанным значением, предназначенным использования. Например, компонент Title (Заголовок) содержит следующие стили по умолчанию:

```

{
  "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"
  }
]
}

```

Значения имен сопоставляются с фактическими словами для отображения в пользовательском интерфейсе, например следующим образом:

```

"COMP_STYLE_FLAT": "Flat",
"COMP_STYLE_HIGHLIGHT": "Highlight",
"COMP_STYLE_DIVIDER": "Divider",

```

Файл `css` предоставляет определения значений классов:

```
.scs-title-default-style {
  color: #333333;
  display: block;
  font-family: "Helvetica Neue", "Helvetica", "Arial", sans-serif;
  font-size: 24px;
  font-weight: normal;
}
```

Например, в файл `design.json` темы можно добавить записи для своего компонента на основе значения `initialData.componentId`, определенного в файле `components.json`:

```
"componentId": "news-article"
```

Соответствующие записи в файле `design.json` будут иметь вид:

```
"componentStyles": {
  "news-article": {
    "styles": [{
      "name": "News Article 1",
      "class": "news-article-default-style"
    },
    {
      "name": "News Article 2",
      "class": "news-article-style-1"
    }
  ]
},
```

Соответствующие записи в файле `design.css` будут иметь вид:

```
.news-article-default-style .scs-image {...}
.news-article-style-1 .scs-image {...}
```

Порядок создания стилей встроенных компонентов

Создайте собственный внешний вид, чтобы задать стиль встроенных компонентов в Oracle Content Management, переопределив и расширив встроенные стили.

Встроенные компоненты получают параметры визуального стиля из двух мест:

- `comp.css` — встроенный CSS-файл, который определяет базовый вид каждого компонента;
- `design.css` — это CSS-файл, который является частью темы, используемой сайтом.

В файле `design.css` можно переопределить и расширить встроенные стили `comp.css`, чтобы создать собственный стиль. В теме файл `design.css` находится в каталоге `designs/default`.

В последующих разделах описаны классы в файле `comp.css`, общие для всех встроенных компонентов, и представлен обзор определения темы:

- [Основы создания стилей компонентов](#)
- [Стили для компонентов](#)
- [Задание свойств компонентов](#)

ОСНОВЫ СОЗДАНИЯ СТИЛЕЙ КОМПОНЕНТОВ

У всех встроенных компонентов сайтов одинаковая структура класса CSS.

Каждый компонент имеет следующие три класса CSS, примененные к крайнему элементу `<div>`:

```
scs-component scs-type design-style
```

type — это тип компонента (например, `image`, `gallery` или `divider`). *design-style* — это выбранный класс стиля компонента, как определено в файле темы.

Основы стилей темы

В файле `design.json` темы перечисляются все стили (`frame`, `shadow`, `highlighted` и т. д.), которые можно применить к компонентам всех типов (таким как `button` или `image`). У каждого стиля есть `display name` и `class name`. Отображаемое имя отображается на вкладке **Стиль** панели **Настройки**. Имя класса указывает на селектор CSS в файле `design.css` темы. Например, запись для компонента `button` выглядит следующим образом:

```
"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"
  }
  ]
},
```

Имена встроенных компонентов переводятся, поэтому отображается ключ для получения имени стиля из пакета ресурсов. Если добавить на страницу компонент `button`, а затем выбрать **Простой** стиль на панели **Настройки > Стиль**, файл `design.json` связывает отображаемое имя **Простой** (ключ `COMP_STYLE_SIMPLE`) с именем класса `scs-button-style-3`. Элемент `button` визуализируется со следующими классами:

```
scs-component scs-button scs-button-style-3
```

Если для заданного компонента стиль не выбран, используется стиль по умолчанию `scs-type-default-style`. В предыдущем примере кнопка визуализируется со следующими классами:

```
scs-component scs-button scs-button-default-style
```

Стиль `scs-component-content`

Для каждого встроенного компонента, в элементе `scs-component <div>`, упомянутом ранее, существует элемент `<div>` с CSS-классом `scs-component-content`. Другими словами:

```
scs-component scs-type design-style  
scs-component-content
```

В файле `design.css` класс `scs-component-content` часто используется для стилизации "рамки" вокруг компонента (например, для применения границы или тени).

Следует отметить, что во встроенном файле `comp.css` общий класс `scs-component-content` определяется со свойствами `position: relative` и `display: inline-block` помимо других свойств CSS.

Хотя стиль `scs-component-content` полезен для стилизации "рамки" вокруг каждого компонента, для полного определения стиля компонента необходимы классы, характерные от компонента. См. [Стили для компонентов](#).

Стили для компонентов

К изображениям, кнопкам, документам, абзацам, заголовкам, картам и другим компонентам можно применить определенные стили.

Компонент "Изображение"

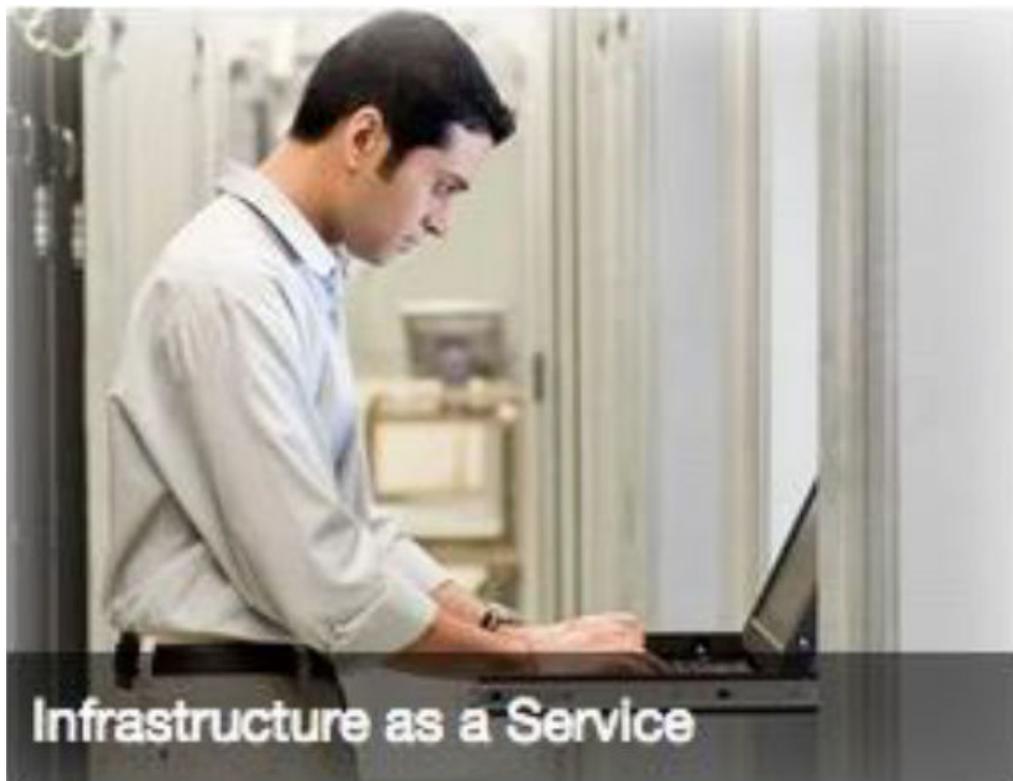
Компонент "Изображение" имеет следующую структуру классов CSS под классом `scs-component-content`:

```
scs-image-container  
scs-image-link  
scs-image-image  
scs-image-caption
```

Класс `scs-image-image` применяется к самому тегу ``. Класс `scs-image-caption` используется для определения стиля надписи, если она присутствует.

Класс `scs-image-link` присутствует, только если к изображению прикреплена ссылка. Ни он, ни класс `scs-image-container`, обычно не требуют применения специальных стилей.

По умолчанию подпись к изображению отображается в виде полупрозрачного наложения, растянутого вдоль нижней части изображения.



```
.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;  
}
```

Чтобы разместить подписи в верхней части изображения и изменить цвета, добавьте дополнительный стиль для компонента "Изображение" в файл `design.json`, а затем определите CSS для него в файле `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;  
}
```

Компонент "Кнопка"

Компонент "Кнопка" имеет следующую структуру классов:

```
scs-button-button  
  scs-button-text
```

Класс `scs-button-button` — это элемент `<div>`, который можно нажимать, в стиле кнопки. Класс `scs-button-text` используется для создания стиля текста внутри кнопки.

Например, протестируйте изменение внешнего вида компонента "Кнопка", добавив для него дополнительный стиль в файл `design.json`, а затем определите CSS для него в файле `design.css`.

Button Linear Color

```
.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;
}
```

Button With Radial Color

```
.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;
}
```

Документ

Компонент "Документ" имеет следующую структуру классов:

```
scs-document-container
  scs-document-cap
    scs-document-title
    scs-document-desc
```

Класс `scs-document-container` охватывает средство просмотра документов и обычно не имеет стиля.

Галерея

Компонент "Галерея" имеет один класс, который охватывает базовый компонент ползунка JSSOR:

```
scs-gallery-container
```

В ползунке JSSOR используется несколько классов, для которых также можно создать стили:

```
jssorb14 (navigator)
jssora02l (left arrow)
jssora02r (right arrow)
jssort07 (thumbnails)
```

Сетка галереи

Классы, используемые для компонента "Сетка галереи", зависят от макета и обрезки, выбранных на панели **Настройки**:

```
scs-gallerygrid-container scs-gallerygrid-layout
  scs-gallerygrid-cell
    scs-image (multiple)
```

В зависимости от настроек обрезки и макета, выбранных для сетки галереи, значение `layout` будет растягиваться, обрезаться, подгоняться или передаваться в потоке.

Класс `scs-gallerygrid-cell` присутствует только для макетов столбцов.

Социальная панель

Компонент "Социальная панель" имеет следующую структуру классов:

```
scs-socialbar-container
  scs-socialbar-icon
```

Класс `scs-socialbar-icon` применяется к каждому тегу `` на социальной панели.

Абзац

Компонент "Абзац" имеет только один класс, охватывающий фактический текст абзаца:

```
scs-paragraph-text
```

Например, чтобы текст, добавляемый вами в компонент "Абзац", выглядел как текст, гравированный на металле, добавьте дополнительный класс стилей в файл `design.json`, а затем определите CSS для него в файле `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;  
}
```

Или, если требуется сделать что-то интересное, используйте такой же пример.



```
.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;
}

```

Заголовок

Компонент "Заголовок" также имеет только один класс, который охватывает фактический текст:

```
scs-title-text
```

Карта

Компонент "Карта" имеет один класс, который охватывает представление карты:

```
scs-map-content
```

У этого класса обычно нет стиля.

Локальный пользовательский компонент

Локальный пользовательский компонент имеет только один класс, охватывающий фактический компонент:

```
scs-custom-component-wrapper
```

У вас есть полный контроль над стилями CSS, которые необходимо использовать для визуализации пользовательского представления для пользовательского локального компонента. Локальный компонент визуализируется встроенным; то есть можно напрямую применять стили CSS, определенные в теме или в файле `design.css`.

Удаленный пользовательский компонент

У удаленного пользовательского компонента только один класс, охватывающий его рамку:

```
scs-app-iframe-wrapper
```

Помимо применения стилей CSS, определенных в настраиваемом удаленном компоненте, Sites SDK можно использовать для извлечения файла `design.css` с сайта хоста.

```

// 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);
}
}
});
```

Таким образом, можно сделать так, чтобы компонент наследовал стиль главного устройства.

Разделитель

Несмотря на то, что для компонента "Разделитель" не существует классов, характерных для компонентов, к тегу `<hr>` можно применить стиль.

Например, можно создать разделитель с точками:

```
.design-style .scs-divider hr {
border-top: 1px dotted #333333;
}
```

Видео, YouTube, Spacer

Для компонентов "Видео", YouTube и Spacer не существует классов компонентов.

Задание свойств компонентов

Можно настроить свойства компонента для использования на сайте.

Компоненты — это отдельные части веб-страницы, которые содержат текст, заголовки, изображения, кнопки, разделители, карты, галереи, видео и т. д. При создании проекта для темы необходимо также указать настройки по умолчанию для компонентов каждого типа. У каждого компонента есть настройки, определяющие его внешний вид и поведение, например размер, выравнивание, промежутки, цвет и границы. Такие настройки зависят от компонента. Можно также выбрать, могут ли свойства компонента изменяться пользователями после того, как он станет доступен на сайте в новой теме.

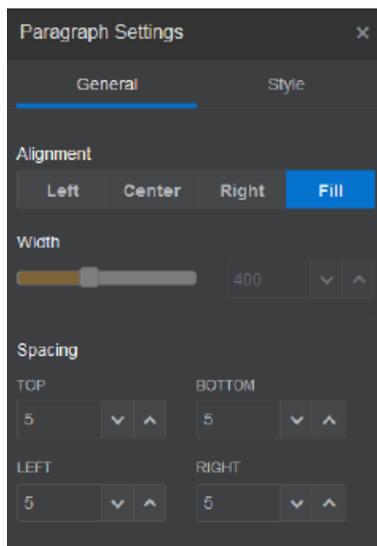
В качестве примера, эти действия иллюстрируют задание настроек для компонента "Абзац":

1. Открыв сайт разработки в режиме редактирования, выберите страницу с компонентом "Абзац" или добавьте компонент "Абзац".
2. Выберите компонент "Абзац", а затем нажмите  в углу компонента и выберите **Настройки**.

Отображается панель **Настройки**, на которой можно выбрать настройки компонента.

 **Примечание.**

Настройки зависят от типа компонента. При работе с локальными или удаленными пользовательскими компонентами отображается ссылка **Пользовательские настройки**.



3. Например, можно изменить настройки выравнивания, ширины, интервала, шрифтов, стиля, цвета, размера и т. д.
Внешний вид компонента изменяется в соответствии с новыми настройками.
4. По завершении нажмите . Ваши настройки применяются к странице.

Компоненты с рендерингом в задаваемых в строках фреймах

Компоненты, визуализируемые во встроенных рамках, можно указать на страницах Oracle Content Management, зарегистрировав и добавив компоненты с внешних серверов (называются **удаленными компонентами**), а также выбрав вариант встроенной рамки при создании локального компонента. Компонент этого типа может расширить функциональные возможности сайтов, например добавив социальный компонент или компонент корзины изъятия.

Для удаленного компонента необходимо указать URL-адрес конечной точки. Для локальных компонентов, визуализированных во встроенных рамках, URL-адреса создаются на основе имени компонента в каталоге компонентов.

- URL-адрес конечной точки: контент виджета извлекается из этого URL-адреса и встраивается во встроенную рамку.
- URL-адрес настроек: этот URL-адрес визуализируется во встроенной рамке для настройки компонента после его перетаскивания на страницу.

Компонент может отображать статические или динамические визуализации данных, а также форму или другой интерактивный пользовательский интерфейс, расширяющий функциональные возможности сайта. См. [Настройки рендеринга компонентов](#).

Компоненты, поставляемые с Oracle Content Management

Набор компонентов, визуализируемых во встроенных рамках, поставляется с Oracle Content Management.

Имя	Тип	Идентификатор
Список папок	scs-app	Список папок
Список файлов	scs-app	Список файлов
Диспетчер документов	scs-app	Диспетчер документов
Кнопка Facebook "Нравится"	scs-app	Кнопка Facebook "Нравится"
Кнопка Twitter "Подписаться"	scs-app	Кнопка Twitter "Подписаться"
Кнопка Twitter "Поделиться"	scs-app	Кнопка Twitter "Поделиться"
Кнопка Facebook "Рекомендовать"	scs-app	Кнопка Facebook "Рекомендовать"

Регистрация компонента

Чтобы удаленный компонент, визуализированный во встроенных рамках, можно было использовать на сайте, необходимо зарегистрировать URL-адрес конечной точки с помощью HTTPS. Эта информация хранится в каталоге компонентов. Конечная точка должна допускать отображение URL-адреса во встроенной рамке. Не задавайте в заголовке `X-Frame-Options="sameorigin"`. Для локальных компонентов эти критерии конечной точки выполняются автоматически, поскольку файлы хранятся на сервере Oracle Content Management.

При регистрации компонента, визуализированного во встроенных рамках, создается новый GUID, представляющий компонент. Если такой компонент зарегистрирован в экземпляре Oracle Content Management несколько раз, он получит несколько GUID, поскольку они представляют каждую регистрацию этого компонента. Если компонент зарегистрирован, его описание сохраняется в Oracle Content Management с указанием GUID. Компонент по-прежнему будет запускаться с удаленной конечной точки. См. [Регистрация удаленного компонента](#).

После регистрации компонента, визуализируемого во встроенной рамке, также создается идентификатор экземпляра. Этот идентификатор экземпляра представляет компонент, зарегистрированный в определенном арендаторе Oracle Content Management. Один и тот же компонент, зарегистрированный несколько раз в одном арендаторе Oracle Content Management или разных арендаторах Oracle Content Management, будет иметь разные идентификаторы экземпляров.

Сохранение настроек удаленных компонентов

Когда URL-адрес настроек для компонента, визуализированного во встроенной рамке, отображается в диалоговом окне панели "Настройки", предоставляются и идентификатор экземпляра, и идентификатор компонента. Это позволяет компоненту самостоятельно выбрать сохранение любых настроек на

собственном сервере, индексируемом по идентификатору экземпляра и идентификатору компонента. Кроме того, комплект Oracle Content Management Sites SDK можно использовать для сохранения до 1,5 КБ данных JSON в модели страницы сайта с учетом идентификатором компонента.

Использование Sites SDK для сохранения настроек имеет два преимущества для компонентов, визуализированных во встроенных рамках:

- Компонент может легко участвовать в версиях страниц, обновлениях страниц и модели публикации сайта.
- Компонент может состоять из конечных точек HTML, которые выполняются в браузере, а не в серверной системе.

См. Комплекты Oracle Content Management SDK.

Об идентификаторе экземпляра и структуре компонентов с рендерингом в задаваемых в строках фреймах

Идентификатор экземпляра компонента — это уникальный идентификатор компонента, который визуализируется во встроенной рамке сайта.

Когда пользователь перетаскивает компонент, визуализированный во встроенной рамке, из каталога компонентов на страницу сайта, сервису Oracle Sites Cloud Server отправляется вызов выделения ресурсов, чтобы создать новый уникальный идентификатор экземпляра компонента. Этот идентификатор уникален, и все такие экземпляры компонентов, выделенные в этом сервисе, получают одинаковый идентификатор экземпляра.

Идентификатор экземпляра компонента содержит дополнительную информацию, которая может служить для защиты настроек и использования компонента, визуализированного во встроенной рамке, чтобы обеспечить поступление обновления настроек компонента из доверенного места.

Параметр идентификатора экземпляра позволяет разработчикам идентифицировать сайт и аутентифицировать вызывающего абонента. Вызывающий абонент аутентифицируется посредством проверки цифровой подписи, которая создается с помощью секретного ключа компонента. Секретный ключ создается в процессе регистрации компонента.

Экземпляр компонента состоит из двух частей, разделенных точкой ("."): данные и структура.

Экземпляр компонента: данные

Часть данных экземпляра компонента, визуализированного во встроенной рамке, представляет собой строку JSON с кодировкой Base64. Структура строки JSON:

```
{
  "instanceid": "BBDC7614F693B75110D811E6C0B77C935FAEC5112E5E",
  "permissions": "",
  "entitlements": "",
  "signdate": "1435426735293",
  "sitedomain": "service1-tenant4.localhost"
}
```

Имя поля	Описание
instanceid	Уникальный идентификатор компонента, визуализированного во встроенной рамке, для арендатора Oracle Content Management.
signdate	Дата создания подписи.
sitedomain	Имя домена экземпляра Oracle Content Management.
permissions	Набор разрешений участника сайта. В режиме редактирования имеет значение "SITE_OWNER"; в противном случае не имеет значения.
entitlements	Список премиальных функций, приобретенных владельцем сайта.

Экземпляр компонента: сигнатура

Часть данных экземпляра компонента сериализована перед подписанием с использованием APP_SECRET_KEY. Этот секретный ключ должен быть создан и показан разработчику во время регистрации компонента. Подпись рассчитывается путем создания хеша части данных экземпляра компонента (сериализованной структуры JSON) с использованием секретного ключа, как показано ниже:

```
$signature = HMAC (serialized JSON structure, APP_SECRET_KEY)
```

При создании подписи используется алгоритм хеширования SHA256. Маркер представляет собой конкатенацию сериализованной структуры JSON и созданного компонента подписи, как показано ниже:

```
$instance = {base64encoded serialized JSON structure}.
{base64encoded $signature}
```

Пример:

```
//base64 encoded serialized object           //signature
eyJpbmN0YW5jZWlkIjoiQTRGOTE3REY5OTZEN0Q3ODBCMjUzODZFOTFEMDA3ODJGMjVBRjY2
Rjc3OTIiLCJzaWduZGF0ZSI6IjE0NDU2MzcwNTk5MTciLCJzaXRlZG9tYWluIjoic2Vydmlj
ZTEtdGVuYW50MS51cy5vcmFjbGUuY29tIiwicGVyYbWlzc2lbnMiOiJTSVRFX09XTkVSIiw
iZW50aXRsZWl1bnRzIjoiIn0=.5p3of7t110wuysF3zpm+YgICSHH8C/BHczdbVZx2VH8=
```

Безопасность удаленных компонентов

Oracle Content Management позволяет сторонним разработчикам интегрировать свои пользовательские компоненты в Oracle Content Management, но с хранением на удаленном сервере.

У каждого удаленного компонента должны быть зарегистрированные настройки и конечные точки рендеринга с Oracle Content Management. Помимо конечных точек разработчикам также необходимо предоставить секретный ключ, уникальный для зарегистрированного компонента.

Oracle Content Management вызывает конечные точки зарегистрированного компонента для реализации контента на странице сайта. Поскольку такие конечные точки доступны из общедоступного Интернета, разработчики должны убедиться, что конечные точки зарегистрированного удаленного компонента вызываются из Oracle Content Management. Для проверки подлинности вызывающего абонента на зарегистрированные конечные точки URL-адреса передается подписанный маркер. Вызывающая сторона аутентифицируется путем проверки цифровой подписи, встроенной в подписанный маркер, с помощью секретного ключа удаленного компонента, который был предоставлен в процессе регистрации.

Формат маркера:

```
{base64 encoded serialized JSON data}.{base64 encoded signature}
```

Пример маркера, переданного зарегистрированным конечным точкам приложения:

```
eyJpbmN0YW5jZWlkIjoiQTRGOTE3REY5OTZEN0Q3ODBCMjUzODZFOTFEMDA3ODJGMjVBRjY2Rjc3O  
TIIiLCJzaWduZGF0ZSI6IjE0NDU2MzcwNTk5MTciLCJzaXRlZG9tYWluIjoic2VydmljZTEtdGVuYW  
50MS51cy5vcnFjbGUuY29tIiwicGVybWlzc2lvdnMiOiJTSVRFX09XTkVSIiwizW50aXRszW11bnR  
zIjoiIn0=.5p3of7t11OwuysF3zpm+YgICSHH8C/BHczdbVZx2VH8=
```

Маркер состоит из двух отдельных частей: данные и подпись, разделенные точкой (".").

Как правило, разработчики должны всегда проверять подлинность маркера в режиме редактирования или предварительного просмотра перед предоставлением доступа к зарегистрированным конечным точкам компонента. Кроме того, при аутентификации вызывающего абонента в конечной точке настроек разработчики должны всегда искать значение `SITE_OWNER` в поле **разрешений** маркера. В поле **разрешений** маркера значение `SITE_OWNER` отображается только в режиме редактирования. Маркер, созданный во время сеанса редактирования, никогда не сохраняется обратно в модель страницы и отключается с помощью маркера времени выполнения со значением `NULL` в поле **разрешений**.

Данные

Часть данных экземпляра — это строка JSON в кодировке Base64. Структура строки JSON:

```
{
  "instanceid": "BBDC7614F693B75110D811E6C0B77C935FAEC5112E5E",
  "permissions": "",
  "entitlements": "",
  "signdate": "1435426735293",
  "sitedomain": "service1-tenant4.localhost"
}
```

Имя поля	Описание
instanceid	Уникальный идентификатор компонента для арендатора Oracle Content Management.
signdate	Дата создания подписи.
sitedomain	Имя домена экземпляра Oracle Content Management.

Имя поля	Описание
permissions	Набор разрешений участника сайта. В режиме редактирования имеет значение "SITE_OWNER"; в противном случае не имеет значения.
entitlements	Список премиальных функций, приобретенных владельцем сайта.

Подпись

Часть данных экземпляра удаленного компонента сериализована перед подписанием с использованием APP_SECRET_KEY. Этот секретный ключ должен быть создан и показан разработчику во время регистрации компонента. Подпись рассчитывается путем создания хеша части данных экземпляра компонента (сериализованной структуры JSON) с использованием секретного ключа, как показано ниже:

```
$signature = HMAC (serialized JSON structure, APP_SECRET_KEY)
```

При создании подписи используется алгоритм хеширования SHA256. Маркер представляет собой конкатенацию сериализованной структуры JSON и созданного компонента подписи, как показано ниже:

```
$instance = {base64encoded serialized JSON structure}.  
{base64encoded $signature}
```

Регистрация удаленного компонента

Чтобы использовать удаленный компонент на сайте, сначала необходимо его зарегистрировать в Oracle Content Management.

Можно регистрировать сторонние удаленные компоненты и компоненты собственной разработки.

Чтобы зарегистрировать удаленный компонент для использования в своем экземпляре Oracle Content Management, выполните указанные ниже действия:

1. Выберите **Разработчик**, а затем нажмите **Смотреть все компоненты**.
2. Нажмите **Создать** и выберите **Зарегистрировать удаленный компонент**.
3. В окне "Регистрация удаленного компонента" введите или выберите нужные сведения, включая описанные ниже.
 - **Имя:** имя компонента, которое отображается для пользователей.
 - **Описание:** описание компонента, которое отображается для пользователей.
 - **URL-адрес компонента:** конечная точка, которая используется в элементе iframe для визуализации содержимого компонента на странице. Он должен соответствовать протоколу HTTPS.
 - **URL-адрес настроек:** конечная точка, которая используется в элементе iframe для визуализации настроек удаленного компонента, добавленного на страницу. Он должен соответствовать протоколу HTTPS.

- **Ширина панели настроек:** используемая по умолчанию ширина панели настроек компонента (в пикселях).
 - **Высота панели настроек:** используемая по умолчанию высота панели настроек компонента (в пикселях).
 - **Ключ:** 192-разрядный AES-ключ, который связан с удаленным компонентом и используется для создания подписанной хеш-лексема при развертывании компонента. Он используется для шифрования и обеспечения безопасности во время чтения и записи настроек компонента.
4. Нажмите **Регистрация**.

После создания удаленного компонента его имя появляется в списке компонентов. Чтобы просмотреть файлы, используемые для регистрации компонента, выберите имя компонента в списке компонентов.

Сведения о регистрации компонента сохраняются в каталоге, который используется сайтами, созданными в том же экземпляре Oracle Content Management, но компонент остается удаленным сервисом.

Для владельца компонента на панель "Пользовательские компоненты" в Site Builder добавляется значок компонента с именем, присвоенным этому компоненту. Если предоставить доступ к компоненту другим пользователям, они также увидят этот компонент на панели "Пользовательские компоненты" в Site Builder.

Удаление компонента

При наличии соответствующих разрешений можно удалить компонент из диспетчера компонентов. Удаленный компонент будет недоступен для дальнейшего использования. При удалении компонента соответствующая папка и все связанные с ней папки и файлы перемещаются в корзину.

Можно удалить компонент из диспетчера компонентов, если он создан вами и вы владелец этого компонента, а также если вам предоставлен доступ к этому компоненту и назначена роль "Управление".

Примечание.

Если компонент используется на сайте или в обновлении (включая сайты и обновления, находящиеся в корзине), удалить этот компонент невозможно.

Удаление компонента:

1. На домашней странице нажмите **Разработчик**.
2. Нажмите **Смотреть все компоненты**.
Отображаются все зарегистрированные в настоящее время компоненты.
3. Выберите имя компонента и выберите **Удалить** в контекстном меню или нажмите  на панели действий.
Отображается запрос на подтверждение действия.
4. Чтобы подтвердить удаление, нажмите **Да**. Чтобы остановить удаление, нажмите **Нет**.
После подтверждения удаления компонент и все связанные с ним папки и файлы перемещаются в корзину.

Удаленная папка компонента остается в корзине до наступления одного из указанных ниже событий.

- Папка восстановлена.
- Папка безвозвратно удалена.
- Квота корзины превышена.
- Корзина автоматически опустошается с интервалом, заданным администратором сервиса. Значение по умолчанию равно 90 дням.

Sites SDK

Компоненты, разработанные для Oracle Content Management, визуализируются как компоненты на сайте. Их можно перетащить в любое место на определенной странице сайта.

Sites SDK обрабатывает все виды взаимодействия между таким компонентом и страницей.

- `Sites.Settings.getProperty(propertyName, callbackFunction)`: предоставляет отклик, чтобы извлечь запрошенное свойство для экземпляра пользовательского компонента.
- `Sites.Settings.setProperty(propertyName, propertyValue)`: сохраняет запрошенное свойство для соответствующего экземпляра настраиваемого компонента.

Примените `Sites.Settings.getProperty` или `Sites.Settings.setProperty` ко всем свойствам пользовательских компонентов, а затем используйте `SitesSDK.publish` и `SitesSDK.subscribe` для прослушивания событий сообщений.

См. [Ссылка на Sites SDK](#).

Настройка файла контроллера

На сайтах Oracle Content Management файл контроллера используется для отображения каждой страницы сайта. Когда браузер отправляет запрос на веб-страницу, сервер отвечает копией файла контроллера. Для каждой запрашиваемой страницы с сервера доставляется одна и та же копия файла контроллера.

- [О файле контроллера](#)
- [Изменение файла по умолчанию controller.html](#)
- [Об объекте SCS](#)
- [Разделы файла контроллера, настраивать которые не нужно](#)
- [Использование маркеров для обеспечения переносимости файлов пользовательского контроллера](#)
- [Примеры файлов пользовательского контроллера](#)

О файле контроллера

Файл контроллера представляет собой небольшую HTML-страницу, которая динамически инициирует последовательность рендеринга для остальной части страницы. Здесь в первую очередь можно применить настройки, чтобы повлиять на поведение каждой страницы сайта.

Основной задачей файла контроллера является предоставление и размещение среды выполнения для JavaScript контроллера. Впоследствии JavaScript контроллера выполняет загрузку и отображение страницы. Настройки в файле контроллера позволяют переопределять и влиять на работу JavaScript контроллера.

Файл контроллера по умолчанию

При первом создании сайта Oracle Content Management с этим сайтом связывается файл контроллера по умолчанию 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>

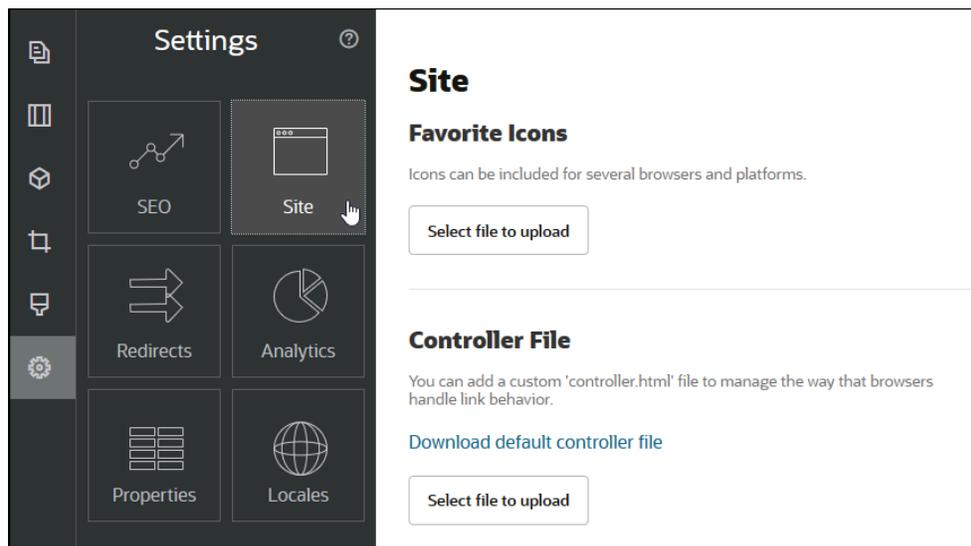
<!-- The following image tag displays an animated circle by default if
it takes too long for a page to display. The wait image must have the
id scsWaitImage. -->

</body></html>
```

Изменение файла по умолчанию controller.html

Выгрузите файл controller.html сайта и измените его, чтобы это повлияло на поведение каждой страницы сайта. Чтобы выгрузить файл сайта controller.html, выполните указанные ниже действия.

1. Откройте сайт, который требуется изменить, в Site Builder и установите для него режим **Редактирование**.
2. Выберите существующее обновление или создайте новое.
3. Нажмите **Настройки** в боковом меню навигации, а затем **Сайт**.



4. В разделе "Файл контроллера" нажмите **Выгрузить файл контроллера по умолчанию**.
5. Сохраните файл controller.html на локальном диске и внесите необходимые изменения.
6. По завершении редактирования вернитесь на страницу настроек сайта и в разделе "Файл контроллера" нажмите **Выбрать файл для загрузки**.
7. Перейдите к измененному файлу контроллера и загрузите его.

Примечание.

Настройки файла контроллера используются только на веб-сайтах. Они не будут использоваться при редактировании сайта или в предварительном просмотре сайта. Чтобы изменения вступили в силу после загрузки пользовательского файла controller.html для сайта, необходимо выполнить обновление сайта и опубликовать его.

Об объекте SCS

JavaScript контроллера использует переменные и свойства, определенные в глобальном объекте SCS, для отображения веб-страницы. Перед включением файла controller.js необходимо определить глобальный объект JavaScript в файле контроллера. В объекте SCS имеются два основных свойства, которые можно использовать для настройки:

- [SCS.sitePrefix](#)
- [SCS.preInitRendering](#)
- [SCS.getDeviceInfo](#)

SCS.sitePrefix

Переменная `SCS.sitePrefix` определяет префикс пути для сайта, находящегося в режиме онлайн. Обычно по умолчанию используется `site/<siteName>/`.

JavaScript контроллера использует это значение для определения веб-страницы, которая будет отображаться. Например, если браузер запрашивает страницу в `/site/SampleSite/products/index.html`, `sitePrefix /site/SampleSite/` позволяет JavaScript вычислить, что на сайте должна отображаться страница `products/index.html`. Обратите внимание, что `sitePrefix` должен быть строковым значением, которое начинается и заканчивается символом `"/`.

Если используется прокси-сервер или другая инфраструктура, например правила сопоставления Akamai и URL-адресов, изменение этой переменной позволяет настраивать префикс сайта. Например, установка для `sitePrefix` значения `/intranet/ExampleSite/` позволит доставлять веб-сайт с использованием такого пути в браузере вместо префикса пути по умолчанию `/site/SampleSite/`.

 **Примечание.**

В качестве префикса `sitePrefix` в JavaScript контроллера по умолчанию используется `/`, если URL-адрес в браузере не совпадает с префиксом сайта. Это позволяет сайту работать с использованием персонализированного домена без индивидуальной настройки.

SCS.preInitRendering

Если это определено, перед выполнением какой-либо из вычислительных логических операций код JavaScript контроллера вызывает функцию `SCS.preInitRendering`. Эта функция полезна для определения в пользовательских файлах контроллера. Она позволяет переопределить основные операции JavaScript контроллера.

Если это определено, `SCS.preInitRendering` должна быть функцией. Она вызывается без каких-либо аргументов, и возвращаемое значение не ожидается и не обрабатывается.

SCS.getDeviceInfo

Функция `SCS.getDeviceInfo` позволяет настраивать логику обнаружения устройств внутри JavaScript контроллера. Это обнаружение используется для определения необходимости доставки мобильной или оперативной версии страницы.

Функция `SCS.getDeviceInfo` не принимает никаких аргументов и возвращает объект JavaScript с двумя свойствами:

- **isMobile** — логическое свойство, указывающее, что текущее устройство является мобильным клиентом, например смартфоном.
- **isIOS** — логическое свойство, указывающее, что текущее устройство работает под управлением операционной системы на базе iOS.

Если пользовательский контроллер не переопределяет `SCS.getDeviceInfo`, используется встроенная реализация по умолчанию.

Разделы файла контроллера, настраивать которые не нужно

Некоторые разделы файла контроллера обрабатываются сервером Oracle Content Management для использования CDN и определения сегментов URL-кэширования. Они указаны в примере ниже.

```
<!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>
```

Использование маркеров для обеспечения переносимости файлов пользовательского контроллера

При создании сайта в файле контроллера используется имя сайта. При этом файл контроллера привязывается к сайту с этим именем и этот файл контроллера не может использоваться другими сайтами.

Использование файла контроллера на сайте с именем, отличным от имени, используемого в файле контроллера, невозможно. Это также относится к сайтам, созданным на основе шаблона сайта с пользовательским файлом контроллера. Следующие динамически оцениваемые маркеры позволяют использовать файл контроллера для нескольких сайтов.

Маркер	Описание
[!--\$SCS_SITE_PREFIX--]	<p>Выполняется оценка префикса сайта для текущего сайта. Примеры значений могут включать:</p> <pre>/site/MySite/</pre> <p>или</p> <pre>/site/authsite/MySecureSite/</pre>
[!--\$SCS_SITE_PATH--]	<p>Выполняется оценка текущего местоположения CDN продукта. Значение не будет содержать символ "/" в конце. Примеры значений могут включать:</p> <pre>/site/MySite/_cache_0000</pre> <p>или</p> <pre>/site/authsite/MySecureSite</pre>
[!--\$SCS_PRODUCT_PATH--]	<p>Выполняется оценка текущего URL-адреса продукта, включая ключ кэша, если применимо. Значение не будет содержать символ "/" в конце. Примеры значений могут включать:</p> <pre>https://www.example.com/cdn/sec/v21.1.2.23</pre> <p>или</p> <pre>https://www.example.com/cdn/sec/v21.1.3.18</pre>

Примеры файлов пользовательского контроллера

Примеры приведены для указанных ниже пользовательских файлов управления.

- [Изменение префикса сайта](#)
- [Настройка графика ожидания](#)
- [Настройка значков сайтов](#)
- [Настройка тегов <noscript> и <meta> для обработчиков без JavaScript](#)
- [Предварительная выборка файлов JavaScript](#)
- [Проверка прав владения сайтом с помощью дополнительной разметки](#)
- [Обнаружение дополняющих устройств](#)
- [Использование маркеров для улучшения переносимости файла controller.htm](#)

Изменение префикса сайта

В следующем примере файл контроллера определяет функцию `preInitRendering`, которая позволяет доставлять сайт с несколькими префиксами.

Примечание.

Для использования этого примера необходимо настроить CDN или другой прокси-сервер для ответа на заданные префиксы. Кроме того, если ни один из дополнительных префиксов, определенных в этой функции, не соответствует URL-адресу браузера, используется поведение по умолчанию.

```
<!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>
```

Настройка графика ожидания

В приведенном ниже примере файл контроллера настраивает график ожидания, который отображается при рендеринге, отложенном из-за сетевых задержек.

Примечание.

В этом примере используется URL-адрес данных для изображения, а не для выполнения отдельного запроса во время рендеринга.

```
<!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>

```

Настройка значков сайтов

В следующем примере файла контроллера определяется пользовательский значок для сайта. Это полезно, если браузеры не загружают значок сайта динамически из объектной модели документа отображаемой страницы.

```

<!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>
```

Настройка тегов <noscript> и <meta> для обработчиков без JavaScript

В приведенном ниже примере файл контроллера настраивает сообщение <noscript> и теги <meta> для обходчиков, которые не обрабатывают JavaScript.

Примечание.

Поскольку файл контроллера работает для каждой страницы веб-сайта, на каждой странице будет отображаться один и тот же метатег.

```
<!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>
```

Предварительная выборка файлов JavaScript

Некоторые браузеры позволяют создавать объявления о ресурсах, необходимых для текущей или последующей навигации. Разметка в файле контроллера может объявлять ресурсы, которые должны быть предварительно выбраны или предварительно загружены.

 **Примечание.**

Поскольку для каждой веб-страницы OCE используются файлы `tare.js` и `renderer.js`, они хорошо подойдут для методов предварительной загрузки и предварительной выборки. Кроме того, если `href` отображается первым в теге `<link>`, его значение может быть назначено сервером OCE для доставки из CDN.

```
<!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>
```

Проверка прав владения сайтом с помощью дополнительной разметки

Обходчики и поисковые системы сторонних поставщиков могут потребовать внесения дополнительной разметки для проверки прав владения сайтом. Это можно ввести в файл контроллера по мере необходимости.

```
<!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>
```

Обнаружение дополняющих устройств

Распознавание новых мобильных устройств или идентификаторов агентов пользователей устройств можно выполнить, настроив код обнаружения устройств в контроллере.

Примечание.

- Функция `getDeviceInfo` вызывается контроллером, чтобы определить, используется ли мобильное устройство. В этом примере переопределяется вызов `getDeviceInfo` по умолчанию.
- Свойство `isMobile` определяет, какой макет будет использоваться при рендеринге страницы.
- Свойство `isIOS` определяет, какой поставщик используется с компонентом Map.
- Встроенный тест `isMobile` по умолчанию:

```
/Mobi|iPhone|iPod|BlackBerry|IEMobile|Opera Mini/  
i.test(userAgent) && !/iPad/i.test(userAgent)
```

- Встроенный тест `isIOS` по умолчанию:

```
/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>
```

Использование маркеров для улучшения переносимости файла controller.htm

По умолчанию имя сайта жестко закодировано в файлы controller.html, что затрудняет переименование сайта или повторное использование одинаковых пользовательских контроллеров на нескольких сайтах. Для устранения этой проблемы можно использовать маркеры. В следующем примере показано, как можно использовать маркеры переносимости.

Примечание.

Теги <link> в примере предназначены только для демонстрации. Базовый controller.html не будет включать их.

```
<!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>
```

Часть V

Разработка сайтов с помощью инструментария других программ

Oracle Content Management предоставляет несколько способов взаимодействия с приложениями, разработанными как внутри Oracle Content Management, так и вне Oracle Content Management с использованием других инструментов сторонних разработчиков.

Использование инструментария OCE

Инструментарий Oracle Content Management OCE и комплекты SDK помогают разрабатывать пользовательские приложения, которые используют контент, управляемый в репозитории Oracle Content Management. Эти приложения можно разрабатывать в Oracle Content Management или с помощью инструментов сторонних разработчиков.

Инструментарий OCE помогает выполнять указанные ниже задачи.

- Настройте свою локальную среду разработки для использования экземпляра Oracle Content Management в целях локальной разработки и тестирования компонентов, шаблонов, тем и макетов контента.
- Создавайте компоненты, шаблоны сайтов и макеты контента на основе образцов, выполняйте их в тестовой оснастке, изучайте их и разрабатывайте компоненты, шаблоны, темы и макеты контента в среде Developer Cloud Service.
- Импортируйте компоненты и шаблоны сайтов, созданные в Oracle Content Management, в проект и среду Developer Cloud Service для управления источниками и дальнейшей разработки.
- Экспорт компонентов, шаблонов и макетов контента из среды Developer Cloud Service для использования в Oracle Content Management
- Копирование существующего компонента, шаблона или макета контента
- Создание модульных тестов
- Оптимизация компонентов
- Развертывание компонентов и шаблонов в Oracle Content Management

Дополнительную информацию можно найти по адресу <https://github.com/oracle/content-and-experience-toolkit#readme>.

Оркестрация взаимодействия

Если для создания компонентов взаимодействия вы используете другие инструменты, а не Oracle Content Management, к ним можно подключить репозитории Oracle Content Management, чтобы создатели контента могли просматривать изменения сайта по мере их работы, а автоматические компоновки могли запускаться при изменении или публикации контента. Такая оркестрация взаимодействия обеспечивает автоматизацию рабочего потока между поставщиками контента и разработчиками сайта для упрощения управления компонентами взаимодействия и их публикации.

Разработка с помощью инструментария OCE

Инструментарий OCE помогает разрабатывать шаблоны сайтов, темы, пользовательские компоненты и макеты контента для Oracle Content Management.

Инструментарий OCE предоставляет для работы собственную среду разработки и позволяет использовать репозитории активов, файлы и папки в Oracle Content Management. Инструментарий OCE содержит инструменты для создания и разработки пользовательских компонентов и шаблонов сайтов, включая темы и макеты контента. Инструментарий включает в себя локальную тестовую оснастку для быстрой итеративной разработки и примеры модульных тестов, с которых можно начать работу.

В указанных ниже темах описывается, как настроить инструментарий OCE и использовать его для разработки на локальном компьютере или в качестве проекта Developer Cloud Service.

- [Настройка инструментария OCE на локальном компьютере](#)
- [Обновление до jQuery 3.5.x](#)
- [Разработка для Oracle Content Management в Developer Cloud Service](#)
- [Распространение изменений из среды тестирования в среду производства с помощью инструментария OCE](#)
- [Создание сайта на основе шаблона и сохранение одинаковых GUID для контента](#)
- [Импорт и экспорт таксономий](#)
- [Импорт и экспорт рекомендаций](#)
- [Разработка пользовательских редакторов полей с помощью инструментария OCE](#)
- [Перенос сайта с одного сервера на другой или обновление](#)
- [Индексирование страниц сайта при помощи инструментария OCE](#)
- [Индексирование многоязычного сайта с помощью инструментария OCE](#)
- [Создание упрощенного компонента для облегчения разработки компонентов](#)
- [Компиляция сайта для повышения производительности во время выполнения для страниц сайта](#)
- [Создание новой задачи перевода сайта или актива на сервере Oracle Content Management](#)
- [Переводите сайт с помощью поставщика переводческих услуг](#)

Настройка инструментария OCE на локальном компьютере

На своем локальном компьютере вы можете выгрузить инструментарий OCE из GitHub.

Чтобы настроить инструментарий OCE, выполните инструкции, приведенные на странице

<https://github.com/oracle/content-and-experience-toolkit/blob/master/README.md>

Завершите настройку и подготовьтесь к использованию инструментария OCE:

1. [Установка зависимых компонентов с помощью команды npm](#)
2. [Использование утилиты командной строки CEC](#)
3. [Тестирование с помощью локальной тестовой оснастки](#)

Установка зависимых компонентов с помощью команды npm

Используйте npm (диспетчер пакетов узлов), чтобы установить зависимости сайтов для своего проекта.

Если для доступа к Интернету используется прокси-сервер, задайте прокси-сервер для npm с помощью команды `npm config`. См. <https://docs.npmjs.com/misc/config>. О том, как настроить прокси-сервер для bower, см. в по ссылке <https://bower.io/docs/config/>.

Примечание.

Убедитесь, что на локальном компьютере установлен модуль Node.js 8.0.0 или более поздней версии (<https://nodejs.org/>).

Использование утилиты командной строки CEC

Межплатформенная утилита командной строки `cec` предоставляет команды для создания шаблонов и компонентов и управления ими.

Перед использованием утилиты командной строки `cec` создайте в локальной среде разработки источник для команд, которые требуют подключения к серверу Oracle Content Management.

Встроенная справка содержит информацию для выполнения команд с примерами. Чтобы просмотреть встроенную справку `cec`, в командной строке можно ввести команды `cec`:

- В окне терминала перейдите к каталогу `cec`.
- Введите `cec`, чтобы отобразить список команд `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> from <zip>.	Imports a template [alias: it]
cec export-template <name> template <name> as a zip file. et]	Exports the [alias:
cec upload-template <name> template <name> to the OCM server. ult]	Uploads the [alias:
cec delete-template <name> template <name> on the OCM server. [alias:]	Deletes the
cec share-template <name> with users and groups on OCM server.	Shares template [alias: stm]
cec unshare-template <name> group access to a template on OCM server.	Deletes user or [alias: ustm]
cec update-template <action> a local template.	Performs action on [alias: ut]
cec describe-template <name> template <name> package. dst]	Describes the [alias:
cec create-template-report <name> usage report for the template <name> package.	Generates an asset [alias: cctr]
 Themes	
cec add-component-to-theme <component> to a theme.	Adds a component [alias: actt]
cec remove-component-from-theme <component> component from a theme. rcft]	Removes a [alias:
cec control-theme <action> <action> on theme on OCM server.	Performs action [alias: ctt]
cec share-theme <name> users and groups on OCM server.	Shares theme with [alias: sth]
cec unshare-theme <name> group access to a theme on OCM server.	Deletes user or [alias: usth]
 Sites	
cec create-site <name> Site <name>.	Creates Enterprise [alias: cs]
cec copy-site <name> Site <name>.	Copies Enterprise [alias: cps]
cec update-site <name> Site <name>.	Update Enterprise [alias: us]
cec transfer-site <name> from one OCM server to another.	Transfers a site [alias: ts]
cec transfer-site-content <name> transfer site content from one OCM server to another.	Creates scripts to [alias: tsc]
cec validate-site <name> <name>.	Validates site [alias: vs]
cec control-site <action> <action> on site on OCM server.	Performs action [alias: cts]
cec share-site <name> users and groups on OCM server.	Shares site with [alias: ss]
cec unshare-site <name> group access to a site on OCM server.	Deletes user or [alias: uss]

<pre> cec get-site-security <name> OCM server. cec set-site-security <name> OCM server. cec index-site <site> of site <site> on OCM server. cec create-site-map <site> site <site> on OCM server. cec create-rss-feed <site> site <site> on OCM server. cec create-asset-report <site> usage report for site <site> on OCM server. cec upload-static-site-files <path> statically from a site on OCM server. cec download-static-site-files <site> files from a site on OCM server. cec delete-static-site-files <site> files from a site on OCM server. cec refresh-prerender-cache <site> cache for a site on OCM server. cec migrate-site <site> OCI IC server to EC server. </pre>	<pre> Gets site security on [alias: gss] Sets site security on [alias: sss] Index the page content [alias: is] Creates a site map for [alias: csm] Creates RSS feed for [alias: crf] Generates an asset [alias: car] Uploads files to render [alias: ulss] Downloads the static [alias: dlss] Deletes the static [alias:] Refreshes pre-render [alias: rpc] Migrates a site from [alias: ms] </pre>
--	--

Assets

<pre> cec download-content OCM server. cec upload-content <name> to a repository on OCM server. cec control-content <action> <action> on channel items on OCM server. cec transfer-content <repository> transfer content from one OCM server to another. cec list-assets server. cec create-digital-asset asset cec update-digital-asset <id> asset cec copy-assets <repository> another repository on OCM server. cec create-asset-usage-report <assets> usage report for assets on OCM server. </pre>	<pre> Downloads content from [alias: dlc] Uploads local content [alias: ulc] Performs action [alias: ctct] Creates scripts to [alias: tc] Lists assets on OCM [alias: la] Creates digital [alias: cda] Updates digital [alias: uda] Copies assets to [alias: ca] Generates an asset [alias: caur] </pre>
---	--

Content

<pre> cec create-repository <name> OCM server. cec control-repository <action> <action> on repositories on OCM server. cec share-repository <name> users and groups on OCM server. cec unshare-repository <name> access to a repository on OCM server. cec create-channel <name> OCM server. cec share-channel <name> </pre>	<pre> Creates a repository on [alias: cr] Performs action [alias: ctr] Shares repository with [alias: sr] Deletes user or group [alias: usr] Creates a channel on [alias: cch] Shares channel with </pre>
--	---

users and groups on OCM server.	[alias: sch]
cec unshare-channel <name>	Deletes user or
group access to a channel on OCM server.	[alias: usch]
cec create-localization-policy <name>	Creates a
localization policy on OCM server.	
[alias: clp]	
cec list-server-content-types	Lists all content
types from server.	[alias: lsct]
cec share-type <name>	Shares type with
users and groups on OCM server.	[alias: st]
cec unshare-type <name>	Deletes user or
group access to a type on OCM server.	[alias: ust]
cec download-type <name>	Downloads types
from OCM server.	[alias: dltp]
cec upload-type <name>	Uploads types to
OCM server.	[alias: ultp]
cec update-type <action>	Performs action
<action> on a type	[alias: utp]
cec download-recommendation <name>	Downloads a
recommendation from the OCM server.	[alias:
dlr]	
cec upload-recommendation <name>	Uploads a
recommendation to the OCM server.	
[alias: ulr]	
cec create-contentlayout <name>	Creates a content
layout based on a content type.	[alias: ccl]
cec add-contentlayout-mapping <contentlayout>	Creates content
type and content layout mapping.	[alias: aclm]
cec remove-contentlayout-mapping <contentlayout>	Removes a content
layout mapping.	[alias: rclm]
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 migrate-content <name>	Migrates content
from OCI IC server to EC server.	[alias: mc]
cec compile-content	Compiles the
content items generating HTML renditions.	[alias:
cmpr]	
cec upload-compiled-content <path>	Uploads the
compiled content to OCM server.	[alias:
ulcc]	
Taxonomies	
cec download-taxonomy <name>	Downloads a
taxonomy from OCM server.	[alias:
dltx]	
cec upload-taxonomy <taxonomy>	Uploads a taxonomy
to OCM server.	[alias: ultx]
cec control-taxonomy <action>	Performs action on
taxonomy on OCM server.	[alias: cttx]
Translation	
cec list-translation-jobs	Lists translation

```

jobs.
    cec create-translation-job <name>
job <name> for a site on OCM server.
    cec download-translation-job <name>
job <name> from OCM server.
    cec submit-translation-job <name>
<name> to translation connection <connection>.
    cec refresh-translation-job <name>
job <name> from translation connection.
    cec ingest-translation-job <name>
<name> from translation connection and ingest.
    cec upload-translation-job <name>
<name> to OCM server.
    cec create-translation-connector <name>
connector <name>.
    cec start-translation-connector <name>
connector <name>.
    cec register-translation-connector <name>
connector.

Groups
    cec create-group <name>
OCM server.
    cec delete-group <name>
OCM server.
    cec add-member-to-group <name>
to an OCM group on OCM server.
    cec remove-member-from-group <name>
groups from an OCM group on OCM server.

Local Environment
    cec create-encryption-key <file>
key to encrypt/decrypt password for servers.
    cec register-server <name>
server.
    cec set-oauth-token <token>
registered server.
    cec list
resources.
    cec execute-get <endpoint>
request to a REST API endpoint on OCM server
    cec install
tree.
    cec develop
server.
    cec sync-server
server.
    cec webhook-server
server.

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:Compl/assets                 Downloads
all files in folder assets of component Compl and save to local folder
src/documents/Compl/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 Compl 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:Comp1/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-Documents-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/Temp1.zip Imports the template Temp1.
```

```
-----
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 Temp1 Exports the template Temp1.
```

```
-----
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  
Documents                        The report will be saved to ~/Documents/  
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 Theme1 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 Theme1 -u user1,user2
cec unshare-theme Theme1 -u user1,user2 -g group1,group2
cec unshare-theme Theme1 -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 Template1      Updates site Site1 using the
  content from template Template1
  cec update-site Site1 -t Template1 -x  Updates site Site1 using the
  content from template Template1 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
<code>[required]</code>	
<code>--destination, -d</code>	The registered OCM server to create or update the site <code>[required]</code>
<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 Sitel -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 Sitel -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 Sitel -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 Sitel -c PageIndex
cec index-site Sitel -c PageIndex -p
cec index-site Sitel -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:

<code>--help, -h</code>	Show Help
<code>--url, -u</code>	<url> Site URL [required]
<code>--changefreq, -c</code>	How frequently the page is likely to change.
<code>--file, -f</code>	Name of the generated site map file
<code>--languages, -l</code>	<languages> The comma separated list of languages used to create the site map
<code>--publish, -p</code>	Upload the site map to OCM server after creation
<code>--toppagepriority, -t</code>	Priority for the top level pages, a decimal number between 0 and 1
<code>--server, -s</code>	<server> The registered OCM server
<code>--newlink, -n</code>	Generate new 19.3.3 detail page link
<code>--noDefaultDetailPageLink, -o</code>	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:

<code>--help, -h</code>	Show Help
<code>--url, -u</code>	<url> Site URL [required]
<code>--query, -q</code>	Query for content items [required]
<code>--limit, -l</code>	The limit of the items returned from the query [required]
<code>--orderby, -o</code>	The order by for the query [required]
<code>--language, -i</code>	The language for the query
<code>--template, -x</code>	The RSS xml template
<code>--javascript, -j</code>	Javascript file that contains functions to process Mustache data
<code>--title, -t</code>	The RSS feed title
<code>--description, -d</code>	The RSS feed description
<code>--ttl</code>	How long the data will last in number of minutes
<code>--file, -f</code>	Name of the generated RSS feed file
<code>--publish, -p</code>	Upload the RSS feed to OCM server after creation
<code>--server, -s</code>	<server> The registered OCM server
<code>--newlink, -n</code>	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                                Upload
content to repository Repol, creating new items, and add to channel
Site1Channel
cec upload-content Site1Channel -r Repol -u                            Upload
    
```

content to repository Repo1, updating existing content to create new versions, and add to channel Site1Channel

```
cec upload-content Site1Channel -r Repo1 -l Site1Collection
```

Upload content to repository Repo1 and add to collection Site1Collection and channel Site1Channel

```
cec upload-content Site1Channel -r Repo1 -p
```

Upload content types from content SiteChannel to the server

```
cec upload-content Site1Channel -r Repo1 -s UAT
```

Upload content to repository Repo1 on server UAT and add to channel Site1Channel

```
cec upload-content Templatel -t -r Repo1 -c channel1
```

Upload content from template Templatel to repository Repo1 and add to channel channel1

```
cec upload-content ~/Downloads/content.zip -f -r Repo1 -c channel1
```

Upload content from file ~/Downloads/content.zip to repository Repo1 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 Repol         List all assets
from repository Repol
cec list-assets -c Channel1      List all assets
from channel Channel1
cec list-assets -r Repol -l Collection1   List all assets
from collection Collection1 and repository Repol
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

```

Тестирование с помощью локальной тестовой оснастки

Запустите пользовательские компоненты, шаблоны и макеты контента в локальной тестовой оснастке, прежде чем импортировать их в Oracle Content Management.

Чтобы запустить локальную тестовую оснастку, выполните указанные ниже действия.

1. В окне терминала введите команду `cd cec`.
2. Введите команду `cec development &` или `cec deploy --server <server-name> &`
3. Откройте браузер по адресу `http://localhost:8085`, чтобы просмотреть компоненты, шаблоны и макеты контента, выполняемые в локальной тестовой оснастке.

4. Компоненты, шаблоны, темы и т. д. можно найти в следующих каталогах:
 - `cec/src/main/components`
 - `cec/src/main/templates`
 - `cec/src/main/themes`

Обновление до jQuery 3.5.x

Как разработчик вы можете использовать инструментарий OCE для выявления неправильно закрытых HTML-тегов.

Обновите до jQuery 3.5.x, чтобы получить исправление безопасности в HTML Parser.

<https://blog.jquery.com/2020/04/10/jquery-3-5-0-released>

Результаты анализа определенных HTML-строк в jQuery 3.5.x отличаются от результатов анализа тех же строк в версии 3.4.x. Эти строки содержат самозакрывающиеся теги, нарушающие стандарт HTML; например: "`<div />`". Синтаксический анализ таких строк, особенно в случае части более крупной последовательности тегов, может по-другому выполняться в версиях 3.5.x по сравнению с версиями 3.4.x.

Можно выполнить команду `cec create-asset-report <site>`, чтобы найти неправильно закрытые теги HTML в JSON-файлах страницы, а также HTML- и JS-файлах компонента.

Разработка для Oracle Content Management в Developer Cloud Service

Инструментарий OCE, интеграция с Developer Cloud Service, помогает разрабатывать шаблоны сайтов, темы, пользовательские компоненты и макеты контента для Oracle Content Management.

Инструментарий OCE позволяет использовать репозитории активов, файлы и папки в Oracle Content Management. Инструментарий OCE содержит инструменты для создания и разработки пользовательских компонентов и шаблонов сайтов, включая темы и макеты контента. Инструментарий включает в себя репозиторий Git, локальную тестовую оснастку для быстрой итеративной разработки и примеры модульных тестов, с которых можно начать работу.

Инструментарий OCE помогает выполнять указанные ниже задачи.

- Настройка локальной среды разработки для использования экземпляра Oracle Content Management в целях локальной разработки и тестирования компонентов, шаблонов, тем и макетов контента
- Создание компонентов, шаблонов сайтов и макетов контента на основе образцов, выполнение их в тестовой оснастке, изучение их и разработка компонентов, шаблонов, тем и макетов контента в среде Developer Cloud Service

- Импорт компонентов и шаблонов сайтов, созданных в Oracle Content Management, в среду Developer Cloud Service для управления источниками и дальнейшей разработки
- Экспорт компонентов, шаблонов и макетов контента из среды Developer Cloud Service для использования в Oracle Content Management
- Копирование существующего компонента, шаблона или макета контента
- Создание модульных тестов
- Оптимизация компонентов
- Развертывание компонентов и шаблонов в Oracle Content Management

В перечисленных ниже разделах описывается настройка Developer Cloud Service для разработки пользовательских компонентов, шаблонов сайтов и тем, а также макетов контента.

1. [Об использовании Developer Cloud Service](#)
2. [Вход в консоль Developer Cloud Service для Oracle Content Management](#)
3. [Создание шаблона в Developer Cloud Service](#)
4. [Добавление инструментария OCE в код проекта в новом репозитории GIT](#)
5. [Тестирование пользовательских компонентов, шаблонов и макетов контента с использованием локальной тестовой оснастки](#)
6. [Объединение изменений](#)

В перечисленных ниже разделах представлена информация об использовании инструментария OCE из Oracle Content Management.

- [Использование утилиты командной строки CEC](#)
- Разработка пользовательских компонентов с помощью Developer Cloud Service
- Разработка шаблонов в Developer Cloud Service
- Разработка макетов контента

Об использовании Developer Cloud Service

Oracle Developer Cloud Service — это облачная платформа для разработки ПО, предоставляемая как услуга (PaaS), и управляемая среда для инфраструктуры разработки приложений. Она обеспечивает стандартизированную интеграцию на основе стандартов открытого исходного кода для разработки, совместного использования и развертывания приложений в Oracle Cloud.

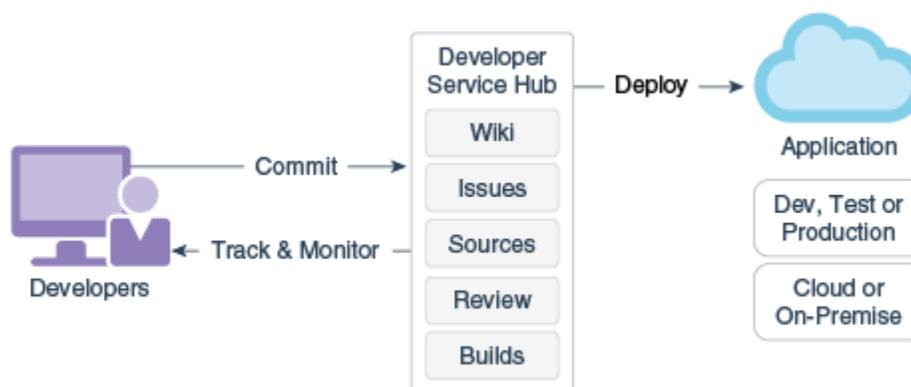
Developer Cloud Service — это коллекция программного обеспечения и услуг, размещенных в Oracle Cloud, которые помогают эффективно управлять жизненным циклом разработки приложений благодаря интеграции с Git, Maven, средствами решения проблем и wiki. С помощью Oracle Developer Cloud Service можно передать исходный код приложения в репозиторий Git в Oracle Cloud, отслеживать назначенные проблемы и дефекты в оперативном режиме, совместно использовать информацию с помощью wiki-страниц, просматривать исходный код и отслеживать сборки проектов. После успешного тестирования проект можно развернуть в Oracle Content Management.

Вход в консоль Developer Cloud Service для Oracle Content Management

Начните разработку пользовательских компонентов для Oracle Content Management на консоли Developer Cloud Service.

Администратор сервисов Oracle Cloud может использовать инструмент My Service Administration для создания облачных сервисов и управления ими. Если вы являетесь администратором экземпляров сервиса для Oracle Content Management и администратором сервиса для Standard Developer Service, вы можете настроить их и начать использовать.

1. Войдите в Oracle Cloud, используя информацию, предоставленную для вашей учетной записи.
2. Войдите в My Service Administration, чтобы создать экземпляр Oracle Content Management и сервис Standard Developer Service и управлять ими.



3. По запросу подтвердите свой адрес электронной почты Oracle Developer Cloud Service.
4. Настройте свой экземпляр Oracle Content Management, используя сведения о подписке для своего сервиса, и перейдите по URL-адресу Oracle Content Management для вашего экземпляра.
5. Перейдите по URL-адресу для Standard Developer Service.
6. Войдите в свою учетную запись Oracle Developer Cloud Service.

Перейдите по URL-адресу Developer Cloud Service и войдите в консоль.

Создание шаблона в Developer Cloud Service

Можно создать проект в Developer Cloud Service с помощью шаблона проекта "Content Experience Cloud" или проект с пустым репозиторием Git и импортировать инструментарий OCE из своего экземпляра Oracle Content Management.

- [Создайте проект Developer Cloud Service с помощью шаблона Oracle Content Management](#)

- [Создание проекта в Developer Cloud Service с помощью инструментария OCE из Oracle Content Management](#)
- [Добавление инструментария OCE в код проекта в новом репозитории GIT](#)

Создайте проект Developer Cloud Service с помощью шаблона Oracle Content Management

Создайте проект для разработки пользовательских компонентов, шаблонов, тем и макетов контента в Developer Cloud Service.

Для создания проекта:

1. После входа в консоль Developer Cloud Service нажмите **Новый проект**.
2. В списке шаблонов выберите **Content Management**, затем нажмите **Далее**.
3. В свойствах в разделе "Свойства проекта" выберите **CONFLUENCE** в поле **Разметка Wiki**.

Создание проекта в Developer Cloud Service с помощью инструментария OCE из Oracle Content Management

Создайте проект для разработки пользовательских компонентов, шаблонов, тем и макетов контента в Developer Cloud Service.

Для создания проекта:

1. После входа в консоль Developer Cloud Service нажмите **Новый проект**.
2. Назначьте имя своему проекту, введите или выберите другие нужные сведения о проекте, а затем нажмите **Далее**.
3. В списке шаблонов выберите **Исходный репозиторий**, затем нажмите **Далее**.
4. В свойствах в разделе "Свойства проекта" выберите **П у с т ы е р е п о з и т о р и и** для исходного репозитория. Нажмите кнопку **Готово**.

Добавление инструментария OCE в код проекта в новом репозитории GIT

Инструментарий OCE можно добавить в новый пустой репозиторий Git для проекта.

1. В разделе **РЕПОЗИТОРИИ** в новом проекте скопируйте URL-адрес HTTP репозитория Git проекта.
2. Откройте окно терминала и введите следующую команду: `git clone <your-project>.git`
 - a. При появлении запроса введите пароль для Developer Cloud Service.
 - b. Если отображается ошибка "git is not a command", установите Git со страницы <https://git-scm.com/downloads>, а затем повторно введите команду `git clone`.
3. `git clone git@github.com:oracle/content-and-experience-toolkit.git`

Инструмент также можно выгрузить по следующей ссылке: <https://github.com/oracle/content-and-experience-toolkit/archive/master.zip>

4. `cp -R content-and-experience-toolkit/sites/cec-components <your-project>`
5. `cd <your-project>`
6. `git add cec-components`
7. `git commit -a -m "<your comments>"`
8. `git push`

Тестирование пользовательских компонентов, шаблонов и макетов контента с использованием локальной тестовой оснастки

Запустите пользовательские компоненты, шаблоны и макеты контента в локальной тестовой оснастке, прежде чем импортировать их в Oracle Content Management.

Чтобы запустить локальную тестовую оснастку, выполните указанные ниже действия.

1. В окне терминала введите команду `cd cec-components`.
Введите `npm start &`
2. Откройте браузер по адресу `http://localhost:8085`, чтобы просмотреть компоненты, шаблоны и макеты контента, выполняемые в локальной тестовой оснастке.

При тестировании компонентов на локальном сервере можно использовать контент из локального шаблона или сервера Oracle Content Management.

Объединение изменений

После создания компонента, шаблона или макета контента или редактирования исходного кода на локальном компьютере необходимо объединить новые и измененные компоненты и шаблоны в хранилище Git проекта.

Чтобы объединить изменения в хранилище Git, в окне терминала введите по порядку следующие команды:

```
cd cec-components git pull
git add .
git status
git commit -a -m "Your comments" git pull
git push
```

Распространение изменений из среды тестирования в среду производства с помощью инструментария OCE

После разработки шаблона сайта интерфейс командной строки (CLI) в инструментарии OCE можно использовать в целях распространения шаблона с этапа разработки на этап тестирования, а затем на этап производства, на серверах Oracle Content Management.

В целях распространения изменений можно использовать команды инструментария для создания сайтов и управления жизненным циклом на серверах разработки, тестирования и производства. Можно вносить изменения в сайты в среде разработки и распространять эти изменения на среды тестирования и производства. Кроме того, этот набор утилит командной строки можно интегрировать в среды создания сценариев для управления развертываниями. С помощью утилит интерфейса командной строки можно создавать новые элементы, такие как активы и компоненты, а также обновлять существующий контент.

Описанные ниже действия показывают, как использовать интерфейс командной строки в инструментарии OCE для перехода из среды разработки в среду тестирования, а затем в среду производства.

1. Настройте серверы разработки, тестирования и производства с одним репозиторием и одной политикой локализации.

Чтобы распространить изменения с сервера разработки на сервер тестирования, а затем на сервер производства, необходимо настроить репозиторий с одинаковым именем и одной и той же политикой локализации всех трех серверах. Для репозитория активов политика локализации по умолчанию — en-US, но можно использовать другую политику, если она одинакова на всех трех серверах.

См. Настройка репозитория активов.

2. Зарегистрируйте свои серверы разработки, тестирования и производства с помощью Oracle Content Management.

Перед распространением изменений на сайт необходимо зарегистрировать каждый из серверов. Сервер можно зарегистрировать с помощью команды `cec register-server`, предоставляемой инструментарием OCE:

```
cec register-server <name>
```

Можно указывать следующие параметры команды:

- `-e <конечная точка>` для URL-адреса сервера;
- `-u <пользователь>` и `-p <пароль>` для подключения к серверу;
- `-t <тип>` — необязательный параметр, используемый для задания типа сервера. Значение по умолчанию: `pod_ec`.

При подключении к арендатору Oracle Content Management используйте только значение `pod_ec`.

Например, следующая команда регистрирует сервер, который выступает арендатором в Oracle Public Cloud:

```
cec register-server DEV -e https://DEV.example.com -u user1 -p <password>
```

Следующая команда регистрирует автономный экземпляр разработки Oracle Content Management:

```
cec register-server DEV -e https://DEV.git.oraclecorp.example.com -u user1 -p <password>
```

После регистрации сервера Oracle Content Management контент, содержащийся в нем, можно отобразить с помощью команды `cec list` инструментария.

Следующая команда отображает контент, содержащийся на сервере разработки:

```
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
  Enterprise    StarterSiteTheme

Templates:
  Name                               Theme                               Type
  StarterTemplate                    StarterTheme                        Standard
```

3. Загрузите шаблон сайта на сервер разработки и создайте сайт на основе этого шаблона.

Шаблон сайта можно создать с помощью команды `cec create-template`, а затем загрузить на сервер разработки. Затем на основе этого шаблона можно создать сайт с помощью команды `cec create-site`. Следующие команды создают шаблон и загружают его:

```
- cec create-template blog -f BlogTemplate
- cec upload-template blog -s DEV
```

Следующая команда создает сайт с именем `blog` на основе загруженного шаблона:

```
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
- site created

```

4. Опубликуйте сайт и переведите его в оперативный режим на сервере разработки.

После создания сайта можно использовать команду `cec control-site`, чтобы опубликовать сайт и перевести его в оперативный режим:

```

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. Чтобы переместить сайт из DEV в UAT, необходимо его упаковать. Модель упаковки для перемещения сайтов между серверами — это шаблон. Создайте новый шаблон на основе сайта, созданного вами на сервере разработки, и выгрузите шаблон.

Команда `cec create-template-from-site` в следующем примере создает шаблон сайта с именем `blog2` на основе шаблона `blog`.

```
cec create-template-from-site blog2 -s blog
```

Выгрузите созданный шаблон с сайта разработки, используя команду `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. Загрузите шаблон для создания типов контента и карт макетов контента.

```
cec upload-template blog2 --server UAT
```

7. Загрузите шаблон, но исключите из него элементы контента (шаблон контента).

```
cec upload-template blog2 --server UAT -x
```

Это необходимо для создания сайта с контентом, которое имеет те же GUID, что и исходный сайт. При создании сайта на основе шаблона с контентом, весь контент нового сайта получает новые GUID. Поскольку мы хотим разрешить обновлять контент, а не создавать новый, необходимо исключить контент из шаблона.

8. Создайте сайт на основе данного шаблона.

```
cec create-site blog -t blog2 -r r -l "en-US" -d "end-US" --server UAT
```

9. Загрузите шаблон контента в канал и коллекцию сайта. Это необходимо сделать, так как он был исключен из шаблона в ходе выполнения шага 7.

```
cec upload-content blog2 -t -r r -c blog -l "blog site" --server UAT
```

10. Опубликуйте сайт и переведите его в оперативный режим на сервере тестирования.

Используйте команду `cec control-site` для публикации сайта и перевода в оперативный режим:

```
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. Если затем внести изменения в свой сайт `blog` на сервере DEV, эти изменения можно распространить на сайт, который уже создан на сервере UAT.
12. Создайте еще один шаблон на основе своего сайта, чтобы шаблон содержал ваши изменения.

```
cec create-template-from-site blog3 -s blog --server DEV
```

13. Выгрузите шаблон.

```
cec download-template blog3 -s DEV
```

14. Загрузите шаблон и создайте на его основе сайт, чтобы распространить изменения на свою тестовую среду.

```
cec upload-template blog3 -s UAT
```

Эта команда создает или обновляет любые измененные компоненты и темы, исключая контент.

15. Теперь с помощью команды update-site выберите контент и обновите страницы.

```
cec update-site blog -t blog3 - UAT
```

Пример:

```
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. Проверьте сайт, чтобы убедиться в распространении изменений.

17. Чтобы перейти от сервера UAT к серверу PROD, выполните те же действия, что и при перемещении сайта от DEV к UAT.
18. Создайте сайт на сервере производства, переведите его в оперативный режим и проверьте изменения.

С помощью команды `cec list` можно просмотреть контент производственного сайта и убедиться, что оно содержит изменения, внесенные в среду разработки. Кроме того, можно проверить сайт, чтобы убедиться в том, что изменения распространены на производство.

Шифрование пароля

При регистрации сервера с помощью инструментария OCE необходимо зашифровать пароль, чтобы сделать сервер доступным для локального использования.

1. Зарегистрируйте сервер Oracle Content Management с помощью команды `cec register-server`, которая содержит пароль в виде обычного текста.
2. Зашифруйте пароль с помощью команды `cec create-encryption-key`.

```
cec create-encryption-key <file> [alias: cek]
Create an encryption key to encrypt/decrypt password for servers.
```

3. Снова зарегистрируйте сервер с помощью ключа шифрования, что делает сервер доступным для локальной разработки и тестирования.

Зашифрованные пароли хранятся в файле подключения к серверу. Пароль расшифровывается при подключении к зарегистрированному серверу.

Регистрация сервера

Сервер можно зарегистрировать в инструментарии OCE.

Для регистрации Oracle Content Management в целях локальной разработки и тестирования используется команда `cec register-server` с ключом шифрования.

При регистрации сервера шифрование пароля делает сервер доступным для использования с инструментарием OCE. См. [Шифрование пароля](#).

Создание отчета об использовании и разрешениях для сайта

Отчет можно создать для того, чтобы проверить и исправить разрешения участников целевого сервера для тестирования сайта для производства.

Примените команду `cec create-asset-report` и выполните проверку следующим образом:

1. Проверьте членство и назначение каналов для всех артефактов сайта:
 - Тема
 - Шаблон
 - Компоненты
 - Тип контента
2. Отметьте проблемы, которые можно решить.

Пример:

```
cec create-asset-report blog1 -s <registered-server> -o
cec create-asset-report trbcent -s <registered-server> -o
```

Отчет создается в JSON-файле, который можно проверить на наличие проблем с использованием и разрешениями. Для исправления разрешений доступны следующие команды:

- `cec share-type`: типы общих файлов для пользователей на сервере Oracle Content Management.
- `cec unshare-type`: удаление доступа к типам для заданных пользователей на сервере Oracle Content Management.
- `cec share-repository`: общий доступ к репозиторию (и типам, используемым репозиторием) на сервере Oracle Content Management.
- `cec unshare-repository`: удаление пользовательского доступа к репозиторию на сервере Oracle Content Management.
- Пример:

```
cec share-repository Repo1 -u <user-name1>,<user-name2> -r manager -
t -s <registered-server>
```

Выгрузка и загрузка документов и папок

На сервере Oracle Content Management возможны выгрузка и загрузка документов и папок.

Для выгрузки и загрузки документов и папок доступны следующие команды:

```
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>
```

С помощью команды `cec-share-folder <name>` можно предоставить пользователям общий доступ к папке на сервере Oracle Content Management и назначить пользователям роль. Укажите сервер с помощью параметра `-s <сервер>` или используйте сервер, указанный в файле `cec.properties`.

Допустимые роли:

- `manager`
- `contributor`
- `downloader`

- viewer

Для загрузок можно указать иерархию папок.

Создание сайта на основе шаблона и сохранение одинаковых GUID для контента

Как разработчик, вы можете использовать команду инструментария OCE для создания сайта Oracle Content Management на основе шаблона с сохранением тех же идентификаторов GUID для контента.

Используйте следующую команду инструментария OCE:

```
update create-site-from-template --reuse-content
```

Эта команда создает сайт на сервере Oracle Content Management и сохраняет идентификаторы контента при создании сайта. Сохранение идентификаторов контента необходимо для нескольких прогонов перехода от тестирования к производству, чтобы не допустить дублирования элементов контента на целевом сервере.

Создание корпоративного шаблона на основе стандартного сайта

Как разработчик, вы можете использовать команду интерфейса командной строки в инструментарии OCE для создания корпоративного шаблона на основе стандартного сайта.

По умолчанию команда `create-template` создает стандартный шаблон, если сайт является стандартным сайтом, и корпоративный шаблон, если сайт является корпоративным сайтом. Корпоративный шаблон также можно создать на основе стандартного сайта.

Выполните команду `cec create-template` с новым параметром `-enterprise`:

```
cec create-template EnterpriseTempl -s StandardSite1 -e
```

Импорт и экспорт таксономий

Используйте команды инструментария OCE для импорта таксономий с локального компьютера на сервер Oracle Content Management или для экспорта таксономий с сервера на локальный компьютер.

Команда `cec download-taxonomy <name>` экспортирует таксономию из Oracle Content Management. Происходит выгрузка таксономии с сервера Oracle Content Management.

В этой команде можно использовать следующие параметры:

- `--status, -t [promoted | published] [required]`: указывает статус таксономии.
- `--id, -i`: указывает идентификатор таксономии, если есть другая таксономия с таким же именем.

- `--server, -s`: укажите зарегистрированный сервер Oracle Content Management или используйте сервер, указанный в файле `cec.properties`.
- `--help, -h`: отображение справки команды.

Ниже приведены некоторые примеры команды `download-taxonomy`.

```
cec download-taxonomy Taxonomy1 -t promoted
```

```
cec download-taxonomy Taxonomy1 -i 6A6DC736572C468B90F2A1C17B7CE5E4 -  
t promoted
```

```
cec download-taxonomy Taxonomy1 -t published -s UAT
```

Команда `cec upload-taxonomy <taxonomy>` импортирует таксономию в Oracle Content Management. Загружает таксономию на сервер Oracle Content Management.

В этой команде можно использовать следующие параметры:

- `--createnew, -c`: создание новой таксономии.
- `--name, -n`: указывает имя новой таксономии.
- `--abbreviation, -a`: указывает сокращение для новой таксономии.
- `--description, -d`: указывает описание новой таксономии.
- `--file, -f`: указывает файл как источник таксономии.
- `--server, -s`: укажите зарегистрированный сервер Oracle Content Management или используйте сервер, указанный в файле `cec.properties`.
- `--T2P`: проверяет, существует ли актив или таксономия с таким же идентификатором в целевом экземпляре Oracle Content Management. В случае значения `true` актив добавляется как новая версия, а таксономия — как черновик (переопределение существующего черновика). В противном случае создается новый актив или таксономия с одинаковыми идентификаторами.
- `--New`: позволяет создать новый актив или таксономию в целевом экземпляре Oracle Content Management.
- `--help, -h`: отображение справки для команды.

Ниже приведены некоторые примеры команды `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

С помощью утилит с интерфейсом командой строки в инструментарии OCE, обеспечивающих переход из среды тестирования в среду производства, можно автоматизировать импорт или экспорт активов вместе с моделью контента и ее зависимостями с исходного сервера Oracle Content Management на целевой сервер Oracle Content Management.

Менеджер или администратор контента может с помощью команд инструментария OCE импортировать или экспортировать таксономии в среде перехода от тестирования к производству. Права менеджера позволяют добавить черновик таксономии. С правами администратора контента можно создать новую таксономию.

Операции жизненного цикла таксономии, такие как повышение уровня, назначение репозиторию и публикация, доступны для перехода от тестирования к производству. Активы вместе с информацией о категоризации и таксономиями можно импортировать в файл импорта или экспортировать из файла экспорта.

Импорт и экспорт рекомендаций

Команды инструментария OCE используются для импорта рекомендаций с сервера Oracle Content Management на локальный компьютер или для экспорта рекомендаций с локального компьютера на сервер.

Команда `cec download-recommendation <name>` экспортирует рекомендацию из Oracle Content Management. Происходит выгрузка рекомендации с сервера Oracle Content Management.

В этой команде можно использовать следующие параметры:

- `--status, -t [promoted | published] [required]`: укажите статус рекомендации.
- `--id, -i`: если у другой рекомендации такое же имя, укажите идентификатор рекомендации.
- `--server, -s`: укажите зарегистрированный сервер Oracle Content Management или используйте сервер, указанный в файле `cec.properties`.
- `--help, -h`: отображение справки команды.

Ниже приведены некоторые примеры команды `download-recommendation`:

```
cec download-recommendation Recommendation1 -t promoted
```

```
cec download-recommendation Recommendation1 -i  
A6DE836572C468B90F2A1C17B7CF5E4 -t promoted
```

```
cec download-recommendation recommendation -t published -s UAT
```

Команда `cec upload-recommendation <name>` импортирует рекомендацию в Oracle Content Management. Загружает рекомендацию на сервер Oracle Content Management.

В этой команде можно использовать следующие параметры:

- `--createnew, -c`: создание новой рекомендации.

- `--name, -n`: укажите имя новой рекомендации.
- `--abbreviation, -a`: укажите сокращение для новой рекомендации.
- `--description, -d`: укажите описание новой рекомендации.
- `--file, -f`: укажите, является ли рекомендация файлом.
- `--server, -s`: укажите зарегистрированный сервер Oracle Content Management или используйте сервер, указанный в файле `cec.properties`.
- `--help, -h`: отображение справки для команды.

Ниже приведены некоторые примеры команды **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

Добавление или удаление контента коллекции

Как разработчик вы можете добавлять контент в коллекции и удалять контент из коллекций с помощью команд инструментария OCE.

Следующая команда `cec-control-content` добавляет все элементы репозитория **Repo1** в коллекцию **Collection1** на зарегистрированном сервере **UAT**:

```
cec control-content add -l Collection1 -r Repo1 -s UAT
```

Следующая команда `cec-control-content` удаляет все элементы из коллекции **Collection1** на зарегистрированном сервере **UAT**:

```
cec control-content remove -l Collection -s UAT
```

Разработка пользовательских редакторов полей с помощью инструментария OSE

Инструментарий OSE поддерживает разработку компонентов типа "Внешний вид поля". Разработчики могут создавать пользовательские редакторы полей и управлять ими.

Для компонента типа "Внешний вид поля" можно выполнить следующие задачи:

- открыть, скопировать или удалить компонент;
- опубликовать компонент или отменить его публикацию;
- экспорт и импорт компонента;
- добавить или удалить элементы компонента;
- просмотреть свойства;
- выбрать логотип компонента.

Список компонентов можно фильтровать по типу "Внешний вид поля".

Для разработки пользовательских редакторов полей доступны следующие команды инструментария OSE:

```
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
```

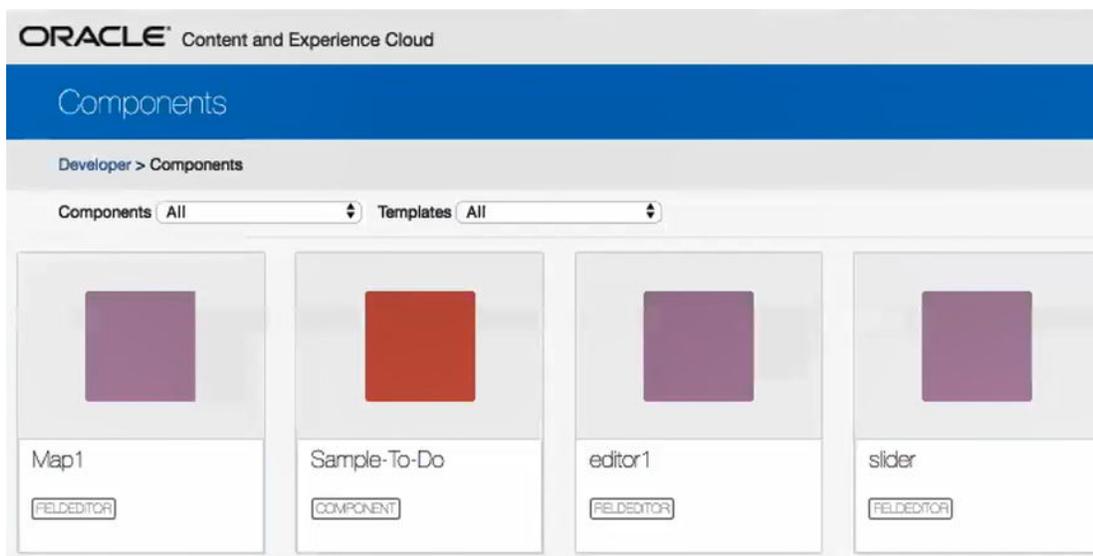
Эти образцы компонентов внешнего вида поля входят в инструментарий OCE:

- TextFieldEditor
- SliderFieldEditor
- MapFieldEditor

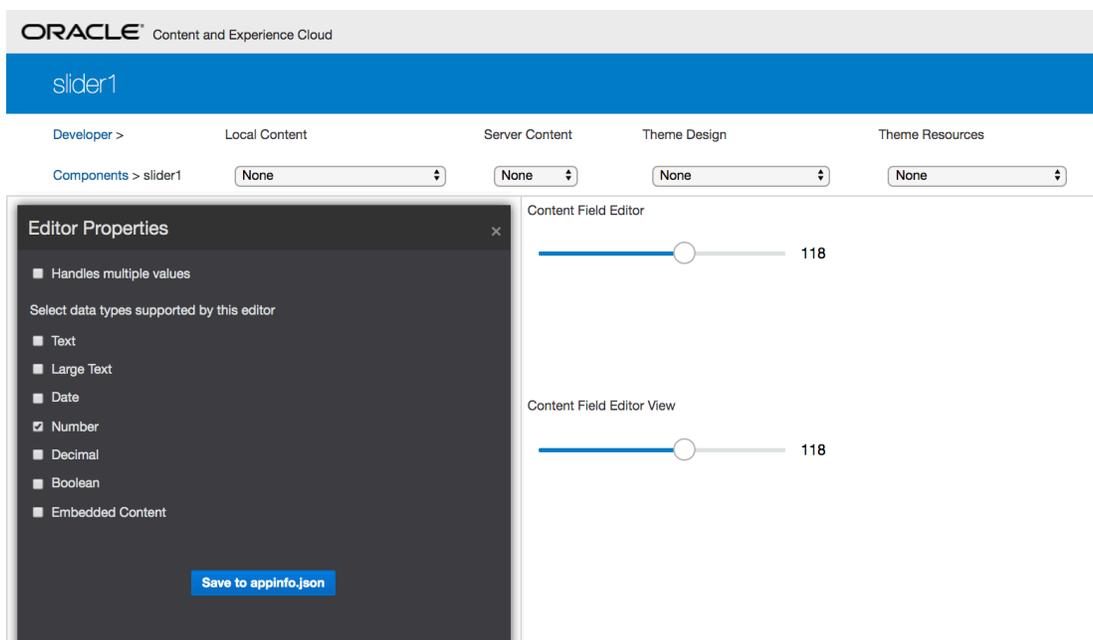
На рисунке ниже показаны команды инструментария OCE, которые можно использовать для разработки образцов компонентов внешнего вида поля.

```
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
```

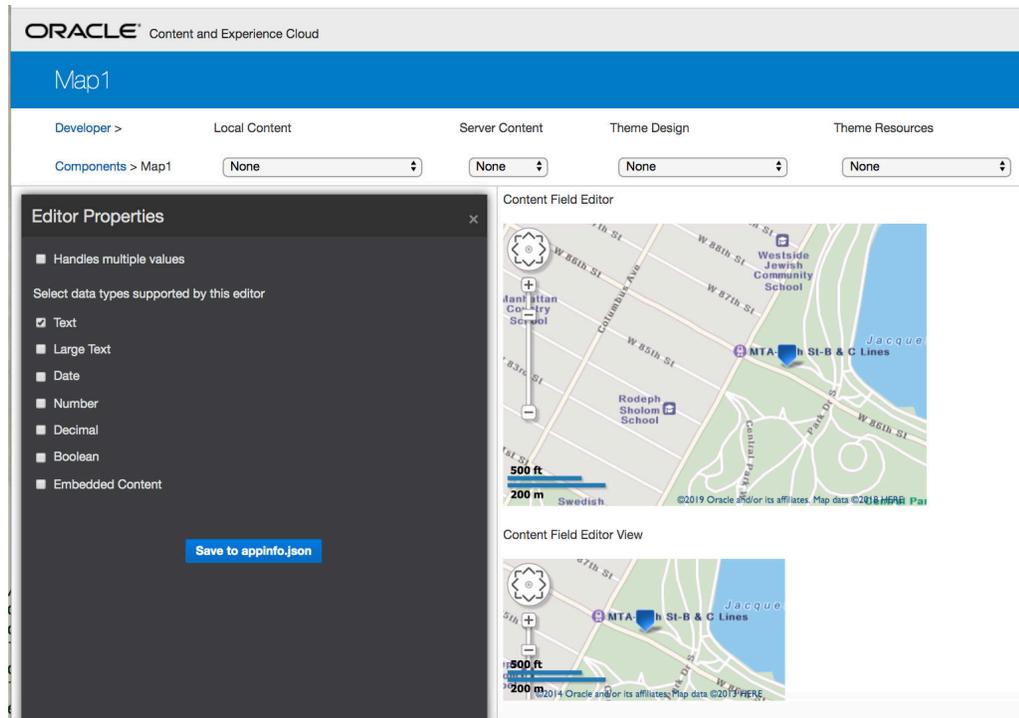
Можно создать на локальном сервере компоненты внешнего вида поля, протестировать их и загрузить в экземпляр Oracle Content Management. На рисунке ниже показаны эти компоненты в узле localhost:8085.



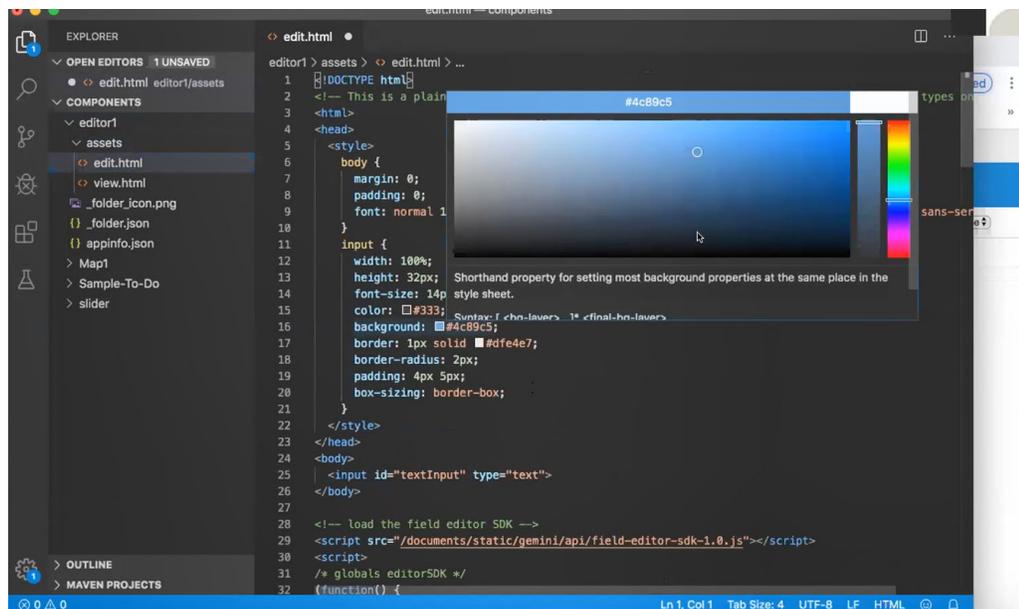
На локальном сервере можно протестировать любой компонент, например ползунок. Здесь можно выбрать свойства компонента и сохранить его.



В случае компонента карты можно нажать на карту, чтобы указать местоположение в качестве значения редактора.



Можно отредактировать HTML-файл компонента, чтобы изменить его настройки, такие как цвет фона.



С помощью команды инструментария OCE можно локально связать редактор полей с полем типа контента:

```
cec add-field-editor editor1 -t SimpleContent -n -c SimpleType -f title
```

```
cec add-field-editor slider -t SimpleContent -n -c SimpleType -f value
```

После завершения настройки и тестирования пользовательские редакторы полей можно загрузить в экземпляр Oracle Content Management с помощью команд инструментария OCE:

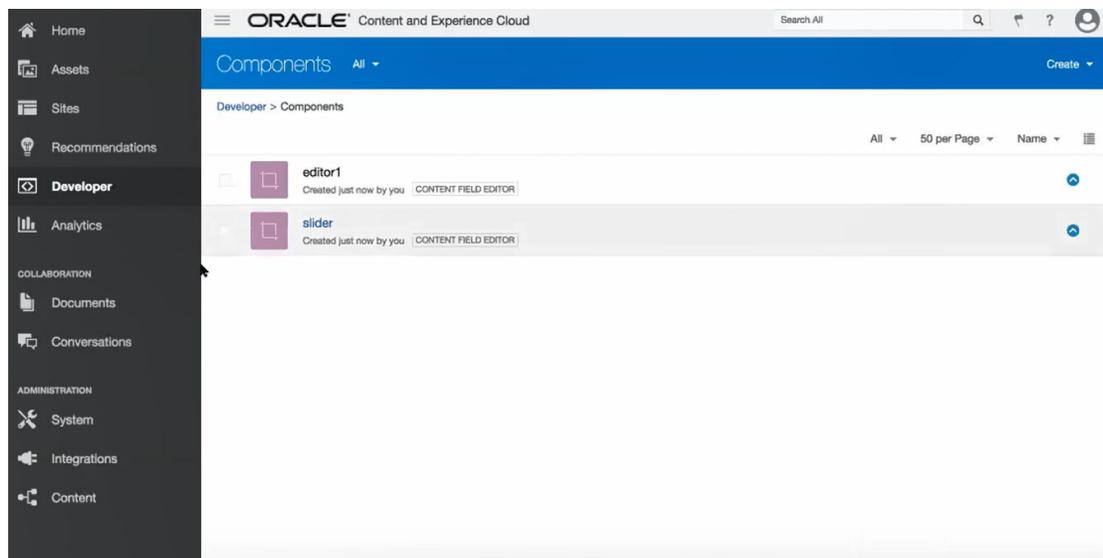
```
# upload editors
cec ulcp editor1,slider -p -s Latest

# upload content

cec cr Repo5 -s

cec upload-content SimpleContent -r Repo5 -s
```

При загрузке пользовательских редакторов полей компоненты импортируются в экземпляр Oracle Content Management.



Перенос сайта с одного сервера на другой или обновление

Разработчик может использовать команду инструментария OCE для создания или обновления сайта и его контента с сервера А на сервер В.

По умолчанию передаются все активы, для передачи только опубликованных активов можно дополнительно задать параметр `-p`. Укажите исходный сервер с помощью параметра `-s <server>`, а сервер назначения с помощью параметра `-d <destination>`. Если на сайте содержатся активы из других репозиториев, можно дополнительно указать сопоставление репозитория, в противном случае активы не будут передаваться.

Для обновления или перевода сайта из тестового в производственный режим можно использовать следующую команду:

```
cec transfer-site site --from server --to server --repository r
--localization-policy l
```

Дополнительные параметры см. в разделе [Использование утилиты командной строки CEC](#).

Передача сайта без элементов контента

Как разработчик вы можете использовать инструментарий OCE для передачи сайта без элементов контента с одного сервера Oracle Content Management на другой.

Для передачи сайта с большим количеством активов контента необходимо разделить сайт и его контент. В команду `transfer-site` можно добавить параметр `--excludecontent` (сокращение `-x`).

Если этот параметр задан, передается только сайт.

Пример:

```
cec transfer-site Site1 -s DEV -d UAT -r Repository1 -l  
LocalizationPolicy1 -x
```

Выгрузка или загрузка элементов контента для сайта в группах

Как разработчик, при перемещении сайта с одного сервера Oracle Content Management на другой, вы можете выгружать или загружать элементы контента сайта в группах.

Для передачи сайта с большим количеством активов контента необходимо разделить сайт и его контент. После перемещения сайта без контента (`cec transfer-site --excludecontent`) выполните команду `cec transfer-site-content` для переноса контента сайта.

Пример:

```
cec transfer-site-content <name>
```

Эта команда создает сценарии для переноса контента корпоративного сайта с одного сервера Oracle Content Management на другой. Эта команда используется для перемещения большого количества элементов контента. Эти элементы передаются пакетами. По умолчанию эта команда не выполняет сценарии, и перемещаются все активы. Можно указать параметр `-p` для переноса только опубликованных активов.

Укажите исходный сервер с помощью параметра `-s <server>`, а сервер назначения с помощью параметра `-d <destination>`.

Параметры

```
--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]
```

Если параметр `--execute` не задан, после завершения команды `cec transfer-site-content` исполните созданный сценарий `<site name>_downloadcontent` для выгрузки контента с исходного сервера и выполните `<site name>_uploadcontent` для загрузки выгруженного контента на сервер назначения.

Примеры

```
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
cec transfer-site-content Site1 -s DEV -d UAT -r Repository1 -p
```

Индексирование страниц сайта при помощи инструментария OCE

Инструментарий OCE можно использовать для создания элементов контента для текста на страницах сайта и включения поиска по страницам сайта.

В указанных ниже разделах описывается процедура индексирования страниц сайта с помощью инструментария OCE.

1. [Создание типа контента для текста страницы сайта](#)
2. [Создание элементов контента индекса страниц с помощью инструментария OCE](#)
3. [Добавление поиска контента на сайт в Oracle Content Management](#)

Создание типа контента для текста страницы сайта

Для типа контента указываются имя, значения обязательных полей и макет контента по умолчанию для типа.

- Имя типа
Укажите любое допустимое имя типа контента.
- Поля
Следующие поля являются обязательными.

Имя поля	Тип поля	Число значений в полях данных	Описание
site	Текст	Одиночные	Имя сайта
pageid	Текст	Одиночные	ID-адрес страницы
pagename	Текст	Одиночные	Имя страницы
pageurl	Текст	Одиночные	URL-адрес страницы
pagedescription	Текст	Одиночные	Описание страницы

Имя поля	Тип поля	Число значений в полях данных	Описание
keywords	Текст	Несколько (без макс.)	Весь текст на странице и значения всех текстовых полей элементов контента на странице, полученных командой инструментария OSE index-site

```

{{#fields}}
<div class="indextype"></div>
<div>
  <a href="{{pageFullURL}}" title="{{pagename}}">{{pagename}}</a>
</div>
{{/fields}}

```

```

content.fields.pageFullURL = SCSRenderAPI.getSitePrefix() +
content.fields.pageurl;

```

- Создайте макет контента для данного типа.

В макете контента должны отображаться имя сайта и URL-адрес перехода на страницу. Например, в `layout.html`:

```

{{#fields}}
<div class="indextype"></div>
<div>
  <a href="{{pageFullURL}}"title="
  {{pagename}}">{{pagename}}</a>
</div>
{{/fields}}

```

- В `render.js` создайте полный URL-адрес страницы:

```

content.fields.pageFullURL =
SCSRenderAPI.getSitePrefix() +
content.fields.pageurl;

```

- Задайте данный макет контента как макет контента по умолчанию для данного типа.

```

content.fields.pageFullURL =
SCSRenderAPI.getSitePrefix() + content.fields.pageurl;

```

Создание элементов контента индекса страниц с помощью инструментария OSE

Для создания элементов контента индекса страниц можно использовать команду инструментария OSE.

Предварительные условия:

- Инструментарий OCE установлен и настроен на локальном компьютере.
- Сайт Oracle Content Management опубликован.
- Элементы контента на странице сайта опубликованы в канале сайта.

В интерфейсе командной строки введите следующую команду инструментария OCE:

```
cec index-site site name -c content type name -p
```

В данной команде *site name* — это имя сайта, *content type name* — это тип контента, созданный для текста страницы; параметр `-p` указывает на публикацию элементов контента индекса страницы после создания.

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
```

Чтобы просмотреть использование, введите `cec index-site -h`

Добавление поиска контента на сайт в Oracle Content Management

На сайт Oracle Content Management можно добавить поиск контента с помощью страницы поиска и поля поиска.

Чтобы добавить на сайт поиск контента, выполните указанные ниже действия.

1. [Добавление страницы поиска к сайту](#)
2. [Добавление поля поиска в тему](#)

Добавление страницы поиска к сайту

На сайт можно добавить страницу поиска, а на эту страницу поиска — компонент "Список контента".

Добавление страницы поиска:

1. Добавьте страницу на сайт и задайте ее как страницу поиска.

2. Добавьте компонент "Список контента" на страницу поиска.
3. Задайте **Тип контента** типу контента индекса страницы, созданному ранее.

Добавление поля поиска в тему

Чтобы поле поиска отображалось на каждой странице сайта, можно добавить поле поиска на HTML-страницу макета темы.

Пример:

```
<div align="center">
<input id="searchonpage" type="text" size="30" placeholder="Search on
page. . ."/>
</div>
```

1. Добавьте поле ввода:

```
<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. Добавьте код JavaScript в конце текста HTML.

Индексирование многоязычного сайта с помощью инструментария OCE

Инструментарий OCE можно использовать в целях индексирования многоязычных сайтов (MLS) для переводов, поиска страниц и элементов контента.

Перед публикацией многоязычного сайта можно создать его индекс и протестировать. Для индексации многоязычного сайта используется команда инструментария OCE `cec index-site`. Перейдите в каталог `cec-components` и выполните эту команду без параметров, чтобы просмотреть справочной информации по данной команде:

```
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
```

Элементы индекса страниц существуют для каждой страницы и каждого языка. Элементы контента индекса страниц, созданные для каждого языка, созданы как переводы элементов индекса страниц на языке по умолчанию. При выполнении запроса на работающем сайте для поиска и списка контента язык выбирается из URL-адреса сайта. Результаты поиска при этом фильтруются автоматически.

Чтобы можно было опубликовать многоязычный сайт, его необходимо проиндексировать и перевести (для этого потребуется задание перевода). См. [Создание новой задачи перевода сайта или актива на сервере Oracle Content Management](#).

Требуется язык по умолчанию (английский). Для всех поддерживаемых языков (обязательных и необязательных) создайте индекс и переводы элементов индекса. Если запустить индексирование второй раз, просто выполняется обновление.

Чтобы индексировать, переводить и публиковать многоязычный сайт с помощью команд инструментария OCE, выполните указанные ниже действия.

1. Создайте для сайта тип контента и сделайте его доступным в репозитории. См. [Создание типа контента для текста страницы сайта](#).
2. Выберите политику проверки.

- a. В левом меню навигации нажмите **Активы**.
 - b. В меню **Активы** выберите **Политики локализации**.
 - c. Выберите политику локализации.
 - d. При необходимости измените политику локализации, чтобы включить языки, которые требуется использовать для индексирования и перевода сайта. Например, если в политике указан только английский язык, можно добавить французский и испанский языки.

Все переводы выполняются с английского языка.
3. Выгрузите задание перевода. Можно переводить только те активы, которые используются на сайте.
 4. **Переведите** сайт. Перевод материалов сайта можно выполнить вручную или с помощью встроенного коннектора.
 5. После завершения перевода загрузите задание перевода.
 6. С использованием команды `cec index-site` выполните индексирование сайта. Укажите `-c <contenttype>`, чтобы задать тип контента индекса страницы.

Также можно указать параметр `-p` для публикации сайта. Затем можно проверить индексирование и перевод перед публикацией изменений на работающем сайте.

Например, следующая команда `cec index-site` создает индекс сайта, использующего английский, французский и испанский языки. Языки, поддерживаемые сайтом, указаны в назначенной политике L10n, включая язык по умолчанию.

```
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. Опубликуйте сайт, чтобы включить переводы.

Создание упрощенного компонента для облегчения разработки компонентов

Используйте инструментарий OCE для создания упрощенного компонента для облегчения разработки.

SimpleHTML — упрощенный компонент, доступный в инструментарии OCE, который позволяет легко начать разработку пользовательских компонентов:

```
cec create-component -f SimpleHTML
```

Также доступен образец компонента JET с которого можно начать:

```
cec create-component MyComp -f JET-CCA-Demo-Card
```

Настройка сервиса компиляции сайтов

Можно создать образ Docker для сервера компиляции, чтобы настроить сервис компиляции сайтов в Oracle Content Management. Этот сервис позволяет определить сервер компиляции для использования перед публикацией.

У сервера компиляции есть конечная точка в форме URL-адреса на странице **Администрирование > Система > Сайты и активы** в поле **URL-адрес конечной точки компиляции**. Здесь можно ввести полный URL-адрес, который требуется зарегистрировать на сервере. Затем нажмите кнопку **Тест**, чтобы проверить конечную точку. См. Настройка URL-адреса конечной точки компиляции.

Сервис компиляции сайтов компилирует страницы сайта, чтобы при доступе к опубликованному сайту возвращались HTML-страницы. Когда сервер Oracle Content Management публикует сайт, для компиляции страниц он вызывает сервис компиляции сайтов, если тот настроен.

После указания конечной точки для сайта на вкладке **Статическая доставка сайта** диалогового окна **Свойства сайта** можно включить автоматическую компиляцию. При публикации сайта он компилируется посредством сервера компиляции.

Сервис компиляции сайтов — это расширение команд инструментария. Команду `cec compilation-server` можно выполнить самостоятельно, но образ Docker позволяет создать сервер компиляции в стандартном режиме, а затем настроить конфигурацию сервиса.

Чтобы создать образ Docker и опубликовать файл Docker:

1. Выгрузите информацию об образе Docker из GitHub.
При выгрузке информации из GitHub вас получаете три дополнительных файла, которые находятся на сервере компиляции. Один из образов Docker — это сервер компиляции, который содержит файл Docker и файлы readme.
2. Команда `run.sh` выполняется после создания образа Docker.
Эта команда выгружает инструментарий OCE, устанавливает его и ваш исходный каталог.
3. Затем можно настроить среду и выполнить команду `run.sh`. Можно указать порт для использования, тайм-ауты и любые другие настройки, которые необходимо изменить, а затем выполнить обновление и использовать команду `.sh`.
4. После выгрузки информации об образе Docker и изменения каталога сервера компиляции можно создать файл Docker.

Если инструментарий OCE уже выгружен, параметр no-cache можно удалить, чтобы не выгружать все файлы повторно.

5. Выведите список образа Docker и других образов.
6. Как только образ Docker станет вам доступен, зарегистрируйте его.
7. Убедитесь, что он работает.
Перейдите к файлу исключений REST API, который позволяет увидеть, какие версии поддерживаются. Этот тест позволяет проверить, что он действительно работает.
8. Теперь необходимо зарегистрировать сервер. Для этого требуется первая часть имени сервера. Все остальное будет обрабатываться внутри.
9. Вернитесь в свой системный каталог, к активам своих сайтов. Здесь можно зарегистрировать конечную точку компиляции и нажать **Сохранить**.
10. Перейдите к свойствам своего сайта и укажите, какие файлы и когда публиковать.

Компиляция сайта для повышения производительности во время выполнения для страниц сайта

Компиляция сайта в Oracle Content Management может улучшить производительность во время выполнения и поведение страниц сайта. Компиляция обеспечивает это путем создания статического HTML-файла для каждой страницы сайта, который будет вести себя точно так же, как и исходная страница.

Обзор компиляции сайта

Файлы метаданных, составляющие "страницу" сайта, объединяются во время компиляции, чтобы избежать запросов сервера, которые обычно требуются во время выполнения. Затраты на рендеринг при каждом просмотре страницы фактически становятся затратами на однократную компиляцию.

Повышение производительности достигается благодаря сокращению количества запросов на этапе выполнения. Сюда может входить исключение всех запросов контента, поскольку результаты этих запросов можно компилировать непосредственно на странице.

Поведение улучшается благодаря немедленному рендерингу страницы. Таким образом, можно избежать таких проблем, как "появление неоформленного контента" или отображение компонентов на странице в неопределенном порядке. Например: нижний колонтитул отображается немедленно, а затем перемещается вниз страницы подобно другим элементам при рендеринге страницы.

Для компиляции страниц на сайте Oracle Content Management необходимо экспортировать сайт как шаблон, а затем использовать инструментарий OCE для компиляции шаблона и загрузки статических страниц, созданных для исходного сайта. Описанные далее действия помогут выполнить процесс компиляции шаблонов, а затем использовать эту модель для компиляции своих сайтов.

Взаимодействие с предварительным рендерингом

Предварительный рендеринг направлен на возврат результатов, подходящих для поисковой системы. Страницы, созданные с помощью предварительного рендеринга,

не должны выполняться подобно исходной странице, и никакой код JavaScript не выполняется. Просто возвращается HTML-код с предварительным рендерингом в браузере для поддержки текстового поиска SEO.

Ожидается, что статические страницы, созданные при компиляции сайта, выполняются в браузере и работают точно так же, как и исходные динамические страницы сайта. Сюда входит динамическое выполнение нескомпилированных компонентов и исполнение запросов времени выполнения, когда пользователю требуются в результатах самые последние данные, а не исправление данных во время компиляции.

Если администратор арендатора включил решение предварительного рендеринга, статические страницы также могут доставляться через сервер предварительного рендеринга. Это позволяет индексам и поисковым роботам обрабатывать части статических страниц, которые по-прежнему могут отображаться динамически, например списки контента.

Запросы контента возвращают элементы контента, которые включают значения полей форматированного текста, поэтому при использовании макетов контента для списков контента не нужно извлекать значения по отдельности.

Рендеринг страниц сайта контроллера

Команда `sec compile-template` делает возможным рендеринг страниц сайта непосредственно в браузере без использования контроллера страниц сайта Oracle Content Management.

Без применения компилятора для создания статических HTML-страниц на сайте Oracle Content Management для рендеринга страниц используется модель контроллера. Это включает в себя ряд запросов на получение информации о сайте, странице сайта и шаблонах, используемых для рендеринга страницы перед ее отображением в браузере.

Примечание.

Требуется больше запросов, таких как получение файла `controller.js`.

Рендеринг скомпилированных страниц сайта

Команда `sec cile-template` позволяет скомпилировать все эти действия в статический HTML-файл и повысить производительность сайтов во время выполнения путем уменьшения или удаления таких запросов сервера. Кроме того, можно еще больше сократить количество запросов ресурсов, которые не нужно изменять во время выполнения, и скомпилировать их в статические страницы.

В процессе компиляции по умолчанию выполняется работу существующего контроллера времени выполнения по созданию физической HTML-страницы для каждого файла `page.json` на сайте. После этого эти страницы можно развернуть на сайте для публикации вместе с сайтом и использования при переходе по соответствующему URL-адресу страницы.

Компиляция шаблонов

Для компиляции сайта нужен доступ к темам и компонентам сайта. Шаблон сайта — это модель упаковки сайта. Шаблон содержит все необходимые ресурсы для миграции и запуска сайта.

Шаблон сайта можно использовать для экспорта сайта в среду инструментария OCE, где можно скомпилировать сайт в шаблоне. В указанных ниже разделах описывается компиляция сайта в шаблоне **BlogTemplate**, который предварительно задан в инструментарии OCE.

Настройка

Команда `cec cile-template` доступна посредством инструментария OCE и устанавливается вместе с установкой этого инструментария. Следуйте стандартным инструкциям по установке инструментария OCE.

После установки инструментария OCE можно выполнять команды `cec` для создания шаблонов.

1. Установите каталог разработки инструментария OCE:

```
mkdir cec-src
cd ./cec-src
cec install

# Start up the CEC Toolkit Development server for testing
cec develop &
```

2. Создайте шаблон.

Предварительно заданный интерфейс `BlogTemplate` обновлен с добавлением настраиваемых компиляторов. Создание шаблона на основе этого источника:

```
# create a new template (must be in the cec-src folder if not already
there)
cd ./cec-src
cec create-template BlogTemplate -f BlogTemplate
```

Предварительно заданный интерфейс **BlogTemplate** расширяет возможности предыдущего шаблона с помощью настраиваемых компиляторов:

- Компиляторы макетов контента

```
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
```

- Компиляторы макетов страниц

```
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
```

Компиляция своих шаблонов

При компиляции шаблона создается статическая HTML-страница для каждой страницы в файле `structure.json` для сайта в шаблоне.

На этапе компиляции метаданные страницы (например, `100.json`) объединяются с макетом страницы (например, `about.html`), чтобы получить HTML-страницу, которая отображается немедленно без использования контроллера во время выполнения. Пользовательские компиляторы позволяют еще больше сократить выполнение кода JavaScript во время выполнения, скомпилировав в странице ранее динамические элементы, такие как элементы навигации и компоненты.

Команда `cec cpile-template` выполняет для каждой страницы сайта следующие действия:

- Чтение метаданных сайта и страниц.
- Чтение макета страницы и примените любого компилятора макетов страниц.
- Для каждой ячейки на странице:
 - Разверните ячейку с сеткой, заданной в файле `page.json`.
 - Для каждого компонента в ячейке:
 - * Примените любой компилятор компонентов.
 - * Вставьте созданную разметку компонента в соответствующее место внутри ячейки.
- Разверните все макрокоманды в скомпилированной разметке и вставьте JavaScript-объект SCS, используемый рендерером Oracle Content Management во время выполнения.
- Сохраните скомпилированную разметку страницы в папке `src/templates/<template>/static`.

Чтобы составить сайт, выполните следующую команду:

```
> 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.
```

Ошибки компиляции

Во время компиляции пользователь получает сообщения трех типов: информационные, предупреждающие и об ошибках.

1. Информационные сообщения касаются проблем, о которых вам следует знать, но эти сообщения, вероятно, ожидаются как часть компиляции:
 - Элементы контента с заполнителями, рендеринг которых осуществляется во время выполнения.
 - Отсутствуют компиляторы макетов страниц. Макеты страниц могут не содержать динамических элементов, поэтому их отсутствие не считается проблемой.
 - Элементы снабженные меткой "рендеринг при доступе", то есть разработчик сайта хочет, чтобы рендеринг этого компонента осуществлялся динамически во время выполнения, даже если страница скомпилирована.
2. Предупреждающие сообщения касаются проблем, которые, скорее всего, могут повлиять на эффективность работы сайта и по возможности должны быть устранены. Элементы этой категории:

- Отсутствуют компиляторы макетов контента или пользовательских компонентов. Без них компоненты динамически отображаются на странице, как это было ранее.
 - Отсутствуют карты макетов контента. Маловероятно, что для рендеринга элементов контента вам захочется использовать системный макет контента по умолчанию.
3. Сообщения об ошибках касаются проблем, которые указывают на сбой компиляции. Страницы по возможности компилируются, но общая компиляция будет закрыта с ошибкой.
- Скорее всего, это вызвано ошибками JavaScript в пользовательских компиляторах. Все ошибки должны быть устранены.

 **Примечание.**

Чтобы сократить количество повторяющихся сообщений, одно и то же сообщение отображается только один раз за компиляцию независимо от того, выполняется ли оно на одной или на нескольких страницах.

Предыдущий пример имеет следующий статус при выходе из компиляции:

```
Compilation completed with 0 errors and 3 warnings.
to display warnings, run with --verbose (-v) option.
```

При повторном выполнении компиляции с параметром "-v" отображается следующий результат:

```
> 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-ydeveloping-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
```

Компиляция страниц сведений

Компиляция страницы сведений объединяет все элементы контента, которые встречаются во время компиляции. Затем программа повторно компилирует страницу сведений для каждого обнаруженного элемента контента, используя значение кода для определения URL-адреса новой страницы сведений.

Вывод предыдущей компиляции осуществляется в двух разделах:

1. Компиляция страниц
2. Компиляция страниц сведений

В предыдущем примере отображались предупреждения об отсутствии записей в карте макетов контента для страницы **Starter-Blog-Author**. Однако нам не нужны страницы сведений для **Starter-Blog-Author**. Страница сведений предназначена только для элементов контента **Starter-Blog-Post**. Чтобы удалить эти ошибки, мы можем исключить элементы контента из компиляции страницы сведений, если в их настройках нет явных ссылок на страницы сведений. Для этого используется следующий параметр:

```
--noDefaultDetailPageLink, -o Do not generate compiled detail page for
items/content lists
    that use the default detail page.
```

Таким образом, повторная компиляция, за исключением создания страницы сведений, при котором используется страница сведений по умолчанию, создает следующие выходные данные:

```
> 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

```

Компиляция определенных страниц

Не требуется компилировать все страницы в шаблоне. Можно выбирать страницы для компиляции. Это полезно в случае работы с определенными страницами, которые требуется отлаживать или обновлять без постоянной повторной компиляции всего сайта.

Чтобы компилировать определенные страницы, укажите параметр `--pages (-p)`, за которым следует список нужных страниц.

Примечание.

Примечание. Если элементы контента на страницах в списке ссылаются на страницы сведений, эти страницы сведений также компилируются, даже если они не включены явно.

```

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

```

Компиляция сайта

Описанные выше действия содержат обзор процесса создания и компиляции локального шаблона. В целом, вы будете компилировать фактические сайты Oracle Content Management.

Чтобы создать сайт Oracle Content Management, необходимо экспортировать его в шаблон, а затем скомпилировать этот шаблон. Пакет шаблона будет содержать сайт, а также любой пользовательский код, необходимый для компиляции сайта в темах и компонентах, экспортируемых вместе с шаблоном.

Предварительные условия компиляции сайта

При выполнении описанных ниже действий предполагается, что вы создали сайт с названием **BlogSite** на сервере Oracle Content Management. Вы можете загрузить предыдущий шаблон **BlogTemplate**, а затем создать сайт из этого шаблона.

Пример:

```
> cec upload-template BlogTemplate --server UAT

> cec create-site BlogSite --template BlogTemplate --repository
<yourRepository> --
localizationPolicy <yourLocalizationPolicy> --defaultLanguage en-US --
server UAT
```

После выбора или создания сайта можно выполнить действия, описанные в перечисленных ниже разделах, чтобы скомпилировать сайт.

Компиляция сайта

Команда `cec create-template-from-site` из инструментария OCE позволяет создать шаблон сайта, чтобы можно было скомпилировать сайт.

```
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]
```

Для компиляции сайта выполните указанные ниже действия.

1. Создайте шаблон на основе сайта.

```
> cec create-template-from-site BlogTemplate --site BlogSite --
includeunpublishedassets -
server UAT
```

2. Выгрузите шаблон:

```
> cec download-template BlogTemplate --server UAT
```

3. Скомпилируйте шаблон:

```
> cec compile-template BlogTemplate --noDefaultDetailPageLink --verbose --server UAT
--channelToken e1bb88cdc1e025c8dd278f6b676877a3
```

Примечание.

Для своего сайта на сервере необходимо получить маркер канала публикации сайта (параметр `--channelToken (-c)`). Затем он будет использоваться для всех запросов на сайте в шаблоне.

4. Загрузите скомпилированные страницы сайта.

Скопируйте скомпилированные статические файлы в статическую папку сайта:

```
> cec upload-static-site-files src/templates/BlogTemplate/static --site BlogSite --server UAT
```

Возврат к режиму без компиляции

Чтобы вернуться к режиму без компиляции, необходимо удалить статические файлы, загруженные на сайт:

```
> cec delete-static-site-files BlogSite --server UAT
```

Если сайт опубликован, повторно опубликуйте его. При этом с сайта удаляется "статическая" папка; сайт не удаляется.

Пользовательские компиляторы

Процесс компиляции можно дополнительно улучшить с помощью индивидуальных компиляторов. Компиляторы можно вызывать для компиляции макета страницы, макета раздела, пользовательского компонента или макета контента на странице, чтобы не нужно было динамически добавлять компоненты во время выполнения.

Если для компонента нет пользовательского компилятора или компилятор компонента не возвращает разметку, то рендеринг компонента осуществляется во время выполнения, как если бы он не был скомпилирован.

Пользовательские компиляторы поддерживаются для указанных ниже компонентов.

- макеты страниц;
- Макеты разделов
- Пользовательские компоненты
- Макеты контента

Следующие образцы демонстрируют компиляторы макетов страниц и макетов контента. В компиляторах макетов разделов и пользовательских компонентов используется та же модель, что и в компиляторе макетов контента.

Если пользовательских компиляторов нет, рендеринг компонента осуществляется посредством файла `render.js`.

Если пользовательский компилятор есть, он вызывается, и полученный HTML-код вставляется в страницу. Пользовательский компилятор также может указывать на необходимость JavaScript-гидратации компонента во время выполнения. В этом случае вызывается функция `hydrate()` компонента в файле `render.js`. Если компоненту не требуется гидратация, файл `render.js` компонента никогда не загружается.

Примечание.

Компиляция — это возможность только времени выполнения (опубликованный сайт). При просмотре сайта в режиме редактирования, навигации или предварительного просмотра страницы работают в обычном режиме и все компоненты всегда динамически добавляются на страницу.

Ограничения

Операция `sec cpile-template` представляет собой приложение NodeJS и выполняется за пределами любого браузера. Поскольку страница не отображается в браузере, объект DOM или окна отсутствует, а библиотеки JavaScript на стороне клиента, такие как JQuery, VueJS или KnockoutJS, не будут работать.

Хотя можно использовать библиотеку, например JSDOM, для создания объекта DOM и выполнения таких библиотек на стороне клиента, в этом нет никаких преимуществ. Для простого синтаксического анализа HTML можно использовать средство синтаксического анализа NodeJS HTML, например Cheerio.

HTML-код, возвращенный разметкой, должен быть допустимым HTML-кодом. Он передается через синтаксический анализатор, и на страницу добавляется только проанализированный HTML-код. Это необходимо для подтверждения того, что скомпилированный HTML-код не содержит несоответствующих тегов, которые могут повредить ячейки.

Отладка настраиваемых компиляторов

При разработке настраиваемых компиляторов потребуется отладка кода.

Команда `sec compile-template` поддерживает параметр `--debug (-d)`, который запускает компилятор с установленным флагом `--inspect-brk`, чтобы отладчик можно было привязать к процессу. Затем можно выполнить стандартную отладку узла для проверки кода.

```
sec 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>

Кроме того, с помощью параметра `--pages (-p)` можно ограничить страницы, которые будут скомпилированы для определенной страницы.

Компиляторы макетов страниц

Компилятор макетов страниц представляет собой JavaScript-модуль NodeJS (CommonJS), который компилирует соответствующий макет страницы.

Компилятор определенного макета страницы определяется связью имен с расширением `-compile.js`:

- `src`
 - `themes`
 - * `<yourTheme>`
 - * `layouts`
 - * `<yourPageLayout>.html`
 - * `<yourPageLayout>-compile.js`

Если для макета страницы отсутствует файл `-compile.js`, то настраиваемая компиляция не применяется.

Компилятор макета страницы должен реализовывать интерфейс `compile()`, который возвращает ожидаемый объект; например `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();

```

Компиляторы компонентов

Все настраиваемые компиляторы поддерживают ту же модель, что и компиляторы страниц, и могут быть созданы для макетов разделов, пользовательских компонентов и макетов контента.

Во время компиляции команда `sec compile-template` ищет файл `cfile.js` в том же месте, что и файл `render.js` для компонента:

- `src`
 - `components`
 - * `<yourComponent>`
 - * `assets`
 - * `render.js`
 - * **`compile.js`**

Если этот файл не существует, компонент не компилируется и его рендеринг будет осуществляться во время выполнения.

Если файл существует, необходимо реализовать интерфейс `compile()`, который возвращает ожидаемый объект. Например, следующий компонент `Starter-Blog-Author-Summary` — это настраиваемый компилятор макета контента:

```

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;

```

Компиляторы макетов контента

Компиляторы макетов контента используются для вывода HTML для макета контента при публикации элемента контента определенного типа.

Во время компиляции команда `scs compile-content` ищет файл `compile.js` в том же месте, что и файл `render.js` для компонента:

- `src`
 - `components`
 - * `<yourComponent>`
 - * `assets`
 - * `render.js`
 - * `compile.js`

Если этот файл не существует, макет не компилируется и его рендеринг будет осуществляться во время выполнения.

Если файл существует, необходимо реализовать интерфейс `compile()`, который возвращает ожидаемый объект. Например, ниже приведен компилятор макетов контента, за которым следует результирующий вывод `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,
        '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
    });
}
};

module.exports = ContentLayout;

```

Полученный файл `layout.html` можно вставить в веб-приложение или в любое другое место, где требуется статический HTML-вывод макета контента. Файл `layout.html` выводится в тот же каталог, что и файл `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

Подобно `SCSRenderAPI`, каждой функции компиляции передается интерфейс `SCSCompileAPI`.

Он содержит следующие свойства и функции.

- Свойства:
 - **navigationRoot**: получает идентификатор узла, который представляет собой корень сайта.
 - **navigationCurr**: получает идентификатор текущего узла страницы.
 - **structureMap**: все узлы иерархии сайта с доступом по идентификатору.
 - **siteInfo**: все свойства сайта.
- Функции:
 - **getContentClient**: получает экземпляр `contentClient` для использования в вызовах Content API.
 - **getCustomSiteProperty**: получает пользовательские свойства во время компиляции сайта.
Пользовательские данные и свойства можно сохранить на уровне сайта в файле `siteinfo.json`, чтобы настроить сайт на основе данных и свойств.

Например, можно изменить текстуру фона страницы, ссылки нижнего колонтитула или имя веб-сайта.

Гидратация компонентов

Под гидратацией понимается процесс добавления сценария JavaScript обратно в скомпилированный HTML-код страницы во время рендеринга HTML-кода в браузере.

Например, если на странице есть два компонента, которые при рендеринге рассматриваются как главный/подчиненный, то для обновления подчиненного компонента необходимо нажать на соответствующий элемент в главном компоненте. Все это обрабатывается кодом JavaScript, исполняемым на странице. Чтобы выполнить эту работу, необходимо *гидратировать* HTML-код двух компонентов после их рендеринга на странице, добавив обработчик событий `on click` к элементам в главном компоненте и прослушиватель в контейнере подчиненного компонента для повторного рендеринга при возникновении события `on click` на основе полезных данных, переданных в событии.

Компиляторы компонентов вставляют HTML-код в страницу. Если для добавления таких объектов, как обработчики событий, необходимо во время выполнения исполнить дополнительный код JavaScript, то у вас есть несколько вариантов: встроенный JavaScript или функция гидратирования. Выбор решения зависит от ваших требований.

Встроенный JavaScript

Можно вставить тег `<script>` непосредственно в возвращенную скомпилированную разметку. Данный сценарий будет выполняться по мере выполнения страницы.

Пример:

```
<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>

```

Функция гидратации

Вместо ввода кода JavaScript в файл `render.js` можно включить функцию гидратации и отметить, что при возврате скомпилированной разметки компоненту требуется *гидратация* во время выполнения. Гидратация позволяет избежать повторяющихся тегов `<script>` и использовать существующий код JavaScript для управления событиями.

Несмотря на то, что файл `render.js` загружен, функция `render()` *не* вызывается во время гидратации. Вызывается только функция `hydrate()`.

Примечание.

Если скомпилированному компоненту не требуется гидратация, файл `render.js` компонента никогда не загружается.

Например, пользовательский компилятор макетов контента возвращается с кодом - `{ 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
});

```

Кроме того, при необходимости пользовательский компилятор может добавлять свойства *hydrate*, которые он ищет во время выполнения. Пример:

```

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}}">
```

Наконец, если компоненту требуется гидратация, во время выполнения загружается файл `render.js` компонента и вызывается функция `hydrate()`, которая передается в контейнере `<div>`, содержащем скомпилированную разметку.

Например, `render.js` — см. функцию `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);
    }
}
```

Публикация

Чтобы скомпилированные статические страницы стали активными после создания и загрузки в статическую папку сайта, необходимо опубликовать или повторно опубликовать сайт. Аналогичным образом, чтобы вернуться к режиму доставки нескомпилированного сайта, необходимо опубликовать или повторно опубликовать сайт после удаления из него статических файлов.

Во время публикации загруженные статические страницы становятся доступными для доставки. Поскольку эти файлы копируются в процессе публикации, производительность операции публикации может снижаться пропорционально количеству файлов.

В операции публикации берется текущий набор статических файлов и делается доступным для доставки. Возможна синхронизация таких файлов с любыми изменениями, которые произошли на динамическом сайте, и они могут отражать динамический сайт. Обновление коллекции статических файлов в соответствующее время остается на усмотрение разработчика сайта.

Приоритет доставки статического сайта

Когда сайт имеет связанные статические файлы, эти файлы доставляются для соответствующих URL-адресов, поступающих на сервер. Если входящий URL-адрес не соответствует статическому файлу, по запросу возвращается файл `controller.html` сайта. Это соответствует существующей динамической модели доставки сайта.

Сайты Oracle Content Management также могут определять перенаправления 301 и 302 посредством связанного JSON-файла. После настройки перенаправления становятся более приоритетными, чем статические файлы. Если URL-адрес соответствует правилу перенаправления и статическому файлу, с сервера доставляется перенаправление.

URL-адрес для доставки сайта оценивается следующим образом:

1. URL-адрес соответствует настроенному перенаправлению?
Если да, выдайте ответ для перенаправления.
2. URL-адрес соответствует статическому файлу?
Если для сайта настроен список статических агентов пользователей для мобильных устройств, а от браузера поступает запрос, соответствующий списку, то следует отправить статический файл для мобильных устройств.
3. В противном случае необходимо доставить динамический файл `controller.html` сайта.

 **Примечание.**

Если с сайтом связаны статические файлы для мобильных устройств, а клиент использует сеть CDN для доставки, то CDN (обычно Akamai) необходимо настроить так, чтобы кэшировать запросы браузеров мобильных устройств отдельно от стандартных запросов настольных устройств.

Если в CDN не настроено отдельное мобильное/стандартное кэширование, то браузеры для мобильных устройств могут принимать стандартные ответы, а браузеры настольных устройств могут получать ответы, предназначенные для браузеров мобильных устройств.

Заголовки кэширования

HTTP-заголовки в ответах веб-серверов помогают определить способ кэширования страниц браузерами. Статические страницы также поставляются с заголовками кэширования для облегчения кэширования в браузере.

Для защищенных сайтов вместе с ответами отправляются следующие заголовки:

- Cache-Control: no-store
- Pragma: no-cache

Для стандартных незащищенных сайтов отправляются указанные ниже заголовки.

- Cache-Control: max-age=300
 - Edge-Control: !no-store,max-age=2592000,downstream-ttl=1800
- Заголовок Edge-Control помогает упростить кэширование CDN.

Если настроить заголовки в одной из этих двух областей, ответ будет содержать пользовательские заголовки вместо стандартных, перечисленных здесь.

Эти ответы можно контролировать на уровне арендаторов или на уровне сайтов.

Страницы сведений

Страницы сведений на сайтах Oracle Content Management позволяют отображать на одной странице информацию о нескольких элементах контента.

Например, для обработки нескольких URL-адресов можно использовать одну и ту же страницу сведений. Каждый из таких URL-адресов отражает одинаковую структуру страниц, но отображает контент, связанный с элементами контента со следующими значениями кода: item1.html, item2.html и item3.html соответственно. В такой ситуации компилятор шаблонов ses может создать четыре файла:

- /detail/item1.html
- /detail/item2.html
- /detail/item3.html
- /detail.html

Последний файл позволяет отображать недавно опубликованный материал на веб-сайте без повторной компиляции и повторной публикации сайта. В данном примере элемент контента со значением кода `item4.html` публикуется после перехода сайта в онлайн-режим. Статическая страница `/detail.html` позволяет динамически отображать новый элемент на сайте. URL-адрес `/detail/item4.html` открывает страницу `detail.html`, но отображается контент, связанный с элементом контента `item4.html`.

Компилятор `sec` создает страницу `detail.html` для отображения элементов контента. По этой причине относительные URL-адреса на скомпилированной странице `detail.html` имеют дополнительные родительские сегменты (`./`). Итак, при наличии прямой ссылки страница `detail.html` отображается неправильно. Поэтому не следует делать ссылку на саму страницу `detail.html` или добавлять ее в навигацию по страницам.

Добавление элементов контента в канал

Команда инструментария OCE `control-content` позволяет добавлять элементы контента в канал на сервере Oracle Content Management.

Команда `control-content <action>` содержит действие `add` для добавления элементов контента в канал Oracle Content Management:

```
sec control-content add -c Channel1 -r Repo1 -s UAT
```

Эта команда добавляет все элементы репозитория `Repo1` в канал `Channel1` на зарегистрированном сервере `UAT`.

Можно указать сервер с помощью параметра `-s <server>` или использовать сервер, указанный в файле `sec.properties`.

Допустимые действия для команды `content-usage`:

- `publish`
- `unpublish`
- `add`
- `remove`

Параметры команды `content-usage`:

- `--channel, -c Channel` [обязательный]
- `--repository, -r Repository` [требуется, когда в качестве `<action>` у к а з а н о `add`]
- `--server, -s` Зарегистрированный сервер Oracle Content Management
- `--help, -h Show help` [логическое значение]

Примеры команд `control-content`:

```
sec control-content publish -c Channel1
```

Publish all items in channel `Channel1` on the server specified in the `sec.properties` file

```
sec 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

Компиляция сайта для мобильных устройств

Инструментарий OCE можно использовать для компиляции макетов веб-страниц сайта для мобильных устройств. Макет для мобильных устройств может отличаться от макета страниц для настольных устройств при одном и том же контенте. Также макеты для мобильных и настольных устройств могут быть одинаковыми.

В редакторе сайтов для мобильных и настольных устройств можно выбрать одинаковый или разные макеты страниц. С помощью инструментария OCE можно скомпилировать статический макет отдельно для мобильных устройств.

Такая страница сайта может выглядеть на мобильном устройстве по-другому. У страницы, рендеринг которой выполнен на мобильном устройстве, может не быть баннера, который есть у страницы с макетом для настольных устройств.

В инструментарии OCE на странице справки по команде `cec compile-template` отображается параметр `targetDevice`, предназначенный для определения целевого устройства при компиляции шаблона сайта:

```
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.
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.
```

При компиляции сайта можно указать назначение компиляции: для настольных устройств `desktop` или для мобильных устройств `mobile`. Файлы для настольных устройств находятся в папке `under static/_files`. Файлы для мобильных устройств размещаются в папке `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
```

После компиляции шаблона для мобильных устройств команда инструментария OCE `upload-static-site-files` поддерживает данные файлы для мобильных устройств.

Жизненный цикл сайта и скомпилированные страницы

При создании шаблона на основе сайта скомпилированные страницы не включаются в шаблон. Это позволит избежать проблемы с доставкой статических страниц, когда разработчик сайта ожидает доставку динамических страниц.

При последующем создании сайта на основе шаблона потребуется скомпилировать страницы сайта и загрузить их на новый сайт.

Создание новой задачи перевода сайта или актива на сервере Oracle Content Management

Используйте инструментарий OCE для создания в Oracle Content Management задания перевода сайта или актива.

Чтобы можно было индексировать многоязычный сайт, необходимо задание перевода. Чтобы создать задания перевода, выполните указанные ниже действия.

1. В верхнем меню страницы "Сайты" нажмите **Перевести**.
2. Введите имя задания в диалоговом окне **Создание задания перевода** и выберите исходный язык по умолчанию, целевые языки и содержимое задания перевода.

По выбору пакет перевода может включать весь контент сайта и целевые активы, только контент сайта или только активы, предназначенные для канала публикации сайта.

Исключите из перевода все элементы контента, сконфигурированные с помощью настройки текста **Не переводить**. Например, названия продуктов обычно не переводятся.

3. Нажмите **Создать**, чтобы создать задание перевода.
4. Используйте команду инструментария OCE для перечисления доступных заданий:

```
cec components> cec list-translation-jobs
Asset translation jobs:
Name                               Status
Source Language Target Languages    Pending
Languages

Site translation jbs:
Name                               Status
Source Language Target Languages    Pending
Languages

demol                               READY      en-
US          fr-FR,es-ES          fr-FR,es-ES

searchdemol                         TRANSLATED en-
US          fr-FR,es-ES
```

5. Выгрузите свое задание перевода:

```
cec components> cec download-translation-job demol
- 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 dmol.zip -l all -t demol-xlate.zip
- target languages: fr-FR,ex-ES
- translation finished: /Users/<user-name>/Dev/webclient/
  developers/sites-toolkit/cec-components/demol-xlate.zip
```

6. Откройте пакет перевода и создайте папки ресурсов для языков, на осуществляется перевод:

```
Unzip demol-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
```

7. Импортируйте задание перевода:

```
cec-components> cec import-translation-job demol-xlate.zip
- Logged in to remote server: <server url>
- file demol-xlate.zip uploaded to home folder, version 1
- importing: percentage 5
- importing: percentage 60
- import demol finished
```

Переводите сайт с помощью поставщика переводческих услуг

Переводом сайта на несколько языков можно управлять с помощью интерфейса командной строки в инструментарии OCE и поставщика переводческих услуг (LSP).

Политика локализации для сайта определяет язык по умолчанию, например английский (США), и один или несколько альтернативных языков сайта, таких как немецкий и французский. Текстовые строки для сайта можно перевести на указанные альтернативные языки. Если изменить язык сайта до перевода, текстовые строки отображаются на языке по умолчанию.

Инструментарий OCE предоставляет следующие возможности перевода в интерфейсе командной строки:

Translation	
cec list-translation-jobs	Lists translation jobs.
[alias: ltj]	[alias:]
cec create-translation-job <name>	Creates a translation job <name> for a site on CEC server.
[alias: ctj]	[alias:]
cec download-translation-job <name>	Downloads translation job <name> from CEC server.
[alias: dtj]	[alias:]
cec submit-translation-job <name>	Submits translation job <name> to translation connection <connection>.
[alias: stj]	[alias:]
cec ingest-translation-job <name>	Gets translated job <name> from translation connection and ingest.
[alias: itj]	[alias:]
cec upload-translation-job <name>	Uploads translation job <name> to CEC server.
[alias: utj]	[alias:]
cec create-translation-connector <name>	Creates translation connector <name>.
[alias: ctc]	[alias:]
cec start-translation-connector <name>	Starts translation connector <name>.
[alias: stc]	[alias:]
cec register-translation-connector <name>	Registers a translation connector.
[alias: rtc]	[alias:]

Команду `cec list-translation-jobs` можно использовать для перечисления заданий перевода, которые уже находятся на сервере. Пример:

```
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
```

Если ввести любую команду `cec` без параметров или с параметром `-h`, предоставляется справка для команды. См. [Использование утилиты командной строки CEC](#).

В указанных ниже разделах представлена информация о переводе сайта с помощью LSP.

1. [Создание задания перевода с помощью инструментария OCE](#)
2. [Перечисление заданий перевода](#)
3. [Создание коннектора перевода](#)
4. [Создание карты сайта для многоязычного сайта](#)
5. [Отправка задания перевода поставщику переводческих услуг](#)
6. [Загрузка задания перевода на сервер](#)

Создание задания перевода с помощью инструментария OCE

Для создания задания перевода сайта в свой локальной системе можно использовать команду инструментария OCE.

Чтобы создать новое задание перевода для сайта, используйте команду `cec create-translation-job`. Эта команда находит все активы сайта и создает ZIP-файл со всем контентом, который необходимо перевести на этом сайте.

```
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
```

Параметры перевода описываются в разделе [Создание новой задачи перевода сайта или актива на сервере Oracle Content Management](#).

Перечисление заданий перевода

Вы можете составить список заданий перевода на сервере, чтобы убедиться, что задание создано и готово к работе.

```
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
```

Обратите внимание, что задание FridayDemo находится в состоянии READY (Готово).

Создание коннектора перевода

Поставщик переводческих услуг (LSP) может помочь перевести сайт. С помощью коннектора перевода поставщику LSP можно отправлять и загружать задания перевода.

Перед отправкой задания перевода необходимо создать коннектор перевода. Чтобы перевести сайт без LSP, можно создать для работы имитацию коннектора перевода. С помощью команды `cec create-translation-connector` создайте коннектор перевода и выполните команду `cec start-translation-connector`, чтобы запустить его:

```
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
```

С помощью инструментария OCE протестируйте коннектор перевода, запустив его посредством ожидаемых API-интерфейсов.

1. Зарегистрируйте коннектор с помощью инструментария OCE.

```
>cec register-translation-connector
```

- Откройте инструментарий и перейдите на страницу "Коннекторы перевода".

```
>http://localhost:8085/public/translationconnections.html
```

- Выполните данные действия на странице проверки коннекторов перевода. В этих действиях используется файл `translationBundle.zip` в папке `/data` в среде коннекторов для проверки коннектора.

Для разработки коннектора перевода для Oracle Content Management можно использовать комплект SDK для коннекторов перевода. Данный комплект SDK служит примером NodeJS-реализации API-интерфейса коннектора перевода. Данный образец принимает ZIP-файл задания перевода Oracle Content Management, переводит все ресурсы в файле и возвращает новый ZIP-файл, содержащий все переводы.

Для использования SDK требуется, чтобы у пользователя был доступ к LSP для фактического перевода строк. В комплект SDK входит имитация сервера LSP для имитации ответов LSP — перед строками просто вставляется имя целевого языка.

Комплект SDK для коннекторов перевода состоит из трех основных модулей.

- Коннектор:** коннектор перевода, который реализует требуемый API-интерфейс коннектора перевода Oracle Content Management.
- Диспетчер заданий:** пример диспетчера заданий на основе файловой системы, который поддерживает состояние заданий коннектора, пока они переводятся поставщиком переводческих услуг.
- Поставщик:** внедрение определенного набора API-интерфейсов, необходимых вашему LSP для отправки документов на перевод и получения переведенных документов.

Можно скопировать имитацию поставщика перевода JS и реализовать в ней все нужные методы.

Создание карты сайта для многоязычного сайта

Используйте инструментарий OCE для создания карты многоязычного сайта и публикации карты на сайте.

С помощью команды `cec create-site map <site>` на сервере Oracle Content Management можно создать карту многоязычного сайта. Пример:

```
cec create-site-map Site1 -u http://www.example.com/site1
```

Эта команда проходит по структуре сайта, создает иерархию карты сайта, которая соответствует иерархии страниц сайта, и создает на сервере Oracle Content Management карту сайта по указанному URL-адресу сайта.

Параметры команды:

```
--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]
```

Допустимые значения параметра <changefreq>:

- always
- hourly
- daily
- weekly
- monthly
- yearly
- never
- auto

Примеры команды `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
```

Для публикации карты сайта создается обновление сайта, карта сайта обновляется, а обновление завершается.

Отправка задания перевода поставщику переводческих услуг

Инструментарий OCE предоставляет ZIP-файл, который можно отправить поставщику переводческих услуг, чтобы начать работу над заданием перевода.

Задание перевода можно отправить поставщику переводческих услуг (LSP) посредством коннектора перевода. Такая отправка занимает некоторое время, поскольку коннектору необходимо распаковать файл и отправить поставщику переводческих услуг все отдельные файлы. Затем LSP может создать проект для вашего задания перевода. После импорта файлов в проект можно начать выбирать файлы для перевода. После этого LSP начинает отслеживать статус переводов.

Чтобы проверить статус, перечислите задания перевода локально с помощью команды `cec list-translation-jobs` без параметров. Когда задание имеет статус `READY TO INGEST` (Готово к принятию), можно выгрузить ZIP-файл от LSP для принятия задания перевода. Коннектор перевода передал ZIP-файл поставщику переводческих услуг, LSP перевел список файлов, а коннектор получил файлы обратно от LSP, в ZIP-файле, который можно выгрузить и принять.

```
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
```

При принятии ZIP-файла задание перевода возвращается из коннектора в ваш инструментарий OCE.

```
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
```

После принятия ZIP-файла при локальном перечислении заданий перевода возвращенное задания перевод имеет статус TRANSLATED (Переведено).

```
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
```

Переведенное задание можно загрузить на сервер Oracle Content Management. Обычно сначала переводится часть работы, после чего она отправляется вам на проверку. Перевод сайта может занять несколько недель: принятие задание перевода, возвращенного поставщиком переводческих услуг, внесение исправлений в перевод и повторная отправка задания перевода.

Загрузка задания перевода на сервер

После завершения задание перевода можно загрузить на сервер Oracle Content Management, а затем проверить перевод на своем сайте.

Используйте команду `cec upload-translation-job` для загрузки ZIP-файла с переводом на сервер.

```
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
```

После загрузки задание перевода имеет на сервере статус INPROGRESS (выполняется):

```
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                          INPROGRESS      en-US
de-DE,fr-FR
```

Чтобы проверить перевод, можно проверить текстовые строки в активах на сайте, перевод которого осуществляется.

Часть VI

Приложения

Доступны указанные ниже приложения.

- [Учебное пособие. Разработка компонентов с помощью Knockout](#)
- [Ссылка на Sites SDK](#)
- [Устранение неполадок](#)

Учебное пособие. Разработка компонентов с помощью Knockout

В этом учебном пособии описывается работа с набором объектов JavaScript, которые используют стандартную функциональность Knockout ViewModel and Template для создания компонента, который хранится в каталоге компонентов Oracle Content Management.

- [Общие сведения и предварительные требования для разработки компонентов с помощью Knockout](#)
- [Шаг 1. Создание компонента](#)
- [Шаг 2. Проверка структуры рендеринга локального компонента](#)
- [Шаг 3. Проверка структура настроек локального компонента](#)
- [Шаг 4. Отображение нового свойства в компоненте](#)
- [Шаг 5. Регистрация триггеров](#)
- [Шаг 6. Срабатывание триггеров](#)
- [Шаг 7. Регистрация действий](#)
- [Шаг 8. Выполнение действий](#)
- [Шаг 9. Создание отдельного заголовка для каждого экземпляра компонента](#)
- [Шаг 10. Использование вложенных компонентов с внутрискрипчным редактированием](#)
- [Шаг 11. Поддержка различных макетов](#)
- [Шаг 12. Определение пользовательских стилей](#)
- [Шаг 13. Рендеринг компонента во встроенной рамке](#)
- [Шаг 14. Использование пользовательских стилей во время рендеринга компонента во встроенной рамке](#)
- [Шаг 15. Интеграция с процессом выполнения на странице операций "Отмена" и "Повтор"](#)
- [Шаг 16. Управление активами](#)
- [Обзор учебного пособия](#)

Общие сведения и предварительные требования для разработки компонентов с помощью Knockout

В данном учебном пособии представлены действия и процедуры проверки для создания образца компонента с помощью объектов JavaScript, использующих стандартные функциональные возможности Knockout JS ViewModel и Template.

Вы должны быть способны взять код, который упоминался в этой процедуре (предоставлен в файлах, предварительно созданных при создании компонента), и обновить только шаблон `.html` и JavaScript `viewModel` только с учетом собственного кода.

 **Примечание.**

Хотя Oracle Content Management не диктует технологию JavaScript, используемую для создания компонентов, реализованная по умолчанию функция JavaScript одинакова для каждой реализации компонента в любой выбранной платформе JavaScript.

Предварительные условия

В данном учебном пособии уделяется внимание только реализации компонентов. Дополнительную общую информацию о компонентах см. в разделе Разработка компонентов.

Для выполнения действий, описанных в данном учебном пособии, необходимо соответствовать следующим требованиям:

- Нужен доступ к экземпляру Oracle Content Management с разрешениями на создание сайтов и компонентов.
- Сервер экземпляров Oracle Content Management синхронизирован с локальным компьютером с помощью настольного приложения Oracle Content Management или пользовательского компонента. См. Разработка пользовательских компонентов с помощью Developer Cloud Service.

Кроме того, необходимо ознакомиться со следующими концепциями и платформами JavaScript:

- Отладка JavaScript в браузере
- JavaScript Closure
- Разработка JavaScript Asynchronous Module Definition (AMD)
- Платформы RequireJS и KnockoutJS

Теперь выполните [Шаг 1. Создание компонента](#).

Шаг 1. Создание компонента

В ходе выполнения этого шага объясняется, как создать пользовательский компонент в Oracle Content Management.

При создании пользовательского компонента его необходимо зарегистрировать для использования платформой Oracle Content Management. Чтобы проинформировать Oracle Content Management о своем компоненте, зарегистрируйте компонент на странице "Компоненты" в Site Builder.

Есть компоненты двух типов, которые требуется зарегистрировать.

- **Локальный компонент:**
 - Это компоненты, файлы которого хранятся на сервере экземпляров Oracle Content Management.

- Основное преимущество заключается в том, что не нужно беспокоиться о проблемах между доменами или между протоколами, так как файлы находятся на вашем сайте.
- Недостаток — невозможность выполнения логики среднего яруса на сервере Oracle Content Management, вследствие чего необходимо использовать API-интерфейсы REST для удаленных серверов, поддерживающих CORS.
- Компонент этого типа может быть встроен непосредственно в страницу, или можно использовать встроенную рамку для рендеринга компонента на странице.
- **Удаленный компонент:**
 - Компонент, файлы которого хранятся на удаленном сервере. Для такого компонента регистрируются только URL-адреса панелей "Рендеринг" и "Настройки".
 - Удаленный компонент обеспечивает преимущество, если у вас есть имеется логика на стороне сервера, которая должна выполняться при создании контента для вашего компонента.
 - Недостаток — для доступа к этим URL-адресам необходимо устранить все междоменные проблемы и проблемы безопасности.
 - Для отображения на странице удаленных компонентов всегда используется встроенная рамка.

Чтобы создать и зарегистрировать локальный компонент, выполните указанные ниже действия.

1. На домашней странице Oracle Content Management щелкните **Разработчик**.
Отображается страница **Разработчик**.
2. Нажмите **Смотреть все компоненты**.
3. В меню выберите **Создать локальный компонент**.
4. Введите имя компонента, например **A_Local_Component**.
5. Введите дополнительное описание.
6. Нажмите **Создать**.

После этого в списке компонентов появится компонент с именем `A_Local_Component`.

Проверка результатов для шага 1

Сейчас, после успешного создания компонента, он должен отображаться в палитре "Компонент" для любого созданного вами сайта. Для проверки создания компонента выполните следующие действия:

1. Создайте сайт с именем `localComponentTest`.
2. Выберите сайт и нажмите **Открыть**.
3. Нажмите **Редактировать**.
4. Создайте обновление для сайта и присвойте ему имя и, при необходимости, описание.
5. Выберите страницу на сайте.

6. Нажмите  на боковой палитре и выберите **Пользовательский**, чтобы отобразить список пользовательских компонентов.
7. В списке "Пользовательский компонент" выберите `A_Local_Component` и перетащите его на страницу.
Теперь вы должны видеть рендеринг по умолчанию созданного локального компонента.
8. Выделите  на баннере для компонента, который вы только что перетащили на страницу.
9. Выберите **Настройки**.
10. Измените выравнивание и задайте стиль для компонента.
11. Закройте панель "Настройки".

В описании указанных ниже действий объясняется, как создать пользовательский компонент и как его изменить в собственных целях. Теперь выполните [Шаг 2. Просмотр структуры локального компонента](#).

Шаг 2. Проверка структуры рендеринга локального компонента

В ходе выполнения этого шага мы проверяем структуру файлов по умолчанию, указанных для локального компонента.

Может показаться, что для простого примера `Hello World` четыре объекта JavaScript и несколько строк кода — это слишком много, но это позволяет предоставить основу для создания более сложных компонентов, а также для взаимодействия с жизненным циклом страниц Oracle Cloud Sites Service.

Чтобы просмотреть структуру локального компонента, выполните указанные ниже действия:

1. На домашней странице Oracle Content Management щелкните **Разработчик**.
Отображается страница **Разработчик**.
2. Нажмите **Смотреть все компоненты**.
3. В меню выберите **Создать локальный компонент**.
4. Введите имя компонента, например `A_Local_Component`.
5. Введите дополнительное описание.
6. Нажмите **Создать**.
После этого в списке компонентов появится компонент с именем `A_Local_Component`.
1. С помощью настольного клиента синхронизации Oracle Content Management найдите компонент и синхронизируйте его с файловой системой.

Если настольного клиентского ПК нет, можно просмотреть все компоненты и выбрать компонент на странице "Компоненты" интерфейса Oracle Content Management, а затем развернуть его, чтобы увидеть файлы.

- Если перечислить файлы, относящиеся к компоненту, можно увидеть следующие файлы:

```
assets
  render.js
  settings.html
appinfo.json
_folder_icon.jpg
```

- Откройте файл `render.js` в каталоге `/assets`.

Основные особенности файла `render.js`:

- он структурирован как AMD-модуль JavaScript, поэтому он может быть "обязательным" элементом страницы;
- Он также содержит ссылки на KnockoutJS и JQuery, которые уже загружены на странице Oracle Content Management.

Рассмотрите структуру файла `render.js`.

В содержимом файла `render.js` есть два объекта JavaScript, которые реализуют требуемые API-интерфейсы компонентов Oracle Content Management: `sampleComponentFactory` и `SampleComponentImpl`. Эти объекты служат примером реализации для создания любых компонентов на основе KnockoutJS. Реализация этих объектов зависит от используемой технологии.

- `sampleComponentFactory`
 - Этот объект возвращается AMD-модулем `render.js`.
 - Это очень простой объект Factory, который реализует единый интерфейс `createComponent()`.
 - В более сложных реализациях может использоваться значение `args`, передаваемое для возврата различных реализаций компонента на основе параметра `viewModel`. По сравнению с Site Builder это позволяет значительно облегчить реализацию компонента для среды выполнения.
- `SampleComponentImpl`
 - Основная функция этого объекта — функция `render`, которая используется для рендеринга компонента на странице.

Для рендеринга компонента Knockout на странице, функция `render` динамически добавляет шаблон на страницу, а затем применяет к шаблону привязки `viewModel`.
 - Остальная часть реализации имеет дело с инициализацией параметра `viewModel` и шаблона, а также с обработкой обмена сообщений между страницей и компонентом.

Последние два объекта в файле `render.js`, `sampleComponentTemplate` и `SampleComponentViewModel`, обеспечивают пользовательскую реализацию компонента. Реализация на основе этих объектов зависит от ваших требований.

- `sampleComponentTemplate`
 - Этот объект обеспечивает создание шаблона KnockoutJS. Он ожидает инициализации всех данных компонента, прежде чем пытаться что-то отобразить.

- SampleComponentViewModel
 - viewModel извлекает информацию, хранящуюся в Oracle Content Management от имени компонента, а затем на основе этих данных выбирает способ правильного расположения компонента.
 - Общие наблюдаемые элементы Knockout, которые используются шаблоном для обработки доступа к метаданным, сохраненными от имени компонента:

```
self.imageWidth = ko.observable('200px');
self.alignImage = ko.observable();
self.layout = ko.observable();
self.showTopLayout = ko.observable();
self.showStoryLayout = ko.observable();
```

- Интеграция триггеров и действий:

Триггер. Функция инициирования триггера Oracle Content Management из компонента, который может быть связан с действиями других компонентов на странице.

```
self.imageClicked = function (data, event) {
  self.raiseTrigger("imageClicked"); // matches appinfo.json
};
```

Действие. Функция обработки обратного вызова, когда компоненту предлагается выполнить действие с заданными полезными данными.

```
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);
      }
    });
  }
};
```

Обратный вызов для выполнения любых зарегистрированных действий по требованию.

```
SitesSDK.subscribe(SitesSDK.MESSAGE_TYPES.EXECUTE_ACTION,
$.proxy(self.executeActionsListener, self));
```

- Подписки на жизненный цикл компонентов:

- * Инициализация компонента: убедитесь, что рендеринг компонента не выполняется, пока не извлечены все данные. Эта операция обрабатывается посредством наблюдаемых элементов Knockout.

```
self.componentLayoutInitialized = ko.observable(false);
self.customSettingsDataInitialized = ko.observable(false);
```

Получите начальные значения для всех требуемых свойств. Эта операция выполняется с использованием обратных вызовов для извлечения данных.

```
SitesSDK.getProperty('componentLayout',
self.updateComponentLayout);
SitesSDK.getProperty('customSettingsData',
self.updateCustomSettingsData);
```

- * Обновление метаданных: обратный вызов выполняется всякий раз, когда изменяются метаданные компонента, сохраненные от имени компонента; например, когда пользователь вызывает панель "Настройки" и обновляет данные.

```
SitesSDK.subscribe(SitesSDK.MESSAGE_TYPES.SETTINGS_UPDATED,
$.proxy(self.updateSettings, self));
```

Примечание.

Поскольку сервер Oracle Content Management для файлов .html всегда задает тип MIME, невозможно загрузить файл .html и использовать требуемый подключаемый модуль "text!" для его загрузки. Поэтому для шаблонов необходимо либо использовать другое расширение, чтобы загрузить его с помощью подключаемого модуля "text!", либо загрузить его в JavaScript, как показано в предварительно заданных данных.

Проверка результатов для шага 2

Теперь у вас должно быть представление о том, как создается структура рендерера пользовательских компонентов. Чтобы проверить его работоспособность, выполните указанные ниже действия.

1. Обновите объект `sampleComponentTemplate` в файле `render.js`, чтобы изменить приведенную ниже строку. Измените этот код:

```
'<!-- ko if: initialized -->'+
```

Вместо него используйте этот код:

```
'<!-- ko if: initialized -->'+
'<div data-bind="text:\'image width is: \' + imageWidth()"></div>' +
```

2. Синхронизируйте или загрузите компонент на сервер экземпляров Oracle Content Management.

3. Измените страницу на сайте и перетащите на пользовательский компонент `A_Local_Component` на данной странице.
В этот момент должна отображаться запись `image width is: 260px`.
4. Откройте панель "Настройки" и нажмите кнопку **Пользовательские настройки**.
5. Измените значение в поле **image Width** на `300px`.
6. В этот момент в компоненте происходят два события:
 - a. Размер изображения по умолчанию увеличивается с 260 до 300 пикселей.
 - b. Добавленный вами текст обновляется с учетом записи `image width is 300px`.

Теперь выполните [Шаг 3. Проверка структура настроек локального компонента](#).

Шаг 3. Проверка структура настроек локального компонента

В ходе выполнения этого шага мы рассмотрим структуру настроек, указанных для локального компонента.

Подобно файлу `render.js`, в каталоге `/assets`, в том же каталоге есть предварительно созданный файл `settings.html`. В файле `settings.html` содержатся все данные пользовательских настроек компонента. В реализации по умолчанию в данных пользовательских настроек имеется свойство `imageWidth`.

Чтобы просмотреть структуру локального компонента, выполните указанные ниже действия:

1. С помощью настольного клиента синхронизации Oracle Content Management найдите компонент и синхронизируйте его с файловой системой.

Если настольного клиента синхронизации нет, компонент можно выбрать на вкладке **Компоненты** веб-интерфейса Oracle Content Management, а затем развернуть его, чтобы увидеть файлы.

2. Если перечислить файлы, относящиеся к компоненту, можно увидеть следующие файлы:

```
assets
  render.js
  settings.html
appinfo.json
_folder_icon.jpg
```

Откройте файл `settings.html` в каталоге `/assets` и просмотрите содержимое. В отличие от файла `render.js` файл `settings.html` использует встроенную рамку к панели "Настройки" в Site Builder. Поэтому ему также необходим доступ к вспомогательным файлам для правильного рендеринга в пределах встроенной рамки. Site Builder необходим для управления сайтом. Таким образом любые ошибки в коде JavaScript могут быть изолированы от Site Builder. Поэтому файл `settings.html` использует встроенную рамку.

Основные области файла `settings.html`:

- Шаблон Knockout для рендеринга панели "Настройки".

```
<!-- 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 -->
```

- Пользовательский обработчик привязки для регулировки высоты встроенной рамки после визуализации панели "Настройки".

```
ko.bindingHandlers.scsCompComponentImpl
```

- Модель Knockout ViewModel для применения к шаблону Knockout.

```
SettingsViewModel
```

Это основные элементы модели SettingsViewModel:

- Подписки на жизненный цикл компонентов.
- Инициализация компонента:
 - Убедитесь, что рендеринг компонента не выполняется, пока не извлечены все данные. Эта операция обрабатывается посредством наблюдаемых элементов Knockout.

```
self.initialized = ko.observable(false);
```

- Убедитесь, что данные обновляются только после завершения подготовки.

```
self.saveData = false;
```

- Получите начальные значения для всех требуемых свойств. Эта операция выполняется с использованием обратных вызовов для извлечения данных.

```
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;
});
```

- Все изменения свойств сохраните в данных пользовательских настроек.

```

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);

```

Чтобы добавить другое свойство, которое требуется захватить, необходимо выполнить несколько действий.

1. Обновите пользовательский интерфейс, чтобы отобразить новое значение.
2. Получите начальное значение как текущее значение, сохраненное для компонента.
3. Сохраните все изменения значения обратно в компонент.

Чтобы добавить в пользовательский компонент еще одно свойство, внесите в файл `settings.html` следующие изменения:

1. Добавьте еще один наблюдаемый элемент для обработки нового свойства. Измените этот код:

```
self.width = ko.observable();
```

Вместо него используйте этот код:

```

self.width = ko.observable();
self.imageBannerText = ko.observable();

```

2. При первом отображении панели настроек необходимо получить любое текущее значение для нового свойства. Измените этот код:

```
self.width(data.width);
```

Вместо него используйте этот код:

```

self.width(data.width);
self.imageBannerText(data.imageBannerText);

```

3. Все изменения сохраняйте в этом новом свойстве. Измените этот код:

```

'width': isNaN(self.width()) ? self.width() : self.width() +
'px'

```

Вместо него используйте этот код:

```
'width': isNaN(self.width()) ? self.width() : self.width() + 'px',
'imageBannerText': self.imageBannerText()
```

4. Добавьте пользовательский интерфейс для отображения нового поля. Измените этот код:

```
<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">
```

Вместо него используйте этот код:

```
<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. Выполните синхронизацию или загрузите файл settings.html.

Если бы эта операция была выполнена сейчас, поле бы отображалось. Однако размер панели "Настройки" не изменяется автоматически. Из-за увеличенного размера панели необходимо также обновить запись регистрации в components.json, указав новый размер.

1. Выгрузите файл appinfo.json, который находится на одном уровне с каталогом assets/ вашего компонента, и обновите размер панели настроек. Измените этот код:

```
"settingsHeight": 90,
```

Вместо него используйте этот код:

```
"settingsHeight": 160,
```

2. Выполните синхронизацию или загрузите файл appinfo.json.

Проверка результатов для шага 3

Теперь новое свойство, добавленное на панель "Настройки", должно быть доступно для просмотра и ввода.

1. Обновите страницу на своем сайте, чтобы приложение Site Builder могло отслеживать изменения в компоненте.
2. Переключите страницу в режим редактирования.
3. Перетащите свой компонент на страницу.
4. Откройте панель "Настройки" для своего компонента.

5. Нажмите кнопку **Пользовательские настройки**.

Для каждого свойства в файле settings.html отображается два поля.

Теперь выполните [Шаг 4. Отображение нового свойства в компоненте](#).

Шаг 4. Отображение нового свойства в компоненте

После завершения этого раздела вы сможете ввести значение нового свойства на панели "Настройки" и увидеть изменение пользовательского компонента с учетом нового значения. Обновления данного свойств также автоматически сохраняются вместе со страницей.

В файле render.js необходимо обновить два объекта JavaScript компонента:

- SampleComponentViewModel
- sampleComponentTemplate

Измените render.js и обновите компонент SampleComponentViewModel, включив в него новое свойство. Измените это свойство:

```
self.showStoryLayout = ko.observable();
```

Взамен используйте следующее:

```
self.showStoryLayout = ko.observable();  
self.imageBannerText = ko.observable();
```

Обновите SampleComponentViewModel, чтобы получить все изменения значений. Измените это свойство:

```
self.imageWidth(customData && customData.width);
```

Взамен используйте следующее:

```
self.imageWidth(customData && customData.width);  
self.imageBannerText(customData && customData.imageBannerText);
```

Измените sampleComponentTemplate, чтобы отобразить новое свойство. Измените это свойство:

```
'<div data-bind="text: \'image width is: \' + imageWidth()"'></div>' +
```

Взамен используйте следующее:

```
'<div data-bind="text: imageBannerText"'></div>' +
```

Синхронизируйте или загрузите компонент на сервер Oracle Content Management.

Теперь компонент изменен для отображения нового свойства. В отличие от панели "Настройки", вложенной во встроенную рамку на странице, поскольку

компонент вставляется непосредственно в страницу. С увеличением размера автоматически увеличивается доступная ему область.

Проверка результатов для шага 4

Чтобы просмотреть новое отображаемое свойство, выполните указанные ниже действия.

1. Обновите страницу на своем сайте, чтобы приложение Site Builder могло отслеживать изменения в компоненте.
2. Переключите страницу в режим редактирования.
3. Перетащите свой компонент на страницу.
4. Откройте панель "Настройки" для своего компонента.
5. Нажмите кнопку **Пользовательские настройки**.
6. Измените Image Banner на Workspace.

Обновление компонента на странице проявляется в появлении надписи Workspace над изображением.

Теперь выполните [Шаг 5. Регистрация триггеров](#).

Шаг 5. Регистрация триггеров

В ходе выполнения этого шага объясняется, как зарегистрировать триггер Oracle Content Management, который можно выбрать с помощью параметра "Действия триггера" на вкладке "Ссылка" панели "Настройки" компонента.

Триггеры — часть межкомпонентного взаимодействия Oracle Content Management. Любой компонент может вызвать любое количество триггеров. Компонент может предоставить триггеру полезные данные, которые затем передаются любому действию, которое выполняется при срабатывании триггера. Пользователи могут выбрать действия, которые должны выполняться для каждого триггера. Наконец, компоненты, созданные для совместной работы, могут автоматически вызывать триггеры для выполнения действий в другом компоненте. Пользователю не нужно определять взаимодействие между такими компонентами.

Для добавленных компонентов триггеры регистрируются как часть данных регистрации компонента. Чтобы добавить триггер, обновите массив свойств "triggers" с каждым триггером, поддерживаемым компонентом. Кроме того, необходимо указать полезные данные, поддерживаемые триггером, чтобы можно было создать пользовательский интерфейс, позволяющий пользователям сопоставить значения в полезных данных со свойствами, поддерживаемыми действием.

Откройте файл appinfo.json и просмотрите запись "triggers": [],:

```
"triggers": [{
  "triggerName": "imageClicked",
  "triggerDescription": "Image clicked",
  "triggerPayload": [{
    "name": "payloadData",
    "displayName": "Trigger Payload Data"
  }]
}],
```

В этой записи отображается следующее:

- Имя триггера (triggerName), "imageClicked", которое должно быть уникальным значением и обычно связывается с пространством имен с помощью идентификатора вашего пользовательского компонента.
- Описание триггера (triggerDescription), "Image clicked", которое используется в диалоговом окне пользовательского интерфейса для отображения триггера.
- Однозначное свойство triggerPayload, "payloadData" для вашего триггера. Пользователи смогут выбирать записи в этих полезных данных и сопоставлять их полям в действии.

Проверка результатов для шага 5

Вы можете просматривать и выбирать свои триггеры, перейдя на вкладку **Ссылка** на панели "Настройки" своего компонента.

1. Обновите страницу на своем сайте, чтобы приложение Site Builder могло отслеживать изменения в компоненте.
2. Переключите страницу в режим редактирования.
3. Перетащите свой компонент на страницу.
4. Откройте панель "Настройки" для своего компонента.
5. Выберите вкладку **Ссылка** в верхней части панели "Настройки".
6. Выберите в списке **Триггеры действий** значение "Тип связи".
7. Нажмите зарегистрированный вами триггер **Image clicked**.
8. В диалоговом окне перетащите действие **Показать оповещение** из раздела "Действия на странице". (Действия на странице — это встроенные действия, предоставляемые платформой Oracle Content Management.)
9. В поле **Сообщение** выберите значение **Полезные данные триггера**, которое представляет собой имя записи в полезных данных, которые отображались при регистрации триггера. Это имя можно заменить любым другим.

Теперь вы можете зарегистрировать триггер и сопоставить его встроенному действию, передавая посредством значения. В ходе выполнения следующего шага мы рассмотрим срабатывание триггера для выполнения действия.

Теперь выполните [Шаг 6. Срабатывание триггеров](#).

Шаг 6. Срабатывание триггеров

В ходе выполнения этого шага мы покажем, как срабатывает зарегистрированный триггер.

В любой момент триггеры могут быть вызваны компонентом. Обычно он срабатывает в результате взаимодействия с пользователем, например, если нажать кнопку или выбрать строку в таблице. Однако компонент может вызвать триггер на основе любого критерия, например, при изменении данных из-за вызова REST.

В этом примере, если нажать изображение, срабатывает триггер, передающий текущее значение свойства whoAreYou.

В файле `render.js` обратите внимание на объект `SampleComponentViewModel`.

Чтобы сработал триггер:

1. Проверьте функцию в объекте `SampleComponentViewModel`, который вызывает Sites SDK для активации триггера.

```
self.raiseTrigger = function (triggerName) {
  SitesSDK.publish(SitesSDK.MESSAGE_TYPES.TRIGGER_ACTIONS, {
    'triggerName': triggerName,
    'triggerPayload': {
      'payloadData': 'some data here'
    }
  });
};
```

2. Теперь для срабатывания триггера необходимо, чтобы какой-то элемент пользовательского интерфейса вызвал данную функцию. Просмотрите файл `render.js` и обновите объект `sampleComponentTemplate`, чтобы получить следующую запись:

```
'<div data-bind="attr: {style: imageStyle, \'data-layout\':
alignImage()}, click: imageClicked">' +
```

В объекте `SampleComponentViewModel` содержится функция JavaScript, которая вызывается, если нажать изображение. Эта функция вызывает Sites SDK, чтобы передать команду на запуск всех действий, определенных для триггера "imageClicked", который представляет собой значение, переданное из привязки `click` в ходе выполнения шага 2. Она также передает объект `triggerPayload` с одним полем: `payloadData`, а также статическое значение "some data here". Эти значения `imageClicked` и `whoAreYou` соответствуют значениям в файле `appinfo.json`, где зарегистрирован триггер (в ходе выполнения предыдущего шага).

В данном примере кода триггер вызывается привязкой данных `click` и передается в имени триггера `imageClicked`. В настоящее время существует три варианта рендеринга компонента `<scs-image>` на основе выбранного пользователем макета. Чтобы убедиться в срабатывании триггера для каждого макета, внесите в файл `render.js` следующие изменения.

- Активация триггеров из разных макетов. Найдите две записи этого кода:

```
'<div data-bind="attr: {style: imageStyle, \'data-layout\':
alignImage()}">' +
```

Измените код на следующее:

```
'<div data-bind="attr: {style: imageStyle, \'data-layout\':
alignImage()}, click: imageClicked">' +
```

- Укажите полезные данные, передаваемые триггерам. Измените этот код:

```
self.raiseTrigger = function (triggerName) {
  SitesSDK.publish(SitesSDK.MESSAGE_TYPES.TRIGGER_ACTIONS, {
    'triggerName': triggerName,
```

```
        'triggerPayload': {  
            'payloadData': 'some data here'  
        }  
    });  
};
```

Вместо него используйте этот код:

```
self.raiseTrigger = function (triggerName) {  
    SitesSDK.publish(SitesSDK.MESSAGE_TYPES.TRIGGER_ACTIONS, {  
        'triggerName': triggerName,  
        'triggerPayload': {  
            'payloadData': self.imageBannerText() // pass banner text  
as payload  
        }  
    });  
};
```

- Выполните синхронизацию или загрузите файл `render.js` на сервер экземпляров Oracle Content Management.

После проверки требуемого кода можно подключить триггер, чтобы пользовательский компонент активировал его при нажатии кнопки.

Проверка результатов для шага 6

Теперь вы можете зарегистрировать действие для выполнения по триггеру, а также выполнить действие при срабатывании триггера:

1. Обновите страницу на своем сайте, чтобы приложение Site Builder могло отслеживать изменения в компоненте.
2. Переключите страницу в режим редактирования.
3. Перетащите свой компонент на страницу.
4. Откройте панель "Настройки" для своего компонента.
5. Выберите вкладку **Ссылка** в верхней части панели "Настройки".
6. В списке **Действия триггера** выберите вариант "Тип ссылки".
7. Нажмите зарегистрированный триггер **imageClicked**.
8. В диалоговом окне перетащите действие **Показать оповещение** из раздела **Действия на странице**.
9. В поле **Сообщение** выберите значение **payloadData**, которое представляет собой полезные данные, введенные при регистрации триггера.
10. Закройте панель "Настройки" и переключите Site Builder в режим предварительного просмотра.
11. Нажмите на изображение в компоненте.
Появится оповещение `no message defined`, так как не указано значение `imageBannerText`.
12. Переключите страницу в режим редактирования и снова откройте панель "Настройки" для данного компонента.
13. Нажмите **Пользовательские настройки** и введите `Workplace`.

14. Закройте панель "Настройки" и переключите страницу в режим предварительного просмотра.
15. Нажмите на изображение в компоненте.

Теперь должны отображаться обновленные полезные данные `Workplace`, которые вызываются из изменения привязки `click`.

При срабатывании триггера может быть выполнено любое количество действий.

Примечание.

При выполнении действия предопределенного порядка нет. Несмотря на то, что каждое действие вызывается в том порядке, в котором оно указано в списке, не нужно ждать его завершения, чтобы вызвать следующее действие. Если действие выполняет асинхронный вызов, оно может не завершиться до выполнения следующего действия.

Теперь выполните [Шаг 7. Регистрация действий](#).

Шаг 7. Регистрация действий

Действия Oracle Content Management вызываются в компонентах при срабатывании триггеров.

Компонент может зарегистрировать любое количество действий, а также определить полезные данные, поддерживаемые действием. Выбирая действия, пользователи могут ввести полезные данные, передаваемые действию.

Как и при регистрации триггеров, действия, поддерживаемые компонентом, можно зарегистрировать в данных регистрации файла `appinfo.json`. Чтобы проверить регистрацию примера действия в компоненте, откройте файл `appinfo.json` и найдите код `"actions"`.

```
"actions": [{
  "actionName": "setImageWidth",
  "actionDescription": "Update the image width",
  "actionPayload": [{
    "name": "imageWidth",
    "description": "Image Width in pixels",
    "type": {
      "ojComponent": {
        "component": "ojInputText"
      }
    },
    "value": ""
  }]
}]
```

Зарегистрированное действие отображается в диалоговом окне "Действие", которое вызывается, если нажать имя триггера на вкладке **Ссылка** панели "Настройка" компонента.

Проверьте результаты шага 7.

1. Обновите страницу на своем сайте, чтобы приложение Site Builder могло отслеживать изменения в компоненте.
2. Переключите страницу в режим редактирования.
3. Перетащите свой компонент на страницу.
4. Перетащите компонент "Кнопка" на страницу.
5. Откройте панель "Настройки" для компонента "Кнопка".
6. На вкладке "Общие" измените метку кнопки на **Нажми меня!**.
7. Выберите вкладку **Ссылка** на панели "Настройки".
8. В списке **Действия триггера** выберите вариант **Тип ссылки**.
9. Нажмите триггер **Нажать кнопку** напротив компонента "Кнопка".
10. В диалоговом окне разверните компонент `A_Local_Component` в левой палитре.
11. Перетащите действие **Обновить ширину изображения** из компонента `A_Local_Component` на страницу.
12. Введите **300px** в поле **Ширина изображения в пикселях**.

Теперь вы знаете, как зарегистрировать действие и как оно будет отображаться в пользовательском интерфейсе. На следующем этапе вы узнаете, как обрабатывать действие в компоненте во время его вызова.

Теперь выполните [Шаг 8. Выполнение действий](#).

Шаг 8. Выполнение действий

После прохождения этой темы вы сможете перетаскивать компоненты на страницу, на которой действия выполняются в пределах компонента. Это позволяет использовать регистрацию действий, созданную в ходе выполнения предыдущего шага.

Чтобы выполнить действие, компонент должен прослушивать сообщение `EXECUTE_ACTION`. Это сообщение также включает передаваемые действию полезные данные, из которых необходимо извлечь ожидаемые значения.

Чтобы прослушивать сообщение `EXECUTE_ACTION`, измените файл `render.js` и обновите объект `SampleComponentViewModel` с учетом следующей записи:

```
SitesSDK.subscribe('EXECUTE_ACTION', $.proxy(self.executeActionsListener, self));
```

При получении сообщения `EXECUTE_ACTION` выполняется соответствующая функция обратного вызова:

```
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);
    }
  });
}
```

В результате создается функция JavaScript для выполнения действия, а затем используется комплект Sites SDK для вызова функции при появлении сообщения EXECUTE_ACTION.

Данный компонент вызывается при любом появлении сообщения EXECUTE_ACTION, и компонент обрабатывает только те действия, для обработки которых он предназначен. Для этого необходимо проверить имя действия, чтобы убедиться, что это то действие, которое компонент может обработать.

Полезные данные для действия — это набор значений. Как правило, значения полезных данных должны быть указаны в массиве.

Примечание.

Поскольку прослушиватель действий представляет собой обратный вызов, необходимо использовать замыкание JavaScript или соответствующим образом привязать данную функцию, чтобы обеспечить доступ к viewModel при выполнении функции.

Проверка результатов для шага 8

1. Обновите страницу на своем сайте, чтобы приложение Site Builder могло отслеживать изменения в компоненте.
2. Переключите страницу в режим редактирования.
3. Перетащите свой компонент на страницу.
4. Перетащите компонент "Кнопка" на страницу.
5. Откройте панель "Настройки" для компонента "Кнопка".
6. На вкладке "Общие" измените метку кнопки на **Нажми меня!**
7. Выберите вкладку **Ссылка** в верхней части панели "Настройки".
8. В списке **Действия триггера** выберите вариант "Тип ссылки".
9. Нажмите триггер **Нажать кнопку** напротив компонента "Кнопка".
10. В диалоговом окне разверните компонент `A_Local_Component` на левой стороне.
11. Перетащите действие **Обновить ширину изображения** из компонента `A_Local_Component` на правую сторону.
12. Введите `300px` в поле **Ширина изображения в пикселах**.
13. Переключите страницу в режим предварительного просмотра.
14. Нажмите кнопку **Нажми меня!**

В этот момент размер изображения увеличивается до 300 пикселей.

 **Примечание.**

Триггеры и действия предназначены для поддержки связи между компонентами. Они не предназначены для создания состояния или управления им. При обновлении страница возвращается в исходное состояние, как если бы не срабатывали триггеры и не выполнялись действия.

Теперь выполните [Шаг 9. Создание отдельного заголовка для каждого экземпляра компонента](#).

Шаг 9. Создание отдельного заголовка для каждого экземпляра компонента

В ходе выполнения этого шага объясняется порядок создания отдельных заголовков для разных экземпляров компонента.

При перемещении компонента на страницу на баннере компонента отображается следующее: `A_Local_Component`. Хотя это нормально, если пользователь перетаскивает на страницу только один из своих компонентов, может потребоваться создать отдельные заголовки, чтобы различать различные экземпляры компонента.

Для обновления заголовка компонента можно использовать Sites SDK. В ходе выполнения этого шага заголовок обновляется на основе свойства `"imageBannerText"`.

Чтобы обновить заголовок, измените файл `render.js` и добавьте в объект `SampleComponentViewModel` следующий код:

```
self.updateDescription = ko.computed(function () {  
    SitesSDK.setProperty('description', self.imageBannerText());  
});
```

Это вычисление на основе Knockout обновляет описание компонента при каждом изменении наблюдаемого элемента `imageBannerText`.

Проверьте результаты шага 9.

1. Обновите страницу на своем сайте, чтобы приложение Site Builder могло отслеживать изменения в компоненте.
2. Переключите страницу в режим редактирования.
3. Перетащите свой компонент на страницу.
4. Откройте панель "Настройки" для своего компонента.
5. Нажмите кнопку **Пользовательские настройки**.
6. Измените **Image Banner** на **Workplace**.

7. Закройте панель "Настройки" и наведите указатель мыши на компонент, чтобы отобразить баннер.

Отображается `A_Local_Component Workplace`.

Теперь выполните [Шаг 10. Использование вложенных компонентов с внутрискрипчным редактированием](#).

Шаг 10. Использование вложенных компонентов с внутрискрипчным редактированием

Компоненты Oracle Content Management внедряются с помощью архитектуры компонента KnockoutJS. Это означает, что если для внедрения компонентов используется KnockoutJS, встроенные компоненты Oracle Content Management можно включить непосредственно в шаблон.

Примечание.

Так как встроенные компоненты Oracle Content Management могут выполняться только на странице Oracle Content Management. Невозможно использовать вложенные компоненты в случае их рендеринга во встроенной рамке.

Чтобы использовать вложенные компоненты, выполните указанные ниже действия.

1. Внедрите свой компонент с помощью KnockoutJS.
2. С помощью RequireJS включите свой компонент и используйте ту же переменную экземпляра Knockout "ko", что создана в Oracle Content Management.

Это необходимо, поскольку Oracle Content Management расширяет Knockout с помощью компонентов. Эти компоненты недоступны, если используется собственный экземпляр KnockoutJS.

В ходе выполнения этого шага вы узнаете о рендеринге компонентов Oracle Content Management ("Изображение", "Абзац" и "Заголовок") в пользовательском компоненте. Пользователь может редактировать его непосредственно на странице и получить доступ к панели "Настройки" для вложенного компонента.

Чтобы узнать, как эти компоненты включены в ваш шаблон, измените файл `render.js` и обратите внимание на объект `sampleComponentTemplate`. Визуализируемый раздел по умолчанию показан здесь:

```
'<!-- 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 -->' +
```

При просмотре вложенного компонента `<scs-image>` можно видеть следующую запись:

```
'<scs-image params="{ scsComponent: { \'renderMode\\': mode,
\'parentId\\': id, \'id\\': \'imageId\\', \'data\\': imageData } }"></scs-
image>' +
```

Данные `scsComponent`, переданные привязке шаблона `params`, включают следующее:

- `renderMode`: относится к режиму работы Site Builder. Эту возможность можно использовать для включения и отключения функций. Например, при использовании компонента `<scs-title>` в режиме редактирования `edit` добавляется редактор форматированного текста.
- `parentId`: этот идентификатор указывает, что рендеринг компонента Oracle Content Management осуществляется как рендеринг вложенного компонента. Все изменения вложенного компонента сохраняются в данных пользовательского компонента.
- `id`: уникальный идентификатор вложенного компонента. Используется для последующего связывания с пространством имен с помощью идентификатора пользовательского компонента.
- `data`: исходные данные для вложенного компонента. Без дальнейших изменений рендеринг компонента будет осуществляться с этими исходными данными.

Значения `id` и `mode` со ссылкой передаются пользовательскому компоненту в объекте `SampleComponentViewModel`. Поэтому для получения этих значений не требуется изменять объект:

```
// Store the args
self.mode = args.viewMode;
self.id = args.id;
```

Синтаксис всех остальных поддерживаемых вложенных компонентов соответствует тому же шаблону, что и для `<scs-paragraph>`; например: `<scs-image>`, `<scs-title>`, `<scs-button>`.

Проверка результатов для шага 10

1. Обновите страницу на своем сайте, чтобы приложение Site Builder могло отслеживать изменения в компоненте.
2. Переключите страницу в режим редактирования.
3. Перетащите свой компонент на страницу.
4. Нажмите текст `As a page author, you can edit. . .` в компоненте и обновите описание с помощью редактора форматированного текста.

5. Переключитесь в режим предварительного просмотра, чтобы просмотреть свое обновление.
6. Переключитесь обратно в режим редактирования.
7. Откройте панель "Настройки" для своего компонента.
8. Перейдите по ссылке **Компоненты**, которая появляется в этот момент, так как обнаружен вложенный компонент.
9. Нажмите **Абзац** — найденный вложенный компонент.

Теперь в своем компоненте можно обновить свойства компонента "Абзац".

Примечание.

Пока не создан экземпляр компонента, платформе Oracle Content Management не известно ни о каких вложенных компонентах в данном шаблоне. Чтобы сообщить Oracle Content Management о скрытых вложенных компонентах, можно использовать API-интерфейс `SitesSDK.setProperty('visibleNestedComponents', [])`; Чтобы скрытые вложенные компоненты отображались по умолчанию, необходимо обновить массив `"nestedComponents": []` в записи регистрации компонента.

Теперь выполните [Шаг 11. Поддержка различных макетов](#).

Шаг 11. Поддержка различных макетов

В ходе выполнения этого шага мы рассмотрим макеты, которые позволяют пользователю изменять способ отображения компонента.

Пользовательский компонент может поддерживать любое количество макетов, которые вы захотите разрешить пользователю выбрать. Каждый из этих макетов изменяет способ отображения пользовательского компонента. Макеты — это еще одно расширение данных регистрации.

Чтобы просмотреть три макета, поддерживаемых в примере кода, просмотрите запись `"componentLayouts"` в файле `appinfo.json`.

```
"componentLayouts": [
  {
    "name": "default",
    "displayName": "IMAGE_LEFT_LAYOUT"
  },
  {
    "name": "right",
    "displayName": "IMAGE_RIGHT_LAYOUT"
  },
  {
    "name": "top",
    "displayName": "IMAGE_TOP_LAYOUT"
  }
],
```

Если открыть панель "Настройки" для пользовательского компонента, отображается возможность переключения между макетами. Чтобы компонент реагировал на изменение выбранных элементов, файл `render.js` содержит код для получения текущего выбранного значения и прослушивания изменений этого значения.

Измените файл `render.js` и посмотрите на объект `SampleComponentViewModel`.

- Существует наблюдаемый элемент `layout`, на который есть ссылка в шаблоне:

```
self.layout = ko.observable();
```

- Существует функция обновления, которая обрабатывается при изменении этого значения:

```
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);
```

- Код инициализации получает исходное значение для макета и вызывает функцию обновления:

```
SitesSDK.getProperty('componentLayout', self.updateComponentLayout);
```

Прослушиватель изменений свойств проверяет наличие изменений этого свойства и вызывает функцию обновления:

```
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));
```

Наконец, объект шаблона `sampleComponentTemplate` содержит код, отражающий изменения этого значения:

```
'<!-- ko if: alignImage() === \'right\' -->' +
```

Вместе эти изменения позволяют выбрать макет на панели "Настройки" и обновить компонент.

Проверка результатов для шага 11

1. Обновите страницу на своем сайте, чтобы приложение Site Builder могло отслеживать изменения в компоненте.
2. Переключите страницу в режим редактирования.
3. Перетащите свой компонент на страницу.
4. Откройте панель "Настройки" для своего компонента.
5. В свойстве "Макет" выберите **Изображение справа**.

В этот момент компонент обновляется для отображения компонента "`<scs-image>`".

Теперь выполните [Шаг 12. Определение пользовательских стилей](#).

Шаг 12. Определение пользовательских стилей

Созданные компоненты обрабатываются как любые другие компоненты в файлах `design.json` и `design.css` из темы, используемой для вашего сайта.

Чтобы добавить собственный стиль для пользовательского компонента, подтвердите значение `id`, используемое при регистрации компонента. В файле `appinfo.json` это был `"id": "hello-world"`.

Используя это значение, измените файл `design.json` темы и добавьте новые стили, которые требуется поддерживать для этого идентификатора `id`. Например, измените файл `/design/default/design.css` из своей темы и добавьте этот код:

```
"hello-world": {
  "styles": [{
    "name": "Plain",
    "class": "hello-world-default-style"
  },
  {
    "name": "Gothic",
    "class": "hello-world-gothic-style"
  }
]
```

Если открыть панель "Настройки" для своего компонента, теперь на вкладке "Стиль" должны отображаться два параметра **Обычный (по умолчанию)** и **Готический**. Однако при переключении между этими параметрами ничего не происходит без фактического определения классов стилей, перечисленных в файле `design.css`.

Измените файл `design.css` темы и добавьте классы каскадной таблицы стилей (CSS) своего стиля. Например, измените файл `/design/default/design.css` из своей темы и добавьте этот код:

```
.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; }
```

Сохраните и синхронизируйте файлы Oracle Content Management на сервере экземпляров.

Проверка результатов для шага 12

1. Обновите страницу на своем сайте, чтобы приложение Site Builder могло отслеживать изменения в компоненте.
2. Переключите страницу в режим редактирования.
3. Перетащите свой компонент на страницу.
4. Откройте панель "Настройки" для своего компонента.
5. Перейдите на вкладку "Стиль".
6. Переключайтесь между стилями **Готический** и **Обычный**, определенными в файле design.json.

Вы заметите, что размер шрифта в компоненте изменяется в соответствии с изменениями, которые происходят между примененными классами CSS для каждого выбора.

Теперь выполните [Шаг 13. Рендеринг компонента во встроенной рамке](#).

Шаг 13. Рендеринг компонента во встроенной рамке

В этом примере показан локальный компонент с рендерингом во встроенной рамке на странице. Также можно выбрать рендеринг компонента во встроенной рамке.

Например, рендеринг компонента во встроенной рамке можно выбрать, если компонент выполняет лишь незначительные обновления страницы, которые требуют повторного создания страницы при любом изменении свойств. Кроме того, рендеринг удаленных компонентов всегда осуществляется во встроенной рамке.

Примеры в этом разделе берутся из файлов, созданных для вас, когда при создании локального компонента вы выбираете вариант **Создать компонент с рендерингом в iframe**. Однако можно взять этот набор файлов и разместить их на удаленном сервере, чтобы они одинаково применялись к удаленным компонентам.

Сходство между компонентами во встроенной и невстроенной рамках

Панель "Настройки"

Поскольку панель "Настройки" всегда помещается на страницу во встроенной рамке, код панели "Настройки" не зависит от того, использует компонент встроенную рамку или нет. Для обоих сценариев использования создается один и тот же код панели "Настройки".

API-интерфейс Sites SDK

API-интерфейс SDK один и тот же для обоих сценариев использования. Один и тот же код используется для создания триггеров, прослушивания действий, получения и задания значений свойств. Хотя некоторые свойства могут быть неприменимы в обоих случаях (например, невозможно задать свойство "height" для компонента, который не использует встроенную рамку), API-интерфейс остается таким же. Следовательно код можно копировать между компонентами обоих типов, а пример кода, рассматриваемый в данном учебном пособии работает в обоих случаях.

Разница между компонентами во встроенной и невстроенной рамках

Структура файлов и зависимости

Если при создании локального компонента выбрать вариант **Создать компонент с рендерингом в iframe**, отображаются следующие файлы, созданные для вас:

```
<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
```

Эти файлы создаются, чтобы можно было немедленно запустить компонент во встроенной рамке на странице. Основные различия между этой структурой и структурой стандартного локального компонента перечислены ниже.

- Зависимости JavaScript
 - Вы получаете полную копию этих файлов для работы своего компонента. Эти файлы необходимы образцу компонента во встроенной рамке для работы. Вы можете добавлять и удалять содержимое этого каталога в соответствии со своими требованиями.
 - Поскольку все содержимое каталога assets вашего компонента передается общедоступному сайт при публикации компонента, все содержимое каталога js будет доступно как в Site Builder, так и во время выполнения.
 - Примечание. Эти файлы созданы для удобства использования. Вам следует взглянуть на консолидацию этих файлов в теме или в другом общедоступном месте, а не создавать отдельные версии этих файлов для каждого своего компонента во встроенной рамке.
- render.html
 - В отличие от файла render.js для стандартных компонентов это полный HTML-документ, представляющий собой AMD-модуль.

Управление высотой компонента

Одна из проблем при использовании встроенной рамки — управление ее высотой. В случае ошибки на странице отображаются полосы прокрутки для компонента, которые могут быть вам и не нужны.

Для управления высотой встроенной рамки компонент должен сообщить странице нужную высоту встроенной рамки. В случае удаленных компонентов могут возникнуть междоменные проблемы, поэтому с помощью обмена сообщениями Sites SDK необходимо запросить страницу задать требуемую высоту встроенной рамки после рендеринга компонента на странице. Для этого используется API-интерфейс `SitesSDK.setProperty('height', {value})`. (См. Комплекты SDK Oracle Content and Experience.)

Например, создайте функцию `setHeight` и пользовательский обработчик привязки, чтобы вызывать его после рендеринга компонента на странице.

- **Функция обновления высоты**

```
// 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');
};
```

- **Пользовательский обработчик привязки для вызова `setHeight` вызывается во время рендеринга компонент на странице или при любом изменении свойств:**

```
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();
  }
};
```

- **Обновление шаблона для вызова обработчика привязки:**

```
<div data-bind="sampleAppSetAppHeight: true"></div>
```

Регистрация триггеров и действий

Хотя записи регистрации триггера/действия для компонентов не во встроенных рамках находятся в файле `appinfo.json`, в случае компонентов во встроенных рамках за предоставление этой информации отвечает сам компонент. Для этого используются два API-интерфейса:

```
SitesSDK.subscribe('GET_ACTIONS', self.getAppActions);
SitesSDK.subscribe('GET_TRIGGERS', self.getAppTriggers);
```

Вот пример использования этих API-интерфейсов.

```
// 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;
};
```

Доступ к стилям темы

Поскольку рендеринг компонента осуществляется во встроенной рамке, у него нет доступа к стилям, доступным в теме. Sites SDK предоставляет API-интерфейс, чтобы извлекать эти стили для применения к элементам внутри встроенной рамки.

Эта тема более подробно рассматривается в разделе [Шаг 14. Использование пользовательских стилей во время рендеринга компонента во встроенной рамке](#).

Смешанный протокол HTTPS и протокол HTTP

Поскольку платформа Oracle Content Management использует протокол HTTPS, все ресурсы, на которые ссылается страница, также должны использовать HTTPS. Ресурсы включают в себя базовый файл .html, который будет визуализироваться во встроенной рамке вместе со всеми файлами, на которые он ссылается.

Это требование к ресурсам относится в основном к удаленным компонентам, однако необходимо знать об этом ограничении. Ресурсы для локальных компонентов, использующих встроенные рамки, предоставляются сервером Oracle Content Management, поэтому эти компоненты уже используют соответствующий протокол.

Теперь выполните [Шаг 14. Использование пользовательских стилей во время рендеринга компонента во встроенной рамке](#).

Шаг 14. Использование пользовательских стилей во время рендеринга компонента во встроенной рамке

У компонентов с рендерингом во встроенной рамке нет прямого доступа к файлу `design.css`. Вместо этого есть дополнительный этап получения URL-адреса `design.css` в компоненте и добавления его на страницу. Затем необходимо обновить компонент, чтобы отразить выбранный пользователем стиль.

Чтобы включить и использовать файл `design.css` в компоненте, необходимо внести изменения в файл `render.html`.

1. Найдите и включите URL-адрес в файл `design.css`.
2. Получите значение выбранного класса стилей при любом его изменении.
3. Обновите шаблон, чтобы он отражал выбранный класс `styleClass`.
4. Отрадите изменения выбранного класса стилей в компоненте.
5. Убедитесь, что при изменении стиля изменяется размер встроенной рамки.

Ниже приведены подробные инструкции по редактированию файла `render.html`.

1. Найдите и включите URL-адрес в файл `design.css`.

Настройте динамическое добавление файла `design.css` в раздел `<head>` страницы. После его загрузки задайте высоту встроенной рамки, так как она могла быть изменена при применении стилей.

Добавьте в объект `viewModel` следующий код:

```
// 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. Получите значение выбранного класса стилей при любом его изменении.

Создайте наблюдаемый элемент для отслеживания изменений значения свойства `styleClass`.

```
self.selectedStyleClass = ko.observable();
```

Обратите внимание, что рендеринг невозможен, пока нет класса стилей. Измените этот код:

```
self.customSettingsDataInitialized = ko.observable(false);
self.initialized = ko.computed(function () {
    return self.customSettingsDataInitialized();
}, self);
```

Вместо него используйте этот код:

```
self.customSettingsDataInitialized = ko.observable(false);
self.styleClassInitialized = ko.observable(false);
self.initialized = ko.computed(function () {
    return self.customSettingsDataInitialized() &&
```

```
self.styleClassInitialized();
}, self);
```

Получите начальное значение для выбранного класса стилей, добавив следующее:

```
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. Обновите шаблон, чтобы он отражал styleClass. Измените этот код:

```
<p data-bind="attr: {id: 'titleId'}, text: titleText"></p>
```

Вместо него используйте этот код:

```
<p data-bind="attr: {id: 'titleId'}, text: titleText, css:
selectedStyleClass"></p>
```

4. Отрадите изменения выбранного класса стилей в компоненте. Измените этот код:

```
if (settings.property === 'customSettingsData') {
    self.updateCustomSettingsData(settings.value);
}
```

Вместо него используйте этот код:

```
if (settings.property === 'customSettingsData') {
    self.updateCustomSettingsData(settings.value);
}
if (settings.property === 'styleClass') {
    self.updateStyleClass(settings.value);
}
```

5. Убедитесь, что при изменении стиля изменяется размер встроенной рамки. Измените этот код:

```
// 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();
```

Вместо него используйте этот код:

```
// 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. Сохраните и синхронизируйте файлы Oracle Content Management на сервере экземпляров.

Проверка результатов для шага 14

1. Обновите страницу на своем сайте, чтобы приложение Site Builder могло отслеживать изменения в компоненте.
2. Переключите страницу в режим редактирования.
3. Перетащите свой компонент на страницу.
4. Откройте панель "Настройки" для своего компонента.
5. Перейдите на вкладку "Стиль".
6. Переключитесь между стилями **Готический** и **Обычный**, определенными в файле design.json.

Вы заметите, что размер шрифта в компоненте изменяется в соответствии с изменениями, которые происходят между примененными классами CSS для каждого выбора.

Теперь выполните [Шаг 15. Интеграция с процессом выполнения на странице операций "Отмена" и "Повтор"](#).

Шаг 15. Интеграция с процессом выполнения на странице операций "Отмена" и "Повтор"

Поскольку в Oracle Content Management сохраняются свойства от имени настраиваемого компонента, изменения этих свойств автоматически становятся частью процесса выполнения на странице операций **Отмена** и **Повтор**.

Чтобы было понятно, что происходит, когда пользователь нажимает кнопки **Отмена** или **Повтор**, эти "события отмены" должны происходить, только когда пользователь фактически сделал что-то на странице. Например, если открыть панель "Настройки пользовательского компонента", свойства на странице не обновляются до тех пор, пока пользователь не внесет изменения в свойство. Простая инициализация свойств на панели "Настройки" не должна вызывать событие обновления.

Если не соблюдать осторожность, это может привести к непредвиденному поведению. Страница будет по-прежнему запускаться, но в ущерб работе пользователей. Например, могут возникнуть следующие ситуации:

- Кнопка **Сохранить** становится активной, если просто открыть панель "Настройки".
- Чтобы эффект был виден, пользователь должен несколько раз нажать **Отмена**.

- стек Redo удаляется, так как компонент записал непредвиденное изменение и обновил стек Redo новым значением.

Образец кода, приведенный в данном учебном пособии для панели "Настройки", содержит пример того, как убедиться, что вы отвечаете письменно, только когда готовы к вызову `saveData`, а не инициализации. Аналогичные действия следует выполнить в самом компоненте, чтобы не обновлять `customSettingsData`, если это не связано с взаимодействием пользователя, хотя обычно это меньше беспокоит.

Теперь выполните [Шаг 16. Управление активами](#).

Шаг 16. Управление активами

В ходе выполнения этого шага описывается и объясняется управление активами, используемыми компонентами.

К активам относятся компоненты и пользовательские компоненты, о которых Oracle Content Management должно быть известно, чтобы управлять жизненным циклом активов.

Папка контента Oracle Content Management

Каждый сайт, созданный в Oracle Content Management, поставляется с собственной папкой `контента`. Это скрытая папка, которая обычно не отображается. Когда сайт опубликован, все файлы в папке `content` также опубликованы в файловой системе.

Например, если выбрать изображения с помощью компонента `Image`, Oracle Content Management создает копию выбранного изображения и помещает ее в папку контента. URL-адрес всегда указывает на скопированную версию изображения, чтобы при удалении исходного изображения не нарушить работу сайта. Это также относится к другим компонентам, предоставляемым платформой Oracle Content Management: "Галерея", "Сетка галереи", "Документ", "Социальная панель", "Выгрузка файлов", а также фоновые изображения для `slots` и `componentGroups`.

Чтобы пользовательский компонент мог участвовать в этом жизненном цикле актива, он должен сообщать Oracle Content Management о любых активах, которыми эта платформа должна управлять от его имени. Поскольку это подразумевает создание копии актива, пользовательский компонент также должен для выбора актива использовать API-интерфейсы Oracle Content Management, чтобы платформе было известно, как управлять им.

Управление URL-адресами

URL-адрес актива зависит от нескольких критериев.

- URL-адрес компонента для времени выполнения отличается от URL-адреса компонента для Site Builder.
- При копировании страницы Oracle Content Management также создает в папке контента копию всех связанных активов, поэтому никогда не будет двух компонентов, указывающих на один и тот же актив в папке контента.
- При перетаскивании группы компонентов на страницу создаются новые копии всех активов, на которые ссылаются компоненты группы компонентов.

Кроме того, хотя относительный URL-адрес может подходить для локального компонента, для удаленных компонентов требуется полный URL-адрес любого актива, которым должна управлять платформа Oracle Content Management от вашего имени, чтобы они могли визуализировать контент встроенных рамок с полным URL-адресом.

Поскольку нельзя полагаться на URL-адрес, который остается статическим, в коде необходимо использовать только ссылки на идентификатор актива и извлекать URL-адрес актива, когда требуется рендеринг актива.

Управление активами

Эти API-интерфейсы Sites SDK доступны для управления активами.

```
SitesSDK.getProperty('componentAssets', callback);
```

- Этот интерфейс позволяет получить массив текущих активов.
- Каждая запись актива состоит из указанных ниже элементов.
 - **id** (идентификатор): уникальный идентификатор актива.
 - **заголовок**: метаданные заголовка Oracle Content Management.
 - **описание**: метаданные описания Oracle Content Management.
 - **fileName** (имя файла): исходное имя выбранного файла. Полезно для отображения на панели "Настройки" пользовательского компонента, чтобы пользователи знали, какой файл они выбрали. Это не имя файла, скопированного в папку контента.
 - **source** (источник): URL-адрес актива с поддержкой макрокоманд. Это значение изменяется со временем и не должно ссылаться на компонент. Однако оно должно быть сохранено как часть актива.
 - **URL**: полный URL-адрес актива на основе контекста, в котором был вызван метод `getProperty()`.

```
SitesSDK.setProperty('componentAssets', [assets]);
```

- Этот интерфейс вызывается для сохранения всех активов, которыми платформа Oracle Content Management должна управлять от вашего имени.
- Если этот интерфейс не вызывается, никакие активы не сохраняются.
- После публикации сайта все активы не из этого массива удаляются.
- Параметр `assets` представляет собой массив активов в том же формате, в котором возвращают данные методы `getProperty` и `filePicker`.

Примечание.

Значение `url` не сохраняется. Это значение динамически создается при запрашивании активов.

```
SitesSDK.filePicker(options, callback);
```

- API-интерфейс для вызова средства выбора файлов для выбора списка активов.
- Он вызывает обратный вызов при успешном выборе активов, передаваемых в массиве выбранных активов.

- В этот момент ничего не сохраняется, а вызов `setProperty('componentAssets', [assets]);` для сохранения выбранных элементов в сочетании с другими активами, которые необходимо сохранить, зависит от компонента.

Пример выбора актива

В этом разделе показано, как выбрать актив, сохранить его идентификатор и извлечь фактические значения из сохраненных активов.

1. Измените файл `settings.js`.
2. Измените объект шаблона, включив в него `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. Измените `viewModel`, чтобы добавить наблюдаемый элемент для хранения идентификатора выбранного актива.

```
self.imageID = ko.observable();
```

4. Измените `viewModel` для управления выбором актива путем вызова средства выбора файлов и отображения имя выбранного актива.

```
//
// 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. Обновите `viewModel`, чтобы активы извлекались до получения `customSettingsData`. Этот код также обуславливает вызов `self.imageName` при изменении наблюдаемого элемента `self.ImageID()`.

```

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. Наконец, обновите функцию `save`, чтобы сохранить `imageID`, и обязательно обновите `componentAssets` вместе со списком активов со ссылками.

```

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);

```

Проверка результатов для выбора актива

1. Обновите страницу на своем сайте, чтобы приложение Site Builder могло отслеживать изменения в компоненте.

2. Переключите страницу в режим редактирования.
3. Перетащите свой компонент на страницу.
4. Откройте панель "Настройки".
5. Нажмите кнопку **Выбрать изображение**.
6. Просмотрите (или загрузите) и выберите изображение.

Обратите внимание, что сохраняется имя изображения, указывающее на выбранное изображение.

7. Закройте панель "Настройки".
8. Снова откройте панель "Настройки".

Обратите внимание, что имя изображения снова отражается.

Пример рендеринга актива

В этом разделе показано, как выбирать активы и визуализировать их в компоненте, а также динамически обновлять компонент при изменении значений на панели настроек.

Примечание.

Хотя здесь показан пример локального компонента, который отображается на странице во встроенной рамке, подобный код будет работать для компонентов, рендеринг которых выполняется в пределах этой страницы.

1. Измените файл `render.js`.
2. Обновите шаблон, включив в него свой актив:

```
<!-- 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. В `viewModel` создайте два наблюдаемых элемента, чтобы получить `imageID` из `customSettingsData` и сохранить `imageURL`, извлеченный из сохраненного списка активов.

```
self.imageID = ko.observable();
self.imageURL = ko.observable();
```

4. Обновите `viewModel`, чтобы при каждом изменении идентификатора изображения `imageID` модель получала соответствующий URL-адрес актива изображения.

```
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. Обновите `viewModel`, чтобы извлекать данный идентификатор из `customSettingsData`.

Проверка результатов для рендеринга актива.

1. Обновите страницу на своем сайте, чтобы приложение Site Builder могло отслеживать изменения в компоненте.
2. Переключите страницу в режим редактирования.
3. Перетащите свой компонент на страницу.
4. Откройте панель "Настройки".
5. Нажмите кнопку **Выбрать изображение**.
6. Просмотрите (или загрузите) и выберите изображение.

Обратите внимание, что сохраняется имя изображения, указывающее на выбранное изображение.

7. Закройте панель "Настройки".

В этот момент должно отображаться выбранное изображение, визуализированное в компоненте.

Теперь выполните [Обзор учебного пособия](#).

Обзор учебного пособия

В этом учебном пособии дается обзор процедуры создания настраиваемого компонента с помощью фабрики компонентов `Knockout`.

Основное назначение данного учебного пособия — показать, как с помощью этой структуры создать любой пользовательский компонент, просто обновив объекты JavaScript `SampleComponentViewModel` и `sampleComponentTemplate`. Объекты `sampleComponentFactory` и `SampleComponentImpl` не изменялись на протяжении всего изучения этого учебного пособия. Вы смогли реализовать эти изменения без взаимодействия со страницей и выполнять следующие задачи:

- Сообщение своему компоненту об изменениях, внесенных с панели "Настройки", и сохранить эти изменения.
- Выполнение триггеров и действий, а также взаимодействие с другими компонентами на странице.
- Создание макетов и использование вложенных компонентов.
- Определение характерных для компонентов стилей.

В данном примере пользовательский компонент разделен на несколько файлов. Это сделано для наглядности данного учебного пособия. Для оптимизации следует правильно упаковать файлы, чтобы избежать многократных выгрузок.

Наконец, хотя данное учебное пособие подходит для компонентов на основе Knockout, если вы хотите создать пользовательские компоненты с помощью другого стека технологий JavaScript, например AngularJS, необходимо повторно реализовать объект `SampleComponentImpl`, чтобы создать соответствующее взаимодействие с новой платформой, а также характерную для новых технологий реализацию фактического компонента. Эта работа выходит за рамки данного учебного пособия.

Ссылка на Sites SDK

Пакет Oracle Content Management JavaScript SDK (Sites SDK) содержит набор функций, которые позволяют компонентам более тесно взаимодействовать с Oracle Content Management.

Sites SDK можно выгрузить с сервера Oracle Content Management:

```
http://{server}/_sitesclouddelivery/renderer/app/sdk/js/sites.min.js
```

Определения API-интерфейса сайтов охватывают все определенные пространства имен глобального объекта `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](#)

Основное назначение Sites SDK — позволить пользователям создавать приложения и управлять ими. Он обладает несколькими функциями:

- Обмен данными между встроенными рамками и границами домена
 - Позволяет компоненту обмениваться данными со страницей независимо от места размещения компонента
 - Включает участие в жизненном цикле модели страницы:
 - * создание компонента,
 - * рендеринг компонента с сохраненными свойствами,
 - * обновление компонента и сохранение этих изменений,
 - * удаление компонента со страницы.
 - Сохраняет и извлекает пользовательские настройки, применяемые в целях настройки компонента для использования на странице
 - Использует связь между компонентами для участия в триггерах и действиях
- Геометрические подсказки для страницы (не применяются к пользовательским компонентам)

- Позволяет динамически изменять размер встроенной рамки на странице при получении соответствующего указания от компонента
- В результате компонент отображается как часть страницы, а страница перемещается в ответ на события изменения размера компонента (полосы прокрутки отсутствуют)
- Осведомленность компонента о моменте его вызова средой Oracle Content Management
 - Связь между компонентом и страницей содержит маркер, подписанный закрытым ключом.
 - Sites SDK позволяет компоненту декодировать маркер и упростить проверку вызывающего абонента.

Функции Sites SDK включаются путем предоставления известного пути к сценарию. Пример:

```
<script type="text/javascript"
src="<sdk_install_dir_path_prefix>/sites.min.js">
</script>
```

В Sites SDK есть следующие глобальные объекты:

- SitesSDK
- SitesSDK.Utils

SitesSDK

Глобальный объект SitesSDK можно использовать в конечных точках рендеринга и настроек.

Команда	Задача
SitesSDK.getProperty(propertyName, callback)	Получает значение именованного свойства от сайта хоста.
SitesSDK.setProperty(propertyName, propertyValue)	Задаёт именованному свойству значение <i>propertyValue</i> .
SitesSDK.getSiteProperty(propertyName, callback)	Получает значение именованного свойства от сайта хоста. Например, он может извлечь текущий дизайн темы, который используется сайтом хоста.
SitesSDK.publish(messageType, payload)	Отправляет сообщение серверу и принимает messageType и объект JSON в качестве полезных данных.

Команда	Задача
SitesSDK.subscribe(messageType, callback)	Прослушиватель сообщений, отправленных с сайта хоста. Ее вызов выполняется асинхронно. Это обратные вызовы JavaScript, поэтому функция должна использовать JavaScript Closure или необходимо должным образом привязать функцию, чтобы обеспечить ее доступ к соответствующему контексту.

SitesSDK.Utils

Пространство имен `SitesSDK.Utils` имеет служебные функции, которые могут вызываться всеми конечными точками компонента Oracle Content Management. Это пространство имен доступно для панели "Настройки" удаленного компонента, но не встроенного локального компонента.

Внимание!

Эти функции официально не поддерживаются компанией Oracle; они предназначены для использования в качестве примеров реализации. Используйте их на свой страх и риск.

Команда	Задача
SitesSDK.Utils.Logger	Возвращает объект регистратора, используемый комплектом Sites SDK. Можно установить один из следующих уровней журнала: <ul style="list-style-type: none"> • debug • log • info • error • warn
SitesSDK.Utils.addSiteThemeDesign(cssUrl)	Добавляет текущий проект темы сайта в компонент.
Utils.getStyle(elem, styleProp)	Служебный метод, который рассчитывает высоту встроенной рамки.

SitesSDK.getProperty('componentAssets', callback)

Эта функция возвращает список активов, которые в настоящее время хранятся на сайте от имени заданного компонента.

Параметры

Имя	Тип	Описание
componentAssets	Строка	Вызывает создателя списка.
callback	Функция	Функция обратного вызова

Синтаксис

```
// 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)

Эта функция получает значение именованного свойства из сайта хоста.

Параметры

Имя	Тип	Описание
propertyName	Строка	Имя свойства.
callback	Функция	Функция обратного вызова.

Связанные свойства

Имя свойства	Описание	Настройки	Вывод
height	Высота рамки	Н/Д	Да
width	Ширина рамки	Н/Д	Да
customSettingsData	Данные настроек	Да	Да
styleClass	Выбранный класс стиля	Да	Да
Любой другой атрибут стиля, например borderColor, borderRadius, alignment, borderStyle, margin и т. д.			

Синтаксис

```
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
        }
    );
});
```

Дополнительные связанные свойства, извлеченные из URL-адреса рендеринга

Помимо использования метода `getProperty()` для извлечения значений свойств, их также содержат другие связанные свойства, которые можно извлечь из URL-адреса рендеринга.

PropertyName	Описание
id	Идентификатор компонента
instance	Зашифрованный маркер
width	Ширина встроенной рамки
height	Высота встроенной рамки
viewMode	Значения режима рендеринга
locale	Язык, заданный на сайте или в браузере

PropertyName	Описание
settingsId	Идентификатор текущих настроек

Дополнительные свойства, извлеченные из URL-адреса настроек

Помимо использования метода `getProperty()` для извлечения значений свойств, их также содержат другие свойства, которые можно извлечь из URL-адреса настройки.

PropertyName	Описание
currCompId	Идентификатор редактируемого компонента
instance	Зашифрованный маркер
width	Ширина встроенной рамки настроек
locale	Язык, заданный на сайте или в браузере
OrigSettingsId	Идентификатор исходных настроек компонента
settingsId	Идентификатор новых настроек

SitesSDK.getSiteProperty(propertyName, callback)

Эта функция получает с сайта хоста значение именованного свойства сайта. Пример: извлечение текущего дизайна темы, который используется сайтом хоста.

Параметры

Имя	Тип	Описание
propertyName	строка	Имя свойства сайта
callback	функция	Функция обратного вызова для считывания значения свойства

Свойства Web-узла

Имя свойства	Описание	Настройки	Вывод
theme	URL-адрес текущего дизайна темы	Н/Д	Да

Синтаксис

```
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])

Эта функция обновляет список активов сайта, хранящихся от имени пользовательского компонента.

Параметры

Имя	Тип	Описание
componentAssets	Строка	Создает список активов компонентов.
assets	Объект JSON	Возвращает список выбранных активов.

Аргументы

Имя свойства	Описание
fileTypes	Список поддерживаемых типов файлов.
multiSelect	Выберите один или несколько файлов.

Синтаксис

```
// 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)

Эта функция задает именованному свойству значение *propertyValue*.

Параметры

Имя	Тип	Описание
propertyName	Строка	Имя свойства.
propertyValue	Объект JSON	Значение свойства.

Связанные свойства

В этой таблице приведен набор связанных свойств, которые можно извлечь с помощью метода `setProperty()`.

Имя свойства	Описание	Настройки	Вывод
height	Высота рамки	Н/Д	Да
width	Ширина рамки	Н/Д	Да
customSettingsData	Данные настроек	Да	Да
styleClass	Выбранный класс стиля	Да	Да
description	Используется в баннере для экземпляра локального компонента на странице		
layout	Возвращает выбранный макет только для локальных компонентов. Можно обновить с панели настроек с учетом значений в файле <code>appinfo.json</code> .		
renderStatus	Позволяет пользователю сообщать о выполнении рендеринга.	true	

Любой другой атрибут стиля, например `borderColor`, `borderRadius`, `alignment`, `borderStyle`, `margin` и т. д.

Синтаксис

```
// 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)

Эта функция возвращает список выбранных файлов.

Параметры

Имя	Тип	Описание
options	Строка	Параметры, чтобы указать выбранные файлы для возврата.
callback	Функция	Функция обратного вызова. Если пользователь отменяет средство выбора, обратный вызов (callback) не выполняется.

Синтаксис

```

// 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)

Эта функция возвращает список выбранных файлов.

Функция `SitesSDK.openDocumentPicker()` принимает один аргумент и возвращает обещание, которое решается для выбранных документов. Доступные варианты соответствуют вариантам для компонента `documentsView`, описанного в разделе "Просмотр документов: встраивание пользовательского интерфейса (компонент `documentsView`) — API-интерфейс версии 2" в документе *Интеграция и расширение возможностей Oracle Content Management*.

Синтаксис

```
SitesSDK.openDocumentPicker({
  selectable: "any",
  layout: "grid"
}).then(function (selection) {
  console.log(selection);
});
```

После извлечения документов Oracle Content Management необходимо указать, что идентификаторы этих документов необходимо сохранять на сайте с помощью свойств `SitesSDK.getProperty('componentAssets', callback)` и `SitesSDK.setProperty('componentAssets', [assets])`. В противном случае они не будут опубликованы на сайте и могут быть удалены.

SitesSDK.openAssetPicker(options)

Эта функция возвращает список выбранных файлов.

Функция `SitesSDK.openAssetPicker()` принимает один аргумент и возвращает обещание, которое решается для выбранных активов. Доступные варианты соответствуют вариантам для компонента `assetsView`, описанного в разделе "Просмотр активов: встраивание пользовательского интерфейса (компонент `assetsView`) — API-интерфейс версии 2" в документе *Интеграция и расширение возможностей Oracle Content Management*.

Синтаксис

```
SitesSDK.openAssetPicker({
  select: "single"
}).then(function (selection) {
  console.log(selection);
});
```

После извлечения активов Oracle Content Management необходимо указать, чтобы идентификаторы этих активов сохранялись на сайте с помощью свойств `SitesSDK.getProperty('componentAssets', callback)` и `SitesSDK.setProperty('componentAssets', [assets])`. В противном случае они не будут опубликованы на сайте и могут быть удалены.

SitesSDK.publish(messageType, payload)

Эта функция отправляет сообщение серверу. Он принимает тип сообщения `messageType` и объект JSON в качестве полезных данных. Тип сообщения распознается и обрабатывается на странице. Если обработчика нет, переданный тип сообщения игнорируется.

Параметры

Имя	Тип	Описание
<code>messageType</code>	строка	Наберите сообщение
<code>payload</code>	Объект JSON	Полезные данные сообщения

Типы сообщений

В этой таблице перечислены типы сообщений, которые известны локальному (с использованием встроенной рамки) или удаленному компоненту и могут быть отправлены с помощью функции `SitesSDK.publish()`.

Тип сообщения	Описание
<code>SETTINGS_UPDATED</code>	Используется на панели "Настройки" для отправки обновленного сообщения "Настройки" в конечную точку рендеринга.
<code>TRIGGER_ACTIONS</code>	Используется в конечной точке рендеринга компонента для инициирования действий на сайте хоста.

Синтаксис

```
// raise trigger
SitesSDK.publish(SitesSDK.MESSAGE_TYPES.TRIGGER_ACTIONS, {
    'triggerName': 'scsChangeSettings',
    'triggerPayload': { 'url':
this.configuration()['url'],'feedcount': this.configuration()['limit'] }
});
```

SitesSDK.subscribe(messageType, callback)

Фактически эта функция является прослушивателем сообщений для сообщений, отправленных с сайта хоста. Ее вызов выполняется асинхронно.

Зарегистрированный обратный вызов вызывается, когда SDK получает сообщение определенного типа. Если обратный вызов возвращает значение, оно возвращается на страницу. Если обратный вызов не передан этой функции, удаляются все зарегистрированные прослушиватели сообщений этого типа.

Это обратные вызовы JavaScript, поэтому функция должна использовать JavaScript Closure или необходимо должным образом привязать функцию, чтобы обеспечить ее доступ к соответствующему контексту.

Параметры

Имя	Тип	Описание
messageType	строка	Тип сообщения <ul style="list-style-type: none"> • SETTINGS_UPDATED • TRIGGER_ACTIONS • EXECUTE_ACTION • GET_ACTIONS • GET_TRIGGERS • COPY_CUSTOM_DATA • PASTE_CUSTOM_DATA
callback	функция	Функция при получении сообщения от сайта хоста.

Типы сообщений

Тип сообщения	Описание
SETTINGS_UPDATED	Подпишитесь на это сообщение, чтобы получать уведомления об изменении любого значения на панели "Настройки". Как правило, компонент отвечает, обновляя свою визуализацию в пользовательском интерфейсе.
TRIGGER_ACTIONS	При публикации этого сообщения вызывается триггер и передаются полезные данные.
EXECUTE_ACTION	Подпишитесь на это сообщение для обработки пользовательских действий. Прослушиватель обычно обрабатывает это сообщение, выполняя указанное действие.
GET_ACTIONS	Подпишитесь на это сообщение, чтобы объявить массив действий, которые компонент может выполнить. Возвращает массив действий.
GET_TRIGGERS	Подпишитесь на это сообщение, чтобы объявить массив триггеров, которые компонент может вызвать. Возвращает массив триггеров.
COPY_CUSTOM_DATA	Подпишитесь на это сообщение, чтобы обрабатывать копирование данных пользовательских настроек в буфер обмена. Возвращает объект, представляющий данные, которые необходимо поместить в буфер обмена.
PASTE_CUSTOM_DATA	Подпишитесь на это сообщение, чтобы обрабатывать синтаксический разбор данных пользовательских настроек из буфера обмена.

Синтаксис

Одни прослушиватели ожидают передачи данных, другие — возврата данных.

Пример 1.

Зарегистрируйте прослушиватель сообщений для сообщений типа EXECUTE_ACTION:

```
SitesSDK.subscribe(SitesSDK.MESSAGE_TYPES.EXECUTE_ACTION,  
self.executeAction,self);
```

Зарегистрированная функция callback должна принимать аргумент, как показано ниже:

```
// 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  
  
}
```

Пример 2.

Прослушиватель COPY_CUSTOM_DATA обычно реализуется с помощью кода, подобного следующему:

```
// listen for COPY_CUSTOM_DATA request  
SitesSDK.subscribe(SitesSDK.MESSAGE_TYPES.COPY_CUSTOM_DATA,
```

Метод copyCustomDataListener() возвращает объект, представляющий данные, которые необходимо скопировать в буфер обмена, например:

```
// Handle Copy Style (save customSettingsData to the clipboard)  
self.copyCustomDataListener = function() {  
    return {  
        prop1: this.prop1(),  
        prop2: this.prop2()  
    };  
};
```

Пример 3.

Запрос PASTE_CUSTOM_DATA принимает данные, но ничего не должен возвращать:

```
// listen for PASTE_CUSTOM_DATA request  
SitesSDK.subscribe(SitesSDK.MESSAGE_TYPES.PASTE_CUSTOM_DATA,  
pasteCustomDataListener);
```

А также:

```
// Handle Paste Style (apply customSettingsData from the clipboard)  
self.pasteCustomDataListener = function(data) {  
    ...  
};
```

SitesSDK.Utils.addSiteThemeDesign(cssUrl)

Эта функция создает элемент `link` в HTML-теге `<head>` текущей страницы. Источник задается для пути `cssUrl`.

▲ **Внимание!**

Функции пространства имен `Utils` официально компанией Oracle не поддерживаются. Они предназначены для использования в качестве примеров реализации. Используйте их на свой страх и риск.

Параметры

Имя	Тип	Описание
<code>cssUrl</code>	строка	URL-адрес текущего дизайна темы. Используйте только с удаленными компонентами. Локальные компоненты (реализованные в <code>iframe</code>) должны извлекать из страницы свойства, а не URL-адрес.

Синтаксис

Эта функция обычно используется в сочетании с извлечением текущего дизайна темы из сайта хоста, как показано в этом примере кода:

```
// 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

Эта функция возвращает объект регистратора, используемый комплектом Sites SDK.

 **Внимание!**

Функции пространства имен `Utils` официально не поддерживаются компанией Oracle. Они предназначены для использования в качестве примеров реализации. Используйте функции `Utils` на свой страх и риск.

Можно задать уровень журнала `debug`, `log`, `info`, `error` или `warning`.

Устранение неполадок

У вас могут возникнуть вопросы о создании сайтов. Вот некоторые ответы.

- [Я пытаюсь создать сайт, но отсутствуют шаблоны](#)
- [Не удается удалить сайт](#)
- [Я не могу открыть дерево сайта и отредактировать страницу](#)
- [Компонент добавлен, но не отображается на странице](#)
- [Компоненты папки, файла и диалога не работают](#)
- [Новая версия изображения не отображается на странице после загрузки](#)
- [Часть содержимого исчезла после изменения макета страницы](#)
- [Не работает добавленный компонент, основанный на другом сервисе](#)
- [На моем корпоративном сайте отображается предупреждение](#)

Я пытаюсь создать сайт, но отсутствуют шаблоны

Прежде чем вы сможете создавать сайты, должны быть установлены шаблоны и к ним должен быть предоставлен доступ.

Шаблоны доступны для использования в Oracle Content Management при условии, что администратор установил их и предоставил к ним общий доступ. Обратитесь к администратору.

Кроме того, если вы создаете шаблон с помощью импорта, копирования или создания из сайта, никто не сможет использовать его, пока вы явно не предоставите общий доступ к шаблону.

См. раздел [Конфигурирование настроек сайтов](#) в документе *Administering Oracle Content Management*.

Не удается удалить сайт

Если вам не удастся удалить сайт, это может быть связано с несколькими причинами:

- Сайт находится в онлайн-режиме.
Сначала переведите сайт в автономный режим, а затем удалите его. Для перевода сайта в автономный режим требуется иметь роль "Управление".
- У вас недостаточно полномочий.
Вы можете удалить сайт, если он создан вами (вы - владелец сайта) или другой пользователь предоставил вам доступ к нему и назначил роль "Соавтор" или "Управление".

Чтобы узнать больше, см. [Управление сайтами и настройки сайтов](#).

Я не могу открыть дерево сайта и отредактировать страницу

По умолчанию Site Builder открывается в режиме предварительного просмотра. В этом режиме сайт можно просматривать, но не редактировать. Вносить изменения можно только в режиме редактирования сайта.

Убедитесь, что переключатель  переведен в положение **Редактировать**.

Компонент добавлен, но не отображается на странице

Существует несколько возможных причин этого.

- Проверьте правильность URL-адреса, связанного с удаленным компонентом (приложением). См. [Регистрация удаленных компонентов](#).
- Редактор заключает удаленные компоненты в HTML-элемент, так называемый встроенный фрейм (тег `iframe`). Во встроенный фрейм можно заключить не все удаленные компоненты.

Информацию о возможности заключения во встроенный фрейм уточняйте у поставщика компонента.

Компоненты папки, файла и диалога не работают

Компоненты, которые обмениваются данными с Oracle Content Management, требуют определенных ресурсов и настроек.

Следующие компоненты требуют доступа к ресурсам в Oracle Content Management:

- Список папок
- Список файлов
- Диспетчер документов
- Диалог

Для компонентов папки и файла требуется доступ к API-интерфейсу REST.

Проверьте следующее:

- У компонента должен быть доступ к конечной точке API-интерфейса для выполнения операций REST с папками и файлами.

По умолчанию конечная точка REST API Oracle Content Management доступна при использовании стандартного адреса URL, указанного для сайта. При использовании настраиваемого URL, возможно, потребуется предоставить доступ к конечной точке API REST `/documents` явным образом.

См. [Добавление URL-адреса сайта](#)

Часть содержимого исчезла после изменения макета страницы

Если выбрать макет с меньшим количеством ячеек или по-другому именованными ячейками, существующее содержимое в других ячейках не будет отображаться в новом макете.

Содержимое не удаляется, оно просто не будет отображаться, пока у выбранного макета не будет ячейки с таким же именем.

Убедитесь, что выбираете макет с таким же количеством областей. Если выбранный макет имеет такое же количество областей, но часть содержимого все равно не отображается, возможно, области имеют разные имена. Для устранения несоответствия свяжитесь с разработчиком темы.

Новая версия изображения не отображается на странице после загрузки

Когда вы выбираете изображение для использования с компонентом, создается уникальная копия, которая сохраняется на сайте.

Загруженные файлы хранятся в репозитории сайта. Если загрузить файл с таким же именем, что и у существующего файла, создается новая версия файла.

Файл изображения, отображаемый на странице, является копией файла, выбранного в репозитории. Этот файл не загружается автоматически. Это запланированное поведение, которое предотвращает непреднамеренное внесение изменений в существующие страницы при обновлении файла изображения в репозитории.

Для обновления изображения, используемого с компонентом, необходимо явным образом выбрать изображение на странице свойств компонента. Чтобы узнать больше, см. [Изображения](#).

Не работает добавленный компонент, основанный на другом сервисе

Можно выполнить интеграцию сайта с различными процессами и сервисами.

Если вы создали компонент на основе интеграции с Process Cloud Service, Oracle Intelligent Advisor (OIA) (ранее Oracle Policy Automation) и другими сервисами, то оба сервиса должны использовать одинаковый домен идентификационных данных. Обратитесь к пользователю, который настроил интеграцию, и проверьте, что оба сервиса используют один домен идентификационных данных.

На моем корпоративном сайте отображается предупреждение

Если корпоративный сайт содержит макеты контента версии 1.0, в Site Builder и при просмотре сайта отображается предупреждение, так как макеты не могут обеспечить визуализацию. Их необходимо обновить до макетов контента версии 1.1.

Такая ситуация может возникнуть в двух случаях.

- Если вы делаете существующий сайт переводимым, он автоматически преобразуется в корпоративный сайт путем назначения языка по умолчанию и политики локализации. Если шаблон сайта содержит макеты контента версии 1.0, такие макеты контента не смогут обеспечить визуализацию.
- При создании корпоративного сайта на основе более старого шаблона, использующего макеты контента версии 1.0, макеты контента не смогут обеспечить визуализацию.

Чтобы устранить эту проблему, обновите макеты контента до версии 1.1. В каталоге компонентов можно создать новый макет контента, чтобы увидеть разницу. Необходимо добавить строку для регистрации версии макета контента и изменить `data.fieldname` на `fields.fieldname`.