Oracle[®] Cloud Oracle Content Management でのサイトの 構築



F27099-22 2021 June

ORACLE

Oracle Cloud Oracle Content Management でのサイトの構築、

F27099-22

Copyright © 2018, 2021, Oracle and/or its affiliates.

著者: Bruce Silver

執筆協力者: Jean Wilson、Sarah Bernau、Kalpana N、Ron van de Crommert

協力者: 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.

目次

はじめに

対象読者	xvi
ドキュメントのアクセシビリティについて	xvi
ダイバーシティ&インクルージョン	xvi
関連リソース	xvi
表記規則	xvii

第 | 部 はじめに

1 Oracle Content Management の概要

Oracle Content Management へのアクセス	1-1
ロールの理解	1-2
アセットの管理	1-2
コンテンツの管理	1-3
コンテンツのコラボレーション	1-3
サイトの構築	1-3
Oracle Content Management の統合および拡張	1-4
スタート・ガイド	1-4
Oracle Cloud Infrastructure への移行	1-4

2 サイトの構築のスタート・ガイド

サイトのスタート・ガイド	2-1
サイト作成プロセスの理解	2-2
サイトの開発について学習	2-4
サイト・ガバナンスの理解	2-5

3 初めての Web サイトの作成

始める前に

3-2



ステップ 1: 環境の設定	3-2
Minimal-NavMenu カスタム・コンポーネントのインポート	3-2
Minimal-NavMenu カスタム・コンポーネントの公開	3-3
サンプル・イメージの追加	3-4
ステップ 2 : Web サイトの設定	3-6
Web サイトの作成	3-6
Web サイトの編集	3-7
ステップ 3 : Web サイトの公開	3-27
上級編	3-30

第 || 部 サイトの作成および編集

4 サイトの作成

サイトの作成	4-1
サイトのコピー	4-4
サイト・リクエストの管理	4-5
サイト・リクエスト詳細の変更	4-6
サイト・リクエスト・ポリシーの表示	4-7

5 サイトの編集

サイト・ビルダー・ページについて	5-1
サイト更新の理解	5-4
更新の使用	5-5
編集のヒントとテクニック	5-6
スタイルおよびフォーマットの使用	5-10
サイト設定のカスタマイズ	5-11
カスタム・サイトのプロパティの追加	5-13
表の操作	5-14
サイト・ファイルのアップロード	5-16

6 サイトでのテンプレートおよびテーマの使用

テンプレートの理解	6-1
サイトからのテンプレートの作成	6-4
テンプレート詳細の変更	6-6
テンプレート・ポリシーの変更	6-7
テンプレートのステータスまたはオーディエンスの変更	6-8



テンプレートの管理	6-9
テンプレートのエクスポートとインポート	6-12
テーマの理解	6-14
テーマの管理	6-15
テーマの公開	6-17

7 カスタム・コンポーネントおよびレイアウトの管理

カスタム・コンポーネントの理解	7-1
レイアウトの理解	7-3
カスタム・コンポーネントおよびレイアウトの使用	7-5
リモート・コンポーネントの登録	7-8
ローカル・コンポーネント、レイアウト、コンテンツ・フィールド・エディタまたはコン テンツ・フォームの作成	7-9
コンポーネントまたはレイアウトのエクスポートまたはインポート	7-11

8 サイト・ページの操作

ページへのナビゲート	8-1
ページの表示	8-1
ページの追加	8-3
ページの移動	8-6
ページの削除	8-7
ページ設定の変更	8-7
ページ・レイアウトの変更	8-10
背景またはテーマの変更	8-12

9 ページ・コンテンツの配置

コンポーネントおよびセクション・レイアウトの追加	9-1
アセットおよびコンテンツ・アイテムの操作	9-4
トリガーおよびアクションの使用	9-4
水平セクション・レイアウトの使用	9-6
2列および3列のレイアウトの使用	9-7
垂直セクション・レイアウトの使用	9-9
タブ付きのセクション・レイアウトの使用	9-10
スライダ・セクション・レイアウトの使用	9-10



10 組込みコンポーネントの使用

基本コンポーネント	10-2
タイトル	10-3
段落	10-4
プレーン・テキスト	10-6
ボタン	10-7
構造コンポーネント	10-9
区切り	10-9
スペーサ	10-9
メディア・コンポーネント	10-10
イメージ	10-10
ギャラリ	10-13
ギャラリ・グリッド	10-16
YouTube ビデオ	10-20
ビデオ	10-21
ドキュメント・コンポーネント	10-22
ドキュメント	10-22
フォルダ・リスト	10-24
ファイル・リスト	10-25
ドキュメント・マネージャ	10-26
プロジェクト・ライブラリ	10-28
ソーシャル・コンポーネント	10-31
ソーシャル・バー	10-31
Facebook の「いいね!」と「おすすめ」	10-32
Twitter の共有とフォロー	10-32
会話コンポーネント	10-33
会話リスト	10-35
プロセス・コンポーネント	10-37
プロセス開始フォーム	10-37
プロセス・タスク・リスト	10-39
タスク詳細フォーム	10-41
コンテンツ・アイテム	10-43
コンテンツ・アイテム・コンポーネント	10-43
コンテンツ・プレースホルダ	10-45
コンテンツ・リスト	10-46
コンテンツ検索	10-51
レコメンデーション	10-52
その他のコンポーネント	10-54



マップ	10-54
ヘッドライン	10-55
記事	10-57
テキスト付きイメージ	10-59
コンポーネント・グループ	10-61
ページでの共有ブラウズの使用	10-62
Oracle Intelligent Advisor	10-63
Oracle Visual Builder	10-64

第 ||| 部 サイトの公開と管理

11 サイトの管理

「サイト」ページについて	11-1
サイトおよびサイト設定の管理	11-3
サイトの説明、ロゴまたは埋込みプロパティの変更	11-7
検索エンジンのプロパティの設定	11-8
検索エンジン最適化のためのプリレンダー・サービスの有効化	11-9
静的サイト配信オプションの設定	11-11
公開時の自動コンパイルの有効化	11-11
コンパイル済サイトのデフォルト・キャッシュ制御ヘッダーのオーバーライド	11-12
コンパイル済アダプティブ・レイアウトをサポートするモバイル User-Agent の指定	11-13
バニティ URL の指定および構成	11-13
共有ブラウズ統合の有効化	11-16
分析追跡の追加	11-17

12 サイトの公開

サイトのオンライン化またはオフライン化	12-1
サイト変更の公開	12-2

13 セキュア・サイト

サイト・セキュリティの理解	13-1
サイト・セキュリティの変更	13-4

14 多言語サイトの操作

多言語サイトの概要	14-1
サイトの翻訳	14-2

サイト翻訳ジョブの管理	14-4
翻訳のロケール	14-5
翻訳のカスタム・ロケール	14-5
URL リダイレクトへのロケール・エイリアスの設定	14-6

15 サイト・リダイレクトまたは URL マッピングの使用

リダイレクトのプラン	15-1
単純な文字列間の一致	15-1
簡易ワイルドカード一致	15-2
サイト・リダイレクトの追加	15-2
JSON ファイル内のリダイレクト・ルールの指定	15-3
サイトへのリダイレクト・ルール・ファイルのアップロード	15-9
サイト URL のマッピング	15-9

16 サイト・パフォーマンスの向上

キャッシュの利用によるパフォーマンスの向上	16-1
ランタイム・キャッシュ	16-2
サイト・ビルダーのキャッシュ	16-3
Above the Fold (ATF)レンダリング	16-3

第 IV 部 サイトの開発

17 デザインおよびスタイルのカスタマイズ

デザインについて	17-1
デザイン・ファイル	17-1
レスポンシブ表の設計	17-4
会話リストのスタイルのカスタマイズ	17-8
フォルダ・リストおよびファイル・リストのスタイルのカスタマイズ	17-10
ソーシャル・バーのアイコンのカスタマイズ	17-11
Oracle Intelligent Advisor のインタビュー・スタイル設定拡張機能の構成	17-12

18 背景の使用の理解

背景およびテーマについて	18-1
背景の実装方法	18-1



設定の格納場所

19 トリガーおよびアクションの設定

トリガーおよびアクションについて	19-1
トリガーの設定	19-2
アクションの設定	19-3

20 テンプレートの開発

テンプレートについて	20-1
基本テンプレート構造	20-2
テンプレートの作成	20-4
テンプレートのエクスポート	20-5
テンプレートのインポート	20-6
スタータ・テンプレートの操作	20-7
Bootstrap または Web サイト・デザイン・テンプレートからのサイト・テンプレートの作成	20-11
Developer Cloud Service によるテンプレートの開発	20-17
Oracle Content Management 用の Developer Cloud Service コンソールへのサインイ	
ン	20-17
Developer Cloud Service でのプロジェクトの作成	20-18
Developer Cloud Service でのテンプレートの作成	20-18
Developer Cloud Service でのテンプレートのコピー	20-19
Developer Cloud Service ヘのテンプレートのインポート	20-20
変更のマージ	20-20
Developer Cloud Service からのテンプレートのエクスポート	20-20

21 テーマの開発

テーマについて	21-1
基本テーマ構造	21-3
サイト・ナビゲーション	21-5
テーマの作成	21-7
テーマのコンポーネントおよびセクション・レイアウトの非表示	21-12
テーマのコンポーネントの位置合せ、幅または間隔オプションの非表示	21-14
コンポーネントとテーマの関連付け	21-15
サイト・レンダリング API	21-18



iх

22 レイアウトの開発

レイアウトについて	22-1
検索エンジン最適化 (SEO)	22-3
components.json ファイルおよびフォーマットの理解	22-5
サイト・ビルダーでのツールバー・グループのカスタマイズ	22-9
スロット内のコンポーネントの制限	22-12
レイアウト・コンテンツの編集可能化	22-15
セクション・レイアウトの作成	22-16
遅延ロードをサポートするセクション・レイアウトの作成	22-17
API を使用したカスタム・セクション・レイアウトの開発	22-18
コンテンツ・レイアウトの開発	22-24
Oracle Content Management を使用したコンテンツ・レイアウトの作成	22-26
コンテンツ・レイアウトへのレイアウト・ビューの受け渡し	22-32
API を使用したサイト詳細ページ URL の生成	22-32
Developer Cloud Service を使用したコンテンツ・レイアウトのローカル開発	22-34
Developer Cloud Service によるコンテンツ・レイアウトの作成	22-34
RequireJS モジュールの定義	22-36
コンストラクタ関数パラメータの構成	22-36
コンテンツ・レイアウトのレンダリング	22-37
Mustache テンプレート内のコンテンツ・レイアウトの編集	22-38
動的 DOM 操作の追加	22-38
design.css ファイル内のスタイルの定義	22-38
参照アイテムの取得	22-39
メディア URL の取得	22-39
トリガーの呼出し	22-39
検索問合せを使用した検索ページへの移動	22-40
マクロの展開とリッチ・テキストのレンダリング	22-41
「詳細」ページへのリンク	22-41
コンテンツ・リスト問合せのマクロの展開	22-41
強力なコンテンツ・レイアウトの開発	22-45
コンテンツ・アイテムのレンダリング	22-45
コンテンツ・レイアウトのデータ構造の標準化	22-46
サンプル・ブログ・テンプレートの作成	22-50
テンプレートへのコンテンツ・レイアウト・マッピングの追加	22-51
ローカル・テスト・ハーネスを使用したコンテンツ・レイアウトのテスト	22-51
ローカル・テスト・ハーネスを使用したテスト	22-51



Oracle Content Management へのコンテンツ・レイアウトを含むテンプレートのイン ポート

23 コンポーネントの開発

コンポーネントについて	23-1
コンポーネントの開発について	23-3
コンポーネントの作成	23-6
Developer Cloud Service によるカスタム・コンポーネントの開発	23-6
Oracle Content Management のカスタム・コンポーネントの開発	23-7
カスタム・コンポーネントの開発	23-8
ユニット・テストの作成および実行	23-10
コンポーネントの最適化(縮小)によるパフォーマンスの向上	23-11
連続統合ジョブの実行	23-11
多言語サイト用の翻訳可能コンポーネントの開発	23-12
「設定」パネルを使用した H1 コンポーネントの構築	23-12
新規ローカル・コンポーネントの作成	23-13
基本的な H1 コンポーネントの構築	23-14
コンポーネントへの CSS の追加	23-17
設定パネルの追加によるヘッダー・テキストの変更	23-18
他のユーザーが H1 コンポーネント・スタイルを選択するためのテーマの更新	23-20
HTML テンプレートを使用した簡易コンポーネントの作成	23-21
HTML コンポーネントの作成	23-21
HTML コンポーネントへのユーザー指定データの追加	23-25
ローカル・コンポーネントとリモート・コンポーネントの比較	23-29
コンポーネント設定のレンダリング	23-30
ローカル・コンポーネントの実装	23-32
コンポーネントのスタイル・クラス	23-33
組込みコンポーネントのスタイリング方法	23-35
コンポーネント・スタイルの基本	23-35
コンポーネント固有のスタイル	23-37
コンポーネント・プロパティの設定	23-44
インライン・フレームでレンダリングされたコンポーネント	23-45
インライン・フレームでレンダリングされたコンポーネントのインスタンス ID および構造 について	23-47
リモート・コンポーネントのセキュリティ	23-48
リモート・コンポーネントの登録	23-50
コンポーネントの削除	23-51
サイト SDK	23-52



22-52

24 コントローラ・ファイルのカスタマイズ

コントローラ・ファイルについて	24-1
デフォルト・コントローラ・ファイル	24-1
デフォルトの controller.html ファイルの変更	24-2
SCS オブジェクトについて	24-3
SCS.sitePrefix	24-3
SCS.preInitRendering	24-4
SCS.getDeviceInfo	24-4
カスタマイズできないコントローラ・ファイル・セクション	24-4
トークンを使用したカスタム・コントローラ・ファイルのポータビリティの許可	24-5
カスタム・コントローラ・ファイルのサンプル	24-6
サイト接頭辞の変更	24-6
待機グラフィックのカスタマイズ	24-7
ファビコンのカスタマイズ	24-9
JavaScript 以外のクローラ用の <noscript>および<meta/>タグのカスタマイズ</noscript>	24-10
JavaScript ファイルのプリフェッチ	24-10
追加のマークアップによるサイトの所有権の確認	24-11
デバイス検出の補強	24-12
トークンを使用した controller.htm ポータビリティの拡張	24-13

第 V 部 他のツールを使用したサイトの開発

25 OCE ツールキットでの開発

ローカル・マシンでの OCE ツールキットの設定	25-1
npm を介した依存関係のインストール	25-2
cec コマンドライン・ユーティリティの 使用	25-2
ローカル・テスト・ハーネスを使用したテスト	25-67
jQuery 3.5.x へのアップグレード	25-68
Developer Cloud Service による Oracle Content Management の開発	25-68
Developer Cloud Service の使用について	25-69
Oracle Content Management 用の Developer Cloud Service コンソールへのサインイ	
ン	25-69
Developer Cloud Service でのプロジェクトの作成	25-70
Oracle Content Management テンプレートを使用した Developer Cloud Service プ	
ロジェクトの作成	25-71
Oracle Content Management からの OCE ツールキットのダウンロードを使用し	
た Developer Cloud Service でのプロジェクトの作成	25-71
新規 GIT リポジトリ内のプロジェクト・コードへの OCE ツールキットの追加	25-71



ローカル・テスト・ハーネスでのカスタム・コンポーネント、テンプレートおよびコン テンツ・レイアウトのテスト	25-72
変更のマージ	25-72
OCE ツールキットによるテストから本番への変更の伝播	25-72
パスワードの暗号化	25-79
サーバーの登録	25-79
サイトの使用状況および権限レポートの作成	25-79
ドキュメントおよびフォルダのダウンロードおよびアップロード	25-80
テンプレートからのサイトの作成およびコンテンツの同じ GUID の維持	25-81
標準サイトからのエンタープライズ・テンプレートの作成	25-81
タクソノミのインポートおよびエクスポート	25-81
レコメンデーションのインポートおよびエクスポート	25-83
コレクション・コンテンツの追加または削除	25-84
OCE ツールキットを使用したカスタム・フィールド・エディタの開発	25-85
あるサーバーから別のサーバーへのサイトの転送または更新	25-90
コンテンツ・アイテムのないサイトの転送	25-90
グループでのサイトのコンテンツ・アイテムのダウンロードまたはアップロード	25-91
OCE ツールキットによるサイト・ページの索引付け	25-92
サイト・ページ・テキストのコンテンツ・タイプの作成	25-92
OCE ツールキットによるページ索引のコンテンツ・アイテムの作成	25-93
Oracle Content Management のサイトへのコンテンツ検索の追加	25-94
サイトへの検索ページの追加	25-94
テーマへの検索フィールドの追加	25-94
OCE ツールキットによる多言語サイトの索引付け	25-95
コンポーネント開発を容易にするための簡潔なコンポーネントの作成	25-98
サイト・コンパイル・サービスのセットアップ	25-98
サイトのコンパイルによるサイト・ページのランタイム・パフォーマンスの改良	25-99
サイトのコンパイルの概要	25-100
プリレンダーとのインタラクション	25-100
コントローラ・サイト・ページのレンダリング	25-100
コンパイル済サイト・ページのレンダリング	25-101
テンプレートのコンパイル	25-101
設定	25-101
テンプレートのコンパイル	25-102
サイトのコンパイル	25-108
サイト・コンパイルの前提条件	25-108
サイトのコンパイル	25-108
カスタム・コンパイラ	25-109
制約	25-110



カスタム・コンパイラのデバッグ	25-110
ページ・レイアウト・コンパイラ	25-111
コンポーネント・コンパイラ	25-112
コンテンツ・レイアウト・コンパイラ	25-114
SCSCompileAPI	25-115
コンポーネント・ハイドレーション	25-116
公開	25-119
静的サイト配信の優先度	25-119
キャッシュ・ヘッダー	25-120
詳細ページ	25-120
コンテンツ・アイテムのチャネルへの追加	25-121
モバイル・デバイスのコンパイル・サイト	25-122
サイトのライフサイクルおよびコンパイル済ページ	25-123
Oracle Content Management サーバーでの新規のサイトまたはアセットの翻訳ジョブの作	
成	25-123
言語サービス・プロバイダでのサイトの翻訳	25-126
OCE ツールキットによる翻訳ジョブの作成	25-127
翻訳ジョブのリスト	25-127
翻訳コネクタの作成	25-128
多言語サイトのサイト・マップの生成	25-129
言語サービス・プロバイダへの翻訳ジョブの送信	25-130
サーバーへの翻訳ジョブのアップロード	25-131

第 VI 部 付録

26 チュートリアル: Knockout でのコンポーネントの開発

Knockout によるコンポーネント開発の概要および前提条件	26-1
ステップ 1: コンポーネントの作成	26-2
ステップ 2: ローカル・コンポーネントのレンダリング構造のレビュー	26-4
ステップ 3 : ローカル・コンポーネント設定の構造のレビュー	26-8
ステップ 4 : コンポーネントでの新規プロパティの表示	26-11
ステップ 5: トリガーの登録	26-13
ステップ 6: トリガーの呼出し	26-14
ステップ 7: アクションの登録	26-17
ステップ 8: アクションの実行	26-18
ステップ 9 : コンポーネントのインスタンスごとの個別タイトルの作成	26-19
ステップ 10: インライン編集によるネストしたコンポーネントの使用	26-20
ステップ 11 : 別のレイアウトのサポート	26-23



ステップ 12 : カスタム・スタイルの定義	26-24
ステップ 13: インライン・フレームでのコンポーネントのレンダリング	26-26
ステップ 14: コンポーネントがインライン・フレームでレンダリングされる場合のカスタ	
ム・スタイルの使用	26-29
ステップ 15: ページの「元に戻す」および「繰返し」動作との統合	26-33
ステップ 16: アセットの管理	26-33
チュートリアル・レビュー	26-39

27 サイト SDK リファレンス

SitesSDK.getProperty('componentAssets', callback)	27-3
SitesSDK.getProperty(propertyName, callback)	27-4
SitesSDK.getSiteProperty(propertyName, callback)	27-5
SitesSDK.setProperty('componentAssets', [assets])	27-6
SitesSDK.setProperty(propertyName, propertyValue)	27-7
SitesSDK.filePicker({options}, callback)	27-8
SitesSDK.openDocumentPicker(options)	27-9
SitesSDK.openAssetPicker(options)	27-10
SitesSDK.publish(messageType, payload)	27-10
SitesSDK.subscribe(messageType, callback)	27-11
SitesSDK.Utils.addSiteThemeDesign(cssUrl)	27-13
SitesSDK.Utils.Logger	27-14

28 トラブルシューティング

サイトを作成しようとしていますが、テンプレートがありません	28-1
サイトを削除できません	28-1
サイト・ツリーを開くこともページを編集することもできません	28-2
コンポーネントを追加しましたが、ページに表示されません	28-2
フォルダ、ファイルおよび会話コンポーネントが動作しません	28-2
ページ・レイアウトを変更したら、コンテンツの一部が表示されなくなりました	28-3
イメージの新規バージョンをアップロードしましたが、ページに表示されません	28-3
別のサービスをベースとするコンポーネントを追加しましたが、機能しません	28-3
エンタープライズ・サイトに警告が表示される	28-3



はじめに

このドキュメントでは、サイト、アセットおよびそれらに関連するすべての構造やポリシーなどのエクスペリエンスを Oracle Content Management で作成および管理する方法について説明します。

対象読者

この公開資料は、Web ブラウザ、デスクトップ同期アプリケーションまたはモバイル・ デバイスを使用してファイルを操作する Oracle Cloud ユーザーを対象としています。

ドキュメントのアクセシビリティについて

オラクルのアクセシビリティについての詳細情報は、Oracle Accessibility Program の Web サイト(http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc)を参照してく ださい。

Oracle Support へのアクセス

サポートをご契約のお客様には、My Oracle Support を通して電子支援サービスを提供 しています。詳細情報は(http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info)か、 聴覚に障害のあるお客様は(http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs)を 参照してください。

ダイバーシティ&インクルージョン

Oracle はダイバーシティ&インクルージョンに積極的に取り組んでいます。Oracle は、ソート・リーダーシップと革新性を高める社員の多様性を尊重し、その価値を重 んじています。従業員、お客様、パートナー様にポジティブな影響をもたらすインク ルーシブな文化を醸成する私たちのイニシアティブの一環として、製品やドキュメン トからインセンシティブな用語を取り除くように努めています。また、Oracle 製品お よび業界標準が進化する中、お客様の既存の技術との互換性を維持する必要性および サービスの継続性確保の要求にも留意しています。このような技術的な制限により、 当社のインセンシティブな用語を削除する取組みは継続中であり、時間と皆様のご協 力が必要となります。

関連リソース

詳細は、オラクル社の次の資料を参照してください。

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

表記規則

このマニュアルでは次の表記規則を使用します。

表記規則	意味
太字	太字は、アクションに関連付けられたグラフィカル・ユーザー・インタフェ ース要素や、本文または用語集で定義されている用語を示します。
イタリック体	イタリック体は、ブック・タイトル、強調、またはユーザーが特定の値を指 定するプレースホルダー変数を示します。
固定幅フォント	固定幅フォントは、段落内のコマンド、URL、サンプル内のコード、画面に 表示されるテキスト、またはユーザーが入力するテキストを示します。





はじめに

このパートでは、Oracle Content Management でのサイトの構築に関する概念情報を示しま す。最初のサイトを作成するための使いやすいチュートリアルも含まれます。次の章が含ま れます。

- サイトの構築のスタート・ガイド
- 初めての Web サイトの作成



1 Oracle Content Management の概要

Oracle Content Management は、オムニチャネルのコンテンツ管理を促進し、エクスペリエンスの配信を加速するクラウドベースのコンテンツ・ハブです。これにより、強力なコラボレーションおよびワークフロー管理機能が提供され、コンテンツの作成と配信が容易になり、顧客と従業員のエンゲージメントが向上します。

Oracle Content Management では、任意のデバイスで内部的および外部的に迅速にコラボレートして、コンテンツの承認およびコンテキスト化されたエクスペリエンスの作成を行うことができます。組込みのビジネスフレンドリなツールにより、魅力的なコンテンツの新しい Web エクスペリエンスを簡単に構築できます。同じコンテンツ・プラットフォームおよび同じプロセスを使用して、すべての関係者にデジタル・エンゲージメントを推進できます。技術的および組織的なボトルネックが解消されるため、魅力的なエクスペリエンスを作成するための障壁がなくなります。

🖍 ノート:

Oracle Content Management Starter Edition の機能セットは制限されています。完 全な機能セットを利用するには、**Premium Edition** にアップグレードします。

Oracle Content Management へのアクセス

Oracle Content Management へのアクセス権が付与された後、インスタンス URL と自分のユ ーザー名に関する詳細が記載されたようこそ電子メールが送られてきます。この情報はサー ビスにログインするために必要であるため、今後の参照用に保持しておくことをお薦めしま す。

Oracle Content Management と対話するための様々な方法があります:

- Web インタフェースは、お気に入りの Web ブラウザからの簡単なアクセスを提供します。クラウド内のコンテンツの管理、他のユーザーとのファイルおよびフォルダの共有、 会話の開始および参加、Web サイトの作成(許可されている場合)などを行うことができます。
- デスクトップ・アプリケーションでは、クラウドとコンピュータ間でファイルおよびフォ ルダの同期を保ちます。独自のファイルおよび共有されているファイルを同期し、常に最 新バージョンにアクセスできるようにすることができます。
- Microsoft Office アドオンにより、Microsoft Word、Excel、PowerPoint および Outlook か ら直接 Oracle Content Management 機能にアクセスできます。
- Android および iOS 用のモバイル・アプリケーションにより、電話やその他のモバイル・ デバイスで簡単にアクセスできます。モバイル・アプリケーションの外観と動作は Web ブラウザのサービスに似ているため、すぐに慣れることができます。クラウド・コンテン ツへのアクセス、ファイルとフォルダの検索およびソート、コンテンツの共有、および会 話の操作を行うことができます。



 REST API および SDK は、Oracle Content Management の機能を Web アプリケーションとモバイル・アプリケーションにプログラムで組み込むための強力なツール を開発者に提供します。

ロールの理解

アクセスできる Oracle Content Management 機能は、割り当てられているロールによって決まります。ユーザー・ロールに応じて異なるオプションが表示されます。標準ユーザーは、ドキュメント、会話およびサイトを操作できます。エンタープライズ・ユーザーは、アセットにアクセスすることもできます。開発者には、テンプレート、テーマ、コンポーネント、レイアウトなど、Web サイトの部分を構築およびカスタマイズするオプションが表示されます。管理者には、サービスの構成、サービスと他のビジネス・アプリケーションの統合およびアセット・リポジトリの設定を行うオプションが表示されます。

Oracle Content Management には、様々なタイプのロールがあります:

- **組織ロール** 組織内のロールにより、実行する必要があるタスクと機能の使用方 法が決まります。
- ユーザー・ロール ユーザー・ロールによって、Oracle Content Management で 表示される機能が制御されます。
- リソース・ロール(権限) リソース(ドキュメント、コンテンツ・アイテム、サイト、テンプレートなど)に関連して表示される内容および実行できる操作は、リソースの共有時に割り当てられたロールによって決まります。

さらに学ぶ...

アセットの管理

Oracle Content Management は、Web サイト、マーケティング資料、電子メール・キャンペーン、オンライン・ストア、ブログなど、様々なチャネルで使用するすべてのアセットを管理するための強力な機能を提供しています。すべてのアセット用の中央コンテンツ・ハブを提供します。そこでは、アセットをリポジトリおよびコレクションに整理し、アセットが使用される方法と場所を定義するルールを作成できます。

アセットの作成および承認プロセスをガイドし、認可されたバージョンのみが使用可 能になるようにする広範な管理およびワークフロー機能もあります。

アセットのタグとフィルタは簡単であるため、必要なアセットを迅速に見つけ、許可 されている場合にそれらを使用できます。また、スマート・コンテンツ機能により、 アセットを使用するときにタグ付けと推奨が自動的に行われます。

アセット・タイプを作成して、ユーザーがアセットを作成する際に収集する必要があ る情報を定義します。デジタル・アセット・タイプでは、デジタル・アセット(ファイ ル、イメージおよびビデオ)に必要なカスタム属性を定義します。コンテンツ・タイプ では、コンテンツの様々な部分を再利用可能な単位にグループ化します。ユーザーは、 それらのアセット・タイプに基づいてデジタル・アセットとコンテンツ・アイテムを 作成して、チャネル間での一貫した使用および表示を確保できます。

さらに学ぶ...



コンテンツの管理

Oracle Content Management では、クラウド内でコンテンツの管理を行うことができます。 すべて 1 箇所で行うことができ、どこからでもアクセスできます。

フォルダ内のファイルをグループ化し、ローカル・コンピュータ上とほぼ同じ方法で一般的 なファイル管理操作(コピー、移動、削除など)を実行できます。また、すべてのファイルがク ラウドに存在するため、どこにいても、またモバイル・デバイスからでもアクセスできます。 デスクトップ・アプリケーションをインストールする場合、すべてのコンテンツはローカル・ コンピュータと自動的に同期できるため、常に最新バージョンを利用できます。

さらに学ぶ...

コンテンツのコラボレーション

クラウド内のすべてのコンテンツでは、ファイルまたはフォルダを共有して、組織の内部ま たは外部の他のユーザーと簡単にコラボレートできます。コンテンツを共有するすべてのユ ーザーは、必要に応じていつでもどこでも最新情報にアクセスできます。フォルダ全体への アクセス権を付与し、特定のアイテムへのリンクを提供できます。共有アイテムへのすべて のアクセスが記録されるため、各共有アイテムがアクセスされた方法とタイミングを監視で きます。

Oracle Content Management の会話では、リアルタイムでのトピックのディスカッションお よびコメントの投稿によって他のユーザーとコラボレートできます。任意のトピックについ て独立した会話を開始し、必要に応じてファイルを追加できます。または、迅速かつ簡単に フィードバックを行うために、特定のファイル、フォルダ、アセットまたはサイトに関する 会話を開始できます。

会話に関連付けられているすべてのメッセージ、ファイルおよび注釈は保持されるため、ディスカッションを簡単に追跡およびレビューできます。また、会話はクラウド内にあるため、 外出中にモバイル・デバイスからそれらを表示して参加することもできます。

さらに学ぶ...

サイトの構築

Oracle Content Management では、概念から起動まで、マーケティングおよびコミュニティ Web サイトを迅速にビルドおよび公開して、魅力的なオンライン・エクスペリエンスを提供 できます。プロセスは完全に統合されています: コンテンツ、コラボレーションおよび創造性 が1つのオーサリングおよび公開環境に結合されています。

迅速に開始するには、すぐに使用可能なテンプレートを使用し、コンポーネント、サンプル・ ページのレイアウトおよびサイトのテーマをドラッグアンドドロップして、定義済の構築ブ ロックからサイトを組み立てます。または、開発者はカスタム・テンプレート、カスタム・ テーマまたはカスタム・コンポーネントを作成して、固有のオンライン・エクスペリエンス を作成できます。

コンポーネントをページ上の指定されたスロットにドラッグ・アンド・ドロップするだけで、 YouTube ビデオ、ストリーミング・ビデオ、イメージ、ヘッドライン、段落、ソーシャル・ メディア・リンク、その他のサイト・オブジェクトを追加できます。ワンタッチでテーマを 切り替え、サイトをリブランドして、最適化され一貫したルック・アンド・フィールを組織 全体に提供できます。



1つ以上の更新を行い、サイトの更新をプレビューし、準備ができたら、1回クリック するだけで更新を公開できます。

サイト・ビルダーでのサイトの作成および公開に加えて、Oracle Content Management では、REST API、React JS、Node JS、その他の Web テクノロジを使用した'ヘッド レス'サイト開発もサポートされます。

さらに学ぶ...

Oracle Content Management の統合および拡張

Oracle Platform-as-a-Service (PaaS)製品として、**Oracle Content Management** は他の **Oracle Cloud** サービスとシームレスに連携します。

Web アプリケーションへの Web UI の埋込みが可能であるため、ユーザーはコンテン ツと直接対話できます。アプリケーション統合フレームワーク(AIF)を使用して、カス タム・アクションを通じてサードパーティ・サービスおよびアプリケーションを Oracle Content Management インタフェースに統合できます。またはコンテンツ・コネクタを 開発して、すでに他の場所で作成したコンテンツを Oracle Content Management に取 り込み、一元的に管理し、複数のチャネルの新規エクスペリエンスで使用できます。

コンテンツとサイトの管理、配信およびコラボレーション用の REST API および SDK の充実したセットにより、Oracle Content Management 機能を Web アプリケーション に組み込むことができます。

クラウド内のコンテンツ SDK およびアセットと対話するクライアント・アプリケーションを作成します。コラボレーション・オブジェクトとのカスタム統合を開発するか、 必要に応じて使用するアセットを取得します。Web サイト、コンテンツ配信ネットワ ーク(CDN)、モバイル・アプリケーションのいずれを使用するかにかかわらず、各チャ ネルに対して最適化されたすべてのコンテンツとアセットのアクセスと 配信が可能で す。

さらに学ぶ...

スタート・ガイド

Oracle Content Management の開始の支援が必要な場合は、Oracle Help Center を参照 してください。ここには、ドキュメント、ビデオ、ガイド・ツアー、開発者情報など の多くのリソースがあります。

必要な場合は、役に立つサポートおよびコミュニティもあります。

Oracle Cloud Infrastructure への移行

Oracle Content Management サブスクリプションが Oracle Cloud Infrastructure (OCI) 上で Infrastructure コンソールを使用してまだ実行されていない場合は、そのネイティ ブ OCI 環境に移行することをお薦めします。これにより、Oracle のクラウド・プラッ トフォームの利点と先進機能を将来にわたって活用できるようになります。

移行は自動ではありません。サービス・リクエストを送信してプロセスを開始する必 要があります。

さらに学ぶ…



2 サイトの構築のスタート・ガイド

Oracle Content Management は、オムニチャネルのコンテンツ管理を促進し、エクスペリエ ンスの配信を加速するクラウドベースのコンテンツ・ハブです。Oracle Content Management では、任意のデバイスで内部的および外部的に迅速にコラボレートして、コンテンツの承認、 デジタル・アセットの管理およびコンテキスト化されたエクスペリエンスの作成を、ビジネ スフレンドリな組込みツールを使用して行うことができます。

- Oracle Content Management の概要
- サイトのスタート・ガイド
- サイト作成プロセスの理解
- サイトの開発について学習
- サイト・ガバナンスの理解
- 初めての Web サイトの作成

ビデオ

サイトのスタート・ガイド

適切な権限があれば、誰でも Oracle Content Management で Web サイトを構築できます。専 用のツール、コードまたはソフトウェアを使用する必要はありません。ユーザー・インタフ ェースがグラフィカルで直観的であり、簡単に使用できます。

N-F:

Oracle Content Management Starter Edition では、1 つのサイトに制限され、サイト・ガバナンスはありません。完全な機能セットが必要な場合は、Oracle Content Management Premium Edition にアップグレードします。

誰がサイトを構築できるかは、いくつかの要因によって決まります。

- サービス管理者がサイトの作成を有効にしているかどうか。
- サービス管理者がサイト・ガバナンスを有効にしているかどうか。サイト・ガバナンスの 理解を参照してください
- サイトの作成がサイト管理者に制限されているかどうか(サイト・ガバナンスが無効になっている場合に可能)。
- サイト管理者がテンプレートを使用可能にしているかどうか。

サイトを作成する場合、テンプレートから開始します。テンプレートには、ユーザーがサイトの使用を開始するにあたり必要なもの、たとえば、サイト・コード・フレームワーク、サンプル・ページとコンテンツを含むデフォルト・サイト、スタイルが設定されたテーマ、イ



メージなどのリソース、カスタム・コンポーネントなどがすべて含まれます。サイト 作成プロセスの理解を参照してください。

エンタープライズ・サイトを作成している場合、そのサイトはリポジトリに関連付け られ、デフォルトの言語を指定するローカリゼーション・ポリシーの定義が必要にな ります。サイトを作成する前に、リポジトリとローカリゼーション・ポリシーの両方 を作成しておく必要があります。サイトに必要なアセットやドキュメントはリポジト リに格納され、アセットで実行できることはリポジトリ・ポリシーによって決定され ます。リポジトリの使用方法の詳細は、「アセット・リポジトリの理解」を参照してく ださい。

サイトを編集する場合、新しい更新を作成するか、既存の更新を使用します。更新内 で、コンテンツの編集と追加、スタイル設定の調整、ページの追加と削除、ページ・ レイアウトの変更、およびページの編成を行うことができます。エディタの操作方法 については、「サイト・ビルダー・ページについて」を参照してください。

更新は1回ですべて準備する必要はありません。自分と自分のチームのメンバーが複数の更新上で同時にかつ別々に作業できます。たとえば、週間ニュースの更新を作業している一方で、他のチーム・メンバーが次回の販売会議のページを追加している場合などです。必要に応じて、更新に対する変更の編集、確認および保存を行い、ベース・サイトと更新をいつでもマージできます。

準備ができたら、サイトを開始します。1回クリックするだけで Web にサイトを公開 できます。つまり、簡単に言えば、概念から起動までです。

サイト作成プロセスの理解

Web サイトを作成して公開するための全体的なプロセスを見てみましょう。

始める前に、サービス管理者がサイトの作成を有効にし、誰がサイトを作成できるか を構成し、サイトの作成に使用するテンプレートをインストールして使用可能にして おく必要があります。サービス管理者は、ユーザーに少なくともテンプレートに対す るダウンロード実行者権限を割り当てる必要があります。ユーザー・ロールについて 学習に関する項、「サイトのスタート・ガイド」および「サイト・ガバナンスの理解」 を参照してください。

アセットおよび多言語サイトの使用が可能なエンタープライズ・サイトを作成する場 合、そのエンタープライズ・サイトはリポジトリに関連付けられます。サイトに必要 なアセットやドキュメントはリポジトリに格納され、アセットで実行できることはリ ポジトリ・ポリシーによって決定されます。エンタープライズ・サイトを作成する前 に、リポジトリを作成しておく必要があります。アセット・リポジトリの理解を参照 してください。

次に、詳細情報へのリンク付きで、サイト作成プロセスの概要を示します。



- テンプレートを選択してサイトに名前を付ける: サイト・ガバナンスが有効になっていて、 サイトの承認が必要な場合、次のステップを完了するにはサイトの承認を待つ必要があり ます。サイトは、公開する準備ができるまでオフラインのままです。
- 2. サイトを作成する: カスタマイズできる空のフレームワークが作成されます。
- 3. 新しい更新を作成するか既存の更新を使用する: 更新は、現在のベース・サイトに対する 変更の名前付きコレクションです。エディタでサイトを表示または編集するたびに、更新 を使用します。サイト更新の理解を参照してください。
- エディタでサイトを開く:ドラッグ・アンド・ドロップのエディタを使用して、サイトに ページおよびコンテンツを追加します。サイト更新の理解を参照してください。
 - **サイトにページを追加する**: 事前定義済レイアウトを選択して、ページのタイプをす ばやく定義します。ページの追加を参照してください。
 - ページにコンポーネントを追加する:ページにテキスト、イメージ、ドキュメントなどをドラッグ・アンド・ドロップします。組込みコンポーネントの使用を参照してください。
 - コンポーネントのコンテンツまたはプロパティを変更する: コンポーネントの間隔、 位置合せおよびその他のプロパティを微調整します。スタイルおよびフォーマット の使用を参照してください。
 - ページに使用されるレイアウトを切り替える(オプション):ページ・デザインをオンザ フライで変更します。ページ・レイアウトの変更を参照してください。
 - サイトに使用されるテーマを切り替える(オプション): サイト全体のルック・アンド・フィールをすばやく変更します。サイトでのテンプレートおよびテーマの使用を参照してください。
 - 検索エンジン・プロパティを設定する: 検索エンジンがサイトの検出を促し、アクセ スが増加します。検索エンジンのプロパティの設定を参照してください。
- 5. サイト更新を適用する: 更新内の変更でサイトを更新します。サイト変更の公開を参照し てください。
 - 他のチーム・メンバーとサイトを共有する:指定したユーザーとサイトを共有し、各 ユーザーにロールを割り当て、サイトで実行できることを決定します。共有の詳細 は、サイト・セキュリティの理解を参照してください。
 - サイト・セキュリティを設定する:公開されたサイトにどのユーザーがアクセスできるかを選択します。サイト・セキュリティの変更を参照してください。



- **サイトを公開する:** 変更をサイトに公開します。サイト変更の公開を参照して ください。
- サイトをオンライン化する: サイト・セキュリティに基づいて、ユーザーに対してサイトを使用可能にします。サイトのオンライン化またはオフライン化を参照してください。

サイトの開発について学習

サイト開発者として、次のように、サイト作成者がサイトの構築に使用するフレーム ワークを定義します:

- 標準テンプレート。サイトのフレームワークとして使用されます。テーマに基づき、サンプル・ページとコンテンツ、カスタム・コンポーネント、およびサイトの 構築を開始するのに必要な他のリソースが含まれています。
- テーマ。サイトの全体のルック・アンド・フィールの定義に使用されます。ロゴ、 スタイル・シート、構成ファイル、およびサイト・ナビゲーションを定義するバッ クグラウンド・コードから構成されます
- カスタム・コンポーネント。特定のタイプのコンテンツのサイト・ページへの追加 に使用されます。これにより、選択したページ・テクノロジを使用してサイト・ペ ージ内に埋め込むことのできる複合要素を開発できます

サイト・プレゼンテーションをサイト・コンポーネントから分離することにより、作 成したサイトが組織の標準とブランディングを維持し、コンテンツ作成者はコンテン ツに集中でき、開発およびサイト作成がより効率的および効果的に行われます。

次に、コンテンツのサイトへの追加時にコンテンツ作成者が使用できる単純なテンプ レートの例を示します。汎用のプレースホルダーがページに配置されていることに注 意してください:





次に、コンテンツの追加後にプレビューしたときに、単純なテンプレートを使用して構築されたサイトの外観の例を示します。どのようにプレースホルダーが特定のコンテンツおよび ナビゲーション要素でカスタマイズされているかに注意してください:



さらに、サイト開発者は Web インタフェースまたは OCE ツールキットを使用してスタイ ル・シート、サイトの構築を作成および維持し、Oracle Content Management と他のサービ ス間に統合を構成します。また、他の従業員と同様に、コンテンツを共有し、会話を開始ま たはこれに参加し、デスクトップまたはモバイルアプリケーションを使用して、他のユーザ ーとコラボレートも行うことができます。

Oracle Content Management を使用できるようにするには、開発者に標準ユーザーまたはエ ンタープライズ・ユーザーのロールが割り当てられている必要があります。標準ユーザー・ ロールを持つ開発者は、コンポーネント、テーマおよび標準テンプレートを作成できます。 エンタープライズ・ユーザー・ロールを持つ開発者は、レイアウトを作成でき、サイトを標 準テンプレートまたはエンタープライズ・テンプレートとして保存することもできます。

サイト開発以外に、Oracle Content Management は、クラウドでの強力で柔軟なバックエンド・コンテンツ管理システム(CMS)としてヘッドレス環境で使用することもできます。

サイト・ガバナンスの理解

サイト・ガバナンスを使用すると、ビジネス・ユーザーは、企業ポリシーに準拠したサイト を簡単に作成できます。また、サイト管理者は、サイトの制御および追跡を一元化された場 所から簡単な方法で行えます。



🖍 ノート:

Oracle Content Management Starter Edition では、1つのサイトに制限され、 サイト・ガバナンスはありません。完全な機能セットが必要な場合は、Oracle Content Management Premium Edition にアップグレードします。

問題

現在、この業界には1つの問題があります。企業は、エクスペリエンス・エクスプロ ージョンに悩まされています。これらのエクスペリエンスは、企業のブランディング とメッセージングに一致するように集約して可視化しながら、迅速に作成して展開す る必要があります。エンタープライズでは、新しいエクスペリエンスを開発して配布 するまでに途方もない時間がかかる気がすることもよくあります。成功のために、迅 速にスケーリングして多くのエクスペリエンスを一貫した方法で作成および管理し、 製品またはブランド・イニシアチブを推進できるように、企業ではパワフルで柔軟な テクノロジを必要としています。新しいエクスペリエンスを迅速かつ効率的に使用で きるようにする必要があり、そうでなければ、市場機会を失い、顧客や見込み客を適 切にエンゲージできなくなります。

エクスペリエンス・エクスプロージョンは、長年にわたる個々のサイロ化されたイニ シアチブの結果であり、その間、各ディビジョン、ブランドおよびチームは、Web コ ンテンツ管理(WCM)システムを中心的な監視なしで入手してきました。これにより、 市場展開に手間取り、セキュリティの脅威に脆弱で、予期せずコストが高騰する複雑 な環境が生み出されています。

エクスペリエンス作成のための一貫したプロセスの欠如は、異種混合された WCM テ クノロジの使用と相まって、組織を非常に大きなセキュリティ・リスクにさらします。 複数のステークホルダーが個々に Web サイトの小さな問題を抱えている場合、セキュ リティなどの中核的な責任は断片化され、エンタープライズは脆弱なまま取り残され ます。異なる Web アプリケーションと IT インフラストラクチャの監視に基づく異な る WCM システムにより提供されるこれらすべてのエクスペリエンスでは、不可能で はないにしろ、すべてを適切に保護することは難しくなります。

さらに、複数の高価な商用 WCM システムを維持するため、重複コストが発生します。 毎年ライセンス更新料がかかり、内部または外部パートナによる継続的なサポート・ コストも徴収されます。

これにより、すべてのユーザーが次のような制約を受けます。

- ビジネスの制約:
 - セルフサービスを使用できません。IT または費用のかかる外部委託に依存します
 - テクニカル・ヘルプに頼らずに更新することができません
 - 作業を管理するためのビジネスに特化したツールがありません
 - すべてのエクスペリエンスとアクティビティを 集約して可視化できません
- IT の制約:
 - すべてのエクスペリエンスに IT の関与が必要です
 - 外部委託されたエクスペリエンスがモニターされないなど、ビジネスにより作 成されるエクスペリエンスに対するガバナンスがありません



- エクスペリエンスの構築に使用されるスタックが断片化します
- 何百ものエクスペリエンス全体でコンテンツとレイアウトに対する変更を管理およびデプロイする必要があります
- ユーザーの制約:
 - チャネル間でのメッセージングに一貫性がありません
 - 情報が古いままです
 - パフォーマンスが貧弱で、チャネルの放棄につながります

ソリューション

Oracle Content Management のガバナンスにより、ビジネス・ユーザーに対するエクスペリ エンスの配布が簡略化および迅速化され、IT 部門は一元化された場所からエクスペリエンス を容易に制御およびトラッキングしつつ、エクスペリエンスのライフサイクル全体を完全に 管理し、企業が必要とする個々の新しいエクスペリエンスの作成および維持のためのコスト を削減できます。ガバナンスは、Oracle Content Management のコアに組み込まれており、 管理者が有効化するのみで済みます。Administering Oracle Content Management のサイト およびアセット設定の構成を参照してください。

	Home	=	Create Site		Cancel	N
		Site				
Ð		SIL	< 😔 💿	5		
			Choose Template Configure Sit	e Add Details		
		Te	Your site will have the following configuration Access Security	a l		
			Anyone without signing in			
믿			Type * Required field			
			Enterprise •	Constant of the second		
			Asset Repository			
		1	Select or create a repository where your site assets will be stored •	and the second second	24	
			Localization Policy *			
			Select or create a policy to determine the languages this site can support $\ ullet$			
			Default Language *			
			Select the default language for your site from the localization policy			

機能

承認済テンプレートのカタログ—IT 開発者は、様々な事業部門のニーズに対応したサイト・テンプレートのセットをテンプレート・カタログに移入できます。使用可能にするテンプレートや、使用可能にするユーザーを制限できます。テンプレート・ポリシーを適用して、新しいサイトが準拠する必要のあるセキュリティのタイプと、新しいサイトが承認を必要とするかどうかを指定できます。これにより、新しいサイトをリクエストするための迅速で簡単な方法が可能になると同時に、ビジネス・ユーザーがブランドおよびセキュ



リティ・ガイドラインに準拠することが保証されます。テンプレートのステータス またはオーディエンスの変更およびテンプレート・ポリシーの変更を参照してくだ さい。

- 効率化されたリクエスト、承認およびプロビジョニング―ビジネス・ユーザーは、 新規サイトを必要な承認および自動プロビジョニングとともに迅速にリクエスト できます。サイトの作成およびサイト・リクエストの管理を参照してください。
- サイト管理—IT 部門は、誰がサイトを作成してデプロイしたかに関係なく、すべてのサイトを1箇所から管理できます。IT ユーザーは、任意のデプロイ済サイトについて、サイト・ステータスのモニターおよびステータスの変更が行えます。サイトおよびサイト設定の管理を参照してください。

メリット

ガバナンスを使用すると、個々の新しいエクスペリエンスの作成および維持のための コストが削減されます。

- ビジネスのメリット:
 - IT に依存せずに迅速にプロビジョニングできます
 - コンテンツとデザインを分離して再利用できます
 - 1つのコンソールでエクスペリエンス、ユーザーおよび権限をグローバルに管 理できます
 - レポートを通じてエクスペリエンス業務のインサイトを取得できます
- IT のメリット:
 - 複雑な実装がありません(クラウドネイティブ・ソリューション)
 - 成長と季節型ピークに応じて自動的にスケーリングできます
 - エクスペリエンスに対するガバナンスにより、安全性と企業ブランディングおよびコンプライアンス標準への準拠が保証されます
- ユーザーのメリット:
 - チャネル間での一貫したメッセージ
 - 最新の情報
 - 最適なパフォーマンス
 - セキュアなエクスペリエンス

プロセス

サイト・ガバナンスを使用してサイトを作成および管理する際に必要なステップは次 のとおりです。

- **1.** 管理者がガバナンスを有効にします。*Administering Oracle Content Management* のサイトおよびアセット設定の構成を参照してください。
- サイト管理者が承認済のテンプレートを使用可能にして、ユーザーがサイトを作成 できるようにします。テンプレートのステータスまたはオーディエンスの変更お よびテンプレート・ポリシーの変更を参照してください。
- **3.** ユーザーが承認済のテンプレートからサイト・リクエストを作成します。サイトの 作成を参照してください。



- 4. サイト管理者がサイト・リクエストを承認し、サイトが自動的に作成されます。サイト管理者がサイト・リクエストを拒否し、リクエストが拒否された理由を説明したノートを添付することもできます。その場合、サイト作成者が問題を修正して、リクエストを再度送信できます。サイト・リクエストの管理を参照してください。
- 5. サイト作成者がサイト作成プロセスを続行して、サイトの編集および公開を行います。サ イト作成プロセスの理解を参照してください。
- 6. 継続的なサイト管理および更新は、「サイト」ページで実行できます。サイト管理者は、 「サイト」ページに*すべての*サイトを表示できます。他のユーザーは、自分で作成したサ イトまたは自分がメンバーになっているサイトを表示できます。サイトおよびサイト設 定の管理を参照してください。



3 初めての Web サイトの作成

Oracle Content Management を使用すれば、初めての Web サイトを短時間で簡単に作成できます。

このチュートリアルでは、Oracle Content Management に用意されている Blank-Template という、すぐに使用可能なテンプレートで Web サイトを設定し、公開する方法を学習しま す。サイト・ビルダーを使用すれば、追加でコーディングする必要がなく、初めての Web サ イトでも非常に簡単に作成できることがわかります。

このチュートリアルは、次の3つの基本ステップで構成されています:

- **1.** 環境の設定
- 2. Web サイトの設定
- 3. Web サイトの公開



始める前に、いくつかの確認事項があります。



始める前に

開始するには、次のアプリケーション・ロールで、Oracle Content Management イン スタンスにアクセスできる必要があります:

- CECContentAdministrator
- CECDeveloperUser

これらのロールがない場合は、サービス管理者に割当てを依頼してください。

Oracle Content Management インスタンスにアクセスできるようになったら、コンテ ンツ管理者としてサインインします。

それでは、始めましょう.

ステップ 1:環境の設定

まず、Minimal-NavMenu カスタム・コンポーネントをインポートし、サンプル・イメ ージをアップロードして環境を設定する必要があります。

そのため、https://www.oracle.com/middleware/technologies/content-experiencedownloads.html にあるアセット・パック OCESamplesAssetPack.zip をダウンロード してください。ダウンロードしたアセット・パックで、次の2つの zip ファイルを含む OCECreateYourFirstSite_data.zip ファイルを探します:

- Minimal-NavMenu.zip: Minimal-NavMenu カスタム・コンポーネントが含まれてい ます
- Minimal-Images.zip: Web サイトで使用するサンプル・イメージが含まれています。

それでは、環境の設定を開始します:

- 1. Minimal-NavMenu カスタム・コンポーネントのインポート
- 2. Minimal-NavMenu カスタム・コンポーネントの公開
- 3. サンプル・イメージの追加

Minimal-NavMenu カスタム・コンポーネントのインポート

Oracle Content Management Web インタフェースにログインしたら、左側のナビゲー ション・メニューで「開発者」、「すべてのコンポーネントの表示」の順にクリックし ます。「開発者」オプションが表示されない場合は、必要なユーザー・ロールがありま せん。

「コンポーネント」ページで「作成」をクリックし、「コンポーネントのインポート」 を選択します。Minimal-NavMenu.zip ファイル(OCECreateYourFirstSite_data.zip ファイルにあります)をアップロードして選択します。



Home	Content and Experience	© © ¤
Assets	Components All -	Create 👻
Sites Recommendations	Developer > Components All -	Create Local Component Register Remote Component
	Name ↑ Туре	Create Content Layout
COLLABORATION		Create Section Layout
Documents		Create Content Field Editor
Conversations		Create Content Form
R Integrations		
f ² Content		
<pre></pre>		



ate -
6.050
= •
_
1

次のステップ: Minimal-NavMenu カスタム・コンポーネントの公開

Minimal-NavMenu カスタム・コンポーネントの公開

今度は、インポートした Minimal-NavMenu コンポーネントを公開する必要があります。

「コンポーネント」ページで、Minimal-NavMenu コンポーネントを選択して、メニュー・バー または右クリック・メニューの「**公開」**をクリックします。



Home	Content and Experience		© © ¤
Assets	Components All -		Create 👻
Sites	Durlans & Commun	11 - 11 - 10 - 10 - 10 - 10 - 10 - 10 -	
Recommendations	Open Copy Publish Export Members	More • All •	50 per Page • Name • 🗮 •
Developer			
Analytics	Name ']'	туре	Status Created L
COLLABORATION	Minimal-NavMenu	Local Component	Just now
Documents			
Conversations			
ADMINISTRATION			
😧 System			
R Integrations			
쥼 ^T Content			
•			

「コンポーネントの公開」ダイアログで、**「確認して続行」**を選択して「**OK」**をクリックします。

4	Publishing the component will make component changes public. All sites using this component will be affected.					
	Your use of Content and Experience Service is subject to the terms and conditions of your agreement(s) with Oracle.					
	Confirm to proceed					

Minimal-NavMenu コンポーネントを公開すると、コンポーネントが公開されたことを 示す通知が、ページ上部に表示されます。

次のステップ: サンプル・イメージの追加

サンプル・イメージの追加

今度は、(作成する Web サイトで使用する)サンプル・イメージを「ドキュメント」に 追加します。

Minimal-Images.zip ファイル(OCECreateYourFirstSite_data.zip ファイルにありま す)のコンテンツを、ローカル・コンピュータの「Minimal-Images」というフォルダに 抽出します。

左側のナビゲーション・メニューで「**ドキュメント」**をクリックし、**「作成」**をクリックして、**「Minimal-Images」**というフォルダを作成します。

Home	Content and Experience	Se	arch		0 1 0	9
	Documents All -				Upload Create	-
Sites						200
Q Recommendations	Select All				Name 👻 🗎	
Developer	Name 个	Version L	.ast Updated 🎗	Updated By	Size	
Analytics	Minimal-Images	t	lust now	You		
COLLABORATION	Minimal-NavMenu.zip	v1 2	? minutes ago	You	5 KB	
Documents						
Conversations						
ADMINISTRATION						
😧 System						
R Integrations						
쥼 ⁷ Content						
•						

ローカル・コンピュータの「Minimal-Images」フォルダのすべてのイメージを、Oracle Content Management に新たに作成した「Minimal-Images」フォルダにアップロードします。

🛆 Home			Current Folder	Search	© © ¤ ©
Assets	Minimal-Images		Share Link	Members Upload	Create - ··· 🔲
圓 Sites	Documents > Minimal-Images		1.11 11 11		
Recommendations	Select All				Name 🔹 🗮 🔹
🔂 Developer	Name 1	Version	Last Updated 11	Updated By	Size
Analytics	Banner1.jpg	v1	Just now	You	335 KB
COLLABORATION	Banner2 ing	v1	lust now	You	311 KB
Documents	C C Connecting		505(1101)	100	51110
Conversations	Logo.png	v1	Just now	You	5 KB
ADMINISTRATION	Powered_by_OCE.png	v1	Just now	You	8 KB
🚱 System					
전 Integrations					
ත් Content					
•					


ノート: テンプレートや Web サイトに使用されているイメージは、Shutterstock 社の イメージです。これらのイメージを再利用する場合は、Shutterstock 社から のライセンスが必要です。これらのイメージは、お客様独自のイメージに置 き換えることもできます。

次のステップ: Web サイトの設定

ステップ 2: Web サイトの設定

Web サイトの作成に必要なものがすべて揃ったので、Web サイトの設定を開始できます。

- 1. Web サイトの作成
- 2. Web サイトの編集

Web サイトの作成

左側のナビゲーション・メニューで、「サイト」、「作成」の順にクリックします。「Blank-Template」を選択して、「次」をクリックします。

「Blank-Template」が表示されない場合は、サービス管理者に連絡してください。サー ビス管理者が、すぐに使用可能なテンプレートを使用できるようにする必要がありま す。一般的に、サービス管理者は、サービスがプロビジョニングされると、Oracle Content Management に用意されているすぐに使用可能なテンプレートをインストー ルします。

必ず、標準の Web サイトの作成を選択し、「次」をクリックしてください。Web サイトの名前(Minimal など)を入力します。「終了」をクリックします。

Create Site					Cancel	Next
(- O Choose Template	2 Configure Site	Add Details	\rightarrow		
Your site will have the following configuration Access Security Everyone Anyone without signing in						
Type Standard	*Required field	ŀ	B	lank		



新しい Web サイトが「サイト」ページに作成され、リストされます。



次のステップ: Web サイトの編集

Web サイトの編集

サイト・ビルダーで、新たに作成した Web サイトを選択し、メニュー・バーまたは右クリック・メニューの「開く」を選択して開きます。サイト・ビルダーで、スイッチ C を「編集」 モードに設定します。更新の名前を入力して、「OK」をクリックします。

編集モードでは、Web サイトに3つのスロットが表示されますが、これはページ上で使用可能な領域です(ページ・レイアウトによって異なります)。ページ上にある+のそれぞれにカー ソルを合せると、ヘッダー、本文、フッターなどのスロットが表示されます。





会社のロゴやナビゲーション・メニューなどの表示には、ヘッダー・スロットを使用 するのが一般的です。本文スロットはページのメイン・コンテンツに使用し、フッタ ー・スロットはコピーライト情報やソーシャル・メディアのリンク、その他の情報に 使用します。

まず、ホーム・ページを作成します。完成すると、次のようなホーム・ページになり ます:



すぐに使用可能なコンポーネントを使用して、ヘッダー・スロットを作成します:



- 1. 左側のサイドバーで、 をクリックし、「シード」をクリックして、Oracle Content Management に用意されているすぐに使用可能なコンポーネントのリストを表示します。
- 2. 左側のサイドバーで、「コンポーネント・グループ」というすぐに使用可能なコンポーネ ントを探します。それをヘッダー・スロットにドラッグ・アンド・ドロップします。

0	Content and Experience	G		0
=	Minimal Update1 🕶 🗠 🗠	Fit to Window 👻 📳 🔗 View 🌑 Edit 🗵 Commit Save 🗏 🤇		D
Ð	< Seeded -	Component Group 열		^
	Find a component	+		
4	Process Start Form			
Ą	Task List	The Body slot is used for the main content of the page.		
٥	Task Details	+		
	Others			
	🔗 Мар	The Footer slot is used for content that goes at the bottom of the page like additional information, links to other resources, a footer logo, etc.		
	Article	+		
	Headline	l	1	
_	Image and Text			
Ð	Component Group			~

3. コンポーネント・グループのメニュー・アイコン をクリックしてから、「設定」をクリ ックします。設定で、「カラー」ドロップダウン・リスト(設定リストの一番下にありま す)をクリックしてから、「その他」をクリックします。#3333333 と入力し、「OK」をクリ ックします。



コンポーネント・グループにイメージ・コンポーネントをドラッグ・アンド・ドロップします。



0	Content and Experience	0 0
=	Minimal Update1 🕶 🖒 🛇	Fit to Window 🔻 👘 🔗 View 🌑 Edit 🗵 Commit Save 🗏 🤤 🛄
Ð	< Seeded -	Image T
		<u>ē</u>
4	Media	Image .
Ą	image	The Rody slot is used for the main content of the page
۵	Gallery	The body side is used for the main content of the page.
	Gallery Grid	+
	YouTube	
	Video	The Footer slot is used for content that goes at the bottom of the page like additional information, links to other resources, a footer logo, etc.
	Documents	
	Document	
Đ	Folder List	

5. イメージ・コンポーネントのメニュー・アイコン をクリックしてから、「設定」 をクリックします。「一般」タブの設定をすべて入力します。

プロパティ	值
選択	「Minimal-Images」フォルダの Logo.png
位置合わせ	左
幅	「幅の設定」 の選択を解除
L	1.2vw
下	30px
左	6vw
右	0





Image Settir	ngs		×
General	St	yle	Link
Mir	Logo.p 191 x 60 Sele	ect	
Title			
Alternate Text			
Caption			
Alignment			
	≘	≣	≡
Width	-	_	_
Set width			
0			
Spacing ⑦		Pottom	
1.2vw		30px	
Left			

6. このロゴ・イメージをホーム・ページにリンクします。「**リンク」**タブで、イメージ・コ ンポーネントの設定をすべて入力します。

プロパティ	值
リンク・タイプの選択	サイト・ページ
ページ	HOME
ターゲット	同じウィンドウで開く

- 7. 左側のサイドバーで、 をクリックし、「カスタム」をクリックしてカスタム・コンポー ネントのリストを表示します。
- Minimal-NavMenu カスタム・コンポーネントを使用して、ホーム・ページにナビゲーション・メニューを追加します。イメージ・コンポーネントの右側にあるコンポーネント・グループに、Minimal-NavMenu コンポーネントをドラッグ・アンド・ドロップします。 Minimal-NavMenu コンポーネントのタイトルをクリックして、その親が先ほど追加したコンポーネント・グループであることを確認します。これは、コンポーネントが Web サイト構造のどこにあるかを確認する便利な方法です。



0	Content	t and Exp	erience																		0	0
=	Minimal	Update	1 -	5	ע					Fit to V	Nindow	•	0 0	Vi	iew 🕻	Edit	Z	Commit	Save		9	
						_		_			Mini	mal-N	avMenu	T								
Ð		0	CE	Mini	mal						но	OME										
					inicii																	
4							The	Body sl	ot is u	sed for	the ma	in coi	ntent of	the	pag	e.						
Ą																						
۲																						
	T	The Foot	er slot	is used	for conter	nt that g	joes at	the bott	tom of	the pa	ige like	additi	ional in	form	natio	n, links	to oth	er resourc	es, a foot	er log	jo, etc.	
											÷											
										10 (m 10 (m)m 10 m 1)
đ																						
	- 101																					

「一般」タブの設定をすべて入力します。

プロパティ	值
位置合わせ	右
上	1.2vw
不	0
左	0
右	6vw

 これでヘッダーの準備ができました。このコンポーネント・グループをカスタム・ コンポーネント・グループとして保存し、後から別の Web サイトのページで使用 できるようにします。コンポーネント・グループのタイトルをクリックし、そのメ

ニュー・アイコン こ をクリックしてから「保存」をクリックします。「コンポーネント・グループの保存」ダイアログの「名前」フィールドに Minimal-Header と入力して、「保存」をクリックします。

Component Group	
	Save
OCE	Copy Style (Component Group)
	Hide
	Delete
	Settings



💡 ヒント:

コンポーネント・グループをクリックしたときに、コンポーネント・グループ のタイトルではなく、イメージや Minimal-NavMenu コンポーネントのタイトル が表示される場合は、イメージまたは Minimal-NavMenu コンポーネントのタイ トルを再度クリックすると、コンポーネント・グループのタイトルが表示され ます。これでコンポーネント・グループのタイトルをクリックできるようにな るので、上のステップを実行します。

次のイメージは、ヘッダー・スロットのイメージ・コンポーネントの親構造を示しています:



10. サイト・ビルダーの右上にある「**保存」**をクリックして、変更を保存します。ヘッダー・ スロットは、次のイメージのようになっています:



本文スロットに進みます:

- 1. 左側のサイドバーで、 をクリックし、「シード」をクリックします。
- 左側のサイドバーから、コンポーネント・グループを本文スロットにドラッグ・アンド・ ドロップします。このコンポーネント・グループとそのコンポーネント(後からコンポー ネント・グループに追加します)を使用して、バナーを作成します。





コンポーネント・グループに2列のセクション・レイアウトをドラッグ・アンド・ドロップします。



0	Content	and Experience												0	0
=	Minimal	Update1 🔹 🖒	S			Fit to Window	•	4	View 🊺 Ed	t 🗵	Commit	Save		0	
Ð	Sec	ction Layouts	0	OCE	Minima	1							Ξ		
	0:0	Horizontal		Section Layout	Two Columns 🗉										
Þ	O	Slider													
₽		Tabs													
0		Three Columns		The Footer sl	ot is used for co	ntent that goes	at the l	bottor	m of the page	ike ado	litional info	ormation, I	links to	o othe	er
		Two Columns				res	sources,	, a foc	oter logo, etc.						
		Vertical													
				[201001	1
đ															

5. 「一般」タブで、セクション・レイアウトの設定をすべて入力します。「カスタム設 定」をクリックして、次の設定を指定します。

プロパティ	值
最初の列の幅 (%)	43
2 番目の列の幅 (%)	57
レスポンシブ・ブレークポ	1,023
イント(ピクセル)	
レスポンシブ動作	最初の列を非表示

「背景」タブの設定をすべて入力します:

プロパティ	值
イメージ	「Minimal-Images」フォルダの Banner1.jpg
位置	中央
スケール	ストレッチ

- 6. 左側のサイドバーで、 **し**をクリックし、シードされたコンポーネントのリストを 表示します。
- シードされたコンポーネントのリストから、2列のレイアウトの2番目の列に、タイトル・コンポーネントをドラッグ・アンド・ドロップします。







8. タイトル・コンポーネントをクリックして、「WELCOME TO THE REVOLUTION」と入 力します。そのテキストを選択して、テキスト・エディタでフォントの色を「白」に設定 します。「一般」タブで、タイトル・コンポーネントの設定をすべて入力します。

プロパティ	值
上 上	6vw
下	1.8vw
左	6vw
右	6vw



左側のサイドバーにあるシードされたコンポーネントのリストから、タイトル・コンポーネントの下にある2列レイアウトの2番目の列に、段落コンポーネントをドラッグ・アンド・ドロップします。「一般」タブの設定をすべて入力します。

プロパティ	值
上	1.8vw
不	6vw
左	6vw
右	6vw

段落コンポーネントをクリックして、次のテキストを入力します:
 "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."

そのテキストを選択して、テキスト・エディタでサイズを 24 に設定します。また、テキスト・エディタでフォントの色を「**白」**に設定します。



これでバナーの準備ができました。このコンポーネント・グループをカスタム・コンポーネント・グループとして保存し、後から別のWebサイトのページで使用できるようにします。コンポーネント・グループのメニュー・アイコンをクリックしてから、「保存」をクリックします。「コンポーネント・グループの保存」ダイアログの「名前」フィールドに Minimal-Banner と入力して、「保存」をクリックします。これで、コンポーネント・グループの名前(Minimal-Banner)が表示されます。





 左側のサイドバーから、すでに追加した Minimal-Banner コンポーネント・グループの下 にある本文スロットに、別のコンポーネント・グループをドラッグ・アンド・ドロップし ます。



- **13.** そのコンポーネント・グループにタイトル・コンポーネントをドラッグ・アンド・ドロップします。
- 14. タイトル・コンポーネントをクリックして、「Welcome to OCE Minimal」と入力します。
- 15.「**一般」**タブで、タイトル・コンポーネントの設定をすべて入力します。

プロパティ	值
上	3vw
下	1.8vw
左	6vw
右	6vw





 左側のサイドバーから、タイトル・コンポーネントの下のコンポーネント・グルー プに、段落コンポーネントをドラッグ・アンド・ドロップします。段落コンポーネ ントをクリックして、次のテキストを入力します:
 "Oracle Content Management は、オムニチャネルのコンテンツ管理を促進し、エ クスペリエンスの配信を加速するクラウドベースのコンテンツ・ハブです。It offers powerful collaboration and workflow management capabilities to streamline

"Oracle Content Management は、Web サイトを作成するためのシンプルで使いや すいツールを提供します。Oracle Content Management が提供する強力な機能を 利用して、Web サイトをすばやく作成できます。"

17.「一般」タブで、段落コンポーネントの設定をすべて入力します。

the creation and delivery of content."

プロパティ	值
上	20px
下	50px
左	6vw
右	6vw





18. 本文スロットが完成しました。コンポーネント・グループをカスタム・コンポーネント・ グループとして保存し、後から別の Web サイトのページで使用できるようにします。コ

ンポーネント・グループのメニュー・アイコン ラします。「コンポーネント・グループの保存」ダイアログの「名前」フィールドに Minimal-Body と入力して、「保存」をクリックします。

19. サイト・ビルダーの右上にある「**保存」**をクリックして、変更を保存します。本文スロットは、次のイメージのようになっています:



フッター・スロットを完成させます。



 左側のサイドバーから、コンポーネント・グループをフッター・スロットにドラッ グ・アンド・ドロップします。コンポーネント・グループの設定で、「色」フィー ルドを#333333 に設定します。



 コンポーネント・グループにイメージ・コンポーネントをドラッグ・アンド・ドロ ップし、「一般」タブで設定をすべて入力します。

値
「Minimal-Images」フォルダの Powered_by_OCE.png
左
「幅の設定」 の選択を解除
0.9vw
0.9vw
6vw
0

 左側のサイドバーから、イメージ・コンポーネントの右側にあるコンポーネント・ グループに、ソーシャル・バー・コンポーネントをドラッグ・アンド・ドロップし ます。





「一般」タブで、ソーシャル・バー・コンポーネントの設定をすべて入力します。

プロパティ	值
上	1.8vw
下	1.8vw
左	0.3vw
右	6vw

「一般」タブで「アイコン」をクリックし、アイコン名をクリックして設定を完了します。

プロパティ	值
URL	https://www.facebook.com/Oracle/ (Facebook)
	https://www.linkedin.com/company/oracle/ (LinkedIn)
	https://twitter.com/Oracle (Twitter)
	https://www.youtube.com/oracle/ (YouTube)
ターゲット	新規ウィンドウで開く

 これでフッターの準備ができました。このコンポーネント・グループをカスタム・コンポ ーネント・グループとして保存し、後から別の Web サイトのページで使用できるように

します。コンポーネント・グループのメニュー・アイコン **存」**をクリックします。「コンポーネント・グループの保存」ダイアログの**「名前」**フィ ールドに Minimal-Footer と入力して、**「保存」**をクリックします。コンポーネント・グル ープは、次のようになっています:

Minimal-Footer	Ξ	
(ja)	Powered by Oracle Content and Experience	📑 in 💟 📴 🗖

5. サイト・ビルダーの右上にある「保存」をクリックして、変更を保存します。



サイト・ビルダーの右上にある をクリックして、Web サイトの最初のページをプレビューします。Web サイトはまだ公開されていないため、今は第三者が見ることはできません。

「HOME」ページの作成が終わりました。「CONTACT US」ページを作成します。完成 すると、次のような問合せ先ページになります:

oce Minimal	HOME CONTACT US
	Want to learn more? Find more learning material on the Headless CMS page. OCE FOR DEVELOPERS
Connect with us:	
Visit the Oracle Cloud Connect Forum to post your queries. Oracle Content and Experience samples are available at: https://www.oracle	.com/middleware/technologies/content-experience-downloads.html
Locations	
Regional Office1: Building Number 1, City1 , Province1 , Country1	
Regional Office2 : Building Number 2, City2 , Province2 , Country2	
Powered by Oracle Content and Experience	f in 🗹 🖬 🗆

様々なスロットにコンポーネントを追加します:

- **1.** 左側のサイドバーで、

 をクリックし、「ページの追加」をクリックします。
- 2. 「ページ名」フィールドに「CONTACT US」と入力し、「閉じる」をクリックしま す。Web サイトに新しいページが追加されました。
- 3. 左側のサイドバーで、**連**をクリックし、「カスタム」をクリックします。
- 4. ヘッダー・スロットに(先ほど作成して保存した)Minimal-Header コンポーネントを ドラッグ・アンド・ドロップします。



0	Content	t and	Expe	rienc	e																																					0)	0
=	Minimal	Up	date1	•	0	Cu												3	Fit	to	Win	dow	•				4		Vie	ew			dit	Z		Com	nit		Save	e	Ę	Q	>	
_	Slot	t Hea	nder	Т							_									_							_	-				-					_	Pars					_	
Ð			Т	he H	eader	slot is	used f	or cor	nten	nt ti	that	nat g	go	oe	es a	at 1	the	to	pc	of t	the	pa	je l	ike	th	e c	on	np	any	y lo	ogo	, th	e n	ienu	ac	ditio	nal	lin	cs, e	etc.				
ħ																																												
4																				_																								
Ą									1	The	he B	Bo	ody	ly s	slo	ot i	is u	sec	d fo	or	the	m	in	cor	nte	nt	of	the	e p	ag	e.													
0																					1																							
	Tł	he Fo	oter	slot	is use	d for a	ontent	that	goe	es al	at th	the	ne t	bo	otte	tom	n of	f th	e p	bag	ge l	like	ade	diti	on	al i	info	ori	ma	tio	n, I	ink	s to	oth	er r	esou	rce	s, a	foo	ter l	logo,	etc.		
Ð																																												1

Minimal-NavMenu	E		
		HOME	CONTACT US

Minimal-NavMenu コンポーネントに、作成したばかりの新しい「CONTACT US」ページ が自動的に表示されます。

0	Content and Experience Ø (2)
=	tinimal Update1 🕶 🖉 🖓 Fit to Window 🕶 📳 🔗 View 🌑 Edit 🗵 Commit 🛛 Save 🗏 🖗 🛄
Ð	OCE Minimal HOME CONTACT US
	Slot Body 1
ţ,	The Body slot is used for the main content of the page.
Ą	+
۲	
	The Footer slot is used for content that goes at the bottom of the page like additional information, links to other resources, a footer logo, etc.
	+
	·
đ	

5. 左側のサイドバーから、Minimal-Banner コンポーネントを本文スロットにドラッグ・アンド・ドロップします。





 「HOME」ページのバナーとは見た目を変え、「CONTACT US」ページにふさわしいものになるようバナーを変更します。コンポーネント・グループ内で、2列のセクション・レイアウトの設定を変更します: 「背景」タブの「イメージ」フィールドで「イメージの選択」をクリックし、「ドキ

「育気」 ダブの「イメーシ」 フィールトで「イメーシの選択」をグリックし、「トキュメント」に作成してある「Minimal-Images」フォルダから Banner2.jpg を選択します。

Select File					Cancel	ок
Minimal-I	Images				Upload	Create
Documents > Minir	mal-Images					5 M 2010 M
					Name 🝷	≣ •
Name 个		Version	Last Updated 1	Updated By	Size	
в	Banner1.jpg	v1	Monday at 3:00 PM	You	335 KB	
В	3anner2.jpg	v1	Monday at 3:00 PM	You	311 KB	
D Mir L	_ogo.png	v1	Monday at 3:00 PM	You	5 KB	
P	Powered_by_OCE.png	v1	Monday at 3:00 PM	You	8 KB	

- 7. 2列のセクション・レイアウト内で、タイトル・コンポーネントと段落コンポーネ ントのテキストを変更します。
 - タイトル・コンポーネントに「Want to learn more?」と入力します。
 - 段落コンポーネントに「Find more learning material on the Headless CMS page.」と入力します。
- 8. 左側のサイドバーで、 をクリックし、 **シード」**をクリックします。



 「CONTACT US」ページのイメージには、バナーの一部として「OCE FOR DEVELOPERS」というボタンがあるので、バナーにこのボタンを追加します。最小バナ ー・コンポーネント・グループ内で、ボタン・コンポーネントを(最近追加した段落コン ポーネントの下にある)2列セクション・レイアウトの2列目にドラッグ・アンド・ドロ ップします。「一般」タブで、ボタン・コンポーネントの設定をすべて入力します。

プロパティ	值
ラベル	OCE FOR DEVELOPERS
上	0.3vw
下	3vw
左	6vw
右	0.3vw

「スタイル」タブで、「カスタマイズ」を選択して設定を完了します。

プロパティ	值
背景色	#c0d600
フォント	 サイズとして 24 を入力します。 色#58595b を入力します。
枠線	なし
ホバー時の色	 「背景」を#e1fa00 に設定します。 「フォント」を#58595b に設定します。 「枠線」を#2222dd に設定します。
コーナー	0

「リンク」タブの設定をすべて入力します。

プロパティ	值
リンク・タイプの選択	Web ページ
URL	http://www.oracle.com/pls/topic/lookup?ctx=cloud&id=content-cloud-headless
ターゲット	新規ウィンドウで開く

Button Settings ×			
General	Style	Link	
Select Link Type:			
Web Page		-	
Link to a specified URL.			
URL			
http://www.oracle.com/pls/topic/look			
Target			
Open in New	Window	-	





- 10. 左側のサイドバーで、 をクリックし、「カスタム」をクリックします。
- **11.** Minimal-Banner コンポーネントの下の本文スロットに、Minimal-Body コンポーネ ントをドラッグ・アンド・ドロップします。
- **12.** Minimal-Body コンポーネント内で、タイトル・コンポーネントと段落コンポーネントのテキストを変更します。
 - a. タイトル・コンポーネントに「Connect with us:」と入力します。
 - b. 段落コンポーネントに、電子メール・アドレスやサポート関連のその他のリン クなど、詳細を入力します:
 "Visit the Oracle Cloud Connect Forum to post your queries."

```
"Oracle Content Management のサンプルは https://www.oracle.com/
middleware/technologies/content-experience-downloads.html で入手できます"
```

- 左側のサイドバーから、先ほど追加した Minimal-Body コンポーネントの下にある 本文スロットに、別の Minimal-Body コンポーネントをドラッグ・アンド・ドロッ プします。
- 14. コンポーネント・グループの設定の「背景」タブで、「色」フィールドを#696969 に設定します。
- **15.** Minimal-Body コンポーネント内で、タイトル・コンポーネントと段落コンポーネ ントを変更します。
 - a. タイトル・コンポーネントに「Locations」と入力します。そのテキストを選択 して、テキスト・エディタでフォントの色を「白」に設定します。
 - **b.** 段落コンポーネントに次のテキストを入力します。 "Regional Office1:

Building Number 1,

City1, Province1, Country1

Regional Office2:

Building Number 2,



City2, Province2, Country2"

そのテキストを選択して、テキスト・エディタでフォントの色を「白」に設定します。

- **16.**「CONTACT US」ページにフッターを追加します。フッター・スロットに Minimal-Footer コンポーネントをドラッグ・アンド・ドロップします。
- 17. サイト・ビルダーの右上にある「保存」をクリックして、変更を保存します。

サイト・ビルダーの右上にある をクリックして Web サイトをプレビューし、問題がないか 確認します。「CONTACT US」ページのロゴ・イメージをクリックしたら、ホーム・ページ に戻れることを確かめます。メニューをテストして、Web サイトのページ間の移動が正常に 行われることを確認します。

Web サイトを公開する準備が整いました。

次のステップ: Web サイトの公開

ステップ 3: Web サイトの公開

Web サイトを作成できたので、ユーザーがオンラインで参照できるように公開します。

1. 問題がなければ、サイト・ビルダーの右上にある「コミット」をクリックして、ベースの Web サイトに変更をコミットします。



2. 「更新のコミット」ダイアログで、「**コミット」**をクリックします。





- 3. 変更がコミットされたら、Web サイトを公開できます。
- 4. サイト・ビルダーを閉じます。
- 5. Oracle Content Management Web インタフェースの左側にあるナビゲーション・ メニューで「サイト」をクリックし、Web サイトを選択します。
- 6. メニュー・バーまたは右クリック・メニューから「**公開」**を選択します。

🗋 Home		© © ¤
Assets	Sites all -	Translation Jobs Create
E Sites		
Q Recommendations	Open View Copy Rename Publish Members More -	50 per Page • Name •
Developer		
ht Analytics	Blank	
COLLABORATION		
Documents	- 🛛 🖸 -	
C Conversations	Minimal Created Monday at 330-PM	
ADMINISTRATION	Standard	
😧 System		
杞 Integrations		
🗗 Content		
<		

- **7.** Web サイトを公開すると、その Web サイトが公開されたことを示す通知が、ページ上部に表示されます。
- 「サイト」ページで Web サイトを再度選択して、メニュー・バーまたは右クリック・メニューから「オンラインにする」を選択します。「オンラインにする」ダイアログで、「確認して続行」を選択して「オンラインにする」をクリックします。





If you make this site online, it will be publicly accessible. Do you want to proceed? Your use of Content and Experience Service is subject to the terms and conditions of your agreement(s) with Oracle.	Bring Online		×
Your use of Content and Experience Service is subject to the terms and conditions of your agreement(s) with Oracle.	If you make this site online, it will be publi proceed?	icly accessible. Do y	ou want to
Confirm to proceed	Your use of Content and Experience Service is subject t agreement(s) with Oracle.	o the terms and condition	s of your
	Confirm to proceed		
		Cancel	Bring Online

9. これで終了です。Web サイトがオンラインになり、第三者が表示できます。

公開 Web サイトを表示するには、メニュー・バーまたは右クリック・メニューから「表示」を選択します。







Web サイトは、組織のニーズに合せてカスタマイズできます。ここでは、開始に役立 つリンクを紹介します:

- スタイルおよびフォーマットの使用
- 編集のヒントとテクニック
- サイトおよびサイト設定の管理
- サイト・ファイルのアップロード
- ページの追加
- ページの移動
- ページ・コンテンツの配置
- ページ設定の変更
- ページ・レイアウトの変更
- 表の操作
- 検索エンジンのプロパティの設定





サイトの作成および編集

このパートでは、テンプレートとテーマを使用してサイトの作成を開始し、コンポーネント とレイアウトを使用してサイトを編集して、コンテンツを整理および追加する方法について 詳細に説明します。次の章が含まれます。

- サイトの作成
- サイトの編集
- サイトでのテンプレートおよびテーマの使用
- カスタム・コンポーネントおよびレイアウトの管理
- サイト・ページの操作
- ページ・コンテンツの配置
- 組込みコンポーネントの使用



4 サイトの作成

適切な権限があれば、誰でも Oracle Content Management で Web サイトを構築できます。専 用のツール、コードまたはソフトウェアを使用する必要はありません。ユーザー・インタフ ェースがグラフィカルで直観的であり、簡単に使用できます。

サイトを作成する場合、テンプレートから開始します。テンプレートには、ユーザーがサイトの使用を開始するにあたり必要なもの、たとえば、サイト・コード・フレームワーク、サンプル・ページとコンテンツを含むデフォルト・サイト、スタイルが設定されたテーマ、イメージなどのリソース、カスタム・コンポーネントなどがすべて含まれます。サイト作成プロセスの理解を参照してください。

Oracle Content Management のサイト・ガバナンスにより、ビジネス・ユーザーに対するエクスペリエンスの配布が簡略化および迅速化され、IT 部門は一元化された場所からエクスペリエンスを容易に制御およびトラッキングしつつ、エクスペリエンスのライフサイクル全体を完全に管理し、企業が必要とする個々の新しいエクスペリエンスの作成および維持のためのコストを削減できます。ガバナンスは、Oracle Content Management のコアに組み込まれており、管理者が有効化するのみで済みます。Administering Oracle Content Management のサイトおよびアセット設定の構成を参照してください。

- サイト作成プロセスの理解
- サイトの作成
- サイトのコピー
- サイト・リクエストの管理
- サイト・リクエスト詳細の変更
- サイト・リクエスト・ポリシーの表示

サイトの作成

サイトを作成するには、テンプレートを選択し、サイトに名前を付けて、コンテンツを追加 します。ガバナンスが有効な場合、コンテンツを追加する前にサイト・リクエストを送信し て、サイト・リクエストの承認後にコンテンツを追加します。

サイトを作成する前に、管理者がサイトの作成を有効にし、テンプレートを使用可能にする 必要があります。「サイト」ページに「**作成」**オプションが表示されない場合、または使用可 能なテンプレートがない場合は、管理者に連絡してください。「サイトのスタート・ガイド」 および「サイト・ガバナンスの理解」を参照してください。

アセットおよび多言語サイトの使用が可能なエンタープライズ・サイトを作成すると、その サイトはリポジトリ、ローカリゼーション・ポリシーおよび(場合により)デフォルト言語に関 連付けられます。リポジトリおよびローカリゼーション・ポリシーがない場合、自分で作成 するか、コンテンツ管理者に作成を依頼します。



💉 ノート:

Oracle Content Management Starter Edition では、翻訳は使用できず、ガバ ナンスも有効にできず、サイトも1つしか作成できません。完全な機能セッ トおよび無制限のサイトが必要な場合は、**Oracle Content Management Premium Edition** にアップグレードします。

- 1. 「サイト」をクリックします。
- 2. 「作成」をクリックします。
- 3. 「テンプレートの選択」ページで、サイトのテンプレートを選択します。
 - ガバナンスが有効な場合、テンプレートにより、作成するサイトが標準のサイト(アセット・リポジトリまたはローカリゼーション・ポリシーを使用しない)なのか、エンタープライズ・サイト(アセット・リポジトリおよびローカリゼーション・ポリシーを使用)なのかが判定されます。
- 4.「サイトの構成」ページに、サイトを作成する前に必要な承認、サイトに必要な最低レベルのセキュリティ、およびサイト・ページのイメージが表示されます。承認が特定のユーザーに制限されている場合は、「承認者の表示」をクリックして、誰がサイト・リクエストを承認できるかを確認します。
 - ガバナンスが有効でない場合、作成するサイトのタイプに「標準」または「エンタープライズ」を選択します。
 - 標準サイトを作成している場合は、「次」をクリックして次のページに移動します。
 - エンタープライズ・サイトを作成している場合は、次のステップを実行します:
 - a. サイトのコンテンツを管理するために使用するデフォルト・リポジトリを選択 します。お客様がリポジトリ管理者で、使用可能なリポジトリがない場合、ま たは新しいリポジトリを使用する必要がある場合は、「新規リポジトリの作成」 を選択し、ステップに従ってリポジトリを作成します。サイトでの複数のリポ ジトリの使用の詳細は、Oracle Content Management でのアセットの管理の複 数のリポジトリへのサイト・アクセスの付与を参照してください。
 - b. サイトのローカリゼーション・ポリシーを選択します。お客様がリポジトリ管理者で、使用可能なポリシーがない場合、または新しいポリシーを使用する必要がある場合は、「新規ローカリゼーション・ポリシーを作成します」を選択し、ステップに従ってポリシーを作成します。
 - c. テンプレートのローカリゼーション・ポリシーに、設定されたデフォルト言語 がない場合、ここで選択します。ローカリゼーション・ポリシーで必要とされ る言語のみが表示されます。
 - d. 次のページに移動するには、「次」をクリックします。
- 5. 「詳細の追加」ページで、次の情報を入力します。
 - サイトの名前を入力します。名前はサイト URL に使用されます。文字、数字、 アンダースコア()およびハイフン(-)を使用できます。URL は大文字と小文字 が区別されます。スペースを入力した場合、ハイフンに自動的に置換されま す。既存のサイトの名前を大/小文字を変えて使用しないでください。たとえ



ば、ABC という名前のサイトが存在する場合、Abc という名前の別のサイトを作成しないでください。

🖉 ノート:

テンプレート、テーマ、コンポーネント、サイトまたはサイト・ページにつ いて、authsite、content、pages、scstemplate_*、_comps、_components、 _compsdelivery、_idcservice、_sitescloud、_sitesclouddelivery、_themes、 _themesdelivery といった名前を使用しないでください。documents およ び sites は、サイト・ページには使用できますが、テンプレート、テーマ、 コンポーネントまたはサイトには使用しないでください。

- エンタープライズ・サイトを作成している場合、テンプレート・ポリシーでは簡易 URL 値のデフォルトのサイト接頭辞の編集が許可されることがあります。この接頭 辞は、コンテンツ・アイテム・スラグ値(ページまたはアセットに固有の URL の一 部)に追加されます。
- オプションで、サイトの説明を入力します。
- オプションで、このサイト・リクエストの理由を入力します。これは、サイト管理者 がリクエストを承認するかどうかを判断するのに役立ちます。
- 6. 「終了」をクリックします。

サイト・リクエストに承認が必要な場合は、それを承認する必要があるユーザーに電子メー ルが送信されます。承認されると、承認されたことと、サイトが作成されることを通知する 電子メールを受信します。リクエストが拒否された場合は、拒否されたことを通知する電子 メールを受信します。フィルタ・メニューの「**リクエスト」**を選択すると、保留中のリクエ ストを「サイト」ページに表示できます。リクエストが拒否された場合、拒否された理由に 関するメッセージが表示されるので、問題を修正してリクエストを再度送信できます。

サイト・リクエストが自動的に承認されるように設定されている場合、サイトは自動的に作 成されます。進行状況バーに、新規サイト名と作成ステータスが表示されます。サイトが作 成されると、サイトのリストに名前が表示され、サイトのステータスがオフラインになりま す。

エンタープライズ・サイトを作成した場合、選択したリポジトリに対応するサイト・コレク ションが作成されます。サイトをユーザーと共有すると、このユーザーには、関連するコレ クションに対する同じ権限が付与されます。

サイトが作成されたら、サイトの共有、セキュリティの変更、コンテンツの追加と編集、サ イトの公開、およびサイトのオンライン化が行えます。サイトおよびサイト設定の管理およ びサイトの編集を参照してください。



サイトのコピー

サイトをコピーして、迅速に開始できます。テーマ、すべての未処理の更新内容、ペ ージ、ページ・コンテンツ、レコメンデーション、その他すべてのアセット(イメージ など)およびポリシーを含む元のサイトのすべてが、指定した新しい名前の新規サイト にコピーされます。

N-F:

複数のリポジトリからのコンテンツを使用するサイトをコピーしている場合、OCE ツールキットを使用して、これを行う必要があります。「OCE ツー ルキットでの開発」および「cec コマンドライン・ユーティリティの使用」 を参照してください

- 1. 「サイト」をクリックします。
- コピーするサイトを選択して、右クリック・メニューの「コピー」、またはアクション・バーの を選択します。
- 3. 「サイトの構成」ページに、サイトを作成する前に必要な承認、サイトに必要な最 低レベルのセキュリティ、およびサイト・ページのイメージが表示されます。次の ステップを実行します。
 - a. 「コピー」ドロップダウン・リストで、ベース・サイトのみをコピーするか、 更新内容を含めるかを選択します。更新内容を含める場合、コピーしたサイト には、元のサイトからの未処理の更新内容がすべて含まれます。標準サイトを コピーしている場合は、矢印をクリックして次のページに移動し、ステップ4 にスキップします。
 - b. エンタープライズ・サイトをコピーしている場合は、デフォルト・アセット・ リポジトリ、ローカリゼーション・ポリシー、必要な言語、およびデフォルト 言語が元のサイトに基づいて設定されます。オプションで、別のリポジトリを 選択して、サイトのコンテンツを管理できます。
 - c. 次のページに移動するには、「次」をクリックします。
- 4. 「詳細の追加」ページで、次の情報を入力します。
 - サイトの名前を入力します。名前はサイト URL に使用されます。文字、数字、 アンダースコア()およびハイフン(-)を使用できます。URL は大文字と小文字 が区別されます。スペースを入力した場合、ハイフンに自動的に置換されま す。既存のサイトの名前を大/小文字を変えて使用しないでください。たとえ ば、ABC という名前のサイトが存在する場合、Abc という名前の別のサイトを 作成しないでください。



💉 ノート:

テンプレート、テーマ、コンポーネント、サイトまたはサイト・ページにつ いて、authsite、content、pages、scstemplate_*、_comps、_components、 _compsdelivery、_idcservice、_sitescloud、_sitesclouddelivery、_themes、 _themesdelivery といった名前を使用しないでください。documents およ び sites は、サイト・ページには使用できますが、テンプレート、テーマ、 コンポーネントまたはサイトには使用しないでください。

- エンタープライズ・サイトをコピーしている場合、オプションで、簡易 URL 値のデ フォルトのサイト接頭辞を編集できます。この接頭辞は、コンテンツ・アイテム・ス ラグ値(ページまたはアセットに固有の URL の一部)に追加されます。
- サイトの説明(オプション)を入力します。
- このサイト・リクエストの理由(オプション)を入力します。これは、サイト管理者が リクエストを承認するかどうかを判断するのに役立ちます。

5. 「終了」をクリックします。

サイト・リクエストの承認が必要な場合、サイトはサイト管理者が承認した後に作成されま す。フィルタ・メニューの「**リクエスト」**を選択すると、保留中のリクエストを「サイト」 ページに表示できます。リクエストが拒否された場合、拒否された理由に関するメッセージ が表示されるので、問題を修正してリクエストを再度送信できます。

サイト・リクエストが自動的に承認されるように設定されている場合、サイトは自動的に作 成されます。進行状況バーに、新規サイト名と作成ステータスが表示されます。サイトが作 成されると、サイトのリストに名前が表示され、サイトのステータスがオフラインになりま す。

エンタープライズ・サイトをコピーした場合、選択したリポジトリに対応するサイト・コレ クションが作成されます。サイトをユーザーと共有すると、このユーザーには、関連するコ レクションに対する同じ権限が付与されます。

サイトが作成されたら、サイトの共有、セキュリティの変更、コンテンツの追加と編集、サ イトの公開、およびサイトのオンライン化が行えます。複数のリポジトリにアクセスできる サイトを共有している場合、デフォルトのリポジトリのみが共有されます。追加のリポジト リは個別に共有する必要があります。サイトおよびサイト設定の管理およびサイトの編集を 参照してください。

サイト・リクエストの管理

サイト・ガバナンスが有効になっている場合、サイトを作成する前に承認を得ることが必要 な場合があります。保留中のサイト・リクエストは、「サイト」ページで「リクエスト」フィ ルタを使用して表示できます。

/ ノート:

Oracle Content Management Starter Edition では、ガバナンスを有効にすることはできません。完全な機能セットおよび無制限のサイトが必要な場合は、**Oracle Content Management Premium Edition** にアップグレードします。



サイト・ガバナンスの理解を参照してください。

保留中のサイト・リクエストを表示するには、「サイト」ページのフィルタ・メニュー で「リクエスト」を選択します。サイト管理者の場合、他のユーザーの承認が必要な リクエストも含めて、システム内のサイト・リクエストがすべて表示されます。承認 者の場合、自分が承認者であるサイト・リクエストがすべて表示されます。すべての ユーザーには、自分が送信したサイト・リクエストが表示されます。

リクエストのリストには、サイトのサムネイル、サイトの名前、誰がリクエストした か、いつリクエストしたか、サイトの説明、リクエストのステータス、およびサイト がログインを必要とするかどうかを示すアイコンが表示されます。

ユーザーのロールに応じて、次のアクションを実行できます。

- 必要な最小限のセキュリティ、使用されるテンプレート、サイトの理由(オプション)、サムネイルなど、リクエストの詳細をさらに表示するには、サイト名をクリックするか、リクエストを選択して、「表示」をクリックします。
- サイト管理者または承認者の場合、リクエストを選択して「承認」をクリックする ことにより、リクエストを承認できます。サイトは承認後に自動的に作成されま す。
- サイト管理者または承認者の場合、詳細を表示して「却下」をクリックすることに より、リクエストを拒否できます。リクエストを拒否する理由を入力し、「却下」 をクリックします。 リクエストのリストでリクエストを選択して「却下」をクリックする方法でもリク エストを拒否できますが、リクエストを却下する理由を追加できません。
- リクエストが失敗または却下された場合、詳細を表示し、リクエストを必要に応じて編集してから再送信できます。
- サイト作成者の場合、リクエストを選択して「削除」をクリックすることにより、 リクエストを削除できます。

🖉 ノート:

サイト・リクエスタが削除されている場合、サイト管理者がサイト・リ クエストを削除できます。

サイトが作成されたら、サイトの共有、セキュリティの変更、コンテンツの追加と編 集、サイトの公開、およびサイトのオンライン化が行えます。サイトおよびサイト設 定の管理およびサイトの編集を参照してください。

サイト・リクエスト詳細の変更

「サイト・リクエスト詳細」タブには、サイト名、説明、使用されるテンプレート、理由、所有者(リクエストを行ったユーザー)およびサイト・イメージが表示されます。

自分でリクエストを作成した場合(リクエスト所有者の場合)、サイト・リクエスト詳細 を更新できます。

サイト・リクエスト詳細を表示または変更するには:

- 1. 「サイト」をクリックし、サイト・メニューで「リクエスト」をクリックします。
- 2. 表示または編集するリクエストをクリックします。



3. 承認者の場合、リクエストを「承認」または「却下」できます。リクエスタの場合、必要 に応じて情報を編集し、リクエストを「再送信」できます。

サイト・リクエスト・ポリシーの表示

サイト・リクエスト・ポリシー・タブには、サイトのタイプ(標準またはエンタープライズ)、 公開されたサイトにアクセスできるユーザー、およびサイトの作成前に必要な承認が表示さ れます。エンタープライズ・サイトでは、サイトで使用されるアセット・リポジトリ、ロー カリゼーション・ポリシー、必須言語およびデフォルト言語も表示されます。

サイト・リクエスト・ポリシーを表示するには:

- 1. 「サイト」をクリックし、サイト・メニューで「リクエスト」をクリックします。
- 2. 表示または編集するリクエストをクリックし、「ポリシー」タブをクリックします。
- 3. 承認者の場合、リクエストを「承認」または「却下」できます。リクエスタの場合、必要 に応じてアセット・リポジトリを変更し、リクエストを「再送信」できます。



5 サイトの編集

エディタについて理解を深め、これで何ができるのか見ていきましょう。

- サイト・ビルダー・ページについて
- サイト更新の理解
- 更新の使用
- 編集のヒントとテクニック
- スタイルおよびフォーマットの使用
- カスタム・サイトのプロパティの追加
- サイト設定のカスタマイズ
- 表の操作
- サイト・ファイルのアップロード

サイト・ビルダー・ページについて

既存の更新を編集するか、サイトの更新を作成する場合、サイト・ビルダーで更新を開きま す。使用可能なページのレイアウトおよびツールについて、少し時間を取って理解するよう にしてください。



次に、いくつか注意する点を示します。



コールア ウト	説明
1	サイトの 名前 がリストされます。
2	現在作業中の更新の名前を 更新 で確認します。 1 つ以上の更新が使用可能な場合、 別の更新に切り替えることができます。
3	現在操作しているバージョンを確認するには、 言語 をチェックします。
4	「元に戻す」 🥙 を使用して、エディタで最後に行った編集または変更を無効にし
	ます。 「再実行」 を使用して、元に戻す処理で無効にした最新の変更を再適用 します。元に戻す処理を複数回使用し、現在の更新内のコンテンツ、スタイルおよ びページ編成の変更などの一連の変更を元に戻すことができます。
5	テスト・プロファイル を使用すると、様々なオーディエンス属性を指定したプロフ ァイルを作成し、そのプロファイルで多様なサイト・ビジターを表して、レコメン デーションをテストすることができます。テスト・プロファイルは、サイト管理者 がレコメンデーションを作成する際に設定されます。
6	プレビュー・オプション を使用して、異なる状況でコンテンツがどのように表示されるかを確認します。任意のページに移動し、現在の更新の変更内容を適用した状態でベース・サイトを表示できます。「ウィンドウに合せる」をクリックして、ディメンションを選択すると、その画面サイズでデバイスに表示されるとおりにページが表示されます。サイズはいくつか用意されていますが、独自のデバイス・サイ
	ズを作成できます。マーキングを表示するには、 くをクリックします。ルーラーの空いている部分をクリックすると、別のサイズではサイトがどのように表示され
	るかを簡単に確認できます。また、 2000 を選択すると、向きによってサイトがモバ イル・デバイスでどのように表示されるかを確認できます。選択された画面サイズ で最適に使用できるよう、対応するデザインのテーマにより、ページ・コンテンツ が自動的に調整されます。
	✓ ノート: サイトのページへのリンクはプレビュー・モードでは 機能しません。
7	ページ上部にある スイッチ を使用して、表示と編集の切替えを行います。サイトを
	プレビューするには、 フーザーに表示されるようにページが表示されます。編集時に使用されるスロッ トおよびコンポーネントの枠線やその他のビジュアル・エイドは表示されません。 ページのリンクを使用してサイト内を移動できます(ナビゲーション・メニューの リンク、テキストのリンクなどを含む)。別のサイトへのリンクはセキュリティの 観点から新規ウィンドウで開きます。現行サイトのページへのリンクは、指定した リンク・ターゲットに応じて同じウィンドウまたは新規ウィンドウで開きます。
8	使用可能な オプション を使用して、変更を処理します。「 コミット」 をクリックし て変更を ベース・サイト にマージするか、「保存」をクリックして変更を 更新 に保 存します。



サイドバーのアイコン(12–18)をクリックして、ページの管理、コンポーネントの追加、テーマの変更などを行います。

コールアウト	クリック対象	目的
12	Ð	ページの編集と追加、ネストされたページ構造の管理、ページの 再編成およびページ設定の変更を行います
13		水平レイアウトや垂直レイアウトなどのセクション・レイアウト を追加します。
14	\Diamond	サイトにアセットを追加します。サイト作成時に選択したリポジ トリ内の任意のアセットを選択できます。
15	4	サイトにコンポーネントを挿入します。テーマ、カスタム、シー ドなど、様々なタイプのコンポーネントから選択することも、使 用可能なコンポーネントをすべて表示することもできます。
16	₽	サイトに別のテーマを選択するか、ページまたはスロットの背景 を変更します。
17	\$	検索エンジン最適化 (SEO) 用のキーワードやサイト・リダイレクト など、サイトの設定を調整します。
18	ē	「サイト」ページに戻ってサイトを管理します。


サイト更新の理解

更新は、現在のベース・サイトへの変更の名前付きコレクションです。変更は、変更 をコミットしてベース・サイトを完全に更新するまで、更新内に留まります。

エディタでサイトを表示または編集するたびに、更新を使用します。エディタで加え る変更は、その更新の一部です。1つ以上の更新を持つことができ、時間とともに更新 に変更が追加され続けます。更新により、サイトに対する編集の管理に柔軟性を持た せることができます。たとえば、サイトの異なる部分に対して、複数のユーザーが自 身の更新上で作業できるようになります。個々の更新を確認して変更し、準備ができ たら、ベース・サイトに更新をコミットできます。更新をオンラインで有効にするに は、依然としてサイトを公開する必要があります。

複数の方法で更新を編成できます。

- ページの名前を持つ更新に対してページ固有の変更
- ユーザーの名前を持つ更新に対して特定のユーザーが加えた変更
- 指定された日付に加えられた変更、または特定のプロジェクトのマイルストン用に 加えられた変更

N-F:

更新では、ベース・サイトに対する更新内の変更内容が表示されます。エデ ィタ内で使用可能な更新を簡単に切り替えられますが、一度に1つの更新し か表示できません。特定のページに対して複数の更新がある場合、同じコン テンツ領域に複数の変更があると、認識できない可能性があります。ページ 上の変更の競合を防ぐには、サイトの特定のページまたは領域に対する個々 の更新をターゲットにします。

現在の更新内の変更をコミットすると、変更がベース・サイトに加えられ、更新は削除されます。サイトにアクセスできるユーザーに対して変更がオンラインで表示され るようにするには、(オンライン化または再公開することによって)サイトを公開する必 要があります。

サイトを編集するには:

- リスト内のサイトを選択して、右クリック・メニューの「開く」、またはアクション・バーの を選択します。
- 2. エディタがプレビュー・モードで開きます。変更を加えるか、サイドバーのナビゲ ーション・オプションを使用する場合、「編集」スイッチのが「編集」に設定 されていることを確認してください。
- 3. サイトに対する最初の更新である場合、更新の名前とオプションの説明を入力し、「OK」をクリックします。名前には、文字、数字、アンダースコア()およびハイフン(-)を使用できます。スペースを入力した場合、ハイフンに自動的に置換されま

す。すでにサイトに対する更新がある場合、リストから更新を選択して 、 をクリ

 特定のページを編集する場合は、左のサイドバーのサイト・ツリーを使用するか、 ページを検索するか、サイト独自のナビゲーションを使用して、ページを見つけま



す。サイト・ツリーを表示するには、左の = をクリックしてから、

をクリックします。

5. 必要に応じて、ページ・コンテンツの追加や変更を行います。「ウィンドウに合せる」を 選択してレイアウト・オプションを使用すると、別のデバイスで異なるサイズを持つペー ジがどのように表示されるかを確認できます。サイズはいくつか用意されていますが、独 自のデバイス・サイズを作成できます。マーキングを表示するには、 をクリックしま す。ルーラーの空いている部分をクリックすると、別のサイズではサイトがどのように表 示されるかを簡単に確認できます。また、 を選択すると、向きによってサイトがモバ イル・デバイスでどのように表示されるかを確認できます。

- 6. サイトを編集した場合、次のいずれかの方法で変更を保存します。
 - 現在の更新に変更を保存するには、「保存」をクリックします。現在の更新で作業を 続行し、後で更新に戻すことができます。
 - ベース・サイトへの変更を適用するには、「コミット」をクリックします。

変更が Web サイト・ユーザーに表示されるようにするには、依然としてサイトを公開す る必要があります。サイトのオンライン化またはオフライン化またはサイト変更の公開 を参照してください。

現在の更新内の変更をコミットすると、変更がベース・サイトに加えられ、更新は削除され ます。エディタでサイトを表示または編集するたびにアクティブな更新を使用する必要があ るため、新しい更新を作成できるサイト・リストに移動します。

更新では、プロセスは直線的に進みます。

- 元のサイト(ベース・サイト)があります。これをバージョン1と呼ぶことにします。
- 更新を作成します。(コミットで)更新をマージすると、更新は既存のサイトに完全に書き 込まれます。ベース・サイトは新規バージョン(バージョン2)になりました。
- 別の更新を作成してマージすると、更新は既存のサイトに完全に書き込まれます。ベース・サイトは新規バージョン(バージョン 3)になりました。

更新では、次のことを覚えていてください。

現在のベース・サイト + マージされた更新 = 新規バージョンのベース・サイト

更新の使用

エディタでサイトを編集するたびに、更新を使用します。エディタで加える変更は、その更 新の一部です。

更新を作成して使用するには:

- 「サイト」ページで、サイトを選択して右クリック・メニューの「開く」を選択するか、 アクション・バーの をクリックします。
- 2. 編集スイッチ を「編集」に設定します。
- 3. サイトに対する最初の更新である場合、更新の名前とオプションの説明を入力し、「OK」 をクリックします。文字、数字、アンダースコア(_)およびハイフン(-)を使用できます。 スペースを入力した場合、ハイフンに自動的に置換されます。既存の更新を使用するに は、リスト内の更新をクリックします。



- エディタの上部バーに、サイトの名前と現在の更新が表示されます。複数の更新が ある場合は、更新のリストから別の更新を選択して、更新を切り替えることができ ます。
- 必要に応じて、ページ・コンテンツの追加や変更を行います。異なるデバイスでペ ージがどのように表示されるかを確認するには、表示サイズの様々なオプションを 使用します。
- 6. 編集が終了したら、「保存」をクリックします。変更をベース・サイトにマージす る準備ができたら、更新をコミットできます。変更が Web サイト・ユーザーに表 示されるようにするには、サイトを公開する必要があります。サイト変更の公開を 参照してください。

編集のヒントとテクニック

ここでは、エディタの使用開始に役立つことを紹介します。

- サイドバーの非表示または表示
- 移動
- ページの管理
- ドラッグ・アンド・ドロップの編集
- コンポーネント
- コンポーネントのサイズおよび間隔の調整
- 編集可能なレイアウト・コンテンツ
- スタイルおよびフォーマット
- 変更を元に戻す

サイドバーの非表示または表示

■をクリックして、ページおよびページ・コンテンツを管理および編集するためのオ プションを表示します。

サイドバーにアクセスするには、 しを「編集」に設定します。ページをプレビュー する際に、サイドバーを非表示にして表示領域を拡大するには、 をクリックします。

移動

サイト・ツリーを使用して別のページを選択するには、サイドバーの をクリックします。ページ・リスト上部の検索ボックスを使用してページを検索します。サイト・ ナビゲーションまたはページ自体のリンクを使用することもできます。

別のページに切り替えると、ページに加えた変更が格納されます。**「保存」**をクリックして、変更を現在の更新に保存することもできます。

ページの管理

ページを追加するには、ページを追加するレベルまたはブランチを選択して「ページの追加」をクリックします。または、子ページを追加するには、



してから^{・へ}をクリックします。「新規ページ」がサイト・ツリーの下部に追加され、ページに名前を付けて他の設定を指定するよう求められます。

- ページを削除するには、ページを選択して
- ページを移動するには、ページを選択して新しい位置にドラッグします。または、ページを切り取って新しい場所に貼り付けることができます。ページを切り取るには、 をクリックしてから をクリックします。ページを貼り付けるには、貼付け先のブランチを選択し、 をクリックしてから をクリックします。
- ページをコピーするには、ページを選択し、
 をクリックしてから
 たクリックしま
 す。ページを貼り付けるには、貼付け先のブランチを選択し、
 をクリックしてから
 をクリックします。
- メタデータ、ヘッダー、フッター、その他のオプションなどのページ設定を変更するには、
 なをクリックします。「ページ設定の変更」を参照してください。

ドラッグ・アンド・ドロップの編集



スロットに複数のアイテムを含めることができ、ページ上のアイテムを新しい場所にドラッ グするだけで移動できます。スロット内の2つのコンポーネント間の境界をクリックしてド ラッグすると、2つのコンポーネントの相対的な幅を調整することもできます。コンポーネン トは、コンポーネントの上に表示されているルーラーに示された次のグリッド線に合せます。

各コンポーネントのサイズは、ピクセルでもスロット内の使用可能な領域の割合としても表示されます。グリッドで定義した以外の値に幅を調整するには、**[Ctrl]**キーを押したまま、コンポーネントの境界をクリックしてドラッグします。





コンポーネント

ページにコンポーネントを配置した後に、アイテムを選択してアイテムのメニュー・ アイコンをクリックし、「設定」を選択すると、位置合せ、間隔およびその他のプロパティを調整できます。メニュー・アイコンではなくコンポーネント名をクリック した場合は、スロットおよびコンポーネント・グループのメニュー・アイコンが表示 され、選択できます(現在のコンポーネントがコンポーネント・グループの一部である 場合)。他のタブの1つを選択した場合は、メニュー・アイコンが表示され、その要素 のプロパティを設定できます。



テーマ・デザイナは、テーマに含めるコンポーネントを指定できるため、一部のコン ポーネントは、あるサイトで使用できても、他のサイトでは使用できない可能性があ ります。テーマ・デザイナは、どのコンポーネントが、指定したページ・レイアウト の指定したスロットで許可されるかを指定することもできます。コンポーネントが特

定のスロットで許可されていない場合、配置バーは色と記号(マイナス)が変化し⁹、 次のようなメッセージが表示されます。



コンポーネントのサイズおよび間隔の調整

多くのコンポーネント(ギャラリやイメージなど)のサイズを異なる CSS 単位に調整したり、コンポーネントの周囲の空きを調整したりすることができます。例:

• px (ピクセル): デフォルト。数値のみが指定されている場合、ピクセル数とみなされます。



- % (パーセント): アイテムを親 HTML 要素のサイズに対する割合に設定します。例: 25%
- em: コンポーネントのサイズを em スペース単位で設定します。例: 20em
- vw: サイズを表示領域の幅に対する割合として設定し、表示領域の幅に応じて変化させま す。例: 10vw

編集可能なレイアウト・コンテンツ

テーマ・デザイナは、ページ・レイアウトにエディタで編集できない、コピーライトなどの 組込みコンポーネントを追加できます。また、デジタル・アセットを含め、単純なテキスト やイメージ・コンテンツを編集可能と指定することもできます。こうすることで、コントリ ビュータがテキストやイメージ・コンテンツを変更することが可能になりますが、場所やそ の他のレイアウト属性の変更はできません。編集可能なテキスト要素には、テキストに太字 や斜体、下線を指定したり、リンクの変更や削除を行ったりするためのオプションがある、 次のようなメニューが表示されます:



編集可能なグラフィック要素には、設定アイコンと、イメージ、タイトルおよび代替テキ ストを指定できるパネルが用意されています。



テキストまたはイメージを囲むフレームは、テキストの長さまたはイメージのサイズに合う ように調整されます。

▲ ノート:

加えられた変更は、現在のページにのみ適用されます。レイアウトが最初にページ に適用されると、元のコンテンツは、レイアウトとともにテーマに格納され、それ がデフォルトになります。

スタイルおよびフォーマット

ほとんどのコンポーネントには、コンポーネントの外観という面から指定するテーマによっ て定義された、1つ以上のベース・スタイルがあります。簡単にスタイルを切り替えたり、オ ーバーライドできます。使用可能なスタイル間で選択するには、コンポーネントの「設定」 パネルを開いて、「スタイル」タブをクリックし、「スタイルの選択」をクリックしてメニュ



ーでスタイルを選択します。スタイル定義に指定したプロパティに独自の値を指定するには、「カスタマイズ」をクリックし、フォーマット・オプションを指定します。

×	
yle	Divider Settings ×
	General Style
-	Choose Style
	O Customize
	Line
	Solid 🔻 1 🗸 🗸
	Corners
	× yle •

変更を元に戻す

を使用して、最後の編集またはエディタで行った変更を無効にします。元に戻す処理を複数回使用して、一連の変更を無効にすることができます。

を使用して、元に戻す処理で無効にした最新の変更を再適用します。元に戻す処理を連続して複数回使用した場合、やりなおす処理を複数回使用できます。

現在の更新内でコンテンツ、スタイルおよびページ編成の変更を元に戻すことができ ます。一部のアクションは元に戻す一連の動作には含まれません。

- 別の更新に切り替えると、元に戻す一連の動作はリセットされるため、以前に作業した更新で行った変更は元に戻せません。更新内では、変更を保存した後でも元に戻すことができます。
- ページの切替えや特定ページのサイズ変更など、エディタで表示を変更した場合、 それらの変更は手動で元に戻す必要があります。
- テキスト・コンポーネント(タイトル、段落など)を編集する場合、テキスト・エディタには独自の元に戻す一連の動作があります。テキスト・エディタを終了すると、そこでの変更を元に戻すことはできません。

スタイルおよびフォーマットの使用

ほとんどのコンポーネントには、コンポーネントの外観という面から指定するテーマ によって定義された、1つ以上のベース・スタイルがあります。簡単にスタイルを切り 替えたり、選択したオプションでスタイルをオーバーライドできます。

- 1. 編集するページに移動し、 「編集」に設定されていることを確認します。
- コンポーネントのベース・スタイルを設定するには、コンポーネントのメニュー・ アイコンをクリックし、「設定」を選択します。「スタイル」タブをクリックします。



- サイトのテーマからスタイルを使用するには、「スタイルの選択」をクリックしてメニューでスタイルを選択します。スタイルは個々のコンポーネントに対して定義されるため、スタイルのリストは変化する可能性があります。たとえば、イメージのスタイル設定は、段落のスタイル設定とは異なります。
- 独自のベース・フォーマット・オプションを指定するには、「カスタマイズ」をクリックし、フォーマット・オプションを指定します。
- ベース・スタイルを1つ以上の同様のコンポーネントにコピーして貼り付けるには、コン ポーネントのメニュー・アイコン をクリックし、componentスタイルのコピーを選 択します。同様のコンポーネントのメニュー・アイコン をクリックし、component スタイルの貼付けを選択します。
- 4. タイトルまたは段落コンポーネント内のテキストをフォーマットするには、テキスト・コンポーネントをクリックします。フォーマット・オプションのあるツールバーが表示されます。フォーマットするテキストを選択し、フォント、色、位置合せなどの任意のオプションを選択します。変更内容はすぐに適用されます。フォーマットを削除するには、テキ

ストを選択して をクリックします。フォーマットの変更は、ベース・スタイル全体に 適用されます。ベース・スタイルを変更すると、オーバーライドした内容がそのまま有効 になります。

サイト設定のカスタマイズ

サイト設定をカスタマイズして、サイト・アイコン、コントローラ・ファイル、サイトマッ プ、ロボット・ファイル、補助ファイルを追加し、マップ・プロバイダを指定できます。

これらの設定は、更新がコミットされるまで更新内に格納されます。公開後、これらのファ イルはテーマのルート・フォルダに格納され、このテーマを使用するすべてのサイトで使用 されます。

サイト全体についてアイコンおよびリンク動作を変更するには:

- 1. 編集するサイトを開きます。
- 2. サイドバーの [@]をクリックし、 「**サイト」**をクリックします。
- 「「お気に入り」アイコン」セクションで、サイトがブラウザまたはモバイル・デバイスなどの異なるプラットフォームで最小化されたときにサイトに使用するイメージを選択します。アイコンは、アクセス可能なデジタル・アセットとして格納する必要があります。「アップロードするファイルの選択」をクリックしてアイコンに移動し、そのアイコンを選択して、「OK」をクリックします。次に示すすべてのブラウザでお気に入りアイコンが機能するよう、カスタマイズする必要があります。
 - Chrome および Safari の場合、次の例で示すように、コントローラ・ファイルでサイト接頭辞を含めてお気に入りアイコンを参照する必要があります。

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

Internet Explorer 11 および Firefox の場合、ページ・テンプレートに同じような入 力をする必要があります。ただし、ページ・テンプレートはテーマの一部であるた め、複数のサイトで使用される可能性があります。そのため、固定のサイト接頭辞を 使用できないので、かわりにトークンを使用します。次の例を参照してください。

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



トークンは、ページが配信されると、サイト接頭辞に置き換えられます。

Internet Explorer と Firefox では、コントローラだけでなく、ページ・テンプレートからもお気に入りアイコンがロードされることに注意してください。そのため、コントローラとページ・テンプレートの両方から同じものが参照されないと、アイコンがフラッシュされます。Chrome と Safari では、コントローラから参照されたアイコンのみがロードされます。

4.「コントローラ・ファイル」セクションで、ブラウザでのリンク・リクエストの処理方法を変更するファイルを追加できます。お気に入りアイコンの参照の他に、Webマスターのサイト検証やソーシャル・メディアでのサイトの共有を目的として、OpenGraphタグやメタデータ・タグを追加することも可能です。デフォルトのコントローラ・ファイルをダウンロードして編集するか、または独自のファイルをアップロードします。ファイルは、アクセス可能なデジタル・アセットとして格納する必要があります。「アップロードするファイルの選択」をクリックしてファイルに移動し、そのアイコンを選択して、「OK」をクリックします。

たとえば、サイトに Facebook の「シェアする」ボタンがある場合、Facebook で 自分のサイトに関する詳細の表示に使用されるよう、次の例のように、メタデータ を入力できます。

<meta property="og:image" content="https://my.domain.com/fbimage.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"/>

次の例のような、Google ウェブマスターの検証タグを追加できます。

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

- 「サイトマップとロボット・ファイル」セクションで、カスタム・サイトマップとロボット・ファイルをアップロードできます。サイトマップとは、Web サイトのURL、および各 URL に関する情報(最終更新時期など)をリストするために使用できる XML ファイルのことです。ロボット・ファイルとは、検索エンジン・ロボットに、Web サイト上のページの索引付け方法を指示するために作成できるテキスト・ファイルのことです。ファイルは、アクセス可能なデジタル・アセットとして格納する必要があります。「アップロードするファイルの選択」をクリックしてファイルに移動し、そのアイコンを選択して、「OK」をクリックします。
- 6.「補助ファイル」セクションで、必要に応じて別のファイル(サイト所有権の確認に 必要なファイルなど)をアップロードできます。ファイルは、アクセス可能なデジ タル・アセットとして格納する必要があります。「アップロードするファイルの選 択」をクリックしてファイルに移動し、そのアイコンを選択して、「OK」をクリッ クします。
- 使用するマップ・コンポーネントとリンクのプロバイダを選択できます。「Oracle Maps」または「Google マップ」を選択します。
- 8. 更新を公開すると、変更も公開されて使用されます。

カスタム・サイトのプロパティの追加

名前/値のペアの形式でカスタム・プロパティをサイトに追加できます。これらのプロパティ はサイトともに格納され、サイトのページ上でスクリプトおよびコンポーネントに対して使 用できるようになります。これにより、基礎となるスクリプトとコンポーネント・コードを 変更することなく、サイトをパラメータ化またはカスタマイズする際に役立ちます。

たとえば、カスタム・プロパティを使用して、ページの背景色を変更したり、検索結果を絞 り込んだり、リストに入力したり、サイト依存の変数を制御できます。

カスタム・サイト・プロパティは、サイトの編集時に設定パネルを使用して追加されます。

- 1. 編集するサイトを開きます。
- 2. サイドバーの[◎]をクリックし、 **□□ 「サイトのプロパティ」**をクリックします。
- 3. 「追加」をクリックします。
- カスタム・サイト・プロパティの名前および値を入力します。最大 50 個のカスタム・サイト・プロパティを追加できます。名前フィールドには 200 文字の制限があり、値フィールドには 2000 文字の制限があります。

=	Events-Site Demo 🔻 en-US (Default) 💌 🍙 🍽					
Ð	Settin	gs Ø	Site Properties			
	~~~7		Custom site properties can be defined site's pages can use these properties t	d for this site. Scripts and components on to tailor their behavior.	the	
Ø	SEO	Site	MarketingCostID	Value	×	
+1			BackgroundColorScheme	LightGreen	×	
·+			SiteGreeting	Welcome to the Event Coordination S	i×	
Ą	Redirects	Analytics	ContactEmail	joe.bloggs@example.com	×	
			ContactName	Joe Bloggs	×	
()) ())			DefaultSearchInfo	Asia	×	
	Site Properties		Configuration I	Blue / Large / Hello	×	
			Add			

- 5. 名前/値のペアの横にある「X」をクリックして、削除します。
- 6. カスタム・サイト・プロパティの追加または削除が終了したら、「**コミット」**をクリックします。

🖋 ノート: 変更は、既存のカスタム・サイト・プロパティにマージされません。変更をカスタ ム・サイト・プロパティにコミットすると、ベース・サイトの既存のカスタム・サ イト・プロパティが上書きされます。



定義すると、カスタム・サイト・プロパティは、サイトやサイト・コンポーネントでのスクリプト記述で使用できます。たとえば、フッター、またはコンテンツ・リストなどの SCSMacro 拡張機能をサポートしているコンポーネントの「追加問合せ文字列」フィールドやタイトルおよびパラグラフ・コンポーネントのトークンなどです。

このスクリプト記述は、実行時および設計時に SCSRenderAPI とともに機能します。 また、カスタム・サイト・プロパティの値は、SCSCompileAPI を介してテンプレー ト・コンパイラのレイアウトおよびコンポーネント・コードに対して使用できます。 この API には SCSRenderAPI、getCustomSiteProperty に似た新しい関数が含まれてお り、レイアウトおよびコンポーネント・コードでカスタム・セクション・プロパティ の値を読み取ることができます。

たとえば、SCSRenderAPIを使用してコールし、ヘッダー、連絡先名および連絡先電 子メールのカスタム・サイト・プロパティを取得するカスタム・セクション・レイア ウトを定義するとします。次のスクリプトでは、DOM に追加したカスタム・サイト・ プロパティを使用して HTML 文字列を作成します。

```
define([
```

```
'iquery'
], function( $ ){
    'use strict';
    function SectionLayout( params ) {
    SectionLayout.prototype = {
        render: function( parentObj ) {
            var html = '';
            try {
                html += '<div>';
                    html += '<h1>' +
SCSRenderAPI.getCustomSiteProperty('SiteGreeting') + '</hl>;
                    html += '<div>For more information, contact <a</pre>
href="mailto:' + SCSRenderAPI.getCustomSiteProperty('ContactEmail')
                         '"> +
SCSRenderAPI.getCustomSiteProperty('ContactName') + '</a></div>';
                html += '</div>';
                $(parentObj).append( html );
            \} catch( e ) {
                console.error( e );
            }
        },
    };
```

カスタム・プロパティは、追加されたカスタム・サイト・プロパティでサイトからテ ンプレートを作成するときやカスタム・サイト・プロパティを含むテンプレートから サイトを作成するときに保持されます。

## 表の操作

段落コンポーネント内に、作成した表または既存の HTML ソースから貼り付けた表を 含めることができます。

- 1. 編集するページに移動し、 「編集」に設定されていることを確認します。
- 2. コンポーネントをページに追加するか、既存の段落コンポーネント内をクリックします。
- 3. 別の HTML ページから表を追加するには、クリックおよびドラッグして表を選択し、コ ピーして段落コンポーネントに貼り付けます。現在のカーソルの位置に新しい表を作成 するには、 ■●をクリックします。行列の数、幅および他のフォーマット・オプションを 選択します。表の上の中央にオプションの「キャプション」を追加したり、自動スクリー ン・リーダーなどの支援技術に追加のコンテキストを提供するオプションの「サマリー」 を追加できます。
- 4. 終了したら「OK」をクリックしてウィンドウを閉じます。これらの表の設定を後で変更 するには、表を右クリックし、「表プロパティ」を選択します。表を削除するには、表を 右クリックし、「表の削除」を選択します。

## 🖋 ノート:

モバイル画面など、様々な表示サイズを作成する場合は、**StarterTheme** の **design.css** ファイルにあるコードを使用してレスポンシブ表を作成すると、モ バイル・デバイスで表示された際に、行データの積上げが可能になります。

5. 特定の行、列またはセルを追加、削除または変更するには、行、列またはセルを右クリックし、オプションのメニューから選択します。例:

	Cell	۲	Insert Cell Before
	Row	۲	Insert Cell After
	Column	Þ	Delete Cells
	Delete Table		Merge Cells
⊞	Table Properties		Merge Right
			Merge Down
			Split Cell Horizontally
			Split Cell Vertically
			Cell Properties

✓ ノート: セルのプロパティは、一度に1つのセルのみ変更できます。

6. セルをマージしたり、分割して複雑な表のレイアウトを作成できます。例:

	My First Table		
		Split Cells	
Merged cells			

- セルを2つのセルに分割するには、セルを右クリックして「セル」を選択し、「セルの水平分割」または「セルの垂直分割」を選択します。
- 水平方向の2つのセルをマージするには、左側のセルを右クリックして「セル」を選択し、「右にマージ」を選択します。
- 垂直方向の2つのセルをマージするには、上のセルを右クリックして「セル」
   を選択し、「下にマージ」を選択します。
- 7. 一般的なフォーマット・メニュー・オプションを使用して、セル内のテキストの書 式および位置合せを変更できます。

## サイト・ファイルのアップロード

Oracle Content Management インタフェースを使用して、サイトで使用するイメージ およびドキュメントをいつでもアップロードできます。背景イメージ、およびイメー ジ、ギャラリおよびドキュメントのコンポーネントの操作中に、サイト・ビルダー内 でファイルをアップロードすることもできます。

サイトでイメージおよびドキュメント・ファイルを使用する場合、サイトに格納され たイメージまたはアクセス可能な別の場所にあるイメージを使用できます。共有され ているイメージまたはローカルまたはネットワークのファイル場所からアップロード するイメージを使用することもできます。

#### ファイルのアップロード

ローカルまたはネットワークの場所から1つ以上のファイルをアップロードするには:

- 1. 「ドキュメント」をクリックし、ファイルを格納する場所に移動します。「作成」を クリックし、現在の場所に新規フォルダを追加します。
- 2. 「アップロード」をクリックします。
- 3. 1つ以上のファイルを検索して選択し、「開く」をクリックします。

#### エディタ内でのファイルのアップロード

背景イメージまたはドキュメントやイメージなどのコンポーネントの操作時に、使用 するファイルが Oracle Content Management の場所にない場合、ファイルを直接アッ プロードできます。

ローカルまたはネットワークの場所から1つ以上のファイルをアップロードしてギャ ラリ・コンポーネントで使用するには、次のステップを実行します。

- ギャラリにイメージを追加するには、そのメニュー・アイコン=をクリックし、
   「設定」を選択して、「一般」タブの「イメージ」をクリックします。
- **2. 「イメージの追加」**をクリックします。
- 3. ファイルを格納する場所に移動するか、「**作成」**をクリックして現在の場所に新規 フォルダを追加します。



- 4. 「**アップロード」**をクリックします。
- 5. 1つ以上のファイルを検索して選択し、「開く」をクリックします。
- 6. リポジトリの1つ以上のイメージを選択し、「OK」をクリックします。

### 🖍 ノート:

ウィンドウに使用可能なすべてのファイルが表示されます。コンテキストに適切なファイル・タイプを選択してください。たとえば、イメージ・ファイルを 選択する場合、有効なイメージ・フォーマット(GIF、JPG、JPEG、PNG また は SVG)のファイルを選択します。ファイルにリンクするには、「サイトにファ イルをコピーするかわりに元のファイルへの参照を使用します」を選択します。 このオプションを選択しない場合、ファイルのコピーがサイトで格納され、サ イトから参照されます。



6

# サイトでのテンプレートおよびテーマの使用

テンプレートには、ユーザーがサイトの使用を開始するにあたり必要なもの、たとえば、サ イト・コード・フレームワーク、サンプル・ページとコンテンツを含むデフォルト・サイト、 スタイルが設定されたテーマ、イメージなどのリソース、カスタム・コンポーネントなどが 含まれます。テーマでは、色スキーム、フォント・サイズ、フォント・タイプ、ページ背景 など、サイトの一般的なルック・アンド・フィール(全体スタイル)を定義します。

#### テンプレート

- テンプレートの理解
- サイトからのテンプレートの作成
- テンプレート詳細の変更
- テンプレート・ポリシーの変更
- テンプレートのステータスまたはオーディエンスの変更
- テンプレートの管理
- テンプレートのエクスポートとインポート

#### テーマ

- テーマの理解
- テーマの管理
- テーマの公開

## テンプレートの理解

テンプレートには、ユーザーがサイトの使用を開始するにあたり必要なもの、たとえば、サ イト・コード・フレームワーク、サンプル・ページとコンテンツを含むデフォルト・サイト、 スタイルが設定されたテーマ、イメージなどのリソース、カスタム・コンポーネントなどが すべて含まれます。

#### デフォルト・テンプレート

Oracle Content Management には、サイトの作成に使用できる多数のテンプレートが用意されています。テンプレートを選択し、サイトに名前を付けるだけで、コンテンツの追加をただちに開始できます。これらのテンプレートは、通常、サービスが初期化される際にサービス管理者によってインストールされます。

テンプレート	説明
空のテンプレート	空のテンプレートはヘッダー、本文、フッターのスロットがある単一のペー ジで、独自の設計を完全に自由に行うことができます。



テンプレート	説明
Collateral Showcase	Collateral Showcase テンプレートには、右揃えの水平メニュー・スタイル、 単純なページ・レイアウト、および関連するビデオ・アセットをリストする 最上位レベルのページが含まれ、その後に特定のビデオをそれぞれ紹介する ネストした詳細ページが含まれます。これは標準メニュー・コンポーネント を特徴としており、ここには現在ログインしているユーザーが表示されま す。
New Product Launch	New Product Launch テンプレートには、右揃えの水平メニューと、ケース・スタディ、機能の詳細、価格設定、企業および連絡先情報のためのレイアウトがあります。ホーム・ページには、カルーセル・バナー・イメージとテキストがあります。
Products and Services Overview	Products and Services Overview テンプレートには、ケース・スタディ、 製品提供の詳細、企業、プライバシ・ポリシーなどのためのレイアウトがあ ります。ホーム・ページには、カルーセル・バナー・イメージとテキストが あります。テンプレートは非常に応答性に優れています。
Starter テンプレート	既成のサイト・ソリューションを独自に作成する場合、Starter テンプレートを使用します。Starter テンプレートには単純ではあるが非常に機能的な 例が用意されており、確認してコンポーネントや対話で拡張できます。サイト・コード・フレームワーク、サンプル・ページやコンテンツを含むデフォ ルト・サイト、スタイルを伴ったテーマ、イメージなどのリソース、および トリガーやアクションの機能を備えたカスタム・コンポーネントが含まれて います。サンプル・ページにはテンプレートの作成に関する情報および詳細 情報を提供するリソースへのリンクが含まれます。
Learn Present Show	これらのテンプレートは相互に類似のものです。これらは応答性に優れ、グ ループ内のカスタム JavaScript メニュー・コンポーネントを特徴としていま す。これらは、多数の標準コンポーネントが含まれたカスタム・コンポーネ ント・グループを特徴としています。ここにもログインしているユーザーが 表示されます。
Access Knowledge Relate	これらのテンプレートは相互に類似のものです。これらは応答性に優れ、グ ループ内のカスタム JavaScript メニュー・コンポーネントを特徴としていま す。これらは、多数の標準コンポーネントが含まれたカスタム・コンポーネ ント・グループを特徴としています。
Share	このテンプレートは、単一の長いページを特徴とするサイトです。これは応 答性に優れ、個別のページではなくページ上の場所に移動するカスタム JavaScript メニューを特徴としています。
JET Starter テンプレー ト	Oracle JET (JavaScript Extension Toolkit)は、オープン・ソースの JavaScript ライブラリのコレクションを使用するモジュール式のオープン・ ソース・ツールキットです。詳細は、Oracle Jet を参照してください。

また、既存のサイトからテンプレートを作成することも、既存のテンプレートをエク スポートし、オフラインで変更し、新しいテンプレートとしてインポートすることも できます。組織では、ユーザーが使用するためのテンプレートが用意されていること があります。

#### テンプレートの動作方法

サイトを作成する場合、テンプレートは次のように使用されます。

- サイト・ガバナンスが有効化されている場合、サイトの作成前に承認が必要になる ことがあります。サイト・ガバナンスの理解を参照してください。
- テンプレート内のデフォルト・サイトが新規サイトにコピーされ、ページの開始点を提供します。
- すべての必要なサポート・ファイルが新規サイトにコピーされます。



- テンプレート・テーマがテーマ・フォルダに存在しない場合は、テーマがテーマ・フォル ダにコピーされます。サイトは、テーマ・フォルダ内の場所からテーマを参照します。テ ーマが存在する場合、新規サイトは単純に既存のテーマを参照します。
- コンポーネント・フォルダに存在しないカスタム・コンポーネントがある場合は、それら がコンポーネント・フォルダにコピーされます。サイトは、コンポーネント・フォルダ内 の場所からコンポーネントを参照します。コンポーネントが存在する場合、新規サイトは 既存のコンポーネントを参照します。

#### カスタム・テンプレートの作成

Web 開発者の場合、テンプレートでは、Web サイトを構築するために必要なすべての要素(サイト、レイアウト、ナビゲーション、サンプル・コンテンツなど)が1つのパッケージにまとめられています。開発者は、コンポーネントおよび対話的な操作をサイトに追加し、ビジネス・ニーズに適合した既製のサイト・ソリューションを提供できます。

テンプレートは、他のフォルダと同様に操作できるフォルダ構造によって表されます。テーマやカスタム・コンポーネントなどのテンプレートの一部の要素は、Oracle Content Management 内の関連付けられている場所から参照されます。たとえば、テンプレートは、 サイトがテーマを参照するのと同じ方法で、関連付けられているテーマを使用可能なテーマ のリストから参照します。

既存のサイトからテンプレートを作成する場合、新規テンプレートでは、サイトのコピーが デフォルト・サイトとして使用されます。テンプレートは、サイトによって使用されるテー マ、およびサイトのページで使用されるカスタム・コンポーネントを参照します。テーマお よびカスタム・コンポーネントは、テンプレートにコピーされませんが、サイトによる参照 と同じ方法で参照されます。

✔ ノート:

テンプレートは、テンプレートが作成されるときに使用されたサイトを反映してい ます。テンプレートの作成に使用されたサイトへのその後の変更は、そのテンプレ ートとともに格納されているサイトでは反映されません。

コンテンツ・モデル(サンプル・コンテンツなど)をサポートするために必要なコンテンツ・レ イアウト、コンテンツ・アイテムおよびデジタル・アセットを含むコンテンツ・モデルを共 有するために使用できるコンテンツ・テンプレートを作成できます。コンテンツ・テンプレ ートは、公開されたコンテンツ・アイテムおよびデジタル・アセットを含むサイトから作成 する必要があります。コンテンツを含むパッケージからサイト・テンプレートをインポート すると、コンテンツ・タイプは作成されません。コンテンツ・テンプレートからサイトを作 成すると、作成されます。したがって、コンテンツ・タイプは、サイト・テンプレートをイ ンポートするユーザーは所有していません。かわりに、コンテンツ・タイプは、そのテンプ レートから最初のサイトを作成するユーザーが所有しています。そのユーザーは、必要に応 じて他のユーザーとテンプレートを共有できます。この機能は、Content Management Cloud サブスクリプション・タイプおよびサービスの開始日によっては使用できない可能性があり ます。

テンプレートをエクスポートすると、テーマやコンポーネントのコピーなど、テンプレートのすべての要素は、ダウンロード可能でオフラインで操作できるテンプレート・パッケージにまとめられます。

オフラインで変更したテンプレートをインポートする場合や、テンプレート、テーマまたは カスタム・コンポーネントの名前または ID がすでに存在する場合、競合を解決するよう求め られます。新規テンプレート、テーマまたはカスタム・コンポーネントを作成するオプショ ンが表示されるか、場合によっては、既存のテンプレート、テーマまたはカスタム・コンポ



ーネントを、インポートしたバージョンで上書きできます。テンプレートの開発を参 照してください。

特定のリポジトリにテンプレートをインポートすることもできます。これを行う場合、既存のアセットを更新または複製する選択肢が用意されています。特定のリポジ トリへのテンプレートのインポートを参照してください。

#### テンプレートの共有

テンプレートを作成した場合、サイトからインポートしたか、コピーしたか、作成し たかには関係なく、自分でテンプレートを共有するまで他のユーザーは使用できませ ん。

### 💉 ノート:

このことは、Oracle Content Management に用意されているテンプレートを 管理者がインストールした場合にも当てはまります。テンプレートが見当た らない場合は、管理者に連絡してください。これらが自分と共有されていな い可能性があります。

テンプレートをユーザーと初めて共有すると、関連付けられたテーマおよび関連付け られたカスタム・コンポーネントは、自動的にユーザーと共有され、ダウンロード実 行者ロールが付与され、それらが使用可能になります(ユーザーがテンプレートからサ イトを作成した場合)。その後、テンプレート内でそのユーザーのロールに変更を加え ても、関連付けられたテーマまたはカスタム・コンポーネントの共有情報は更新され ません。

サイト・ガバナンスが有効化されている場合、テンプレート・ポリシーを介してテン プレートを使用可能にします。「サイト・ガバナンスの理解」および「テンプレート・ ポリシーの変更」を参照してください。

## サイトからのテンプレートの作成

他のサイトの開始点として使用するサイトがある場合は、そのサイトからテンプレートを作成できます。

### ✓ ノート:

サイトからテンプレートを作成するには、そのサイトに対するダウンロード 実行者、コントリビュータまたはマネージャ・ロールが必要です。 また、管理者が「作成」メニューのオプションを有効にする必要があります。 テンプレート・ページに「作成」メニューがない場合は、管理者に連絡して ください。

次のステップで、サイトからテンプレートを作成する方法を示します。オフラインで 作成または変更したテンプレート・パッケージをインポートすることもできます。テ ンプレートのエクスポートとインポートを参照してください。



**Oracle Content Management Web** インタフェースを使用して、サイトからテンプレートを作成するには:

- 1. 「サイト」をクリックし、使用するサイトを選択します。
- 2. 右クリック・メニューの「テンプレートの作成」を選択するか、アクション・バーの レートの作成」を選択するか、アクション・バーの
- テンプレートの名前を入力します。文字、数字、アンダースコア(_)およびハイフン(-)を 使用できます。スペースを入力した場合、ハイフンに自動的に置換されます。
   テンプレート、テーマ、コンポーネント、サイトまたはサイト・ページについて、 authsite、content、pages、scstemplate_*、_comps、_components、_compsdelivery、_ idcservice、_sitescloud、_sitesclouddelivery、_themes、_themesdelivery といった名前 を使用しないでください。documents および sites は、サイト・ページには使用できます が、テンプレート、テーマ、コンポーネントまたはサイトには使用しないでください。
- 4. テンプレートのオプションの説明を入力します。
- 5. 公開されていないコンテンツ・アイテムおよびデジタル・アセットをテンプレートに含め るかどうかを選択します。
- 6. 準備ができたら、「テンプレートの作成」をクリックします。

テンプレートが作成されると、名前がテンプレートのリストに表示されます。すべてのテン プレートを表示するには、「開発者」をクリックしてから「すべてのテンプレートの表示」を クリックします。「表示」アイコンをクリックしてリストからオプションを選択することによ り、テンプレートがどのように表示されるかを制御できます。



テンプレートに関連付けられたフォルダおよびファイルを表示するには、右クリック・メニ ューから「開く」を選択するか、アクション・バーのBをクリックします。テンプレートの名 前、説明およびその他の詳細を表示または変更するには、テンプレート名をクリックするか、 右クリック・メニューの「詳細」を選択するか、アクション・バーの をクリックします。 新規テンプレートでは、サイトのコピーがデフォルト・サイトとして使用されます。テンプ レートは、サイトによって使用されるテーマ、およびサイトのページで使用されるカスタム・ コンポーネントを参照します。テーマおよびカスタム・コンポーネントは、テンプレ ートにコピーされませんが、サイトによる参照と同じ方法で参照されます。

テンプレートがエンタープライズ・サイトから作成された場合、元のサイトで使用されていたローカリゼーション・ポリシーおよびデフォルト言語が、テンプレートから 作成された新規サイトでデフォルトで選択されます。

テンプレートは、テンプレートが作成されるときに、それを作成するために使用され たサイトを反映しています。テンプレートの作成に使用されたサイトが将来的に変更 されても、そのテンプレートとともに格納されているサイトでは反映されません。

### テンプレート詳細の変更

テンプレート詳細タブにはテンプレート名、作成者、説明、テンプレート・テーマ、 テンプレートに含まれているカスタム・コンポーネント、およびテンプレートのプレ ビュー・イメージが表示されます。サイト・ガバナンスが有効化されている場合、テ ンプレートのステータス(サイトの作成時にテンプレートを使用できるかどうか)も表 示されます。

適切な権限を持っている場合、名前や説明などのテンプレート・プロパティを変更ま たは更新できます。テンプレートのプレビュー・イメージを追加または削除すること もできます。

テンプレート詳細を更新できるのは、テンプレートを作成した(テンプレートの所有者 である)場合か、サイト管理者である場合か、他のユーザーがあなたとテンプレートを 共有していて、あなたにコントリビュータまたはマネージャ・ロールを付与している 場合です。

プレビュー・イメージを追加した場合、ファイルは、テンプレートのアセット・フォ ルダに格納されます。これらのファイルは、関連する Web サイトに表示されません が、テンプレート所有者はテンプレート自体の情報を指定できます。

テンプレート詳細を表示または変更するには:

- 1. 「開発者」をクリックし、「すべてのテンプレートの表示」をクリックします。
- 2. テンプレートを選択して、右クリック・メニューの「詳細」を選択するか、アクション・バーの をクリックします。
- 3. 必要に応じて情報を編集します。情報を編集できない場合、ユーザーにはコントリ ビュータまたはマネージャ・ロールがありません。

プレビュー・イメージの場所がすべて埋まっている場合、新しいイメージを追加す るにはイメージを削除する必要があります。プレビュー・イメージを削除するに

は、イメージのとをクリックします。

5. 「保存」をクリックします。



## テンプレート・ポリシーの変更

テンプレート・ポリシー・タブには、テンプレートのタイプ(標準またはエンタープライズ)、 必要な承認、およびテンプレートから作成されたサイトの最小限のセキュリティが表示され ます。テンプレートのステータス(サイトの作成時にテンプレートが使用可能かどうか)も表 示されます。

テンプレート・ポリシー・タブが表示されるのは、サイト・ガバナンスが有効で、かつユー ザーがサイト管理者である場合です。サイト・ガバナンスの理解を参照してください。

テンプレート・ポリシーを表示または変更するには:

- 1. 「開発者」をクリックし、「すべてのテンプレートの表示」をクリックします。
- 2. テンプレートを選択して、右クリック・メニューの「詳細」を選択するか、アクション・ バーの たっします。
- 3. 「ポリシー」タブを選択します。
- 4. このテンプレートから作成されたサイト・リクエストはサイト管理者の承認が必要なのか、自動的に承認されるのか、特定のユーザーの承認が必要なのかを選択します。承認を特定のユーザーに制限する場合、承認者として追加するユーザーまたはグループの名前または電子メールを入力し始め、検索結果からユーザーまたはグループを選択します。承認者を削除するには、名前の横の「X」をクリックします。
- 5. このテンプレートから作成されたサイトに必要な最小限のセキュリティを選択します。 サイト作成者は、必要に応じて、自分のサイトにこれより高いレベルのセキュリティを選 択できます。
  - 特定のサービス・ユーザー Oracle Content Management のこのインスタンスへのサインインが可能な選択されたユーザーのみがサイトにアクセスできます。サイトの作成後、サイト作成者が Oracle Content Management ユーザーを選択します。サイト・セキュリティの変更を参照してください。
  - 特定のクラウド・ユーザー ドメインへのサインインが可能な選択されたユーザーの みがサイトにアクセスできます。サイトの作成後、サイト作成者がクラウド・ユーザ ーを選択します。サイト・セキュリティの変更を参照してください。
  - サービス・ユーザー Oracle Content Management のこのインスタンスへのサインインが可能なすべてのユーザーがサイトにアクセスできます。
  - クラウド・ユーザー ドメインへのサインインが可能なすべてのユーザーがサイトに アクセスできます。
  - **全員** 誰もがサインインせずにサイトにアクセスできます。
- 6. テンプレートを使用しているサイトの期限が切れるタイミングを決定するために有効期限ポリシーを選択します。サイト管理者は、期限切れのサイトをオフラインにするかどうか、または削除するかどうかを指定できます。サイト所有者および管理者は、サイトの有効期限が切れる前に電子メールで通知を受け取り、有効期限を延長する機会を与えられます。有効期限を延長しない場合は、サイトがオフラインになるか削除されたときに電子メールで通知されます。その時点で、必要に応じて有効期限を延長し、サイトをオンラインに戻すかごみ箱から復元できます。有効期限ポリシーは次のとおりです:
  - なし
  - 1年
  - 2年



カスタム

「カスタム」を選択すると、1年より短いか2年より長い有効期限を設定できます。

💉 ノート:

サイトが作成された後、サイト管理者は**「サイトのプロパティ」**ダイ アログでサイトの有効期限ポリシーを変更できます。

- エンタープライズ・テンプレートを編集している場合、簡易 URL 値のサイト接頭 辞の作成方法を選択できます。この接頭辞は、コンテンツ・アイテム・スラグ値 (ページまたはアセットに固有の URL の一部)に追加されます。サイト名に基づい て接頭辞を自動的に生成するか、ユーザーに接頭辞を入力させることができます。
- 8. エンタープライズ・テンプレートを編集している場合、サイトに必要な言語を決定 するために使用するローカリゼーション・ポリシーを選択します。
- 9. 終了したら、「保存」をクリックします。

## テンプレートのステータスまたはオーディエンス の変更

テンプレート・オーディエンス・タブには、テンプレートのステータス(サイトの作成 時にテンプレートを使用できるかどうか)と、サイトを作成するためのテンプレートを 使用できるユーザーが表示されます。

テンプレート・オーディエンス・タブが表示されるのは、サイト・ガバナンスが有効 で、かつユーザーがサイト管理者である場合です。サイト・ガバナンスの理解を参照 してください。

テンプレート・オーディエンスを表示または変更するには:

- 1. 「開発者」をクリックし、「すべてのテンプレートの表示」をクリックします。
- 2. テンプレートを選択して、右クリック・メニューの「詳細」を選択するか、アクション・バーの を をクリックします。
- 3. 「オーディエンス」タブを選択します。
- 4. テンプレートのステータスを変更するには、ステータス・スイッチをクリックします。テンプレートからサイトを作成するには、テンプレートをユーザーに対してアクティブにする必要があります。デフォルトでは、アクティブ化したテンプレートは、すべてのサイト作成者が使用できます。テンプレートを使用できるユーザーを変更する場合、このタブで設定を変更します。
- 5. このテンプレートを使用してサイトを作成できるユーザーを選択します。
  - 特定の個人―選択されたユーザーのみがこのテンプレートを使用してサイト を作成できます。このテンプレートを使用できるようにするユーザーまたは グループの名前または電子メールを入力し始め、検索結果からユーザーまたは グループを選択します。ユーザーまたはグループを削除するには、名前の横の 「X」をクリックします。
  - **全員**—サイトを作成できるすべてのユーザーがこのテンプレートを使用できます。

6. 終了したら、「保存」をクリックします。

# テンプレートの管理

テンプレート・フォルダは、他のフォルダと同じようにコピー、名前変更および削除できま すが、テンプレートをインポートまたは共有する際には特別な配慮が必要です。

「開発者」、「すべてのテンプレートの表示」の順に選択して、「テンプレート」ページを開き ます。「テンプレート」ページのアクション・バーまたは右クリック・メニューのオプション を使用して、次のいずれかのタスクを実行します。

タスク	
テンプレートの作 成	他のサイトの開始点として使用するサイトがある場合は、そのサイトからテンプ レートを作成できます。
	ノート: サイトのダウンロード実行者、コントリビュータまたはマネージャ・ロールを持っている必要があり、管理者がオプションを有効にしている必要があります。テンプレート・ページに「作成」メニューがない場合は、管理者に連絡してください。また、複数のリポジトリからのコンテンツを使用するサイトからテンプレートを作成している場合、OCE ツールキットを使用して、これを行う必要があります。「OCE ツールキットでの開発」および「cec コマンドライン・ユーティリティの使用」を参照してください。
	テンプレート・ページで、 <b>「作成」</b> をクリックし、 <b>「既存のサイトから作成」</b> を選 択します。サイトを選択し、テンプレートの名前を指定して、 <b>「作成」</b> をクリック します。サイトからのテンプレートの作成を参照してください。
テンプレート・ファ イルの表示および 編集	<ul> <li>テンプレートを選択して右クリック・メニューの「開く」を選択するか、アクション・バーのかをクリックすると、テンプレートを構成するフォルダとファイルを検索できます。</li> <li>Web開発者は、個々のテンプレート・ファイルをダウンロードおよび変更したり、デスクトップ・アプリケーションを使用してローカル・システムで行った変更を同期したりできます。また、関連するテーマとカスタム・コンポーネントを含むテンプレート・パッケージをエクスポートして、それを適切な開発環境でオフラインのまま操作できます。</li> <li>テンプレートは、他のフォルダと同様に操作できるフォルダ構造によって表されます。</li> <li>テンプレートをエクスポートすると、テーマやコンポーネントのコピーなど、テンプレートのすべての要素は、ダウンロード可能でオフラインで操作できるテンプレート・パッケージにまとめられます。テンプレートの開発を参照してください。</li> </ul>
サイトの作成	テンプレートからサイトを作成するには、テンプレートを選択して右クリック・
	メニューの <b>「サイトの作成」</b> を選択するか、アクション・バーの <b>じ</b> をクリックします。サイトの作成を参照してください。

タスク	説明
テンプレート名の	名前を変更するテンプレートを選択して、右クリック・メニューの <b>「名前変更」</b>
変更	を選択するか、アクション・バーの ・ をクリックします。同じサーバー上の他のテンプレートとは異なる名前をテンプレートに対して入力します。名前を指定する際のガイドラインは、「サイトからのテンプレートの作成」を参照してください
テンプレートのコ ピー	既存のテンプレートをコピーし、コピーに変更を加えることによって、テンプレ ートを作成できます。
	<ul><li>ノート:</li><li>テンプレートをコピーする場合、テンプレートに関</li></ul>
	連付けられている共有情報はコピーされません。
	コピーするテンプレートを選択して、右クリック・メニューの <b>「コピー」</b> を選択
	するか、アクション・バーの をクリックします。同じサーバー上の他のテン プレートとは異なる名前をテンプレートに対して入力します。名前を指定する際 のその他のガイドラインは、「サイトからのテンプレートの作成」を参照してくだ さい
	進行状況バーに、新しいテンプレート名とコピー・ステータスが表示されます。
テンプレートの削 除または復元	適切な権限を持っている場合、テンプレート・フォルダとそのコンテンツを削除 できます。テンプレートを削除すると、テンプレート・フォルダとその関連フォ ルダおよびファイルはごみ箱に移動されます。
	テンプレートを削除または復元できるのは、テンプレートを作成した(テンプレー トの所有者である)場合か、他のユーザーがあなたとテンプレートを共有してい て、あなたにコントリビュータまたはマネージャ・ロールを付与している場合で す。
	✓ ノート:
	テンプレートを削除しても、関連付けられているテ ーマおよびカスタム・コンポーネントは削除されま せん。
	テンプレートを削除するには、削除するテンプレートを選択して、右クリック・
	メニューの「 <b>削除」</b> を選択するか、アクション・バーの テンプレートをごみ箱に移動するよう求められます。削除したテンプレートは、 次までごみ箱にあります。 • テンプレートを復元します。

- テンプレートを完全に削除します。
- ごみ箱の割当て制限に達しました。
- ごみ箱は、サービス管理者が設定した間隔に基づいて自動的に空になります。 デフォルト値は **90**日です。

テンプレートを復元するには、「テンプレート」ドロップダウン・リストの「ごみ 箱」を選択し、テンプレートを選択して、メニュー・バーまたは右クリック・メ ニューから「復元」を選択します。



タスク	説明
テンプレートのエ クスポートまたは インポート	テンプレートをエクスポートしオフラインで変更して、その後新規テンプレート としてインポートするか、既存テンプレートを置き換えることができます。また、 テンプレートをエクスポートして別のインスタンスに移動し、そこでインポート することもできます。
	テンプレートをエクスポートする際、テンプレートをフォルダに単一の.zipファ イルとしてコピーします。テンプレート・パッケージは、フォルダから直接ダウ ンロードして解凍し、個別のファイルを操作できます。ファイルの操作が完了し たら、テンプレート・パッケージを含む.zipファイルを作成し、それをインポー トして、元のテンプレートを上書きするか、新規に作成します。
	テンプレートのエクスポートとインポートを参照してください。
テンプレートの共 有およびメンバ ー・ロールの管理	テンプレートの所有者またはマネージャである場合、テンプレートを他のユーザーと共有し、ユーザーがテンプレートで実行できることを定義するロールを割り 当てることができます。
	ガバナンスが有効 <i>ではない</i> 場合、ユーザーに任意のロールを割り当てると、テン プレートからサイトを作成する権限がそのユーザーに付与されます。ガバナンス が有効 <i>である</i> 場合、テンプレートをアクティブ化して、ユーザーがテンプレート・ オーディエンスに含まれ、テンプレートからサイトを作成する権限が付与されて いることを確認する必要があります。テンプレートのステータスまたはオーディ エンスの変更を参照してください。
	テンプレートのテーマとカスタム・コンポーネントは、個別オブジェクトとして 格納され、テンプレートで参照されます。そのため、テーマとカスタム・コンポ ーネントのメンバーシップは、個別に管理されます。ユーザーとテンプレートを 共有する場合、そのテンプレートに対してユーザーにどのロールを付与しても、 関連する任意のテーマおよびカスタム・コンポーネントに対するダウンロード実 行者ロールが自動的に付与され、テンプレートからサイトを作成する場合にユー ザーは必ずそれらのオブジェクトを使用できます。
	共有するテンプレートを選択して、右クリック・メニューまたはアクション・バ ーの <b>「メンバー」</b> を選択します。
	<ul> <li>1つ以上のユーザー名または電子メール・アドレスを入力し、次のロールのいず れかを割り当てます。</li> <li>参照者:参照者はテンプレートのフォルダとファイルを表示できますが、内容 の変更はできません。</li> </ul>
	• <b>ダウンロード実行者:</b> ダウンロード実行者は、ファイルをダウンロードして自分のコンピュータに保存することもできます。
	<ul> <li>コントリビュータ: コントリビュータは、テンプレート詳細およびテンプレート・ファイルの編集、新規ファイルのアップロード、ファイルの削除およびテンプレート自体の削除を行うこともできます。</li> </ul>
	<ul> <li>マネージャ:マネージャはユーザーの追加とそのロールの割当てを行うこと もできます。テンプレートの所有者(作成者)は自動的にマネージャ・ロールを 割り当てられます。</li> </ul>
テンプレート詳細	テンプレート詳細を表示するには、テンプレートを選択して、右クリック・メニ
の 衣 示 ま た は 編 集	ューの「詳細」を選択するか、アクション・バーの をクリックします。テン プレート詳細には、テンプレート名、作成者、サポート・ファイル、説明、テー マ、カスタム・コンポーネントおよびサムネイルが含まれます。適切なテンプレ ート・ロールを持っている場合、詳細を編集できます。テンプレート詳細の変更 を参照してください。
テンプレート・ポリ シーの表示または 編集(ガバナンスが 有効な場合)	ガバナンスが有効で、かつユーザーがサイト管理者である場合、必要な承認、最 小限のセキュリティ、サイトの有効期限、ローカライズなどのテンプレート・ポ リシーを表示できます。テンプレートを選択して、右クリック・メニューの「詳
"日刈'み物口 <b>】</b>	<b>細」</b> を選択するか、アクション・バーの <b>し</b> をクリックして、 <b>「ポリシー」</b> タブを 選択します。テンプレート・ポリシーの変更を参照してください。



タスク	説明
サイト作成に対するテンプレートの有効化と、サイトを	ガバナンスが有効で、かつユーザーがサイト管理者である場合、サイト作成でテ ンプレートを使用可能にし、サイトを作成するためのテンプレートを使用できる ユーザーを管理できます。アクティブ(サイト作成で使用可能)なテンプレートの
YF成するためのテ ンプレートを使用	横にはのが表示されます。
できるユーザーの 管理 <b>(</b> ガバナンスが	ステータスまたはオーディエンスを変更するには、テンプレートを選択して、右
有効な場合)	クリック・メニューの <b>「詳細」</b> を選択するか、アクション・バーの <b>Lo</b> をクリックします。
	<ul> <li>サイト作成でテンプレートを使用可能にするには、ステータス・スイッチを クリックしてそれをアクティブ化します。</li> </ul>
	<ul> <li>サイトを作成するためのテンプレートを使用できるユーザーを変更するには、「オーディエンス」タブをクリックします。テンプレートのステータスまたはオーディエンスの変更を参照してください。</li> </ul>

# テンプレートのエクスポートとインポート

テンプレートをエクスポートしオフラインで変更して、その後新規テンプレートとし てインポートするか、既存テンプレートを置き換えることができます。また、テンプ レートをエクスポートして別の Oracle Content Management インスタンスに移動し、 そこでインポートすることもできます

#### テンプレートのエクスポート

テンプレートをエクスポートする際、基本的に、テンプレートを Oracle Content Management のフォルダに単一の.zip ファイルとしてコピーします。テンプレート・パ ッケージは、フォルダから直接ダウンロードして解凍し、個別のファイルを操作でき ます。テンプレート・ファイルの操作が完了したら、テンプレート・パッケージを含 む.zip ファイルを作成し、サイトにインポートして、元のテンプレートを上書きする か、新規に作成します。

/ノート:

テンプレートをエクスポートした場合、テンプレートの共有情報は含められ ません。

テンプレートをエクスポートするには:

- 1. 「開発者」をクリックし、「すべてのテンプレートの表示」をクリックします。
- テンプレートを選択し、メニュー・バーまたは右クリック・メニューから「エクスポート」を選択します。
- 3. 既存のフォルダに移動するか、「**作成」**をクリックして名前およびオプションの説 明を指定し、新しいフォルダを作成します。
- 4. チェック・ボックスを選択して「OK」をクリックし、フォルダを選択します。

テンプレートのインポート

### / ノート:

テンプレートを別のサーバーにインポートした場合、デフォルト・サイトの一部の リンクが新しいサーバー・コンテキストでは無効になることがあります。コンテン ツをサイトに直接コピーするのではなく、サイトでイメージまたは他のコンテンツ への参照リンクを使用している場合、そのコンテンツは新しいサーバーでは使用で きません。コンテンツを新しいサーバーにコピーしても、コンテンツの内部 ID が異 なるため、リンクは無効になります。テンプレートをインポートする際、無効な参 照リンクが含まれているページが通知されます。

テンプレート・パッケージをインポートするには:

- 1. 「開発者」をクリックし、「すべてのテンプレートの表示」をクリックします。
- 2. 「作成」をクリックして、「テンプレート・パッケージのインポート」を選択します。
- テンプレート・パッケージがアップロード済の場合、テンプレート・パッケージが含まれ るフォルダに移動して、フォルダを開きます。まだテンプレート・パッケージをアップロ ードしていない場合、それをアップロードするフォルダに移動するか、新しいフォルダを 作成します。「アップロード」をクリックし、テンプレート・パッケージを検索および選 択して「開く」をクリックします。
- 4. テンプレートを使用するには、テンプレート・ファイル名の横にあるチェック・ボックス を選択し、「OK」をクリックします。テンプレート、それに関連付けられたテーマおよび カスタム・コンポーネントに対して新しいフォルダが作成されます。テンプレート、テー マまたはカスタム・コンポーネントの名前または ID が存在する場合、競合を解決するよ う求められます。状況に応じて、新しいテンプレート、テーマまたはコンポーネントを作 成するか、インポートしたバージョンで既存のアイテムを上書きできます。

#### 特定のリポジトリへのテンプレートのインポート

リポジトリを選択し、テンプレート・パッケージをこれに限定してインポートできます。

- 1. 「コンテンツ」をクリックし、メニューから「リポジトリ」を選択します。
- テンプレート・パッケージのインポート先のリポジトリを選択し、「コンテンツのインポート」をクリックします。
- テンプレート・パッケージがアップロード済の場合、テンプレート・パッケージが含まれ るフォルダに移動して、フォルダを開きます。まだテンプレート・パッケージをアップロ ードしていない場合、それをアップロードするフォルダに移動するか、新しいフォルダを 作成します。「アップロード」をクリックし、テンプレート・パッケージを検索して選択 し、「OK」をクリックします。
- リポジトリ内にすでに存在するアセットを新しいリビジョンで更新するか、アセットの複 製を作成するかを選択し、「インポート」をクリックします。
- 5. 「詳細」をクリックし、インポートされたアセットおよびコンテンツ・タイプのリストを 表示します。終了する場合は「OK」をクリックします。



### テーマの理解

テーマでは、色スキーム、フォント・サイズ、フォント・タイプ、ページ背景など、 サイトの一般的なルック・アンド・フィール(全体スタイル)を定義します。テーマは、 サイト内のページ間に視覚的な統一感を与えます。デザインを調整し、コンテンツを 追加して、スタイル、ブランドおよびビジョンを売り込むサイトを作成します。

H	lome	Products <del>-</del>	About	Contact	Privacy Policy	
Slot	slot-con	itent 들				
					_	
L						f У in 🕫 🗖

テーマの内容は次のとおりです。

- ページ・レイアウト
- カスケード・スタイル・シート(CSS)
- スタイル・シートのバリエーション
- 構成ファイル
- サイト・ナビゲーションを定義するバックグラウンド・コード

Web サイトごとに1つのテーマを使用します。テンプレートからサイトを作成する と、テーマがテンプレートから継承されます。サイトのテーマはいつでも変更できま す。

Oracle Content Management には、使用開始時に使用できるテーマを含む多数のテン プレートが用意されています。新しいテーマを作成するには、既存のテーマをコピー します。テーマ・ファイルをダウンロードおよび変更したり、デスクトップ・アプリ ケーションを使用してローカル・システムで行った変更を同期したりできます。テー マを作成する他の方法の詳細は、「テーマの開発」を参照してください。

開発者は、コンポーネントのある一部を使用するテーマも作成できます。このような テーマのいずれかを使用してサイトを作成する場合、そのテーマで使用可能なコンポ ーネントのみが表示されます。そのため、選択したテーマでコンポーネントが有効で あるかどうかを心配する必要はありません。シード・コンポーネントも引き続きテー マ・コンポーネントと一緒に使用できます。

サイトで新規の非公開テーマを使用している場合、サイトを初めてオンラインにした 際に、テーマはサイトとともに自動的に公開されます。テーマに変更を加え、オンラ イン・サイトを更新して変更を表示する場合は、テーマを明示的に公開する必要があ ります。テーマを明示的に公開できるのは、テーマ所有者またはマネージャ権限を持 つユーザーのみです。



### 🖍 ノート:

テーマへの変更を公開すると、テーマを使用するすべてのオンライン・サイトに変 更が反映されます。たとえば、テーマのデフォルト・フォントを変更してテーマを 公開すると、そのテーマを使用するすべてのサイトで新しいデフォルト・フォント が使用されます。

## テーマの管理

テーマ・フォルダは、他のフォルダと同じようにコピー、名前変更および削除できますが、 テーマは複数のサイトによって共有されることがあるため、特別な配慮が必要です。

「開発者」をクリックしてから「すべてのテーマの表示」をクリックすると、すべてのテーマを表示できます。「テーマ」ページのメニュー・バーまたは右クリック・メニューのオプションを使用して、これらのタスクを実行します。

タスク	説明
新しいテーマの作 成	新しいテーマを作成するには、既存のテーマをコピーします。テーマを作成する 他の方法の詳細は、「テーマの開発」を参照してください。
テーマのコピー	既存のテーマをコピーし、コピーに変更を加えることによって、新しいテーマを 作成できます。

🖊 ノート:

テーマをコピーする場合、テーマに関連付けられた 共有情報はコピーされません。

コピーするテーマを選択して、メニュー・バーまたは右クリック・メニューから 「コピー」を選択します。別のテーマで使用されていない名前を入力します。文 字、数字、アンダースコア()およびハイフン(-)を使用できます。スペースを入力 した場合、ハイフンに自動的に置換されます。「コピー」をクリックします。 テンプレート、テーマ、コンポーネント、サイトまたはサイト・ページについて、 authsite、content、pages、scstemplate_*、_comps、_components、 _compsdelivery、_idcservice、_sitescloud、_sitesclouddelivery、_themes、 _themesdelivery といった名前を使用しないでください。documents および sites は、サイト・ページには使用できますが、テンプレート、テーマ、コンポー ネントまたはサイトには使用しないでください。

タスク	説明
テーマの共有	管理者が共有を有効にしている場合、テーマの所有者またはマネージャは、自分 のテーマを他のユーザーと共有できます。テーマを共有する場合、ユーザーがテ ーマで実行できることを定義するロールを割り当てます。
	★ ノート: 誰かがユーザーとテンプレートを共有している場合、そのテンプレートに対してどのロールを付与されていても、関連する任意のテーマに対するダウンロード実行者ロールが自動的に付与され、テンプレートからサイトを作成する場合にユーザーは必ずそのテーマを使用できます。
	<ul> <li>共有するテーマを右クリックして「共有」を選択し、「メンバーの追加」をクリックします。1つ以上のユーザー名または電子メール・アドレスを入力し、次のロールのいずれかを割り当てます。</li> <li>参照者:参照者はテーマのフォルダとファイルを表示できますが、内容の変更はできません。</li> <li>ダウンロード実行者:ダウンロード実行者は、ファイルをダウンロードして自分のコンピュータに保存することもできます。</li> </ul>
	<ul> <li>コントリビュータ: コントリビュータは、テーマの編集、新規ファイルのアップロード、ファイルの削除およびテーマ自体の削除を行うこともできます。</li> <li>マネージャ: マネージャはユーザーの追加とそのロールの割当てを行うこともできます。テーマの作成者(所有者)は自動的にマネージャ・ロールを割り当てられます。</li> </ul>
テーマの編集	Web 開発者は、個々のテーマ・ファイルをダウンロードおよび変更したり、デス クトップ・アプリケーションを使用してローカル・システムで行った変更を同期 したりできます。 また、関連するテーマとカスタム・コンポーネントを含むテンプレート・パッケ ージをエクスポートして、それを適切な開発環境でオフラインのまま操作できま す。テーマの開発を参照してください。 テーマに変更を加えた場合、そのテーマを使用するサイトに変更が反映されるよ
テーマの公開	<ul> <li>うに、テーマを公開する必要があります。</li> <li>サイトで新規の非公開テーマを使用している場合、サイトを初めてオンラインにした際に、テーマはサイトとともに自動的に公開されます。テーマに変更を加え、オンライン・サイトを更新して変更を表示する場合は、テーマを明示的に公開する必要があります。所有者である場合、またはマネージャ・ロールを割り当てられている場合、テーマを公開できます。</li> <li>リストのテーマを選択してメニュー・バーの「公開」(以前に公開されているテー</li> </ul>
	マの場合は <b>「再公開」)</b> をクリックします。公開済アイコン ⁽¹⁾ がリストのテーマの 横に追加されます。テーマの公開を参照してください。

タスク	説明
テーマの削除また は復元	適切な権限を持っている場合、テーマ・フォルダとそのコンテンツを削除できま す。テーマを削除すると、テーマ・フォルダとその関連フォルダおよびファイル はごみ箱に移動されます。
	テーマを削除または復元できるのは、テーマを作成した(テーマの所有者である)場 合か、他のユーザーがあなたとテーマを共有していて、あなたにコントリビュー タまたはマネージャ・ロールを付与している場合です。
	✓ ノート:
	サイトで使用されているテーマは削除できません。
	テーマを削除するには、削除するテーマを右クリックして <b>「削除」</b> を選択します。 テーマをごみ箱に移動するよう求められます。削除したテーマは、次までごみ箱 にあります。 • テーマを復元します。 • テーマを復元します。

- テーマを完全に削除します。
- ごみ箱の割当て制限に達しました。
- ごみ箱は、サービス管理者が設定した間隔に基づいて自動的に空になります。
   デフォルト値は 90 日です。

テーマを復元するには、「テーマ」ページのメニュー・バーにある「ごみ箱」をクリックしてからリストのテーマを右クリックし、「復元」を選択します。

## テーマの公開

テーマでは、サイトの一般的なルック・アンド・フィールを定義します。テーマを更新して、 テーマを使用するサイトの外観を変更できます。

サイトで新規の非公開テーマを使用している場合、サイトを初めてオンラインにした際に、 テーマはサイトとともに自動的に公開されます。テーマに変更を加え、オンライン・サイト を更新して変更を表示する場合は、テーマを明示的に公開する必要があります。テーマへの 変更を公開するには、テーマの所有者であるか、マネージャ・ロールを持っている必要があ ります。

/ ノート:

テーマへの変更を公開すると、テーマを使用するすべてのオンライン・サイトに変 更が反映されます。テーマへの更新を公開する前に、オフラインで変更内容をテス トして、関連付けられているサイトへの影響を理解していることを確認してくださ い。

テーマを公開するには:

- 1. 「開発者」をクリックし、「すべてのテーマの表示」をクリックします。
- 2. テーマのリストから既存のテーマを選択します。
- 3. メニュー・バーまたは右クリック・メニューから「公開」をクリックします。



**4. 「確認して続行」**をクリックし、「**OK」**をクリックします。公開済アイコンへがリストのテーマの横に追加されます。



# 7 カスタム・コンポーネントおよびレイアウトの 管理

開発者は、カスタム・コンポーネントおよびレイアウトを作成および管理できます。サイト・ コントリビュータは、サードパーティのコンポーネント(アプリケーション)およびコンポーネ ント・グループを登録できます。

カスタム・コンポーネントには、エディタで作成するコンポーネント・グループと、この項 で説明するオプションを使用して作成するローカルおよびリモート・コンポーネントが含ま れます。カスタム・レイアウトには、ページ上のスロット内のコンポーネントを配置するた めの*セクション・レイアウト*、およびコンテンツ・アイテム内のフィールドを配置するため の*コンテンツ・レイアウト*が含まれます。

- カスタム・コンポーネントの理解
- レイアウトの理解
- カスタム・コンポーネントおよびレイアウトの使用
- リモート・コンポーネントの登録
- ローカル・コンポーネント、レイアウト、コンテンツ・フィールド・エディタまたはコン テンツ・フォームの作成
- コンポーネントまたはレイアウトのエクスポートまたはインポート

サイトでどのようにコンポーネントを使用するかについて学習するには、ページ・コンテン ツの配置を参照してください。

個々のコンポーネントの使用方法の詳細は、組込みコンポーネントの使用を参照してください。

独自のコンポーネントの作成方法の詳細は、「コンポーネントの開発」を参照してください。

## カスタム・コンポーネントの理解

コンポーネントは、Web ページの個々の部分です。Web ページを参照すると、何が表示されますか。おそらく、いくつかのタイトル、テキストの段落、およびサイトの他のページへのリンクが表示されているでしょう。イメージ、ボタン、区切り、マップおよびギャラリが表示されることもあります。これらの各アイテムがコンポーネントです。

コンポーネントをページに追加するには、

なたクリックして、使用するコンポーネントのタイプを選択します。

パネルからコンポーネントをドラッグし、ページのスロットにドロップします。これで終了 です。タイトル、段落、イメージおよびその他のコンポーネントをページの必要な場所にド ラッグ・アンド・ドロップします。

リモート・コンポーネント(アプリケーション)を簡単に登録して組み込み、コンポーネント・ マネージャのオプションを使用して独自のコンポーネントを作成することもできます。「開発 者」をクリックし、「すべてのコンポーネントの表示」をクリックします。



「作成」をクリックして、関連付けられたオプションを選択し、新しいローカル・コン ポーネントを作成するか、リモート・コンポーネントを登録します。このようにして 作成および共有したコンポーネントは、エディタ内のカスタム・コンポーネント・パ ネルにリストされます。

独自のコンポーネントの作成方法の詳細は、「コンポーネントの開発」を参照してくだ さい。

#### リモート・コンポーネント

使用する予定のサードパーティ・コンポーネント(アプリケーション)がある場合、それ を登録すればサイトで使用できます。これは非常に簡単です。

リモート・コンポーネントを登録する際、リモート・コンポーネント自体の URL、および Web 作成者がリモート・コンポーネントに指定できる設定用に 2 番目の URL を 指定します。

リモート・コンポーネントを登録したら、他のユーザーとこれを共有できます。所有 している、または共有している登録済リモート・コンポーネントは、すべてサイト・ ビルダーの「カスタム・コンポーネント」パネルにリストされます。

他のコンポーネントと同じ方法で、エディタ内でリモート・コンポーネントのプロパ

ティを変更できます。メニュー・アイコン = をクリックし、「設定」を選択します。 「カスタム」ボタンを使用すると、リモート・コンポーネントを登録したときに指定し た設定 URL が開きます。

/ ノート:

リモート・コンポーネントは、Oracle Content Management をホスティング しているサーバー以外のサーバーでホストされているため、セキュリティの 観点から、インライン・フレームで囲まれます(iframe 要素を使用)。すべて のリモート・コンポーネントがインライン・フレームで囲まれるわけではあ りません。囲むことができるかどうかを知るには、プロバイダに確認してく ださい。

ローカル・コンポーネント

開発者は、Oracle Content Management で提供されるものと同じ機能にアクセス可能 なコンポーネントを作成できます。コンポーネントをページに直接挿入するか、iframe 要素を使用してインライン・フレームで囲むことができます。

「作成」をクリックしてローカル・コンポーネントを作成する際、完全に機能するサン プル・コンポーネントが、指定した名前および一意の識別子でコンポーネント・リス トに追加されます。開発者は、サンプルを変更し、独自のソリューションを作成でき ます。

独自のコンポーネントの作成方法の詳細は、「コンポーネントの開発」を参照してくだ さい。



### レイアウトの理解

ページ・レイアウトにより、ページ上のスロットおよびコンテンツを配置します。セクション・レイアウトにより、スロット内のコンテンツを配置します。コンテンツ・レイアウトにより、コンテンツ・アイテム内のフィールドを配置します。

一般に、レイアウトにより、コンテンツのプレゼンテーションを指定しますが、コンテンツ 自体は指定しません。コンテンツをプレゼンテーションから分離すると、同じコンテンツを 別の方法で提示したり、コンテンツに触れずにプレゼンテーションを変更することが容易に なります。

#### ページ・レイアウト

サイトにページを追加する場合、そのページに使用するレイアウトを選択します。各レイア ウトにはページ上の領域(スロットと呼ばれる)があり、ここでコントリビュータがコンテンツ をドラッグ・アンド・ドロップできます。ページ・レイアウトにより、ページ上のスロット の数と位置を定義します。レイアウトには、ページ上に事前定義および配置されたコンテン ツを含めることもできます。このコンテンツは、会社のロゴのように、静的であり、編集が 不可能な場合があります。また、コントリビュータが変更できるが位置や外観は変更できな い見出しのような、最小限の編集が可能なコンテンツもあります。

ページ・レイアウトはテーマに格納されます。テーマには、1つ以上のページ・レイアウトを 含めることができます。開発者は、既存のテーマをコピーおよび変更して新しいテーマを作 成できます。レイアウトの開発を参照してください。



#### セクション・レイアウト

セクション・レイアウトの場合、セクション・レイアウトに追加されたコンテンツが自動的 に編成されるため、コントリビュータがページ上でコンテンツのフォーマット設定に時間を 費やさずにコンテンツを追加しやすくなります。たとえば、セクション・レイアウトでは、 コンテンツを複数の列または垂直リストに自動的に編成できます。サイト・コントリビュー タは、1つ以上のセクション・レイアウトを1つのスロットに追加してコンテンツを編成でき ます。



Slot slot2					
Section Layout Three Columns 📃					
o" Image	o" Image	o" Image			

開発者は、提供されているレイアウトに加えて、別のセクション・レイアウトを作成 することにより、レイアウトに関する特定の問題を解決したり、コントリビュータの ためにオーサリングを簡素化できます。テーマ・デザイナは、複数のセクション・レ イアウトをページ・レイアウト内の1つのスロットに組み込むこともできます。レイ アウトの開発を参照してください。

#### コンテンツ・レイアウト

エンタープライズ・ユーザーの場合、用意されているコンテンツ・タイプとレイアウ トに基づいてコンテンツ・アイテムを作成および使用できます。この方法で構築され たコンテンツを使用すると、コントリビュータが、サイト・ビルダーの外部でコンテ ンツ・アイテムのコンテンツを結集できるようになります。コンテンツ・タイプに関 連付けられた複数のコンテンツ・レイアウトを使用すると、サイト・デザイナが、結 集したコンテンツに触れずにコンテンツ・アイテムを様々なコンテキストで表示でき るようになります。開発者は、用意されているデフォルトのレイアウトから新しいレ イアウトを作成できます。「レイアウトの開発」を参照してください。




### カスタム・コンポーネントおよびレイアウトの使用

開発者は、カスタム・コンポーネントおよびレイアウトを作成および管理できます。サイト・ コントリビュータは、サードパーティのコンポーネント(アプリケーション)およびコンポーネ ント・グループを登録できます。

カスタム・コンポーネントには、エディタで作成するコンポーネント・グループと、次に説 明するオプションを使用して作成するローカルおよびリモート・コンポーネントが含まれま す。使用する予定のサードパーティ・アプリケーション(リモート・コンポーネント)がある場 合、それを登録すればサイトで使用できます。開発者は、Oracle Content Management で提 供されるものと同じ機能にアクセス可能なローカル・コンポーネントを作成することもでき ます。

カスタム・レイアウトには、ページ上のスロット内のコンポーネントを配置するための*セク* ション・レイアウト、およびコンテンツ・アイテム内のフィールドを配置するための*コンテ* ンツ・レイアウトが含まれます。

「コンポーネント」ページのメニュー・バーまたは右クリック・メニューのオプションを使用 して、次のタスクを実行します。

タスク	説明
リモート・コンポー ネントの登録	リモート・コンポーネントを登録する際、リモート・コンポーネント自体の URL、 および Web 作成者がリモート・コンポーネントに指定できる設定用に 2 番目の URL を指定します。
	コンポーネントを作成する前に、管理者が「 <b>作成」</b> メニューのオプションを有効 にする必要があります。コンポーネント・ページに「 <b>作成」</b> メニューがない場合 は、管理者に連絡してください。
	「作成」をクリックして、関連付けられたオプションを選択し、リモート・コンポ ーネントを登録します。リモート・コンポーネントの登録を参照してください。
ローカル・コンポー ネントまたはレイ アウトの作成	ローカル・コンポーネントまたはレイアウトを作成する場合、完全に機能するサ ンプル・コンポーネントまたはレイアウトが提供され、これを独自のコンポーネ ントまたはレイアウトの作成の基準として使用できます。
	コンポーネントまたはレイアウトを作成する前に、管理者が <b>「作成」</b> メニューの オプションを有効にする必要があります。コンポーネント・ページに <b>「作成」</b> メ ニューがない場合は、管理者に連絡してください。
	「作成」をクリックし、関連付けられたオプションを選択して、新しいローカル・ コンポーネントまたはレイアウトを作成します。ローカル・コンポーネント、レ イアウト、コンテンツ・フィールド・エディタまたはコンテンツ・フォームの作 成を参照してください。
	コンポーネントを作成する他の方法の詳細は、「コンポーネントの開発」を参照し てください。



タスク	説明
コンポーネントま たはレイアウトの コピー	既存のコンポーネントまたはレイアウトをコピーし、コピーに変更を加えること によって、新しいコンポーネントまたはレイアウトを作成できます。
	✓ ノート: コンポーネントまたはレイアウトを作成または登録 した後にその名前を変更することはできません。コ ンポーネントまたはレイアウトをコピーして、その コピーに別の名前を指定することはできます。その 他すべての登録情報(リモート・コンポーネントのキ ー値を含む)は保持されます。共有情報は、登録情報 から独立しており、コピーされません。
	コピーするコンポーネントまたはレイアウトを右クリックし、 <b>「コピー」</b> を選択し ます。名前を入力し、 <b>「コピー」</b> をクリックします。文字、数字、アンダースコア ()およびハイフン(-)を使用できます。スペースを入力した場合、ハイフンに自動 的に置換されます。
コンテンツ・フィー ルド・エディタの作 成	コンテンツ・タイプを作成する際にプロモートおよび使用できるコンテンツ・フ ィールド・エディタを作成し、データ・フィールドの外観を制御できます。メデ ィアおよびリファレンスを除くすべてのデータ型がサポートされています。コン テンツ・タイプを作成する際にコンポーネントを使用できるようにするには、プ ロモートする必要があります。
コンポーネントま たはレイアウトの 共有	コンポーネントまたはレイアウトを他の Oracle Content Management ユーザー と共有できます。コンポーネントまたはレイアウトを共有できるのは、所有者で ある場合か、マネージャ・ロールを割り当てられている場合です。コンポーネン トまたはレイアウトを共有する場合、ユーザーがコンポーネントまたはレイアウ トで実行できることを定義するロールをユーザーに割り当てる必要があります。 共有するコンポーネントまたはレイアウトを右クリックし、「共有」を選択して、 「メンバーの追加」をクリックします。
	<ul> <li>1 つ以上のユーザー名または電子メール・アドレスを入力し、次のロールのいずれかを割り当てます。</li> <li>参照者: 参照者は、カスタム・コンポーネントおよびレイアウトのリスト内のコンポーネントまたはレイアウトをエディタで表示および使用できます。また、テーマのフォルダとファイルを表示できますが、内容の変更はできません。</li> </ul>
	<ul> <li>ダウンロード実行者: ダウンロード実行者は、ファイルをダウンロードして自分のコンピュータに保存することもできます。</li> <li>コントリビュータ: コントリビュータは、コンポーネントまたはレイアウトの編集、新規ファイルのアップロード、ファイルの削除およびコンポーネントまたはレイアウト自体の削除を行うこともできます(サイト内で使用中でない場合)。</li> </ul>
	<ul> <li>マネージャ:マネージャはユーザーの追加とそのロールの割当てを行うこともできます。コンポーネントまたはレイアウトの作成者(所有者)にはマネージャ・ロールが自動的に割り当てられます。</li> </ul>

タスク	説明
コンポーネントま たはレイアウトの 編集	Web 開発者は、個々のコンポーネントまたはレイアウト・ファイルをダウンロー ドして変更したり、デスクトップ・アプリケーションを使用して、ローカル・シ ステムで加えた変更を同期できます。
	また、コンポーネントまたはレイアウトを個々にエクスポートするか、カスタム・ コンポーネントやセクション・レイアウトを含むテンプレート・パッケージの一 部としてエクスポートし、任意の開発環境でオフラインでそれを操作することも できます。
	コンポーネントまたはレイアウトに変更を加えた場合、そのコンポーネントまた はレイアウトを使用するサイトに変更内容が反映されるように、コンポーネント またはレイアウトを公開する必要があります。
コンポーネントま たはレイアウトの エクスポートまた はインポート	<ul> <li>コンポーネントまたはレイアウトをエクスポートしてオフラインで変更し、その 後新規コンポーネントまたはレイアウトとしてインポートするか、既存のコンポ ーネントまたはレイアウトを置き換えることができます。また、コンポーネント またはレイアウトをエクスポートして別のインスタンスに移動し、そこでインポ ートすることもできます。コンポーネントまたはレイアウトを個々にエクスポー トするか、カスタム・コンポーネントおよびレイアウトを含むテンプレート・パ ッケージの一部としてエクスポートできます。</li> <li>コンポーネントまたはレイアウトをエクスポートする際、コンポーネントまたは レイアウトを1つの.zip ファイルとしてフォルダにコピーします。そのコンポー ネントまたはレイアウト・パッケージをフォルダから直接ダウンロードして解凍 し、個々のファイルを操作できます。コンポーネントまたはレイアウト・ファイ ルの操作が完了したら、コンポーネントまたはレイアウト・パッケージを含む.zip ファイルを作成し、それをインポートして、元のコンポーネントまたはレイアウ トを上書きするか、新しいものを作成します。</li> <li>コンポーネントまたはレイアウトのエクスポートまたはインポートを参照してく ださい。</li> </ul>
コンポーネントま たはレイアウトの 公開	サイトで新規の非公開コンポーネントまたはレイアウトを使用している場合、サ イトを初めてオンラインにした際に、コンポーネントまたはレイアウトがサイト とともに自動的に公開されます。コンポーネントまたはレイアウトに変更を加 え、オンライン・サイトを更新して変更内容を表示する場合は、コンポーネント またはレイアウトを明示的に公開する必要があります。コンポーネントまたはレ イアウトを公開できるのは、所有者である場合か、マネージャ・ロールを割り当 てられている場合です。 リスト内のコンポーネントまたはレイアウトを選択し、メニュー・バーの「公開」 (以前に公開されているコンポーネントまたはレイアウトの場合は「再公開」)をク リックします。公開済アイコン がリストのコンポーネントまたはレイアウト の横に表示されます。

#### タスク 説明

コンポーネントま 適切な権限を持っている場合、コンポーネントまたはレイアウト・フォルダとそ たはレイアウトの のコンテンツを削除できます。コンポーネントまたはレイアウトを削除すると、 削除または復元 コンポーネントまたはレイアウト・フォルダとそのすべての関連フォルダおよび ファイルがごみ箱に移動されます。

> コンポーネントまたはレイアウトを削除または復元できるのは、コンポーネント またはレイアウトを作成した(コンポーネントまたはレイアウトの所有者である) 場合か、他のユーザーがコンポーネントまたはレイアウトをあなたと共有し、コ ントリビュータまたはマネージャ・ロールをあなたに付与している場合です。

コンポーネントまたはレイアウトがサイトまたは更 新(ごみ箱内のサイトまたは更新を含む)で使用され ている場合、削除することはできません。

コンポーネントまたはレイアウトを削除するには、削除するコンポーネントまた はレイアウトを右クリックし、「削除」を選択します。コンポーネントまたはレイ アウトをごみ箱に移動するよう求められます。削除したコンポーネントまたはレ イアウトは、次のときまでごみ箱にあります。

- コンポーネントまたはレイアウトを復元する。
- コンポーネントまたはレイアウトを完全に削除する。
- ・ ごみ箱の割当て制限に達しました。
- ごみ箱は、サービス管理者が設定した間隔に基づいて自動的に空になります。
   デフォルト値は 90 日です。

コンポーネントまたはレイアウトを復元するには、「ごみ箱」をクリックし、リス ト内のコンポーネントまたはレイアウトを右クリックして、「復元」を選択しま す。

### リモート・コンポーネントの登録

サイトでリモート・コンポーネントを使用するには、最初にそれを Oracle Content Management に登録する必要があります。

サードパーティのリモート・コンポーネントおよび自分が開発したリモート・コンポ ーネントを登録できます。

┛ ノート: リモート・コンポーネントを登録する前に、管理者が「**作成」**メニューのオ プションを有効にする必要があります。コンポーネント・ページに「**作成」** メニューがない場合は、管理者に連絡してください。

Oracle Content Management で使用するためにリモート・コンポーネントを登録する には:

- 1. 「開発者」をクリックし、「すべてのコンポーネントの表示」をクリックします。
- 2. 「作成」をクリックして、「リモート・コンポーネントの登録」を選択します。



第7章 ローカル・コンポーネント、レイアウト、コンテンツ・フィールド・エディタまたはコンテンツ・フォームの 作成

- **3.** 「リモート・コンポーネントの登録」ウィンドウで、次のような情報を入力または選択します。
  - **名前**: ユーザーが参照するコンポーネントの名前。
  - 説明: ユーザーが参照するコンポーネントの説明。
  - コンポーネント URL: ページにコンポーネント・コンテンツをレンダリングするために、iframeで使用されるエンドポイント。HTTPS である必要があります。
  - 設定 URL: ページに追加されるリモート・コンポーネントの設定をレンダリングする ために、iframe で使用されるエンドポイント。HTTPS である必要があります。
  - 設定幅: コンポーネント設定パネルのデフォルトの幅を設定します(ピクセル単位)。
  - 設定高さ: コンポーネント設定パネルのデフォルトの高さを設定します(ピクセル単位)。
  - キー: リモート・コンポーネントに関連付けられ、コンポーネントのプロビジョニン グ時に署名済ハッシュ・トークンの作成に使用される 192 ビットの AES キー。これ は暗号化して、コンポーネント設定のセキュアな読取りおよび書込みを確保するため に使用されます。
- 4. 「登録」をクリックします。

リモート・コンポーネントが作成されると、名前がコンポーネントのリストに表示されま す。コンポーネントのリストでコンポーネント名をクリックすると、コンポーネントを登 録するのに使用したファイルを検索できます。

コンポーネントの登録情報は同じ Oracle Content Management インスタンスに作成されたサ イトで使用されるカタログに格納されますが、コンポーネントはリモート・サービスのまま となります。

コンポーネント所有者として、コンポーネント・アイコンは、エディタ内の「カスタム・コ ンポーネント」パネルに、コンポーネントに割り当てた名前で追加されます。コンポーネン トは他のユーザーと共有でき、そのユーザーのエディタ内の「カスタム・コンポーネント」 パネルにコンポーネントが表示されます。

## ローカル・コンポーネント、レイアウト、コンテン ツ・フィールド・エディタまたはコンテンツ・フォ ームの作成

開発者がローカル・コンポーネントまたはレイアウトを作成する場合、完全に機能するサン プルが提供され、これを独自のコンポーネントまたはレイアウトに変更できます。

コンポーネントまたはレイアウトを作成する前に、管理者が「**作成」**メニューのオプション を有効にする必要があります。コンポーネント・ページに「**作成」**メニューがない場合は、 管理者に連絡してください。

開発者は、Oracle Content Management で提供されるものと同じ機能にアクセス可能なコン ポーネントを作成できます。

サンプル・ローカル・コンポーネントまたはレイアウトを作成するには:

- 1. 「開発者」をクリックし、「すべてのコンポーネントの表示」をクリックします。
- 2. 「作成」をクリックし、作成するコンポーネントまたはレイアウトのタイプを選択します。



第7章 ローカル・コンポーネント、レイアウト、コンテンツ・フィールド・エディタまたはコンテンツ・フォ ームの作成

> 3. コンポーネントまたはレイアウトの名前を入力します。別のコンポーネントまた はレイアウトで使用されている名前は使用できません。

文字、数字、アンダースコア(_)およびハイフン(-)を使用できます。スペースを入力した場合、アンダースコアに自動的に置換されます。

テンプレート、テーマ、コンポーネント、サイトまたはサイト・ページについて、 authsite、content、pages、scstemplate_*、_comps、_components、 _compsdelivery、_idcservice、_sitescloud、_sitesclouddelivery、_themes、 _themesdelivery といった名前を使用しないでください。documents および sites は、サイト・ページには使用できますが、テンプレート、テーマ、コンポーネント またはサイトには使用しないでください。

- 4. オプションで、コンポーネントまたはレイアウトの説明を入力します。
- コンポーネント・タイプの場合、コンポーネントをページに直接挿入するには「デフォルト」を、(iframe 要素を使用して)コンポーネントをインライン・フレームで 囲むにはサンドボックスを、またはテンプレートを使用してレンダリングのみを行うには「テンプレート」を選択します。

### /ノート:

「テンプレート」を選択した場合、ローカル・コンポーネントは Knockout ベースではないため、ページに配置されている場合、ネストされたコン ポーネントを使用できません。たとえば、タイトルまたは本文テキスト の編集は、ページに表示されているコンポーネント・テキストを選択す るのではなく、コンポーネントの設定パネルのカスタム設定を通じて実 行する必要があります。

6. 「作成」をクリックします。

進行状況バーに、作成ステータスが表示されます。コンポーネント、レイアウトま たはフォームが作成されると、名前がコンポーネントのリストに表示されます。コ ンポーネントのリストでコンポーネントまたはレイアウトの名前をクリックする と、コンポーネントまたはレイアウトを構成するフォルダとファイルを確認できま す。

- 7. コンポーネントまたはレイアウトに割り当てられているデフォルト・アイコン以外 のアイコンを選択するには:
  - a. リストからコンポーネントまたはレイアウトを選択します。
  - **b. 「プロパティ」**をクリックします。
  - c. 「コンポーネント・ロゴ」タブをクリックします。
  - d. ロゴのギャラリでロゴをクリックし、「完了」をクリックします。

独自のコンポーネントの作成方法の詳細は、「コンポーネントの開発」を参照してくだ さい。

コンポーネント、レイアウト、コンテンツ・フィールド・エディタまたはコンテンツ・ フォームをカスタマイズしたら、他のユーザーと共有したり、次の方法で使用できる ようにプロモートしたりできます。

 カスタム・コンポーネント: サイト・ビルダーを使用する際に、コンポーネント・ アイコンがエディタの「カスタム・コンポーネント」パネルに、コンポーネントに 割り当てた名前で追加されます。

ORACLE

- セクション・レイアウト: サイト・ビルダーを使用する際に、セクション・レイアウト・ アイコンがエディタの「セクション・レイアウト」パネルに、レイアウトに割り当てた名 前で追加されます。
- コンテンツ・レイアウト: コンテンツ管理者は、レイアウトを1つ以上のコンテンツ・タイプに割り当てて、デフォルト・ビューとして使用されるようにすることも、サイト設計者がそのタイプのコンテンツ・アイテムがページにどのように表示されるかを指定する際にサイト・ビルダーで選択できるレイアウトのリストに追加することもできます。
- コンテンツ・フィールド・エディタ: コンテンツ・タイプを作成する際にコンテンツ・フィールド・エディタを使用できるようにするには、これを選択し、「プロモート」をクリックして、確認してから「OK」をクリックします。プロモートされると、コンテンツ・タイプを作成する際、コンテンツ・フィールド・エディタを使用して、メディアおよびリファレンスを除くすべてのデータ・フィールドの外観を制御できます。
- コンテンツ・フォーム: コンテンツ・タイプを作成する際にコンテンツ・フォームを使用できるようにするには、これを選択し、「プロモート」をクリックして、確認してから「OK」をクリックします。プロモートすると、コンテンツ・フォームをあらゆるコンテンツ・タイプやアセット・タイプと使用できるようになります。

## コンポーネントまたはレイアウトのエクスポートま たはインポート

コンポーネントをエクスポートしオフラインで変更して、その後新規コンポーネントとして インポートするか、既存コンポーネントを置き換えることができます。また、コンポーネン トをエクスポートして別の Oracle Content Management インスタンスに移動し、そこでイン ポートすることもできます。

#### エクスポート

コンポーネントをエクスポートする際、基本的に、コンポーネントを Oracle Content Management のフォルダに単一の.zip ファイルとしてコピーします。コンポーネントは、フォ ルダから直接ダウンロードして解凍し、個別のファイルを操作できます。コンポーネント・ ファイルの操作が完了したら、コンポーネントのフォルダおよびファイルを含む.zip ファイル を作成し、それをインポートして、元のコンポーネントを上書きするか、新規に作成します。

/ ノート:

コンポーネントをエクスポートした場合、コンポーネントの共有情報は含められま せん。

コンポーネントをエクスポートするには:

- 1. 「開発者」をクリックし、「すべてのコンポーネントの表示」をクリックします。
- コンポーネントまたはレイアウトを選択し、メニュー・バーまたは右クリック・メニュー から「エクスポート」を選択します。
- 3. フォルダに移動するか、「新規」をクリックして新規フォルダを作成し、名前およびオプ ションの説明を指定して、「作成」をクリックします。フォルダを開くには、フォルダ・ アイコンまたはフォルダ名をクリックします。
- 関連付けられているフォルダのチェック・ボックスをクリックしてフォルダを選択し、 「OK」をクリックします。



コンポーネントまたはレイアウトのパッケージ・ファイルは、そのコンポーネント 名またはレイアウト名および.zip 拡張子を使用して、選択されたフォルダに作成さ れます。

#### インポート

コンポーネント、レイアウトまたはコンテンツ・フィールド・エディタをインポート または作成する前に、管理者が「作成」メニューのオプションを有効にする必要があ ります。コンポーネント・ページに「作成」メニューがない場合は、管理者に連絡し てください。

コンポーネント、レイアウトまたはコンテンツ・フィールド・エディタをインポート するには:

- 1. 「開発者」をクリックし、「すべてのコンポーネントの表示」をクリックします。
- 「作成」をクリックして、「コンポーネントのインポート」を選択します。このオプションを選択し、コンポーネント、セクション・レイアウトまたはコンテンツ・レイアウトをインポートします。
- コンポーネント・パッケージまたはレイアウト・パッケージがアップロード済の場合、コンポーネントまたはレイアウトが含まれるフォルダに移動して、フォルダを開きます。まだパッケージをアップロードしていない場合、コンポーネントまたはレイアウトをアップロードするフォルダに移動するか、新しいフォルダを作成します。「アップロード」をクリックし、コンポーネント・パッケージまたはレイアウト・パッケージを検索して「開く」をクリックします。
- 4. コンポーネント・ファイルまたはレイアウト・ファイル名の横にあるチェック・ボックスを選択し、「OK」をクリックします。新しいコンポーネントまたはレイアウトが作成され、コンポーネント・リストに追加されます。コンポーネントまたはレイアウトの名前またはIDがすでに存在する場合、競合を解決するよう求められます。状況に応じて、新しいコンポーネントまたはレイアウトを作成するか、インポートしたバージョンで既存のコンポーネントまたはレイアウトを上書きできます。



# 8 サイト・ページの操作

ページを作成し、レイアウトを選択して、サイト内のページの配置を決定しましょう。

- ページへのナビゲート
- ページの表示
- ページの追加
- ページの移動
- ページの削除
- ページ設定の変更
- ページ・レイアウトの変更
- 背景またはテーマの変更

### ページへのナビゲート

ページの階層を表示およびナビゲートするには:

- 1. 編集するサイトを開きます。
- Eをクリックして、ページおよびページ・コンテンツを管理および編集するためのオプ ションを表示します。
- 3. をクリックして、ページの最初のレベルをリストします。
- 4. あるページをクリックし、ページを表示します。

矢印(>)が付いているページには、ネストした階層のページがあります。ページをクリッ クすると、ネストしたページが表示されます。 ページ・リスト上部の検索ボックスを使用して、必要なページを検索することもできま す。

5. 特定のページの設定を表示または変更するには、そのページの ^②をクリックします。ペ ージ設定の変更を参照してください。

### ページの表示

サイトのページを表示するには、エディタでサイトを開き、プレビュー、サイズ、方向に関 する様々なオプションを使用します。

エディタを使用中にページの表示方法を変更するには:

- 1. 編集するサイトを開きます。
- 2. ページに移動します。



- エディタでページをプレビューするには、 が「プレビュー」に設定されていることを確認します。これにより、サイトにアクセスしたユーザーに表示されるようにページが表示され、編集時に使用されるビジュアル・エイドは表示されません。
- ベース・サイトに適用される現在の更新を別のブラウザ・ウィンドウでプレビュー するには、
   をクリックします。

/ ノート: サイトのページへのリンクはプレビュー・モードでは機能しません。

5. 特定の画面サイズでデバイスに表示されるページをプレビューするには、上部のメニュー・バーにある画面に合せるをクリックして、リストから画面ディメンションのセットを選択します。選択された画面サイズで最適に使用できるよう、対応するデザインのテーマにより、ページ・コンテンツが自動的に調整されます。



- カスタム・サイズを作成するには、「ウィンドウに合せる」をクリックして、「デバ イス・プリセットの作成」を選択します。必要なフィールドに入力します。カスタ ム・サイズを削除するには、サイズの隣にある■をクリックします。ルーラー をアクティブ化して、ルーラーの空いている部分をクリックすると、別のサイズを 簡単に表示できます。
- 7. 縦と横でページの向きを変更するには、





### ページの追加

サイト内のページはフォルダに似た階層(*サイト・ツリー*)で構造化されています。ページの追加、ページ設定の定義、およびサイト内のページの配置の決定を行うことができます。

ページの追加または外部ページへのリンクの追加を選択できます。

- Web ページ: サイトにあるページとコンテンツ。ページの名前指定とそのコンテンツの 定義、サイト内のページの配置指定、現在のサイトのコンテキストにおけるページの動作 方法の指定を行います。
- 外部リンク:ページは URL で指定された別の場所から参照されます。ページの名前指定、 サイト内の配置指定、および現在のサイトのコンテキストにおけるページの動作方法の指 定を行うことができます。ライブ・サイトからページを使用しているため、リンクしたペ ージのコンテンツを変更できません。

サイト・ツリーで、外部 URL にリンクするページはパージ名の前にWMがあります。

#### Web ページの追加

サイトに Web ページを追加するには:

- 1. 編集するサイトを開きます。
- ページを追加するレベルまたはブランチを選択して「ページの追加」をクリックします。 または、子ページを追加するには、 をクリックしてから なクリックします。「新規 ページ」がサイト・ツリーの下部に追加され、ページに名前を付けて他の設定を指定する よう求められます。
- 3. ページ・タイプとして「Web ページ」を選択します。別のサイトからページを再利用す るかわりにページでコンテンツを管理する場合に、このオプションを使用します。
- ページに名前を付けます。文字、数字、アンダースコア()およびハイフン(-)を使用できます。
   テンプレート、テーマ、コンポーネント、サイトまたはサイト・ページについて、authsite、content、pages、scstemplate_*、_comps、_components、_compsdelivery、_idcservice、_sitescloud、_sitesclouddelivery、_themes、_themesdeliveryといった名前を使用しないでください。documents および sites は、サイト・ページには使用できますが、テンプレート、テーマ、コンポーネントまたはサイトには使用しないでください。



- ページ URL を指定します。デフォルトで、ページ URL にはページ名が使用されます。スペースはハイフンに自動的に置換されます。デフォルト以外の URL を使用するには、「オーバーライド」をクリックして URL で使用するファイル名を追加します。文字、数字、アンダースコア()およびハイフン(-)を使用できます。必ず、ファイル拡張子を含めてください。デフォルトのファイル拡張子は.html です。
- ページ・レイアウトを選択します。これは、コンテンツではなくページの一般的な 構造を定義します。レイアウトには、コンテンツを配置できる1つ以上の名前付き のスロットが含まれます。ページ・レイアウトの数とタイプは、サイトに関連付け られたテーマによって異なります。
- 7. モバイル・ページ・レイアウトを選択します。これは、モバイル・デバイスで表示 する際のページの一般的な構造を定義します。
- ページのタイトルを指定します。これはページ自体には表示されませんが、ページ が表示される際にブラウザのタイトル・バーまたはブラウザ・タブに表示されま す。ページのタイトルを追加するには、エディタを使用してタイトル・コンポーネ ントを追加します。
- 9. オプションの説明を追加します。この説明はページには表示されませんが、他のコントリビュータまたは自身の使用のためにページに関する情報を追加できます。
- 10. 検索エンジンがページのコンテンツを識別するためのカンマ区切りのオプションのキーワードを指定します。キーワードは、ページのテキストに表示されない、またはイメージに表示される用語または概念を識別するのに有用です。キーワードはページに表示されませんが、検索エンジンではそれらを使用してサイトを検索および特定できます。適切な説明、キーワードおよびシノニムによって、Webサイトへのアクセスを増加できます。
- アナリティクスまたはトラッキング用のページ・ヘッダーのスクリプト記述または タグをオプションで追加します。ヘッダーまたはフッターで使用する任意のコー ドを検証して、適切に動作し、サイトにセキュリティ・リスクをもたらさないこと を確認してください。
- 12. アナリティクスまたはトラッキング用のページ・フッターのスクリプト記述または タグをオプションで追加します。デフォルトでフッターに含まれるのは、プライバ シ・ポリシーへのリンクを含むサイトの Cookie の使用に関して表示されるポップ アップのテキストです。開発者の場合、表示されるテキストを編集できます。ヘッ ダーまたはフッターで使用する任意のコードを検証して、適切に動作し、サイトに セキュリティ・リスクをもたらさないことを確認してください。
- 13. オプション:次の使用可能なページ・オプションのいずれかを使用できます。
  - エラー・ページ: エラーによりリクエストされたページを表示できない場合に、 デフォルトのエラー・ページのかわりにこのページを表示します。デフォルト では、このオプションを選択すると、「ナビゲーションからページを非表示」 および検索エンジンのすべてのオプションも選択されます。エラー・ページが ナビゲーションまたは検索エンジンの結果に含まれるようにする場合は、これ らのオプションの選択を解除できます。
  - **ナビゲーションからページを非表示**: サイトの自動生成されるナビゲーション にページが含まれません。
  - 詳細ページ: このページを使用して、選択したコンテンツ・アイテムの詳細な 情報を別のページで表示します。構造化コンテンツを使用できるのは、エンタ ープライズ・ユーザーのみです。デフォルトでは、このオプションを選択する と、「ナビゲーションからページを非表示」も選択されます。ナビゲーション にディテール・ページを含める必要がある場合は、このオプションの選択を解 除できます。

構造化コンテンツ・アイテムがディテール・ページを使用するように構成されている 場合、ユーザーが構造化コンテンツ・アイテムの詳細を表示するためにリンクをクリ ックすると、コンテンツ・アイテムの詳細情報を示すディテール・ページが表示され ます。

- 検索ページ: このページを使用して、検索の結果を表示します。このページは、コン テンツ検索コンポーネントの「リンク」設定で選択できます。コンテンツ検索を参照 してください。デフォルトでは、このオプションを選択すると、「ナビゲーションか らページを非表示」も選択されます。検索ページがナビゲーションに含まれるように する場合は、このオプションの選択を解除できます。
- 14. オプション: 検索エンジン最適化(SEO)オプションを選択します。
  - ページを検索エンジンから隠す: 検索結果に表示されないように、ページのコンテン ツの索引を作成しないよう検索エンジンに通知します。
  - ページ・リンクを検索エンジンから隠す: ページのリンクをフォローしないように、 また結果としてリンク先の索引を作成しないように検索エンジンに通知します。
  - 検索エンジンのページ・キャッシュを無効化: このページをキャッシュしないよう検 索エンジンに通知します。
  - ページの説明を検索エンジンから隠す:検索結果にページの後の(上で指定した)説明 を含めないよう検索エンジンに通知します。
- オプション: このページに、参照セッションで使用する「共有ブラウズ」ボタンを表示す るかどうかを選択します。共有ブラウズ・セッションでは、Oracle Cobrowse Cloud Service を使用して、サイト・ビジターとの画面共有エクスペリエンスを管理します。「共 有ブラウズ統合の有効化」および「ページでの共有ブラウズの使用」を参照してくださ い。
- オプション: サイトの分析追跡スニペットをオーバーライドし、このページに固有のスニペットを追加します。Web 分析追跡用の JavaScript 追跡コードのスニペットを追加すると、Google、Adobe または Oracle Infinity などの外部のアナリティクス・プロバイダとの統合が容易になります。分析追跡の追加を参照してください。
- 17. 現在の更新における保留中のすべての変更を保存するには、「保存」をクリックします。

#### リンク・ページの追加

サイトに外部リンク・ページを追加するには:

- 1. 編集するサイトを開きます。
- 2. ページを追加するサイト・ツリーのレベルのページに移動します。ページは現在のレベル に追加されますが、サイト・ツリーの別の場所に簡単に移動できます。
- 3. 「ページの追加」をクリックします。「新規ページ」がサイト・ツリーの下部に追加され、 ページに名前を付けて他の設定を指定するよう求められます。
- 4. ページ・タイプとして「**外部リンク」**を選択します。
- 5. ページに名前を付けます。文字、数字、アンダースコア()およびハイフン(-)を使用でき ます。この名前は、ページを識別するためにサイト・ツリーで使用されます。スペースは ハイフンに自動的に置換されます。
- 6. ページへの完全な URL を指定します。たとえば、https://www.example.com/ sharedpage.htm です



- 7. ページを新規ブラウザ・ウィンドウまたはタブで開く場合は、「新規ウィンドウで リンクを開く」を選択します。このオプションを選択しないと、ページは現在のウ ィンドウで開かれ、サイト・ページおよびナビゲーションと置き換わります。
- 8. 必要に応じて、任意の使用可能なオプションを選択します。
  - エラー・ページ: エラーによりリクエストされたページを表示できない場合に、 デフォルトのエラー・ページのかわりにこのページを表示します。デフォルト では、このオプションを選択すると、「ナビゲーションからページを非表示」 も選択されます。エラー・ページがナビゲーションに含まれるようにする場合 は、このオプションの選択を解除できます。
  - ナビゲーションからページを非表示: サイトの自動生成されるナビゲーション にページが含まれません。
  - 詳細ページ: このページを使用して、選択したコンテンツ・アイテムの詳細な 情報を別のページで表示します。構造化コンテンツを使用できるのは、エンタ ープライズ・ユーザーのみです。デフォルトでは、このオプションを選択する と、「ナビゲーションからページを非表示」も選択されます。ナビゲーション にディテール・ページを含める必要がある場合は、このオプションの選択を解 除できます。 構造化コンテンツ・アイテムがディテール・ページを使用するように構成され ている場合、ユーザーが構造化コンテンツ・アイテムの詳細を表示するために リンクをクリックすると、コンテンツ・アイテムの詳細情報を示すディテー ル・ページが表示されます。
  - 検索ページ: このページを使用して、検索の結果を表示します。このページは、 コンテンツ検索コンポーネントの「リンク」設定で選択できます。コンテンツ 検索を参照してください。デフォルトでは、このオプションを選択すると、「ナ ビゲーションからページを非表示」も選択されます。検索ページがナビゲーションに含まれるようにする場合は、このオプションの選択を解除できます。
- 9. 現在の更新における保留中のすべての変更を保存するには、「保存」をクリックします。

### ページの移動

ページをサイト・ツリーの現在のレベルの別の場所にドラッグ・アンド・ドロップするか、ページを切り取って別のレベルに貼り付けることができます。

階層には、必ずトップ・ページがあります。これはサイトのホーム・ページであり、 他のすべてのページがその下にリストされます。ページを特定のレベルにドラッグ・ アンド・ドロップして、ページの順序を変更できます。ページを切り取って階層内の 別の場所に貼り付けることもできます。

ページにネストしたページを含めることができます。ネストしたページを含むページ を移動すると、ネストしたページも移動し、移動したページの下にネストされたまま になります。使用するテーマが自動ナビゲーションに含まれている場合、ページの順 序を変更すると、ナビゲーション・メニューにリストされる順序も変更されます。

サイト・ツリーの別の場所にページを移動するには:

- 1. 編集するサイトを開きます。
- 2. 移動するページに移動します。
- **3.** ページをサイト・ツリーの現在のレベルの別の場所に移動するには、ページ名をクリックし、ページ・リストの別の場所にドラッグしてドロップします。



4. ページをサイト・ツリーの異なるレベルに移動するには、移動するページをクリックして そのページのをクリックします。をクリックして現在の場所からページを切り取るか、をクリックして現在の場所にページのコピーを残します。ページを配置するレベルに移動し、をクリックします。そのレベルのページを選択すると、貼り付けたページはその下にネストされます。をクリックして、現在の場所にページを貼り付けます。ページの順序を変更するには、サイト・ツリーの現在のレベルの別の場所にページをドラッグ・アンド・ドロップします。

### ページの削除

更新の一部としてページを削除できます。

削除されたページは、**「保存」**をクリックしたときに更新の一部として記録されます。更新が 公開されるまで、ページはベース・サイトから削除されません。コントリビュータまたはマ ネージャ・ロールを持つユーザーであれば、ページを削除できます。

/ ノート:

ページを削除する更新を公開すると、ページは完全に削除され、復元できません。 多言語サイトからページを削除する場合、サイトが公開される際にページの翻訳済 バージョンも削除されます。ページの追加または削除の管理が懸念される場合は、 サイトに対してページを追加または削除する際に専用の更新を使用して、これらの アクションを分離できます。

サイトからページを削除するには:

- 1. 編集するサイトを開きます。
- 2. 削除するページに移動し、 をクリックします。
- 3. 「OK」をクリックして削除を確定します。現在の更新における保留中のすべての変更を保存するには、「保存」をクリックします。

### ページ設定の変更

名前、タイトル、URL、ヘッダー、フッター、その他のオプションなどのページ設定を変更 できます。

- 1. 編集するサイトを開きます。
- 3. 必要に応じて設定を変更します。
  - ページ・タイプ:



- Webページ:サイトにあるページとコンテンツ。ページの名前指定とその コンテンツの定義、サイト内のページの配置指定、現在のサイトのコンテ キストにおけるページの動作方法の指定を行います。
- 外部リンク:ページは URL で指定された別の場所から参照されます。ペ ージの名前指定、サイト内の配置指定、および現在のサイトのコンテキストにおけるページの動作方法の指定を行うことができます。ライブ・サイトからページを使用しているため、リンクしたページのコンテンツを変更できません。

サイト・ツリーで、外部 URL にリンクするページはパージ名の前にのが あります。

- 「Webページ」を選択した場合、次の設定が表示されます。
- ページ名: 文字、数字、アンダースコア()およびハイフン(-)を使用できます。 テンプレート、テーマ、コンポーネント、サイトまたはサイト・ページについ て、authsite、content、pages、scstemplate_*、_comps、_components、 _compsdelivery、_idcservice、_sitescloud、_sitesclouddelivery、_themes、 _themesdelivery といった名前を使用しないでください。documents および sites は、サイト・ページには使用できますが、テンプレート、テーマ、コン ポーネントまたはサイトには使用しないでください。
- ページ URL: デフォルトで、ページ URL にはページ名が使用されます。スペースはハイフンに自動的に置換されます。デフォルト以外の URL を使用するには、「オーバーライド」をクリックして URL で使用するファイル名を追加します。文字、数字、アンダースコア()およびハイフン(-)を使用できます。必ず、ファイル拡張子を含めてください。デフォルトのファイル拡張子は.htmlです。
- ページ・レイアウト:ページ・レイアウトは、コンテンツではなくページの一般的な構造を定義します。レイアウトには、コンテンツを配置できる1つ以上の名前付きのスロットが含まれます。ページ・レイアウトの数とタイプは、サイトに関連付けられたテーマによって異なります。
- モバイル・ページ・レイアウト: モバイル・ページ・レイアウトは、モバイル・ デバイスで表示する際のページの一般的な構造を定義します。
- ページのタイトル:ページのタイトルはページ自体には表示されませんが、ペ ージが表示される際にブラウザのタイトル・バーまたはブラウザ・タブに表示 されます。ページのタイトルを追加するには、エディタを使用してタイトル・ コンポーネントを追加します。
- ページの説明: 説明はページには表示されませんが、他のコントリビュータまたは自身の使用のためにページに関する情報を追加できます。
- キーワード:検索エンジンがページのコンテンツを識別するためのカンマ区切りのオプションのキーワードを指定します。キーワードは、ページのテキストに表示されない、またはイメージに表示される用語または概念を識別するのに有用です。キーワードはページに表示されませんが、検索エンジンではそれらを使用してサイトを検索および特定できます。適切な説明、キーワードおよびシノニムによって、Webサイトへのアクセスを増加できます。
- ページ・ヘッダー: アナリティクスまたはトラッキング用のページ・ヘッダーのスクリプト記述またはタグをオプションで追加します。ヘッダーまたはフッターで使用する任意のコードを検証して、適切に動作し、サイトにセキュリティ・リスクをもたらさないことを確認してください。



- ページ・フッター: アナリティクスまたはトラッキング用のページ・フッターのスク リプト記述またはタグをオプションで追加します。デフォルトでフッターに含まれ るのは、プライバシ・ポリシーへのリンクを含むサイトの Cookie の使用に関して表 示されるポップアップのテキストです。開発者の場合、表示されるテキストを編集で きます。ヘッダーまたはフッターで使用する任意のコードを検証して、適切に動作 し、サイトにセキュリティ・リスクをもたらさないことを確認してください。
- ページ・オプション:
  - エラー・ページ: エラーによりリクエストされたページを表示できない場合に、 デフォルトのエラー・ページのかわりにこのページを表示します。デフォルトで は、このオプションを選択すると、「ナビゲーションからページを非表示」および 検索エンジンのすべてのオプションも選択されます。エラー・ページがナビゲー ションまたは検索エンジンの結果に含まれるようにする場合は、これらのオプシ ョンの選択を解除できます。
  - **ナビゲーションからページを非表示**: サイトの自動生成されるナビゲーションに ページが含まれません。
  - 詳細ページ: このページを使用して、選択したコンテンツ・アイテムの詳細な情報を別のページで表示します。構造化コンテンツを使用できるのは、エンタープライズ・ユーザーのみです。デフォルトでは、このオプションを選択すると、「ナビゲーションからページを非表示」も選択されます。ナビゲーションにディテール・ページを含める必要がある場合は、このオプションの選択を解除できます。構造化コンテンツ・アイテムがディテール・ページを使用するように構成されている場合、ユーザーが構造化コンテンツ・アイテムの詳細を表示するためにリンクをクリックすると、コンテンツ・アイテムの詳細情報を示すディテール・ページが表示されます。
  - 検索ページ: このページを使用して、検索の結果を表示します。このページは、 コンテンツ検索コンポーネントの「リンク」設定で選択できます。コンテンツ検 索を参照してください。デフォルトでは、このオプションを選択すると、「ナビゲ ーションからページを非表示」も選択されます。検索ページがナビゲーションに 含まれるようにする場合は、このオプションの選択を解除できます。
- SEO オプション:
  - ページを検索エンジンから隠す: 検索結果に表示されないように、ページのコン テンツの索引を作成しないよう検索エンジンに通知します。
  - ページ・リンクを検索エンジンから隠す:ページのリンクをフォローしないよう に、また結果としてリンク先の索引を作成しないように検索エンジンに通知しま す。
  - 検索エンジンのページ・キャッシュを無効化: このページをキャッシュしないよう検索エンジンに通知します。
  - ページの説明を検索エンジンから隠す:検索結果にページの後の(上で指定した) 説明を含めないよう検索エンジンに通知します。
- 共有ブラウズ:管理者がシステムで共有ブラウズを有効にしており、このサイトで共 有ブラウズが有効になっている場合、このページに参照セッションで使用する「共有 ブラウズ」ボタンを表示するかどうかを選択するオプションが表示されます。共有ブ ラウズ・セッションでは、Oracle Cobrowse Cloud Service を使用して、サイト・ビ ジターとの画面共有エクスペリエンスを管理します。共有ブラウズ統合の有効化お よびページでの共有ブラウズの使用を参照してください。
- 分析オプション: このサイトに分析追跡スニペットが含まれている場合、このページのスニペットをオーバーライドできます。オーバーライド・スイッチをクリックしてスクリプトを編集します。詳細は、分析追跡の追加を参照してください。サイト・ス

ニペットを編集すると、スクリプトが変更されたことを示すメッセージが表示 されます。カスタマイズ内容を削除するには:最新のサイト・スクリプトにリ ストアします。

「外部リンク」を選択した場合、次の設定が表示されます。

- ページ名: 文字、数字、アンダースコア()およびハイフン(-)を使用できます。
   この名前は、ページを識別するためにサイト・ツリーで使用されます。スペースはハイフンに自動的に置換されます。
- リンク URL: ページへの完全な URL を指定します。たとえば、https:// www.example.com/sharedpage.htm です。
- 新規ウィンドウでリンクを開く:ページを新規ブラウザ・ウィンドウまたはタブで開く場合は、このオプションを選択します。このオプションを選択しないと、ページは現在のウィンドウで開かれ、サイト・ページおよびナビゲーションと置き換わります。
- ページ・オプション:
  - エラー・ページ: エラーによりリクエストされたページを表示できない場合に、デフォルトのエラー・ページのかわりにこのページを表示します。 デフォルトでは、このオプションを選択すると、「ナビゲーションからページを非表示」も選択されます。エラー・ページがナビゲーションに含まれるようにする場合は、このオプションの選択を解除できます。
  - ナビゲーションからページを非表示: サイトの自動生成されるナビゲーションにページが含まれません。
  - 詳細ページ:このページを使用して、選択したコンテンツ・アイテムの詳細な情報を別のページで表示します。構造化コンテンツを使用できるのは、エンタープライズ・ユーザーのみです。デフォルトでは、このオプションを選択すると、「ナビゲーションからページを非表示」も選択されます。ナビゲーションにディテール・ページを含める必要がある場合は、このオプションの選択を解除できます。
     構造化コンテンツ・アイテムがディテール・ページを使用するように構成されている場合、ユーザーが構造化コンテンツ・アイテムの詳細を表示するためにリンクをクリックすると、コンテンツ・アイテムの詳細情報を示すディテール・ページが表示されます。
  - 検索ページ:このページを使用して、検索の結果を表示します。このページは、コンテンツ検索コンポーネントの「リンク」設定で選択できます。 コンテンツ検索を参照してください。デフォルトでは、このオプションを 選択すると、「ナビゲーションからページを非表示」も選択されます。検索 ページがナビゲーションに含まれるようにする場合は、このオプションの 選択を解除できます。
- 4. 現在の更新における保留中のすべての変更を保存するには、「保存」をクリックします。

### ページ・レイアウトの変更

レイアウトは、どのようにコンテンツをページに配列するかを定義します。異なるレ イアウトに、異なる数の名前付き*スロット*(ページの幅に広がる領域)を含めることがで きます。スロットには、1つ以上のコンテンツ・タイプを含めることができます。

各テーマには複数のページ・レイアウトがあります。サイトにページを追加する場合、 そのページに使用するレイアウトを選択します。各レイアウトには、コンテンツをド



ラッグ・アンド・ドロップできるスロットが含まれます。これらのスロットにどんなコンテ ンツを配置するかは、自由に選択できます。タイトル、テキストおよびドライバからマルチ メディア、ギャラリおよびソーシャル・メディアまで何でも使用できます。スロット内のコ ンテンツを配置できますが、ページのスロットの数や配置は変更できません。これを行うに は、新規ページ・レイアウトを使用する必要があります。

┛ ノート:

あるレイアウトを別のものと交換できます。ただし、注意が必要です。より少ない または異なる名前付きスロットを含むレイアウトを選択する場合、他のスロットの 既存のコンテンツは新しいレイアウトに表示されません。コンテンツは削除されま せんが、選択したレイアウトに同じ名前のスロットがないかぎり、コンテンツを表 示できません。

次の図は、サンプルのページのレイアウトを示しています。ページ・レイアウトの空のスロットと、スロットにタイトル、イメージおよびテキストが追加された完成ページが表示され ています。



ページに対して異なるレイアウトを選択するには:

- 1. 編集するサイトを開きます。
- 2. ページに移動し、 をクリックします。
- 「ページ・レイアウト」フィールドに移動し、メニューから異なるレイアウトを選択します。ページ・レイアウトの数とタイプは、サイトに関連付けられたテーマによって異なります。



4. 現在の更新における保留中のすべての変更を保存するには、「保存」をクリックします。

### 背景またはテーマの変更

ページ、ページ上の個々のスロットおよびスロット内のセクション・レイアウトの背 景色とイメージを指定できます。サイトのテーマ全体を変更することもできます。

スロットはページの幅に広がる領域で、1つ以上のタイプのコンテンツを含めることが できます。スロットに指定した背景は、スロット全体およびスロット内のすべてのコ ンポーネントに適用されます。

#### 背景の変更

背景は、相互に重なり合っています。スロットの背景を指定すると、ページに指定し た背景の上に重なります。ほとんどのイメージおよび色の場合、上側のレイヤーが下 側のレイヤーを効果的にオーバーライドします。背景色またはイメージで透明度を使 用する場合、下側のレイヤーの色は、透けて見えるか、上側のレイヤーで使用される 色と混合されます。

段落、タイトルなどの一部のコンポーネントは、事前に定義されたスタイルの一部と して、またはカスタマイズされたスタイルとして背景色を指定できます。コンポーネ ントに背景色を指定すると、この背景色もすべてのセクション・レイアウト、スロッ トまたはページ背景の上に重なります。スタイルおよびフォーマットの使用を参照し てください。

ページ、スロットおよびセクション・レイアウトの背景オプションは類似しています が、アクセス方法が多少異なります。

- 1. 編集するサイトを開きます。
- 2. 編集するページに移動し、 しか「編集」に設定されていることを確認します。
- 3. ページまたはスロットの背景を変更するには:

  - ページ全体の背景を指定するには、「ページを完成」をクリックし、
     シクリックします。ページ上のスロットの背景を指定するには、スロットを選択し、
     をクリックします。
- 4. セクション・レイアウトの背景を変更するには:
  - セクション・レイアウトのメニュー・アイコン
     をクリックし、「設定」を選 択します。
  - 「背景」をクリックします。ページ、スロットおよびセクション・レイアウトの背景オプションは同じです。
- 5. 背景にイメージを使用するには、「イメージを選択」をクリックします。アクセス 可能な任意の場所からイメージを使用できます。共有されているイメージまたは ローカルまたはネットワークのファイル場所からアップロードしたイメージを使 用することもできます。
- 6. イメージを選択して、「OK」をクリックします。ウィンドウに使用可能なすべての ファイルが表示されます。コンテキストに適切なファイル・タイプを選択してくだ



さい。たとえば、イメージ・ファイルを選択する場合、有効なイメージ・フォーマット (GIF、JPG、JPEG、PNG または SVG)のファイルを選択します。

- 7. イメージの設定を調整するには:
  - 「位置」を使用してページ上またはスロット内にイメージを配置します。
  - 「スケール」オプションを選択して、イメージの表示を調整します。
    - **フィット**: イメージをゆがませずにイメージ全体が使用可能な領域にフィットするように、イメージがスケーリングされます。
    - ストレッチ: 2 つの寸法(幅または高さ)の長い方が使用可能な領域にフィットするようにスケーリングされ、短い方の寸法は使用可能な領域を埋めるようにストレッチされます。
    - なし: イメージはそのフル・ピクセル解像度で使用され、使用可能な領域にフィットしない場合は均一に切り取られます。
  - 「繰返し」オプションを選択して、小さすぎて使用可能な領域を埋められないイメージを並べて表示します。「スケール」オプションとして「ストレッチ」を選択した場合、このオプションは適用されません。
  - ユーザーがページをスクロールしている間イメージを固定したままにするには、「ペ ージとともにスクロールしない」を選択します。
- 8. 背景の色を選択するには、「色」メニューから色を選択するか、「その他」をクリックして 全範囲の色から色を選択します。
  - 色を選択して色範囲表示にその色のバリエーションを表示するには、右側にあるスペクトラム・バーをクリックします。画面には、右上隅に選択した色と、左に向かって白の分量が多くなり下に向かって黒の分量が多くなっている、選択した色の色調が表示されます。
  - 色のバリエーションを選択するには、色範囲表示内でクリックするか、選択ポイント を新しい色の位置にクリックしてドラッグします。現在のカラー・ボックスと6桁の カラー・コードにより、選択した色が示されます。
  - 透明度(アルファ・チャネル)を調整する場合、透明度を増やすには透明度スライダを クリックして左にドラッグします。アルファ値0%では、色が透明化されるため、下 側のレイヤーの色とコンテンツが完全に透けて表示されます。アルファ値100%で は、色が完全に不透明化されるため、下側のレイヤーの色とコンテンツが遮断されま す。その他の値では、下側のレイヤーのイメージと色は、透けて見えるか、異なる度 合いで上側のレイヤーの色と混合されます。
  - 現在の色の選択を適用するには、「選択」をクリックします。

#### テーマの変更

テーマの変更は、「保存」をクリックしたときに更新の一部として記録されます。更新が公開 されるまで、テーマの変更はサイトに適用されません。



### 💉 ノート:

すべてのテーマが相互に互換性があるわけではありません。異なるテーマ に、異なるナビゲーション、スタイル名またはページ・レイアウト情報を含 めることができます。たとえば、別のレイアウト名のテーマを選択した場合、 既存のページに、そのテーマでは使用できないレイアウトが使用されること があります。それらのページは空白で表示されます。ページ・コンテンツは 削除されませんが、選択したテーマに同じ名前のレイアウトとスロットがな ければ表示されません。互換性のないテーマを選択した場合は、テーマを元 のテーマに再度変更して、ページ・コンテンツを復元できます。 確実に互換性のあるテーマに切り替えてください。不明な場合は、テーマの 開発者に確認してください。

サイトに使用されるテーマを変更するには:

- 1. 編集するサイトを開きます。
- サイドバーの 

   セクリックし、
   かをクリックします。
- 3. 使用可能なテーマのリストからいずれかを選択し、「OK」をクリックします。テーマが現在の更新に適用されます。現在の更新における保留中のすべての変更を保存するには、「保存」をクリックします。



9

### ページ・コンテンツの配置

コンポーネントおよびコンテンツ・アイテムによって、ユーザーが必要とする機能やコンテ ンツが提供され、セクション・レイアウトを使用すると、それらを自動的に配置できます。

- コンポーネントおよびセクション・レイアウトの追加
- アセットおよびコンテンツ・アイテムの操作
- トリガーおよびアクションの使用
- 水平セクション・レイアウトの使用
- 垂直セクション・レイアウトの使用
- 2列および3列のレイアウトの使用
- タブ付きのセクション・レイアウトの使用
- スライダ・セクション・レイアウトの使用

個々のコンポーネントの使用方法の詳細は、組込みコンポーネントの使用を参照してください。

コンポーネントおよびレイアウトを管理する方法について学習するには、「カスタム・コンポ ーネントおよびレイアウトの管理」を参照してください。

独自のコンポーネントの作成方法の詳細は、「コンポーネントの開発」を参照してください。

## コンポーネントおよびセクション・レイアウトの追 加

*コンポーネント*は、Web ページの個々の部分です。*セクション・レイアウト*により、それに 追加されたコンテンツが自動的に編成されるため、コントリビュータがページ上でコンテン ツのフォーマット設定に時間を費やさずにコンテンツを追加しやすくなります。

#### コンポーネントの追加

ページにコンポーネントを追加するには:

- 1. 編集するページに移動し、 「編集」に設定されていることを確認します。
- 2. りをクリックして、次のコンポーネント・タイプのいずれかをクリックします。
  - 「テーマ」をクリックすると、サイトに関連付けられたテーマで使用するために選択したコンポーネントのリストが表示されます。
  - 「カスタム」をクリックすると、ユーザーと共有されたカスタム・コンポーネントの リストが表示されます。
  - 「シード」をクリックすると、サービスで使用可能なデフォルト・コンポーネントの リストが表示されます。



- 「すべて」をクリックすると、ユーザーと共有されたすべてのコンポーネント が表示されます。
- 3. 特定のアイコンをクリックしてホールドし、ページ上の必要な場所にドラッグしま す。ページにアイテムをドラッグする際に、使用可能なスロット、セクション・レ

イアウトおよび既存のアイテムの境界が表示されます。配置アイコン

は垂直配置バー (左または右に配置)は、新しいコンテンツをどこに配置するかを 示します。セクション・レイアウトまたはコンテンツ・アイテムの周囲を実線で囲 む枠線は、そのアイテムがドロップ可能であり、自動的に配置されることを示しま す。スロットに複数のアイテムを含めることができ、ページ上のアイテムを新しい 場所にドラッグするだけで移動できます。

- 4. 適切な場所に移動したら、ページにアイテムをドロップします。
- 5. アイテムのプロパティを 調整するには、アイテムのメニュー・アイコン マクし、「設定」を選択します。アイテムによっては、テキストの追加、イメージ へのリンクの追加、別のサイトまたはマップへの URL の指定、アイテムのサイズ 変更またはその他のアクションを行う必要があります。

各コンポーネントには、サイズ、位置合せ、間隔、色、枠線などの設定があります。 これらにより、コンポーネントの外観と動作方法を定義します。たとえば、段落の設 定には、フォント・タイプやフォント・サイズなどの機能があり、テキストがどのよ うに表示されるかを決定します。

コンポーネントのプロパティを調整するには、コンポーネントを選択し、 の「メニ ユー」をクリックして「設定」を選択します。コンポーネントごとに、次のように異 なる種類の設定があります。

設定	説明
—般	「一般」設定には、間隔、位置合せおよびコンポーネント間で共通 の設定が含まれます。
スタイル	スタイルは、外観を管理するデフォルト値の名前付きセットです。 スタイルはテーマで定義されます。テーマごとに異なるスタイル を設定でき、テーマ内で、コンポーネントごとに異なるスタイル を設定できます。また、コンポーネントの特定のインスタンスに、 スタイル設定を手動で指定することもできます。
リンク	「リンク」設定には、コンポーネントによって使用されるイメー ジ、ドキュメントまたは他のリソースの場所が含まれます。アク ションを実行するボタンなどのコンポーネントの場合、リンク設 定には、コンポーネントによってサポートされるトリガーおよび アクションも含まれます。
コンポーネント	記事コンポーネントなどの他の標準コンポーネントを含むコンポ ーネントは、個別のコンポーネントのリストを提供し、これらの 各コンポーネントの設定へのアクセスをユーザーに提供します。
カスタム	「カスタム」設定は、コンポーネントに対して一意であり、標準の 設定グループから分離されて表示されます。たとえば、リモート・ コンポーネントは、一意の URL でプリファレンスを格納でき、カ スタム設定としてそれらを表示できます。

#### セクション・レイアウトの使用

セクション・レイアウトは、コンテンツを複数の列、垂直リストまたは一連のタブ付 き領域に自動編成できます。サイト・コントリビュータは、1つ以上のセクション・レ



イアウトを1つのスロットに追加してコンテンツを編成できます。ページにレイアウトを追加するには:

- 1. 編集するページに移動し、 **し**が「**編集」**に設定されていることを確認します。
- 2. をクリックして、ページにセクション・レイアウトを挿入します。使用するレイアウトのタイプを選択し、コンポーネントの配置に使用するのと同じアイテム配置手順に従います。

Slot slot2 Section Layout Three Columns			
[o]	[]	[]	
Image	Image	Image	

配置アイコン ・ は、セクション・レイアウトの移動方向(上、下、左、右)を示します。セクション・レイアウトまたはコンテンツ・アイテムを囲む実線の枠は、そのアイテムをドロップできることと、アイテムが自動的に配置されることを示しています。



エンタープライズ・ユーザーが構造化コンテンツ・アイテムを使用する場合は、コンテンツ・ アイテムのリストをページに追加するときにセクション・レイアウトを割り当てることがで きます。これにより、セクション・レイアウトがページ上のアイテムを自動的にフォーマッ トします。

開発者は、提供されているレイアウトに加えて、別のセクション・レイアウトを作成することにより、レイアウトに関する特定の問題を解決したり、コントリビュータのためにオーサリングを簡素化できます。テーマ・デザイナは、複数のセクション・レイアウトをページ・レイアウト内の1つのスロットに組み込むこともできます。レイアウトの開発を参照してください。



### アセットおよびコンテンツ・アイテムの操作

エンタープライズ・ユーザーの場合、*サイト・コレクション*または関連付けられてい るリポジトリに格納されているデジタル・アセットおよびコンテンツ・アイテムをサ イトに含めることができます。コレクションは、サイトで使用できるリポジトリ内の アセットの一部です。

デジタル・アセットまたはコンテンツ・アイテムをページに追加するには、

ジトリが使用されている場合、使用するリポジトリを選択します。サイトでの複数の リポジトリの使用の詳細は、Oracle Content Management でのアセットの管理の複数 のリポジトリへのサイト・アクセスの付与を参照してください。

アセットをフィルタして、必要なものを見つけることができます。 選択肢を絞り込む方法を選択します。アセットが表示されていない場合、アセットが サイト・コレクションに含まれていない可能性があります。参考まで、表示するサイ ト・コレクションだけではなく、リポジトリ内のすべてのコレクションを表示するよ うにフィルタを変更してみてください。詳細は、アセットの検索、フィルタおよびソ ートを参照してください。

デジタル・アセットまたはコンテンツ・アイテムをパネルからドラッグし、ページ上 のスロットにドロップします。段落コンポーネントのカーソルがある場所にイメージ を直接埋め込むことが可能で、オプションで、代替テキストの入力やイメージの高さ と幅の設定、位置合せの設定を行うことができます。

サイト・コレクションからページにアイテムを追加すると、自動的に適切なタイプの コンポーネントに配置されます。たとえば、イメージのデジタル・アセットを追加す ると、自動的にイメージ・コンポーネントに配置されます。コンテンツ・アイテムを 追加すると、自動的にコンテンツ・アイテム・コンポーネントに配置されます。また は、先にイメージやコンテンツ・アイテムのコンポーネントを追加し、後で、コンテ ンツ・アイテムを「コンテンツ」パネルからコンポーネントヘドラッグすることもで きます。また、コンポーネントの**「設定」**を選択し、**「選択」**をクリックして、アセッ トまたはドキュメント・リストからイメージを選択することも可能です。

## トリガーおよびアクションの使用

ボタン・コンポーネントでは、ページ・コンポーネントの表示または非表示とメッセ ージの表示など1つ以上のアクションを開始できます。特定のコンポーネント(フォル ダ・リストおよびファイル・リストなど)では、ユーザーが行う選択に基づいてコンパ ニオン・コンポーネントのアクションを開始できます。

たとえば、ボタンを押す(トリガー)と、ユーザーは別のページまたは外部 URL が示さ れ、この変化を通知するアラートが表示される(アクション)ようにボタンを構成できま す。ユーザーへの表示は、一緒に使用されるコンポーネントの種類によって変わりま す。



Configure Trigger Actions			
Select the actions for the trigger: Click	< on Button		
Page Actions	Show Alert	×	•
Navigate to Site Page	Message		
Navigate to External URL	You are going to the About page.	•	-
Show Alert	And		
Show or Hide a Component	Navigate to Site Page	×	
Show or Hide a Slot	Page		
	About Target	•	
	Open in Same Window	•	

2 つ以上のアクションを使用する場合、操作の順序を検討し、アクションを実行する順序に配置します。上の例では、最初にアラート・アクションをリストします。これにより、ページ にリダイレクトされる前にメッセージを読んで終了する時間がユーザーに与えられます。最初にリダイレクト・アクションをリストすると、ユーザーがメッセージを読む前に、メッセ ージが新しいページに置き換わる可能性があります。

#### ページ・アクション

トリガーおよびアクションをサポートするすべてのコンポーネントは、次のページ・アクシ ョンをサポートします。

- サイト・ページに移動:現在のサイトのページを選択します。
- 外部 URL に移動: 外部ページまたはサイトへの完全な URL を指定します。
- アラートの表示: 指定されたメッセージがウィンドウに表示されます。
- コンポーネントの表示または非表示:現在のページのコンポーネント・リストから、表示、 非表示または切替えを行うコンポーネントを選択します。
- **スロットの表示または非表示**: 現在のページのスロット・リストから、表示、非表示また は切替えを行うスロットを選択します。

#### コンポーネント固有のアクション

ページ・アクションの他に、コンポーネントでは独自のアクションを定義できます。これら のアクションにより、コンポーネントは他のコンポーネントとやりとりし、コンパニオン・ コンポーネントのアクションを開始できます。たとえば、ユーザーがフォルダ・リスト・コ ンポーネントでフォルダを選択すると、ファイル・リスト・コンポーネントにより選択した フォルダ内のファイルを表示できます。この場合、フォルダ・リスト・コンポーネントは**フ** 



**ォルダ選択済**トリガーをサポートし、ファイル・リスト・コンポーネントは**ファイル** の表示アクションをサポートしています。

アクションをサポートするコンポーネントをページに追加すると、コンポーネントと サポートするアクションが使用可能なアクションのリストに追加されます。一部のコ ンポーネントはアクションまたは特定のアクションのみをサポートします。一部のコ ンポーネントはトリガーを提供しますが、アクションをサポートしません。

独自のコンポーネントの作成方法について学習するには、コンポーネントの開発を参 照してください。

たとえば、ボタン・コンポーネントに1つ以上のアクションを指定するには、次のス テップを実行します。

- 1. ボタンのメニュー・アイコン をクリックし、「設定」を選択します。
- 2. 「設定」パネルで、「リンク」をクリックします。
- 3. 「リンク・タイプの選択」をクリックし、「トリガー・アクション」を選択します。
- 4. 「使用可能なトリガー」の下の「ボタンをクリック」をクリックします。
- 5. 「トリガー・アクションの構成」ウィンドウで、左側の列からアクションをクリッ クしてドラッグし、「何かをする」というラベルの付いたスロットにドロップしま す。

### 水平セクション・レイアウトの使用

セクション・レイアウトを使用して、レイアウトに追加するコンポーネントの間隔や 配置が自動的に決定されるようにすることができます。

水平レイアウトでは、追加されたアイテムが水平線方向に順番に配置されます。ページ幅が増加または減少すると、レイアウトは比例して変化します。デフォルトでは、 アイテムは1行に収まり、均等な水平スペースが割り当てられます。

ページにレイアウトを追加するには:

- 1. 編集するページに移動し、 **し**が「**編集」**に設定されていることを確認します。
- 2. ページにレイアウトを追加します。
- 他のコンテンツをセクション・レイアウトに配置するには、コンテンツをレイアウトにドラッグ・アンド・ドロップします。

レイアウトが、実線の枠線および「アイテムの追加」と表示されるバナーで強調表 示されます。

 Add Item	

レイアウトにさらにアイテムを追加でき、レイアウトによってそれらが適宜フォー マットされます。他のセクション・レイアウトを追加して、高度なレイアウトを作 成することもできます。



次に、レイアウトでのアイテムの順序を表す番号付きテキスト・コンポーネントを含む水 平レイアウトを示します。



- コンポーネントとその外観を編集するには、そのコンポーネントのメニュー・アイコン
   をクリックして、「設定」を選択します。デフォルトの比例サイズ設定を使用しない 場合は、個々の領域の幅を設定できます。位置合せ(左揃え、中央揃えまたは右揃え)を設 定することもできます。
- 5. 「一般」タブを使用して、レイアウト内の個々のコンポーネントの設定を変更します。 コンポーネント名をクリックして、そのコンポーネントの設定を表示します。
- 6. 「背景」タブを使用して、レイアウトの背景設定を変更します。

背景またはテーマの変更を参照してください。

### 2列および3列のレイアウトの使用

複数列のセクション・レイアウトを使用して、レイアウトに追加するコンポーネントの間隔 や配置が自動的に決定されるようにすることができます。

ページにレイアウトを追加するには:

- 1. 編集するページに移動し、 しか「編集」に設定されていることを確認します。
- 2. ページにレイアウトを追加します。
- 他のコンテンツをセクション・レイアウトに配置するには、コンテンツをレイアウトにドラッグ・アンド・ドロップします。
   レイアウトが、実線の枠線および「アイテムの追加」と表示されるバナーで強調表示されます。



レイアウトにさらにアイテムを追加でき、レイアウトによってそれらが適宜フォーマット されます。他のセクション・レイアウトを追加して、高度なレイアウトを作成したり、コ ンポーネント・グループを追加することもできます。

イメージ・コンポーネントがプレースホルダになった2列レイアウトを次に示します。

Section Layout Two Columns	
o"	o"
Image	Image

アイテムを追加すると、そのアイテムは追加の行に表示されます。それぞれに最大 2つのアイテムが入ります。

- コンポーネントとその外観を編集するには、そのコンポーネントのメニュー・アイ コン
   ケリックして、「設定」を選択します。
- 5. 「一般」タブを使用して、レイアウト内の個々のコンポーネントの設定を変更しま す。

コンポーネント名をクリックして、そのコンポーネントの設定を表示します。

- 6. 「背景」タブを使用して、レイアウトの背景設定を変更します。
- 7. 「カスタム設定」を選択して、表示されるコンテンツの追加のデフォルトを設定し ます。
  - 最初の列の幅(%): 列の幅を、レイアウトに使用可能な領域のパーセントで指定 します。
  - 2番目の列の幅(%): 列の幅を、レイアウトに使用可能な領域のパーセントで指定します。
  - 3番目の列の幅(%): 列の幅を、レイアウトに使用可能な領域のパーセントで指定します。
  - レスポンシブ・ブレークポイント(ピクセル):使用可能な表示サイズが変わると コンテンツが自動的に再フォーマットされるレスポンシブ・ページ設計の場 合、セクション・レイアウトが標準の2列レイアウトと次で指定する「レスポンシブ動作」オプションの間で切り替わる幅(ピクセル)を指定します。
  - **レスポンシブ動作**: 使用可能な表示サイズが**レスポンシブ・ブレークポイント** の値より小さいときにレイアウトをどのように変更するかを選択します。
    - アクションなし: レイアウトの動作を調整しません。
    - 列の積上げ:1列目のすべてのアイテム、2列目のすべてのアイテム、それ 以降の列のアイテムの順に、アイテムを上から下に1つの列内に配置しま す。
    - 最初の列を非表示: 他の列の領域が大きくなるように、最初の列のコンテンツを非表示にします。
    - 2番目の列を非表示:他の列の領域が大きくなるように、2番目の列のコン テンツを非表示にします。



- 3番目の列を非表示:他の列の領域が大きくなるように、3番目の列のコンテンツ を非表示にします。
- 両方の列を非表示: レイアウト内のすべてのコンテンツを非表示にします。
- 2番目の列を最初の列の下に移動:1列目のすべてのアイテム、2列目のすべての アイテムの順に、アイテムを1つの列内に配置します。
- 最初の列を2番目の列の下に移動:2列目のすべてのアイテム、1列目のすべての アイテムの順に、アイテムを1つの列内に配置します。

### 垂直セクション・レイアウトの使用

セクション・レイアウトを使用して、レイアウトに追加するコンポーネントの間隔や配置が 自動的に決定されるようにすることができます。

垂直レイアウトでは、追加されたアイテムが順番に縦1列に配置されます。

ページにレイアウトを追加するには:

- 1. 編集するページに移動し、**し**が「**編集」**に設定されていることを確認します。
- 2. ページにレイアウトを追加します。
- 他のコンテンツをセクション・レイアウトに配置するには、コンテンツをレイアウトにド ラッグ・アンド・ドロップします。

レイアウトが、実線の枠線および「アイテムの追加」と表示されるバナーで強調表示されます。



レイアウトにさらにアイテムを追加でき、レイアウトによってそれらが適宜フォーマット されます。他のセクション・レイアウトを追加して、高度なレイアウトを作成することも できます。

次に、レイアウトでのアイテムの順序を表す番号付きテキスト・コンポーネントを含む垂 直レイアウトを示します。

Section Layout	Vertical	T	
1			
2			
3			
4			



- コンポーネントとその外観を編集するには、そのコンポーネントのメニュー・アイ コン
   たクリックして、「設定」を選択します。
- 5. 「一般」タブを使用して、レイアウト内の個々のコンポーネントの設定を変更しま す。

コンポーネント名をクリックして、そのコンポーネントの設定を表示します。

6. 「背景」タブを使用して、レイアウトの背景設定を変更します。

### タブ付きのセクション・レイアウトの使用

タブ付きのセクション・レイアウトを使用すると、レイアウトに追加するコンポーネ ントの間隔や配置を作成できます。

ページにレイアウトを追加するには:

- 1. 編集するページに移動し、 「編集」に設定されていることを確認します。
- ページにレイアウトを追加します。デフォルトでは、レイアウトにタブが1つ追加 されます。他にもタブを追加する場合は、「新規タブ」をクリックします。
- タブにその他のコンテンツを配置するには、そのコンテンツをタブにドラッグ・アンド・ドロップします。

レイアウトにさらにアイテムを追加でき、レイアウトによってそれらが適宜フォー マットされます。他のセクション・レイアウトを追加して、高度なレイアウトを作 成したり、コンポーネント・グループを追加することもできます。

- 4. タブとその外観を編集するには、メニュー・アイコン を アイコン を クリックして、「設定」
- 5. タブの背景の設定を変更するには、「背景」セクションを使用します。 タブにイメージを使用したり、位置を変更したりすることが可能です。背景または テーマの変更を参照してください。
- 6. タブの外観に関するその他の設定を変更するには、「スタイル」セクションを使用 します。細線、フレームなどのスタイルを選択したり、枠線やタブの角をカスタマ イズしたりします。

### スライダ・セクション・レイアウトの使用

スライダ・セクション・レイアウトを使用すると、新しいコンテンツを表示するため にどちらかにスライドされるまで表示されるコンテンツを作成できます。ユーザー は、スライド・セクションの下にあるナビゲーション・ドットをクリックしてスライ ド間を移動します。

ページにスライダ・レイアウトを追加するには:

- 1. 編集するページに移動し、 **し**が「**編集」**に設定されていることを確認します。
- ページにレイアウトを追加します。デフォルトでは、レイアウトにスライドが1つ 追加されます。他にもスライドを追加する場合は、「新しいスライド」をクリック します。

ORACLE

 コンテンツを特定のスライドに配置するには、スライダにそのスライドを表示し、コンポ ーネントまたはコンテンツ・アイテムをドラッグ・アンド・ドロップします。

必要に応じて、引き続きスライドへのアイテムの追加やアイテムの配置を行うことができ ます。他のセクション・レイアウトを追加して、高度なレイアウトを作成したり、コンポ ーネント・グループを追加することができ、これにより必要なコンテンツを配置できま す。

- **4.** スライド・レイアウトやその外観を編集するには、メニュー・アイコン て、**「設定」**を選択します。
- 5. スライド領域全体の背景の設定を変更するには、「背景」セクションを使用します。 背景またはテーマの変更を参照してください。イメージを使用したり、位置を変更したり することが可能です。
- 6. 外観に関するその他の設定を変更するには、「スタイル」セクションを使用します。細線、 フレームなどのスタイルを選択したり、枠線やスライドの角をカスタマイズしたりしま す。
- 7. 個々のスライドの設定を変更するには、スライドの名前をクリックして設定を開きます。 必要に応じて、各スライドの背景やスタイルなどを変更します。



10

## 組込みコンポーネントの使用

コンポーネント、セクション・レイアウト、デジタル・アセットおよび構造化コンテンツを 使用すると、ユーザーが必要とするコンテンツや機能を柔軟に提供できます。

この項では、個々のコンポーネントの使用について説明します。コンポーネントに関する他 の情報は、次を参照してください。

- サイトでコンポーネントを使用する方法について学習するには、「ページ・コンテンツの 配置」を参照してください。
- コンポーネントの管理方法について学習するには、カスタム・コンポーネントおよびレイ アウトの管理を参照してください。
- 独自のコンポーネントの作成方法について学習するには、コンポーネントの開発を参照し てください。

#### 基本コンポーネント

- タイトル
- 段落
- プレーン・テキスト
- ・ ボタン

#### 構造コンポーネント

- 区切り
- スペーサ

#### メディア・コンポーネント

- ・ イメージ
- ギャラリ
- ギャラリ・グリッド
- YouTube ビデオ
- ビデオ

#### ドキュメント・コンポーネント

- ドキュメント
- フォルダ・リスト
- ファイル・リスト
- ドキュメント・マネージャ
- プロジェクト・ライブラリ



ソーシャル・コンポーネント

- ソーシャル・バー
- Facebook の「いいね!」と「おすすめ」
- Twitter の共有とフォロー
- 会話コンポーネント
- 会話リスト

#### プロセス・コンポーネント

- プロセス開始フォーム
- プロセス・タスク・リスト
- タスク詳細フォーム

#### コンテンツ・アイテム

- コンテンツ・アイテム・コンポーネント
- コンテンツ・プレースホルダ
- コンテンツ・リスト
- コンテンツ検索
- レコメンデーション

#### その他のコンポーネント

- マップ
- ヘッドライン
- 記事
- テキスト付きイメージ
- コンポーネント・グループ
- ページでの共有ブラウズの使用
- Oracle Intelligent Advisor
- Oracle Visual Builder

## 基本コンポーネント

基本コンポーネントは、テキスト・コンテンツの構築ブロックを提供します。

- タイトル
- 段落
- プレーン・テキスト
- ボタン



### タイトル

タイトル・コンポーネントを追加およびフォーマットするには:

- 1. 編集するページに移動し、 **し**が「**編集」**に設定されていることを確認します。
- 2. ページにコンポーネントを追加します。
- タイトル・コンポーネントをクリックして、タイトルのテキストを入力します。テキスト は、コンポーネントのデフォルトのスタイルにフォーマットされます。
- 4. タイトル・テキスト内にリンクを追加するには:
  - a. リンク・テキストとして使用するテキストを入力および選択して、 CMをクリックします。
  - b. 「リンク・タイプの選択」をクリックし、次のオプションのいずれかを選択します。
    - Webページ:外部ページまたはサイトへの完全な URL を指定し、どこでリンクを 開くかを選択します。
    - サイト・ページ:ページ・ピッカーを使用して現在のサイトのページを選択し、 どこでリンクを開くかを選択します。追加の URL パラメータを形式 key1=value1&key2=value2 で指定できます。空の値がサポートされていま す(例: key1=&key2=value2)。URL アンカーも指定できますが、特殊な「アン カー」セクション・レイアウトを追加して、アンカー・リンクを解決するターゲ ット・サイト・ページに配置し、トリガー・アクションを定義する際に使用する レイアウト設定に同じアンカー名を指定する必要があります。

### 🖉 ノート:

URL アンカーを使用するために必要な「アンカー」セクション・レイ アウトは、OCE ツールキットで配布されます。ツールキットの取得方 法は、「OCE ツールキットでの開発」を参照してください

ファイルのダウンロード: 選択したファイルをリポジトリからダウンロードします。ファイルを選択します。

デジタル・アセットを選択する場合、特定のレンディションを選択できます。レ ンディションを選択しない場合、元のサイズが使用されます。サイトの公開時に 最新バージョンのアセットを公開する場合、「最新バージョンのアセットを使用」 を選択します。「最新バージョンのアセットを使用」を選択しない場合は、最新の ドラフト・バージョンではなく最新の公開済バージョンが使用されます(ある場 合)。

- ファイル・プレビュー: 選択したファイルを、グレー表示された非アクティブな バージョンのページ上のオーバーレイでプレビューし、オプションでダウンロー ドします。
- コンテンツ・アイテム: 関連付けられたアセット・リポジトリからコンテンツ・ アイテムを選択し、表示する詳細ページを選択して、同じまたは新規ウィンドウ に開くページをターゲット設定します。


- 電子メール: 有効な電子メール・アドレス(およびオプションとして件名)を 指定します。生成されるメッセージが開かれて、デフォルトの電子メー ル・クライアントを通じて送信されます。
- **マップ**: 有効なアドレスまたは座標を入力し、マップをデスクトップおよびモバイル・ブラウザのどこで開くかを選択します。
- **電話**: 有効な電話番号を入力します。

リンクを削除するには、リンク・テキストの任意の場所をクリックして、 をクリックします。

デフォルトのフォーマットを変更する場合は、フォーマットするテキストを選択し、フォーマット・ツール・バーの任意のオプション(フォント、色、位置合せなど)を選択します。テーマによっては、これらのオプションがすべて表示されない場合もあります。



変更内容はすぐに適用されます。これらのフォーマットの変更は、ベース・スタイ ル全体に適用されます。ベース・スタイルを変更すると、オーバーライドした内容 がそのまま有効になります。

6. これらのオプションにより適用されたフォーマットを削除するには、テキストを選 択し、 ▲をクリックします。

### 段落

段落コンポーネントを追加およびフォーマットするには:

- 1. 編集するページに移動し、 **し**が「**編集」**に設定されていることを確認します。
- 2. ページにコンポーネントを追加します。
- 段落コンポーネントをクリックし、テキストを入力します。テキストは、コンポーネントのデフォルトのスタイルにフォーマットされます。
- 4. 段落にビデオ(□→)やイメージ(□→)、コンテンツ・アイテム(□→)を追加するには、 適切なアイコンをクリックします。追加するアセットを選択し、必要に応じて設定 を調整します。たとえば、コンテンツ・アイテムを挿入し、様々なビューを選択し て、段落内の高度なレイアウトを制御します。
- 5. 段落コンポーネント内に段落を追加するには、[Enter]を押します。
- 6. 段落内にリンクを追加するには:
  - a. リンク・テキストとして使用するテキストを入力するか選択して、 20 をクリックします。



- **b. 「リンク・タイプの選択」**をクリックし、次のオプションのいずれかを選択します。
  - Web ページ:外部ページまたはサイトへの完全な URL を指定し、どこでリンクを 開くかを選択します。
  - サイト・ページ:ページ・ピッカーを使用して現在のサイトのページを選択し、 どこでリンクを開くかを選択します。追加の URL パラメータを形式 key1=value1&key2=value2 で指定できます。空の値がサポートされていま す(例: key1=&key2=value2)。URL アンカーも指定できますが、特殊な「アン カー」セクション・レイアウトを追加して、アンカー・リンクを解決するターゲ ット・サイト・ページに配置し、トリガー・アクションを定義する際に使用する レイアウト設定に同じアンカー名を指定する必要があります。

💉 ノート:

URL アンカーを使用するために必要な「アンカー」セクション・レイ アウトは、OCE ツールキットで配布されます。ツールキットの取得方 法は、「OCE ツールキットでの開発」を参照してください

ファイルのダウンロード: 選択したファイルをリポジトリからダウンロードします。ファイルを選択します。

デジタル・アセットを選択する場合、特定のレンディションを選択できます。レ ンディションを選択しない場合、元のサイズが使用されます。サイトの公開時に 最新バージョンのアセットを公開する場合、「最新バージョンのアセットを使用」 を選択します。「最新バージョンのアセットを使用」を選択しない場合は、最新の ドラフト・バージョンではなく最新の公開済バージョンが使用されます(ある場 合)。

- ファイル・プレビュー: 選択したファイルを、グレー表示された非アクティブな バージョンのページ上のオーバーレイでプレビューし、オプションでダウンロー ドします。
- コンテンツ・アイテム: 関連付けられたアセット・リポジトリからコンテンツ・ アイテムを選択し、表示する詳細ページを選択して、同じまたは新規ウィンドウ に開くページをターゲット設定します。
- 電子メール: 有効な電子メール・アドレス(およびオプションとして件名)を指定します。生成されるメッセージが開かれて、デフォルトの電子メール・クライアントを通じて送信されます。
- マップ: 有効なアドレスまたは座標を入力し、マップをデスクトップおよびモバイル・ブラウザのどこで開くかを選択します。
- **電話**: 有効な電話番号を入力します。

リンクを削除するには、リンク・テキストの任意の場所をクリックして、 🗱 をクリックします。

テキストの任意の部分に対してデフォルトのフォーマットを変更する場合は、フォーマットするテキストを選択し、フォーマット・ツール・バーの任意のオプション(フォント、色、位置合せなど)を選択します。テーマによっては、これらのオプションがすべて表示されない場合もあります。





変更内容はすぐに適用されます。これらのフォーマットの変更は、ベース・スタイ ル全体に適用されます。ベース・スタイルを変更すると、オーバーライドした内容 がそのまま有効になります。

 これらのオプションを使用して適用されているフォーマットを削除するには、テキ ストを選択して、
 タリックします。

### プレーン・テキスト

テキスト・コンポーネントを使用して、ページにテキストを追加し、テーマに定義されたスタイルのみを使用してテキストをフォーマットできます。コンテンツの編集中 はフォーマット・ツール・バーを使用できないため、選択されたスタイルをカスタム・ フォーマットでオーバーライドできません。

これは、見出しなどの要素に統一されたフォーマットを使用する場合に有用です。テ ーマで定義されたスタイルを更新し、関連付けられているサイト全体のテキストを自 動的に更新できます。これを妨げるフォーマット・オーバーライドは伴いません。

テキスト・コンポーネントを追加するには:

- 1. 編集するページに移動し、 **し**が「**編集」**に設定されていることを確認します。
- 2. ページにコンポーネントを追加します。
- 段落コンポーネントをクリックし、テキストを入力します。テキストには、コンポ ーネント(または段落要素)のデフォルト・スタイルのフォーマットが適用され ます。

追加の段落を追加するには、単に[Enter]を押すだけです。

💉 ノート:

他のソースからコンテンツをコピーしてテキスト・コンポーネントに貼 り付けることができますが、イメージおよびベースの HTML タグは削除 されます。

- 4. テキスト・コンポーネントのベース・スタイルを変更するには:
  - a. コンポーネントのメニュー・アイコン をクリックし、「設定」を選択しま す。
  - b. 「スタイルの選択」をクリックして、メニューからスタイルを選択します。



リストには、段落および見出し用の標準の HTML タグが含まれます。選択したスタイ ルによって関連付けられているタグがコンテンツに割り当てられます。

コンポーネントは選択したベース・スタイルを反映します。

### ボタン

ボタンを使用してページのリンクまたはその他の機能がすぐにわかるようにします。

Professional	١
\$10	
per month	
10 users	
<ul> <li>30 GB storage</li> </ul>	l
<ul> <li>Email priority support</li> </ul>	
Free update	
BUTTON SITE PAGE: PROFESSIONAL	
Start Free Trial	

ページにボタンを追加するには:

- 1. 編集するページに移動し、 **し**が「**編集」**に設定されていることを確認します。
- 2. ページにコンポーネントを追加します。
- 3. ボタンのプロパティを調整するには、そのメニュー・アイコンをクリックし、「設定」 を選択します。

ボタンのテキスト**(**ラベル**)**、ボタンのサイズ、位置合せおよびその他の表示オプションを 指定できます。

🖉 ノート:

「幅」および「高さ」フィールドを 0 (ゼロ)に設定すると、ボタンはラベルに指定したテキストにフィットするサイズに自動設定されます。

背景色、フォント、枠線およびその他の設定を指定するには、「スタイル」タブをクリッ クします。現在のテーマの事前定義済スタイルから選択できます。テーマによっては、 「カスタマイズ」をクリックして独自の値を指定することもできます。

- 4. リンクまたはその他のアクションをボタンに関連付けるには:
  - a. 「設定」パネルで、「リンク」をクリックします。
  - **b. 「リンク・タイプの選択」**をクリックし、次のオプションのいずれかを選択します。
    - リンクなし: ユーザーがボタンをクリックした際に、アクションを実行しません。



- Webページ:外部ページまたはサイトへの完全な URL を指定し、どこでリンクを開くかを選択します。
- サイト・ページ:ページ・ピッカーを使用して現在のサイトのページを選択し、どこでリンクを開くかを選択します。追加の URL パラメータを形式 key1=value1&key2=value2 で指定できます。空の値がサポートされています(例: key1=&key2=value2)。URL アンカーも指定できますが、特殊な「アンカー」セクション・レイアウトを追加して、アンカー・リンクを解決するターゲット・サイト・ページに配置し、トリガー・アクションを定義する際に使用するレイアウト設定に同じアンカー名を指定する必要があります。

N-F:

URL アンカーを使用するために必要な「アンカー」セクション・ レイアウトは、OCE ツールキットで配布されます。ツールキッ トの取得方法は、「OCE ツールキットでの開発」を参照してくだ さい

ファイルのダウンロード: 選択したファイルをリポジトリからダウンロードします。ファイルを選択します。

デジタル・アセットを選択する場合、特定のレンディションを選択できま す。レンディションを選択しない場合、元のサイズが使用されます。サイ トの公開時に最新バージョンのアセットを公開する場合、「最新バージョン のアセットを使用」を選択します。「最新バージョンのアセットを使用」を 選択しない場合は、最新のドラフト・バージョンではなく最新の公開済バ ージョンが使用されます(ある場合)。

- ファイル・プレビュー: 選択したファイルを、グレー表示された非アクティブなバージョンのページ上のオーバーレイでプレビューし、オプションでダウンロードします。
- コンテンツ・アイテム: 関連付けられたアセット・リポジトリからコンテンツ・アイテムを選択し、表示する詳細ページを選択して、同じまたは新規ウィンドウに開くページをターゲット設定します。
- 電子メール: 有効な電子メール・アドレス(およびオプションとして件名)を 指定します。生成されるメッセージが開かれて、デフォルトの電子メー ル・クライアントを通じて送信されます。
- トリガー・アクション:ボタンが押されたときに実行する1つ以上のページ・アクションを選択します。次のようなページ・アクションがあります。
  - サイト・ページに移動:現在のサイトのページを選択します。
  - **外部 URL に移動**: 外部ページまたはサイトへの完全な URL を指定します。
  - **アラートの表示**:指定されたメッセージがウィンドウに表示されます。
  - コンポーネントの表示または非表示:表示、非表示または切替えを行う
     現在のページのコンポーネントを選択します。
  - スロットの表示または非表示:表示、非表示または切替えを行う現在のページのスロットを選択します。



- **マップ**:有効なアドレスまたは座標を入力し、マップをデスクトップおよびモバ イル・ブラウザのどこで開くかを選択します。
- **電話**: 有効な電話番号を入力します。

トリガーおよびアクションの使用を参照してください。

## 構造コンポーネント

構造コンポーネントを使用すると、ページ上のコンテンツを区切ることができます。

- 区切り
- スペーサ

### 区切り

区切り(横線)を使用して列またはページに視覚的区切りを作成します。

ページに区切りを追加するには:

- 1. 編集するページに移動し、 **し**が「**編集」**に設定されていることを確認します。
- 2. ページにコンポーネントを追加します。

Before		After	
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.		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.	
Get in touch		Get in touch	

3. 区切りのプロパティを調整するには、そのメニュー・アイコン を選択します。

左と右の間隔オプションを使用して、区切りの水平配置および長さを指定できます。 線の色、太さおよびその他の設定はコンポーネント・スタイルで定義されます。現在のテ ーマの事前定義済スタイルから選択するか、「カスタマイズ」をクリックして独自の値を 指定できます。

### スペーサ

ページの「余白」を管理することで、ページが視覚的によりアピールされ、読みやすくなり ます。スペーサ・コンポーネントを使用すると、スタイルまたはその他のコンポーネントで 定義された間隔をオーバーライドせずに余白を追加できます。

ページ内のコンポーネント間の垂直間隔を追加するには:

- 1. 編集するページに移動し、 // 「編集」に設定されていることを確認します。
- 2. ページにコンポーネントを追加します。



Our Privacy Policy	SPACER 🕂 =
This policy lets you know how we collect and use your Personal Information, how you can control its use, and describes	SPACER
our practices regarding information collected from computer or mobile	
software applications and sites that link or refer to this policy.	
Privacy Policy Lorem issum dolor sit amet. consectetur adisicino elf. Duis a rhonous ex.	

3. スペーサの高さを調整するには、そのメニュー・アイコン <br/>
こをクリックし、「設定」を選択します。

## メディア・コンポーネント

メディア・コンポーネントにより、イメージおよびビデオをページに配置できます。

- ・ イメージ
- ギャラリ
- ギャラリ・グリッド
- YouTube ビデオ
- ビデオ

イメージ

イメージは、参照者の注意を引き、参照者にページの別のコンテンツ領域を探索する よう招待できます。

ページにイメージを追加するには:

- 1. 編集するページに移動し、 **し**が「**編集」**に設定されていることを確認します。
- 2. ページにコンポーネントを追加します。イメージ・コンポーネントには、使用する イメージを選択するまでプレースホルダ・イメージが表示されます。

IMAGE		1
	$\boxed{\bigcirc}$	
	IMAGE	

3. イメージを選択するには、メニュー・アイコン <br/>
をクリックし、「設定」を選択して、「イメージ」フィールドの横の「選択」をクリックします。



 サイト・リポジトリからイメージ(ユーザーと共有されているドキュメント・フォルダの イメージ)を選択するか、イメージをドキュメント・フォルダにアップロードします。

/ノート:

ウィンドウに使用可能なすべてのファイルが表示されます。コンテキストに適切なファイル・タイプを選択する必要があります。たとえば、イメージ・ファイルを選択する場合、有効なイメージ・フォーマット(GIF、JPG、JPEG、PNGまたは SVG)のファイルを選択する必要があります。

a. 使用するイメージを検索して選択します。

デジタル・アセットが表示されない場合、 ^前をクリックして、コレクション・フィルタを「**すべて」**に変更します。

- b. ドキュメント・フォルダからイメージを選択した場合、ファイルをサイトにコピーせずに、そのファイルにリンクできます。ファイルにリンクするには、「サイトにファイルをコピーするかわりに元のファイルへの参照を使用します」を選択します。このオプションを選択しない場合、ファイルのコピーがサイトで格納され、サイトから参照されます。元のファイルにリンクすることで、コンテンツを複製せずに済みます。リンクを使用すれば、ファイルの権限が変更されたり、表示が制限されるような場合でも、サイトのビジターがコンテンツを表示できます。
- **c.**「OK」をクリックします。
- d. デジタル・アセットを選択した場合、特定のレンディションを選択できます。レンディションを選択しない場合、元のサイズが使用されます。サイトの公開時に最新バージョンのアセットを公開する場合、「最新バージョンのアセットを使用」を選択します。
- 5. イメージを編集するには、 をクリックし、次のいずれかのアクションでイメージを編 集します。
  - イメージを切り取るには、「「切取り」をクリックします。切取りツールバーで事前定義されたイメージ比率の1つを選択するか、イメージの切取りハンドルを目的の位置までドラッグします。完了したら、切取りツールバーで「切取り」をクリックします。
  - イメージを回転または反転するには、 「回転」をクリックします。回転ツールバーで、カスタム回転度を入力する、ボタンを使用してイメージを左または右に回転する、あるいはイメージを水平方向または垂直方向のいずれに反転するかを選択します。
  - イメージにウォーターマークを追加するには、 「ウォーターマーク」をクリックします。テキストをイメージに追加し、テキスト・サイズ、スタイル、色および不透明度を、ウォーターマーク・ツールを使用して必要に応じて変更します。
  - イメージ・フォーマットを変更するには、⁽²⁾「オプション」をクリックし、「フォー マット」ドロップダウン・リストから新しいフォーマットを選択します。
  - 背景色を変更するには、 「オプション」をクリックし、「背景色」ドロップダウン・メニューからオプションを選択します。



• .jpg または.webp (Google Chrome ブラウザに対応)を編集中の場合、イメージ

品質を変更してより小さなファイル・サイズを作成できます。 🥨 「オプショ ン」をクリックし、「品質」ボックスに新しいパーセンテージを入力します。

- 変更を元に戻すか、やりなおすには、[→]または [⊂]をクリックします。行った すべての変更を削除するには、「リセット」をクリックします。
- イメージの拡大率を変更するには、ズーム・コントロールを使用します (⁻ - - +)。
- 6. 「設定」メニュー の「一般」パネルを使用して、キャプションを追加したり、 間隔、位置合せ、スタイル、その他のオプションを変更したりします。
- 7. リンクまたはその他のアクションをイメージに関連付けることができます。
  - a. 「設定」パネルで、「リンク」をクリックします。
  - **b.** 次のいずれかのオプションを選択します。
    - リンクなし: ユーザーがイメージをクリックした際に、アクションを実行しません。
    - Webページ:外部ページまたはサイトへの完全な URL を指定し、どこでリンクを開くかを選択します。
    - サイト・ページ:ページ・ピッカーを使用して現在のサイトのページを選択し、どこでリンクを開くかを選択します。追加の URL パラメータを形式 key1=value1&key2=value2 で指定できます。空の値がサポートされています(例: key1=&key2=value2)。URL アンカーも指定できますが、特殊な「アンカー」セクション・レイアウトを追加して、アンカー・リンクを解決するターゲット・サイト・ページに配置し、トリガー・アクションを定義する際に使用するレイアウト設定に同じアンカー名を指定する必要があります。

♪ ノート:

URL アンカーを使用するために必要な「アンカー」セクション・ レイアウトは、OCE ツールキットで配布されます。ツールキッ トの取得方法は、「OCE ツールキットでの開発」を参照してくだ さい

ファイルのダウンロード: 選択したファイルをリポジトリからダウンロードします。ファイルを選択します。

デジタル・アセットを選択する場合、特定のレンディションを選択できま す。レンディションを選択しない場合、元のサイズが使用されます。サイ トの公開時に最新バージョンのアセットを公開する場合、「最新バージョン のアセットを使用」を選択します。「最新バージョンのアセットを使用」を 選択しない場合は、最新のドラフト・バージョンではなく最新の公開済バ ージョンが使用されます(ある場合)。

ファイル・プレビュー: 選択したファイルはページ上にオーバーレイとして表示されます。

- コンテンツ・アイテム: 関連付けられたアセット・リポジトリからコンテンツ・ アイテムを選択し、表示する詳細ページを選択して、同じまたは新規ウィンドウ に開くページをターゲット設定します。
- 電子メール: 有効な電子メール・アドレス(およびオプションとして件名)を指定します。生成されるメッセージが開かれて、デフォルトの電子メール・クライアントを通じて送信されます。
- イメージ・プレビュー: 選択したイメージはページ上にオーバーレイとして表示 されます。
- **マップ**:有効なアドレスまたは座標を入力し、マップをデスクトップおよびモバ イル・ブラウザのどこで開くかを選択します。
- **電話**: 有効な電話番号を入力します。

### ギャラリ

イメージ・ギャラリを使用して、一連のイメージを表示します。イメージが自動的に繰り返 し表示されるように選択することも、ユーザーがイメージを手動で進められるように選択す ることもできます。

ページにイメージ・ギャラリを追加するには:

- 1. 編集するページに移動し、 **し**が「**編集」**に設定されていることを確認します。
- ページにコンポーネントを追加します。ギャラリ・コンポーネントには、使用するイメージを選択するまでプレースホルダ・イメージが表示されます。
- 3. ギャラリに1つ以上のイメージを追加するには、そのメニュー・アイコン こ、「設定」を選択して、「一般」タブの「イメージ」をクリックします。
- 4. 「イメージの追加」をクリックします。



5. 1つ以上のイメージを選択します。



#### 💉 ノート:

ウィンドウに使用可能なすべてのファイルが表示されます。コンテキストに適切なファイル・タイプを選択する必要があります。たとえば、イメージ・ファイルを選択する場合、有効なイメージ・フォーマット(GIF、JPG、JPEG、PNG または SVG)のファイルを選択する必要があります。

a. 使用するイメージを検索して選択します。

デジタル・アセットが表示されない場合、³³⁷をクリックして、コレクション・フィルタを**「すべて」**に変更します。

- b. ドキュメント・フォルダからイメージを選択した場合、ファイルをサイトにコ ピーせずに、そのファイルにリンクできます。ファイルにリンクするには、「サ イトにファイルをコピーするかわりに元のファイルへの参照を使用します」を 選択します。このオプションを選択しない場合、ファイルのコピーがサイトで 格納され、サイトから参照されます。元のファイルにリンクすることで、コン テンツを複製せずに済みます。リンクを使用すれば、ファイルの権限が変更さ れたり、表示が制限されるような場合でも、サイトのビジターがコンテンツを 表示できます。
- **c. 「OK」**をクリックします。
- d. デジタル・アセットを選択した場合、特定のレンディションを選択できます。 レンディションを選択しない場合、元のサイズが使用されます。サイトの公開時に最新バージョンのアセットを公開する場合、「最新バージョンのアセットを使用」を選択します。

選択したイメージは、イメージのリストに追加されます。イメージをドラッグ・ア ンド・ドロップしてリスト内のイメージの順序を変更します。各イメージのデフォ ルトのタイトルは、拡張子のないファイル名です。

6. 特定のイメージのタイトル、説明またはその他のオプションを変更するには、リス トのイメージをクリックして変更を加えます。

リンクまたはその他のアクションをギャラリのイメージに関連付けることができ ます。

- a. 特定のイメージの「設定」パネルで、「リンク」フィールドをクリックします。
- b. 次のいずれかのオプションを選択します。
  - リンクなし: ユーザーがイメージをクリックした際に、アクションを実行しません。
  - Webページ:外部ページまたはサイトへの完全な URL を指定し、どこでリンクを開くかを選択します。
  - サイト・ページ:ページ・ピッカーを使用して現在のサイトのページを選択し、どこでリンクを開くかを選択します。追加の URL パラメータを形式 key1=value1&key2=value2 で指定できます。空の値がサポートされています(例: key1=&key2=value2)。URL アンカーも指定できますが、特殊な「アンカー」セクション・レイアウトを追加して、アンカー・リンクを解決するターゲット・サイト・ページに配置し、トリガー・アクションを定義する際に使用するレイアウト設定に同じアンカー名を指定する必要があります。



🔊 ノート:

URL アンカーを使用するために必要な「アンカー」セクション・レイ アウトは、OCE ツールキットで配布されます。ツールキットの取得方 法は、「OCE ツールキットでの開発」を参照してください

ファイルのダウンロード: 選択したファイルをリポジトリからダウンロードします。ファイルを選択します。

デジタル・アセットを選択する場合、特定のレンディションを選択できます。レ ンディションを選択しない場合、元のサイズが使用されます。サイトの公開時に 最新バージョンのアセットを公開する場合、「最新バージョンのアセットを使用」 を選択します。「最新バージョンのアセットを使用」を選択しない場合は、最新の ドラフト・バージョンではなく最新の公開済バージョンが使用されます(ある場 合)。

- コンテンツ・アイテム: 関連付けられたアセット・リポジトリからコンテンツ・ アイテムを選択し、表示する詳細ページを選択して、同じまたは新規ウィンドウ に開くページをターゲット設定します。
- 電子メール: 有効な電子メール・アドレス(およびオプションとして件名)を指定します。生成されるメッセージが開かれて、デフォルトの電子メール・クライアントを通じて送信されます。
- **マップ**:有効なアドレスまたは座標を入力し、マップをデスクトップおよびモバ イル・ブラウザのどこで開くかを選択します。
- **電話**: 有効な電話番号を入力します。
- c. イメージの設定パネルに戻るには、「戻る」をクリックします。更新する別のイメージを選択するためにイメージ・リストに戻るには、再度「戻る」をクリックします。
- d. 個々のイメージの更新が完了したら、「戻る」をクリックしてギャラリ・オプション を指定します。
- 7. ギャラリのイメージのサイズ設定およびスケーリングを行うには:
  - a. 幅を使用して、スロット内のギャラリの幅(ピクセル単位)を指定します。幅を指定する場合、「フィル」以外の位置合せオプションをクリックします。幅を設定した後に、「フィル」を使用して指定した幅にイメージを拡大することができます。
  - b. 次の「スケーリング」オプションを選択して、ギャラリのイメージの表示を調整しま す。
    - 切取り:2つの寸法(幅または高さ)の短い方が使用可能な領域にフィットするようにスケーリングされ、長い方の寸法はイメージをストレッチしないで切り取られます。
    - **フィット**: イメージをゆがませずにイメージ全体が使用可能な領域にフィットするように、各イメージがスケーリングされます。
    - ストレッチ: 2 つの寸法(幅または高さ)の長い方が使用可能な領域にフィットするようにスケーリングされ、短い方の寸法は使用可能な領域を埋めるようにストレッチされます。
    - **なし**: イメージはそのフル・ピクセル解像度で使用され、使用可能な領域にフィットしない場合は均一に切り取られます。
- 8. ギャラリ内を移動するためにユーザーに役立つには:



- **a.** 「ナビゲーション」方法を選択します。
  - サムネイル: ギャラリ内のイメージのリストをギャラリの下に順に表示します。ユーザーがサムネイル・イメージをクリックすると、ギャラリ内の関連するイメージに移動できます。
  - インデクサ: ギャラリ内の各イメージを表す一連のボタンを ギャラリの下に表示します。ユーザーがボタンをクリックすると、ギャラ リ内の関連するイメージに移動できます。
  - なし: 視覚的なナビゲーションは表示されません。ユーザーは右または左にスワイプして、隣り合ったイメージを表示できます。この手動の方法は常に使用できます。
- b. 「前/次を表示」をクリックして、ギャラリ内の前または次のイメージに進むた めの矢印アイコンを各イメージに含めます。
- 9. 「キャプションの表示」ボックスを選択して、イメージのキャプションを表示また は非表示にするように選択できます。
- 10. ギャラリでイメージが自動的に繰り返し表示されるようにするには、「自動再生」 を選択し、「遷移時間」および「時間の表示」を指定します。

ユーザーは設定した手動ナビゲーション機能をすべて継続して使用できます。

### ギャラリ・グリッド

ギャラリ・グリッドを使用して、行および列に一連のイメージを一度に表示します。 ページにイメージ・ギャラリを追加するには:

- 1. 編集するページに移動し、 **し**が「**編集」**に設定されていることを確認します。
- ページにコンポーネントを追加します。ギャラリ・グリッド・コンポーネントには、使用するイメージを選択するまでプレースホルダ・イメージが表示されます。
- 3. ギャラリに1つ以上のイメージを追加するには、そのメニュー・アイコン = をク リックし、「設定」を選択して、「一般」タブの「イメージ」をクリックします。
- 4. 「イメージの追加」をクリックします。
- 5. 1 つ以上のイメージを選択します。

#### 💉 ノート:

ウィンドウに使用可能なすべてのファイルが表示されます。コンテキストに適切なファイル・タイプを選択する必要があります。たとえば、イメージ・ファイルを選択する場合、有効なイメージ・フォーマット(GIF、JPG、JPEG、PNGまたはSVG)のファイルを選択する必要があります。

a. 使用するイメージを検索して選択します。

デジタル・アセットが表示されない場合、¹¹¹をクリックして、コレクション・ フィルタを**「すべて」**に変更します。



- b. ドキュメント・フォルダからイメージを選択した場合、ファイルをサイトにコピーせずに、そのファイルにリンクできます。ファイルにリンクするには、「サイトにファイルをコピーするかわりに元のファイルへの参照を使用します」を選択します。このオプションを選択しない場合、ファイルのコピーがサイトで格納され、サイトから参照されます。元のファイルにリンクすることで、コンテンツを複製せずに済みます。リンクを使用すれば、ファイルの権限が変更されたり、表示が制限されるような場合でも、サイトのビジターがコンテンツを表示できます。
- **c.**「OK」をクリックします。
- d. デジタル・アセットを選択した場合、特定のレンディションを選択できます。レンデ ィションを選択しない場合、元のサイズが使用されます。サイトの公開時に最新バー ジョンのアセットを公開する場合、「最新バージョンのアセットを使用」を選択しま す。

選択したイメージは、イメージのリストに追加されます。イメージをドラッグ・アンド・ ドロップしてリスト(およびグリッド)内のイメージの順序を変更します。各イメージの デフォルトのタイトルは、拡張子のないファイル名です。

6. 特定のイメージのタイトル、説明またはその他のオプションを変更するには、リストのイ メージをクリックして変更を加えます。

リンクまたはその他のアクションをギャラリのイメージに関連付けることができます。

- a. 特定のイメージの「設定」パネルで、「リンク」フィールドをクリックします。
- b. 次のいずれかのオプションを選択します。
  - リンクなし: ユーザーがイメージをクリックした際に、アクションを実行しません。
  - Web ページ:外部ページまたはサイトへの完全な URL を指定し、どこでリンクを 開くかを選択します。
  - サイト・ページ:ページ・ピッカーを使用して現在のサイトのページを選択し、 どこでリンクを開くかを選択します。追加の URL パラメータを形式 key1=value1&key2=value2 で指定できます。空の値がサポートされていま す(例: key1=&key2=value2)。URL アンカーも指定できますが、特殊な「アン カー」セクション・レイアウトを追加して、アンカー・リンクを解決するターゲ ット・サイト・ページに配置し、トリガー・アクションを定義する際に使用する レイアウト設定に同じアンカー名を指定する必要があります。

#### ▲ ノート:

URL アンカーを使用するために必要な「アンカー」セクション・レイ アウトは、OCE ツールキットで配布されます。ツールキットの取得方 法は、「OCE ツールキットでの開発」を参照してください

ファイルのダウンロード: 選択したファイルをリポジトリからダウンロードします。ファイルを選択します。

デジタル・アセットを選択する場合、特定のレンディションを選択できます。レ ンディションを選択しない場合、元のサイズが使用されます。サイトの公開時に 最新バージョンのアセットを公開する場合、「最新バージョンのアセットを使用」 を選択します。「最新バージョンのアセットを使用」を選択しない場合は、最新の ドラフト・バージョンではなく最新の公開済バージョンが使用されます(ある場 合)。



- コンテンツ・アイテム: 関連付けられたアセット・リポジトリからコンテンツ・アイテムを選択し、表示する詳細ページを選択して、同じまたは新規ウィンドウに開くページをターゲット設定します。
- 電子メール: 有効な電子メール・アドレス(およびオプションとして件名)を 指定します。生成されるメッセージが開かれて、デフォルトの電子メー ル・クライアントを通じて送信されます。
- イメージ・プレビュー: 選択したイメージはページ上にオーバーレイとして表示されます。
- マップ:有効なアドレスまたは座標を入力し、マップをデスクトップおよびモバイル・ブラウザのどこで開くかを選択します。
- 電話: 有効な電話番号を入力します。
- c. イメージの設定パネルに戻るには、「戻る」をクリックします。更新する別の イメージを選択するためにイメージ・リストに戻るには、再度「戻る」をクリ ックします。
- d. 個々のイメージの更新が完了したら、「戻る」をクリックしてギャラリ・オプ ションを指定します。
- 7. グリッドのイメージを配置するには、「レイアウト」オプションを使用します。
  - メーソンリー
  - 列
  - カスタム

これらの各オプションについては、次のステップで説明します。

8. 使用可能な領域内の行にイメージを自動的に配置するには、「メーソンリー」レイ アウトを選択します。

生成される行の高さは均一ですが、列の幅は統一されていません。



- a. すべてのイメージを指定された高さ(ピクセル単位)に比例してスケーリングす るには、「高さ」を指定します。
- b. 行内のイメージ間のスペースを増やすまたは減らすには、「イメージ間隔」を 指定します。
- 9. 行と列にイメージを配置するには、「列」レイアウトを選択します。
  - a. 次の「スケーリング」オプションを選択して、グリッド内のイメージの表示を 調整します。



- 切取り:2つの寸法(幅または高さ)の短い方が使用可能な領域にフィットするようにスケーリングされ、長い方の寸法はイメージをストレッチしないで切り取られます。
- **フィット**: イメージをゆがませずにイメージ全体が使用可能な領域にフィットするように、各イメージがスケーリングされます。

たとえば、次のグリッドでは4列を使用し、6つのイメージをフィットするようにス ケーリングしています。



次にイメージが切り取られた同じグリッドを示します。



- b. 「アスペクト比」を指定して、グリッドのセルの形を決定します。
  - **四角形**: アスペクト比は 1:1 です。
  - **ランドスケープ**: アスペクト比は 16:9 です。
  - ポートレート: アスペクト比は 9:16 です。
  - カスタム: アスペクト比の独自の数値を指定します。
- **c. 「列」**の数を指定します。

列幅を揃えて作成するように、グリッドは自動的に調整されます。

- d. 行と列内のイメージ間のスペースを増やすまたは減らすには、「**イメージ間隔」**を指 定します。
- **10.** 指定したイメージ・サイズと幅に基づいて行と列にイメージを配置するには、「**列」**レイ アウトを選択します。



- a. 次の「スケーリング」オプションを選択して、グリッド内のイメージの表示を 調整します。
  - 切取り:2つの寸法(幅または高さ)の短い方が使用可能な領域にフィット するようにスケーリングされ、長い方の寸法はイメージをストレッチしな いで切り取られます。
  - **フィット**: イメージをゆがませずにイメージ全体が使用可能な領域にフィットするように、各イメージがスケーリングされます。
- **b. 「画像の高さ」**および「**イメージの幅」**を指定して、グリッドのセルの形を決定します。

指定した寸法でセルを作成するように、グリッドは自動的に調整されます。

- c. 行と列内のイメージ間のスペースを増やすまたは減らすには、「イメージ間隔」 を指定します。
- **11.「位置合わせ」、「幅」**および「間隔」オプションを指定して、スロット内にグリッドを配置します。

幅を使用して、スロット内のギャラリの幅(ピクセル単位)を指定します。幅を指定 する場合、「フィル」以外の位置合せオプションをクリックします。幅を設定した 後に、「フィル」を使用して指定した幅にイメージを拡大することができます。

### YouTube ビデオ

ページに動画および画像のエンゲージメントを追加するために、ストリーム YouTube ビデオを組み込みます。

YouTube でホストされているビデオ以外のビデオを使用するには、ビデオを参照して ください。

YouTube ビデオをページに追加するには:

- 1. 編集するページに移動し、 **し**が「**編集」**に設定されていることを確認します。
- ページにコンポーネントを追加します。コンポーネントでは、使用するビデオを選 択するまで、プレースホルダ・イメージが表示されます。
- 使用する YouTube ビデオを指定して、表示プロパティを調整するには、YouTube ビデオのメニュー・アイコン をクリックして、「設定」を選択します。
- **4.** 使用するビデオを検索し、URL をコピーして「YouTube URL」フィールドに貼り 付けます。
- 5. 次の表示オプションを任意で指定します。
  - **コントロールの表示**: 画面上およびデバイスのオプションを有効にして、ユー ザーが手動でビデオの再生を制御できるようにします。
  - **情報の表示**: ビデオの左上隅に一時的にビデオの説明を表示します。
  - 自動再生: ビデオの再生を自動的に開始します。
  - **ループ**: 完了後に自動的にビデオを繰り返します。
- 6. ビデオに定義された「アスペクト比」(自動)を使用するか、他のアスペクト比を選 択してビデオ表示の形を決定します。
- 7. 任意の位置合せまたは間隔オプションを指定してビデオを配置します。



## ビデオ

Oracle Content Management からビデオを含めて、動きや視覚的な連動をページに追加します。

Oracle Content Management は、ビデオをページに追加するためのいくつかのオプションを 提供しています。YouTube ビデオ(そのサイトでホストされている)を埋め込むことも、Oracle Content Management で格納、管理および提供されているビデオを追加して、Video Plus が提 供する自動コード変換および最適化されたストリーミングを利用することもできます。これ は、サイトが異なる機能を持つ複数のデバイスに提供される場合に便利です。システム管理 者が Video Plus を有効にしていない場合、標準のビデオが選択肢となります。

YouTube でホストされたビデオを使用するには、YouTube ビデオを参照してください。

**Oracle Content Management** ドキュメントまたはアセット・リポジトリに格納されているビデオを使用するには:

- 1. 編集するページに移動し、「編集」スイッチ( )が「編集」に設定されていることを確認します。
  - a. デジタル・アセットとして格納されたビデオを使用するには、「**アセット」**パネルを 開き、これによって使用するビデオを検索してから、これを選択してページにドラッ グします。



- b. または、コンポーネント・リストからビデオ・コンポーネントをページに追加します。コンポーネントでは、使用するビデオを選択するまで、プレースホルダ・イメージが表示されます。
- 「アセット」パネルからアセットをドラッグするか、コンポーネント・リストからビデオ・ コンポーネントをドラッグした場合、メニュー・アイコン()を使用してビデオの表示 プロパティを調整するか、ビデオ・コンポーネントのビデオに移動します。メニューをク リックして、「設定」を選択します。



- ビデオ・コンポーネントのビデオを選択するには、「選択」をクリックして「ドキュメント」および「デジタル・アセット」タブを開き、Oracle Content Management から追加するビデオに移動します。親エントリを選択して、「OK」をクリックします。
  - a. Video Plus が有効になっている場合、様々なリポジトリ、検索チャネル、コレ クションおよびキーワードを選択して、アセットをフィルタすれば、ビデオに 移動したときに検索を絞り込むことができます。
  - b. 標準のビデオを使用する場合、タイプ MP4 のビデオ・ファイルを選択します。 MP4 ビデオ形式は、サポートされているすべてのブラウザに共通です。また、 一部の標準のビデオには事前定義済のプレビュー・イメージがあります。プレ ビュー・イメージとして空白画面が表示されないものです。空白画面のかわり にグラフィックを使用する場合は、ビデオ設定ダイアログで「選択」をクリッ クし、Oracle Content Management からプレビュー・イメージを選択します。
- 4. 「アセット」パネルからビデオをドラッグするか、設定ダイアログから選択してビ デオを選択したら、設定ダイアログを使用して表示オプションを指定します:
  - **コントロールの表示**: 画面上およびデバイスのオプションを有効にして、ユー ザーが手動でビデオの再生を制御できるようにします。
  - 自動再生: ビデオの再生を自動的に開始します。
  - **ループ**: 完了後に自動的にビデオを繰り返します。
  - **ミュート**: ページのロード時にビデオを自動的にミュートします。
- 5. 任意の幅、位置合せまたは間隔オプションを指定してビデオを配置します。
- 6. 「スタイル」をクリックして、ビデオの枠線のカスタム・スタイルを選択または定 義します。

## ドキュメント・コンポーネント

ファイルおよびフォルダにアクセスし表示するためのコンポーネントを見てみましょう。

- ドキュメント
- フォルダ・リスト
- ファイル・リスト
- ドキュメント・マネージャ
- プロジェクト・ライブラリ

### ドキュメント

複数ページ・ドキュメントとスライド・プレゼンテーションをページから直接表示で きます。





ページにドキュメントを追加するには:

- 1. 編集するページに移動し、 *「編集」*に設定されていることを確認します。
- ページにコンポーネントを追加します。ドキュメント・コンポーネントには、表示するドキュメントを指定するまでプレースホルダ・イメージが表示されます。
- 3. ドキュメントを選択するには、そのメニュー・アイコンをクリックし、「設定」を選択して、「ドキュメント」フィールドの横の「選択」をクリックします。

ドキュメントは、自分がアクセスできるサイト・リポジトリまたは別のリポジトリに格納 されている必要があります。共有されているドキュメントまたはローカルまたはネット ワークのファイル場所からアップロードしたドキュメントを使用することもできます。

4. ドキュメントを選択して「OK」をクリックします。

#### 🖋 ノート:

ウィンドウに使用可能なすべてのファイルが表示されます。コンテキストに適切なファイル・タイプを選択する必要があります。たとえば、ドキュメント・ファイルを選択する場合は、TXT や DOC などの有効なドキュメント形式のファイルを選択する必要があります。

- 5. 「設定」パネルを使用して、キャプションを追加したり、間隔、位置合せ、スタイル、その他のプレゼンテーション・オプションを変更します。
- 6. ユーザーがドキュメント全体を移動しやすくするには:
  - a. 「ページ番号の表示」をクリックして、各ページの下にページ番号を表示します。
  - **b.**「ナビゲーション」方法を選択します。
    - サムネイル:ドキュメントの下にドキュメント内のページのリストが順番に表示 されます。ユーザーは、サムネイル・イメージをクリックして関連ページに移動 します。



- インデクサ:ドキュメントの下に一連のボタン
   を表示して、ドキュメント内の各ページを表します。ユーザーは、ボタンをクリックして 関連ページに移動します。
- なし: 視覚的なナビゲーションは表示されません。ユーザーは、左右にス ワイプして隣接ページを表示できます。この手動の方法は常に使用できま す。
- c. 「前/次を表示」をクリックして、ドキュメント内の前または次のページに進め る矢印アイコンを各ページに含めます。

### フォルダ・リスト

フォルダ・リストを使用して、Oracle Content Management アカウントから、指定したフォルダ内のフォルダをリストできます。

ページ上の1つ以上のファイル・リストまたはドキュメント・マネージャ・コンポー ネントとともにこのコンポーネントを使用すると、フォルダ・リストで選択されたフ ォルダのコンテンツをこれらのコンポーネントで自動的に表示できます。

ページにフォルダ・リスト・コンポーネントを追加するには:

- 1. 編集するページに移動し、 **し**が「**編集」**に設定されていることを確認します。
- 2. ページにコンポーネントを追加します。
- 3. コンポーネントとその外観を編集するには、メニュー・アイコン こを
  クリックして、「設定」を選択します。
- 4. 「カスタム設定」を選択して、表示されるコンテンツに関するデフォルトの詳細を 設定します。
- 5. 「フォルダの選択」の横にある「選択」をクリックし、表示に使用するフォルダを 変更します。ドキュメント・リポジトリのホーム・ページ以外のフォルダを選択す る必要があります。終了したら、「戻る」をクリックします。

#### 🖉 ノート:

フォルダ・リストでは、すべてのユーザーにダウンロード実行者アクセ ス権が付与されます。ユーザーは、自分のロールに関係なくファイルを 表示およびダウンロードできます。サイトのビジターがコンポーネント に指定された権限より上位の権限を持っている場合、その権限でコンポ ーネントに設定された権限がオーバーライドされます。

- 6. 次から選択して、表示されるコンテンツの追加のデフォルトを設定します。
  - デフォルト選択項目から選択:「Oracle Documents フォルダ」を選択した場合、リストに選択したとおりに表示されるようにフォルダを選択します(必要な場合)。
  - フォルダ名ヘッダーの表示: 埋込みコンポーネントのヘッダーにフォルダ名を 表示するには、これを選択します。
  - フォルダのソート: アイテムを最初にどのように表示するかを選択します(名前のアルファベット順またはアイテムの最終更新日順)。



- **サブフォルダの表示**: スライド・バーを使用して、表示されるサブフォルダの数を制限します。
- 7. 「一般」タブを使用して、間隔、位置合せおよびその他の表示オプションの変更を行いま す。
- 8. 「スタイル」タブを使用して、事前定義済スタイルまたは独自のカスタム選択でコンポー ネントを含むフレームをフォーマットします。

### ファイル・リスト

ファイル・リストを使用して、Oracle Content Management アカウントの指定したフォルダ からファイルのビューを表示できます。

ページ上の1つ以上のフォルダ・リスト・コンポーネントとともにこのコンポーネントを使 用すると、フォルダ・リストで選択されたフォルダのコンテンツをファイル・リスト・コン ポーネントで自動的に表示できます。また、ユーザーがリスト内のファイルをクリックした ときに1つ以上のアクションが実行されるように、コンポーネントを構成することもできま す。たとえば、選択したファイルを、別のブラウザ・ウィンドウ、"lightbox"オーバーレイ、 またはページ上のドキュメント・マネージャ・コンポーネントでプレビューできます。

ページにファイル・リスト・コンポーネントを追加するには:

- 1. 編集するページに移動し、 **し**が「**編集」**に設定されていることを確認します。
- 2. ページにコンポーネント・グループを追加します。
- 3. コンポーネントとその外観を編集するには、メニュー・アイコン = をクリックして、「設 定」を選択します。
- 4. 「カスタム設定」を選択して、表示されるコンテンツに関するデフォルトの詳細を設定し ます。
- 5. 「フォルダの選択」の横にある「選択」をクリックし、表示に使用するフォルダを変更し ます。ドキュメントのホーム・ページ以外のフォルダを選択する必要があります。終了し たら、「戻る」をクリックします。

#### 🖉 ノート:

ファイル・リストでは、すべてのユーザーにダウンロード実行者アクセス権が 付与されます。ユーザーは、自分のロールに関係なくファイルを表示およびダ ウンロードできます。サイトのビジターがコンポーネントに指定された権限よ り上位の権限を持っている場合、その権限でコンポーネントに設定された権限 がオーバーライドされます。

- 6. 次から選択して、表示されるコンテンツの追加のデフォルトを設定します。
  - 表示オプション: リストされたファイルに何の詳細が表示されるかを選択します。
    - フォルダ名ヘッダー: ヘッダーにフォルダ名を表示します。
    - ファイルの説明:指定された場合、ファイルの説明を表示します。
    - ファイル・セパレータ: それぞれのファイルを線で区切ります。
    - ダウンロード・アイコン:ダウンロード・アイコンを組み込んで、ユーザーが必要に応じてファイルをダウンロードできるようにします。

- 最終更新: ファイルが最後に更新された日付を表示します。
- **ファイル・サイズ**: ファイルのサイズを表示します。
- **イメージ**: ファイルのコンテンツのサムネイル・イメージを表示します。
- 「トリガー」&「アクション」:フォルダ・リスト・コンポーネントで選択されたフォルダのコンテンツとともにファイル・リストを自動リフレッシュするかどうかを選択します。ファイルが選択されている場合、ファイル選択済トリガーの有効化を選択することもできます。「リンク」タブを使用して、アクションとファイル選択済トリガーを関連付けます。
- ファイルのソート: アイテムの最初に表示される方法を選択します(名前のア ルファベット順またはアイテムの最終更新日順)。
- ファイルの表示: すべてのファイルを表示するか、スライダ・バーを使用して ファイルの数を制限するかを選択します。
- 7. 「一般」タブを使用して、間隔、位置合せおよびその他の表示オプションの変更を 行います。
- 8. 「スタイル」タブを使用して、事前定義済スタイルまたは独自のカスタム選択でコンポーネントを含むフレームをフォーマットします。
- 「リンク」タブを使用して、アクションとファイル選択済トリガーを関連付けます。 たとえば、選択したファイルを lightbox オーバーレイでプレビューするようにファ イル・リスト・コンポーネントを構成するには、次の手順を実行します。
  - a. ファイル・リスト設定ウィンドウで「ファイルが選択された際にトリガーをア クティブ化します」を選択していることを確認します。
  - b. ファイル・リスト設定の「リンク」タブで、**ファイル選択済**トリガーをクリックします。
  - c. 「Lightbox プレビュー」ページ・アクションをクリックしてアクション・リストにドラッグします。
  - d. 「ファイル ID」または「URL」フィールドで、「ファイル・リンク」を選択しま す。

「ファイル・リンク」では、すべてのサイト・ビジターがファイルをプレビュ ーし、必要に応じてダウンロードできるように、ダウンロード実行者権限を含 む参照リンクを使用します。「ファイル ID」を使用すると、メンバー・リンク が使用されます。メンバー・リンクを使用すると、登録済のユーザーのみが選 択されたファイルをプレビューし、必要に応じてダウンロードできます。

ユーザーがファイル・リスト内のファイルをクリックすると、グレー表示された非 アクティブなバージョンのページでファイル・プレビューが開きます。

#### ドキュメント・マネージャ

ドキュメント・マネージャを使用して、Oracle Content Management のホーム・ページまたはファイルのビューを提供できます。

ドキュメント・マネージャ・コンポーネントをページに追加するには:

- 1. 編集するページに移動し、 **し**が「**編集」**に設定されていることを確認します。
- 2. ページにコンポーネントを追加します。



- 3. コンポーネントとその外観を編集するには、メニュー・アイコンをクリックして、「設 定」を選択します。
- 4. 「カスタム設定」を選択して、表示されるコンテンツに関するデフォルトの詳細を設定し ます。
- 5. 「フォルダの選択」の横にある「選択」をクリックし、表示に使用するフォルダを変更し ます。

フォルダを選択するには、フォルダ名の横のボックスを選択します。フォルダを開くに は、フォルダ名をクリックします。パスのフォルダ名をクリックしてそのフォルダに戻る か、「ホームに移動」をクリックしてホーム・フォルダに戻ります。終了したら、「戻る」 をクリックします。

6. 「フォルダ・アクセスの選択」をクリックし、ビジターに付与するアクセス・ロールを選択します。

ビジターは、自分のロールおよび次の手順で有効にする**「参照オプション」**に基づいてフ ォルダ・コンテンツを表示および操作できます。

- **メンバーのアクセス**: ビジターは、会話、注釈またはアイテムのカスタム・プロパティの表示など、フォルダのメンバーが使用可能な任意の機能を使用できます。
- **参照者**: 参照者はファイルおよびフォルダを参照できますが、内容の変更はできません。
- **ダウンロード実行者**: ダウンロード実行者は、ファイルをダウンロードして自分のコンピュータに保存することもできます。
- **コントリビュータ**: コントリビュータは、ファイルの変更、ファイルの更新、新規フ ァイルのアップロード、およびファイルの削除を行うこともできます。

フォルダ・アクセスを設定する際には次の点を考慮してください。

- サイト作成者は、自分が持っているアクセス権より高いアクセス権をフォルダに付与 することはできません。たとえば、作成者がフォルダのダウンロード実行者アクセス 権を持っている場合、サイトのビジターにコントリビュータ権限を付与することはで きません。
- コンポーネントのフォルダに設定された権限で、ビジターの権限を補強できます。たとえば、ビジターにフォルダの参照者権限がある(または権限がない)場合、コンポーネントは選択されたロールに基づいてより高い権限を付与できます。これらの引き上げられた権限は、そのコンポーネント内でのみ有効です。
- サイトのビジターがコンポーネントに指定された権限より上位の権限を持っている 場合、その権限でコンポーネントに設定された権限がオーバーライドされます。
- フォルダに付与された権限は、そのフォルダ内にネストしているフォルダおよびファ イルに適用されます。
- 7. 次から選択して、表示されるコンテンツの追加のデフォルトを設定します。
  - レイアウト: フォルダおよびファイルの初期グリッド、リストまたは圧縮リスト・レイアウトを選択します。ユーザーは、完了した埋込みコンポーネントを表示する際にレイアウトを変更できます。
  - **カラー・スキーム**: 埋込みフォルダ・リストに使用可能なカラー・スキームの1つを 選択します。
  - **ソート順序**: アイテムが最初にどのように表示されるか(名前のアルファベット順、またはアイテムの最終更新日時順)を選択します。



- 参照オプション: アイテムの選択時にユーザーが使用できるオプションを選択します。たとえば、ユーザーにファイルの表示、ファイルのダウンロード、ファイルとフォルダの共有またはコピー、またはファイルの削除を許可することを選択できます。ユーザーがファイルおよびフォルダで実行できる操作を制限する場合は、このリストのオプションを選択解除します。前の手順で「メンバーのアクセス」を選択した場合は、会話、注釈またはカスタム・プロパティが表示されるサイド・ペインを表示することを選択できます。
- ビューア・オプション: ユーザーがファイルを表示する方法を選択します。埋込みフレーム内または別のタブ(またはブラウザ設定によってはウィンドウ)でファイルを表示することを許可できます。サムネイルの表示または非表示、ビデオの表示方法のカスタマイズも行うことができます。
- ズーム・コントロールの表示: フォルダの埋込みビューにスライダ・バーまた はズーム・コントロールを表示するかどうかを選択します。
- ビューアのフィット・モード: ファイルをページー杯、ページの幅一杯または 元のサイズで表示することを選択できます。
- トリガーとアクション: フォルダ・リスト・コンポーネントでも使用されている場合に、ファイル・リストをリフレッシュすることを選択します。
- 8. 「一般」タブを使用して、間隔、位置合せおよびその他の表示オプションの変更を 行います。
- 9. 「スタイル」タブを使用して、事前定義済スタイルまたは独自のカスタム選択でコ ンポーネントを含むフレームをフォーマットします。

ドキュメント・マネージャが追加された後、参照者はサイトのフレームに埋め込まれ ている選択されたフォルダのビューを参照します。ユーザーは、提供された表示オプ ションを使用して、フォルダとファイルのリスト方法を変更できます。ユーザーがフ ァイルまたはフォルダを選択すると、メニュー・バーまたは右クリック・メニューで 表示されるオプションを選択して、指定したロールおよび「参照オプション」で許可 されたアクションを実行できるようになります。

会話がアイテムに関連付けられている場合は、アイテムとともに が表示されます。 これをクリックすると、注釈とコメントを表示できる会話ペインが開きます。独立し た会話(フォルダに関連付けられていない会話)は会話コンポーネントを使用して追加 する必要があります。

アイテムにカスタム・プロパティがある場合は、ペインでプロパティを表示できます。 「その他」をクリックし、「カスタム・プロパティ」を選択してプロパティ・ペインを 開きます。

### プロジェクト・ライブラリ

プロジェクト・ライブラリを使用して、元のフォルダまたはその場所を変更する必要 なくリポジトリ内の別の場所からフォルダをアセンブルできます。プロジェクト・ラ イブラリ・コンポーネントを使用して、各フォルダに異なる権限レベルを割り当て、 様々なプロジェクト・チーム・ロールに対応することもできます。

たとえば、プロジェクト・ライブラリ内のフォルダにビジター・ロール(参照者、ダウ ンロード実行者またはコントリビュータ)の1つを付与する場合、ビジターはロールに 関連付けられている権限でフォルダ・コンテンツを参照し、操作できます。メンバー・ アクセスを指定した場合、メンバーのみがプロジェクト・ライブラリにリストされて いる内容を参照できます。メンバーは、元のフォルダで自分に対して指定されている 権限でフォルダ・コンテンツを操作します。



このコンポーネントをページ上の1つ以上のファイル・リストまたはドキュメント・マネー ジャ・コンポーネントとともに使用する場合、これらのコンポーネントはプロジェクト・ラ イブラリで選択されたフォルダのコンテンツを自動的に表示できます。

プロジェクト・ライブラリ・コンポーネントをページに追加するには:

- 1. 編集するページに移動し、 *「編集」*に設定されていることを確認します。
- 2. ページにコンポーネントを追加します。
- 3. コンポーネントとその外観を編集するには、メニュー・アイコン をクリックして、「設 定」を選択します。
- 4. 「カスタム設定」を選択して、会話の選択と表示オプションの選択を行います。
- タイトルを指定します。デフォルト・タイトルは「プロジェクト・ライブラリ」です。下の「タイトル」表示オプションを選択または選択解除することで、タイトルを表示するか 非表示にするかを選択できます。
- 6. カラー・スキームを選択します。

#### 🖉 ノート:

トリガーとアクションを使用してプロジェクト・ライブラリをドキュメント・ マネージャ・コンポーネントに関連付ける場合、プロジェクト・ライブラリに 選択したカラー・スキームは、ドキュメント・マネージャ・コンポーネントで 表示するように選択されたフォルダにも適用されます。これは、プロジェクト・ ライブラリをフォルダ・リストまたはファイル・リスト・コンポーネントに関 連付ける場合には当てはまりません。これらのコンポーネントはカラー・スキ ームをサポートしていないためです。

- 7. フォルダのリストの横の「追加」をクリックして、既存のフォルダを追加または新規フォ ルダを作成します。
  - a. 1つ以上のフォルダに移動して選択するか、「作成」をクリックしてフォルダを作成 します。このウィンドウで選択する各フォルダは、プロジェクト・ライブラリに追加 されます。

表示オプションやソート・オプションなど、ツール・バーにリストされている任意の オプションを使用できます。

- **b.** 終了したら、**「戻る」**をクリックします。
- リストを整理するには、フォルダを選択し、「上に移動」または「下に移動」をクリック してリストの順序内で場所を変更するか、「削除」をクリックしてフォルダをリストから 削除します。
- ユーザーのロールに基づいてフォルダへのアクセス権を指定するには、フォルダのリストからフォルダを選択し、「フォルダ権限の選択」をクリックしてアクセス・ロールを選択します。
  - メンバーのアクセス:フォルダに対する権限を持つ登録済ユーザーのみがプロジェクト・ライブラリにリストされているフォルダを参照できます。メンバーは、元のフォルダで自分に対して指定されている権限でフォルダ・コンテンツを操作します。
  - 参照者: 参照者はファイルおよびフォルダを参照できますが、内容の変更はできません。



- **ダウンロード実行者**: ダウンロード実行者は、ファイルをダウンロードして自 分のコンピュータに保存することもできます。
- コントリビュータ:コントリビュータは、ファイルの変更、ファイルの更新、 新規ファイルのアップロード、およびファイルの削除を行うこともできます。

フォルダ・アクセスを設定する際には次の点を考慮してください。

- サイト作成者は、自分が持っているアクセス権より高いアクセス権をフォルダ に付与することはできません。たとえば、作成者がフォルダのダウンロード実 行者アクセス権を持っている場合、サイトのビジターにコントリビュータ権限 を付与することはできません。
- コンポーネントのフォルダに設定された権限で、ビジターの権限を補強できます。たとえば、ビジターにフォルダの参照者権限がある(または権限がない)場合、コンポーネントは選択されたロールに基づいてより高い権限を付与できます。これらの引き上げられた権限は、そのコンポーネント内でのみ有効です。
- サイトのビジターがコンポーネントに指定された権限より上位の権限を持っている場合、その権限でコンポーネントに設定された権限がオーバーライドされます。
- フォルダに付与された権限は、そのフォルダ内にネストしているフォルダおよびファイルに適用されます。
- 「メンバーのアクセス」権限があるフォルダの場合、「ドキュメント・マネージャで 会話ペインを表示」をクリックしてフォルダまたはコンテンツに関連付けられてい る会話を表示できます。

このオプションを選択し、ページでドキュメント・マネージャ・コンポーネントを 構成して選択されたフォルダを表示する場合、ユーザーは会話アイコンをクリック してフォルダに関連付けられている会話を表示できます。このオプションを選択 しない場合、会話アイコンは表示されません。

- 11. 終了したら、ウィンドウを閉じます。
- **12.**「一般」タブを使用して、間隔、位置合せおよびその他の表示オプションの変更を 行います。
- **13.**「スタイル」タブを使用して、事前定義済スタイルまたは独自のカスタム選択でコンポーネントを含むフレームをフォーマットします。
- 14.「リンク」タブを使用して、「選択されたフォルダ」トリガーにアクションを関連付けます。たとえば、ドキュメント・マネージャ・コンポーネントもページに追加する場合、ドキュメント・マネージャ・コンポーネントを使用して、プロジェクト・フォルダで選択されたフォルダのコンテンツを表示できます。
  - a. プロジェクト・ライブラリ設定の「リンク」タブで、「選択されたフォルダ」 トリガーをクリックします。
  - **b.** 「トリガー・アクションの構成」ウィンドウで、「Documents マネージャ」を クリックします。
  - c. 「ドキュメントの表示」アクションをクリックしてアクション・リストにドラ ッグします。
  - d. 「フォルダ ID」または「URL」フィールドで、**「選択したフォルダ」**を選択しま す。

ユーザーがプロジェクト・ライブラリのフォルダをクリックすると、フォルダ・コ ンテンツがページのドキュメント・マネージャ・コンポーネントに表示されます。

## ソーシャル・コンポーネント

ソーシャル・コンポーネントを使用すると、ユーザーが相互のつながりを維持し、コミュニ ケーションを取ることができます。

- ソーシャル・バー
- Facebook の「いいね!」と「おすすめ」
- Twitter の共有とフォロー
- 会話コンポーネント
- 会話リスト

### ソーシャル・バー

**Facebook** や Twitter などの人気ソーシャル・メディアのアイコンおよびリンクが簡単に追加 されます。ソーシャル・バーにはデフォルトでいくつかのターゲットが含まれていますが、 ソーシャル・バーに対してアイテムを追加および削除できます。

SOCIAL BAR	Ξ
f У in 🕫 🖸	

ソーシャル・バーを追加および変更するには:

- 1. 編集するページに移動し、 いうが「編集」に設定されていることを確認します。
- ページにコンポーネントを追加します。ソーシャル・バーには、デフォルトで含まれているアイコンが表示されます。
- ソーシャル・バーのアイコンを追加または変更するには、そのメニュー・アイコン =を クリックし、「設定」を選択して、「一般」タブの上部にある「アイコン」をクリックしま す。

ソーシャル・アイコンを削除するには、名前の横の「X」をクリックします。アイコンを 追加する場合、自分がアクセス権を持つサイト・リポジトリまたは別のリポジトリにその アイコンが格納されている必要があります。共有されているイメージまたはローカルま たはネットワークのファイル場所からアップロードしたイメージを使用することもでき ます。

- 4. ローカルまたはネットワークの場所から1つ以上のアイコンをアップロードするには:
  - a. パネルの上部にある「アイコンの追加」をクリックします。
  - **b.** イメージを格納するリポジトリの場所にナビゲートするか、「キをクリックして現在の場所に新規フォルダを追加します。
  - **c. ふ** をクリックします。
  - d. イメージ・ファイルを検索して選択し、「開く」をクリックします。
     イメージ・ファイルがリポジトリの現在の場所にアップロードされます。
- 5. リポジトリの1つ以上のイメージを選択し、「OK」をクリックします。



- a. 使用するイメージを見つけてクリックします。
- b. リポジトリ内のファイルにリンクするには、「サイトにファイルをコピーする かわりに元のファイルへの参照を使用します」を選択します。このオプション を選択しない場合、ファイルのコピーがサイトで格納され、サイトから参照さ れます。元のファイルにリンクすることで、コンテンツを複製せずに済みま す。リンクを使用すれば、ファイルの権限が変更されたり、表示が制限される ような場合でも、サイトのビジターがコンテンツを表示できます。
- **c.**「OK」をクリックします。

選択したイメージは、イメージのリストに追加されます。イメージをドラッグ・ア ンド・ドロップしてリスト内のイメージの順序を変更します。各イメージのデフォ ルトのタイトルは、拡張子のないファイル名です。

**6.** 特定のイメージのターゲット URL、タイトル、説明またはその他のオプションを 変更するには、リストのイメージをクリックして変更を加えます。

個々のアイコンの更新が完了したら、**「戻る」**をクリックして一般オプションを指 定します。

7. ソーシャル・バーのすべてのアイコンに対してサイズ、間隔、方向および位置合せ を指定します。

### Facebook の「いいね!」と「おすすめ」

Facebook の「いいね!」ボタンをページに追加すると、参照者がサイトを気に入った ことを容易に Facebook で表現できるようになります。

- 1. 編集するページに移動し、 **し**が「**編集」**に設定されていることを確認します。
- 2. ページにコンポーネントを追加します。
- アプリケーションとその外観を編集するには、アプリケーションのメニュー・アイ コン
   たクリックして、「設定」を選択します。
- Facebook ページのサイトにリンクをポストしやすいように、Facebook いいね設定または Facebook おすすめ設定を選択して URL を設定し、オプションで「共有」ボタンを追加します。
- 5. 「一般」タブを使用して、間隔、位置合せおよびその他の表示オプションの変更を 行います。
- 6. 「スタイル」タブを使用して、事前定義済スタイルまたは独自のカスタム選択でア プリケーションを含むフレームをフォーマットします。

### Twitter の共有とフォロー

参照者が Twitter アカウントですぐにリンクを共有できるように、「Twitter 共有」ボタ ンをページに追加できます。

- 1. 編集するページに移動し、 **し**が「**編集」**に設定されていることを確認します。
- 2. ページにコンポーネントを追加します。
- アプリケーションとその外観を編集するには、アプリケーションのメニュー・アイ コンをクリックして、「設定」を選択します。



- Twitter フォロー設定を選択して、Twitter のユーザー名を設定し、ユーザー名を表示する かどうかを選択して、「フォロー」ボタンのサイズを選択します。Twitter 共有設定を選択 して、次のデフォルトを設定します。ユーザーはボタンを使用する際に、値を変更できま す。
  - 共有 URL: ユーザーが共有できるサイトの URL。
  - **ツイート・テキスト**:ページに関するツイートのテキスト。
  - 経由@: ツイートに使用された Twitter ユーザー・アカウント。
  - レコメンデーション@: Twitter のお薦めのために使用された Twitter ユーザー・アカウント。
  - ハッシュタグ#: ポストに使用するハッシュタグ。
  - 件数: 共有件数の表示(垂直または水平)。
  - **大きいボタン**: アプリケーションで大きいボタンを使用するか小さいボタンを使用す るかを選択します。
- 5. 「一般」タブを使用して、間隔、位置合せおよびその他の表示オプションの変更を行いま す。
- 6. 「スタイル」タブを使用して、事前定義済スタイルまたは独自のカスタム選択でアプリケ ーションを含むフレームをフォーマットします。

### 会話コンポーネント

会話を使用して、あるトピックに関するディスカッションをサイトから直接促進できます。

🖉 ノート: サイトで会話が機能するには、指定されたユーザーまたは Oracle Content Management ユーザー・ロールを持つユーザーに限られたセキュアなサイトである ことが必要です。サイト・セキュリティの変更を参照してください。

ページに会話コンポーネントを追加するには:

- 1. 編集するページに移動し、 **し**が「**編集」**に設定されていることを確認します。
- 2. ページにコンポーネントを追加します。
- 3. コンポーネントとその外観を編集するには、メニュー・アイコン <br/>
  をクリックして、「設定」を選択します。
- 4. 「カスタム設定」を選択して会話を選択し、プレゼンテーション・カラー・スキームを選択します。
- 5. 「会話の選択」の横にある「選択」をクリックして既存の会話を選択するか、新しい会話 を作成します。
  - a. 使用可能な会話のリストから会話を選択するか、「作成」をクリックして新しい会話 を作成して名前を付けます。

リストには、アクセスできる会話がすべて含まれています。 リストはフィルタできます。



- すべて: ミュートとマークされたものを除くすべての会話が表示されます。
- **お気に入り**: お気に入りとマークされた会話が表示されます。
- ミュート: ミュートされ、会話のリストから除外された会話が表示されます。
- **クローズ**: クローズとマークされた会話がすべて表示されます。

リストをソートすることもできます。

- 最終更新: 更新時間が最も新しいものから最も古いものの順に会話をリストします。
- 名前: アルファベット順に会話をリストします。
- **未読**: 未読の会話を先に(未読のコメントが最も多いものから最も少ないものの順に)リストします。
- b. 終了したら、「戻る」をクリックします。
- c. 独立した会話(ファイルやフォルダが基になっていないもの)を選択した場合 は、その会話を表示するユーザーにアクセス権限を設定できます。
- d. カラー・スキームを選択します。
- e. ページ上の会話リスト・コンポーネントとともに会話コンポーネントを使用す るには、「会話リスト・コンポーネントでの選択に基づいて会話を自動リフレ ッシュします」を選択します。

ユーザーがリストから会話を選択すると、その会話が会話コンポーネントに表 示されます。

- f. 終了したら、ウィンドウを閉じます
- 6. 「一般」タブを使用して、間隔、位置合せおよびその他の表示オプションの変更を 行います。
- 7. 「スタイル」タブを使用して、事前定義済スタイルまたは独自のカスタム選択でコ ンポーネントを含むフレームをフォーマットします。

公開されたサイトでは、ビジターにはサイトのフレームに会話が表示されます。ユー ザーは会話内を移動してコメントを読んだり、コメントに返答できます。

### 💉 ノート:

ビジターが会話のメンバーとして明示的に追加されていない場合、コメントを読んだり、コメントに返答したりできますが、下の図の1から4までのア イテムは表示されません。

		0 0	23	
Site	Feedback	3	Ð	
	tenant1,admina 2 hours ago I'd like to get your feedback on the your comments here.	e new site! Please p	♥	-4
	S Reply 🕐 Like 🚥 More			-5
B I	<u>U</u>   <u>Z</u> =	Ø	Post	-6

- メニュー・バー(1)には、会話の各参加者に関する情報が含まれます。ユーザー・アイコンをクリックし、ステータス情報を表示たり、ユーザーと協働するためのオプションを表示します。数字のアイコンには、総参加者数が表示されます。アイコンをクリックすると、完全なリストが表示されます。
- 「ユーザーの追加」(2)をクリックし、ユーザーを会話に追加します。
- 「その他のオプション」(3)をクリックし、会話を操作するためのオプションのリストを表示します。オプションのリストは、ユーザーのロールによって異なります。フォルダおよびファイルに関連付けられている会話の場合、オプションのリストは、フォルダまたはファイルに設定されている権限によっても異なります。
- 「**フラグ」(4)**をクリックし、通知フラグを割り当てることで特定のユーザーにアラートを 通知します。ユーザーは、プリファレンスで指定したとおりに電子メールでアラートが通 知されます。
- コメント・オプション(5)を使用して、会話内の特定のコメントに対するアクション(コメントへの返信または LIKE、コメントの編集または削除など)を実行します。
- コメントを追加または編集する場合、編集オプション(6)を使用して太字や下線などの基本的な書式を追加または削除したり、添付を追加したり、会話にコメントを投稿します。 サイトのビジターが会話のメンバーでない場合、添付を追加できません。

### 会話リスト

会話を使用して、あるトピックに関するディスカッションをサイトから直接促進できます。

ページ上の1つ以上の会話リスト・コンポーネントとともにこのコンポーネントを使用する と、会話リストで選択された会話のコンテンツを会話リスト・コンポーネントで自動的に表 示できます。 💉 ノート:

サイトで会話または会話リストが機能するには、指定されたユーザーまたは Oracle Content Management ユーザー・ロールを持つユーザーに限られたセ キュアなサイトであることが必要です。サイト・セキュリティの変更を参照 してください。

ページに会話リスト・コンポーネントを追加するには:

- 1. 編集するページに移動し、 「編集」に設定されていることを確認します。
- 2. ページにコンポーネントを追加します。
- 3. コンポーネントとその外観を編集するには、メニュー・アイコン こをクリックして、「設定」を選択します。
- 4. 「カスタム設定」を選択して、会話の選択と表示オプションの選択を行います。
- 5. タイトルを指定します。デフォルトのタイトルは「会話リスト」です。下の「タイ トル」表示オプションを選択または選択解除することで、タイトルを表示するか非 表示にするかを選択できます。
- 6. 会話のリストの横にある「追加」をクリックして、既存の会話を追加するか、新し い会話を作成します。
  - a. 使用可能な会話のリストから1つ以上の会話を選択するか、「作成」をクリックして新しい会話を作成し、名前を付けます。
  - b. 終了したら、「戻る」をクリックします。
  - c. 会話リストを表示するユーザーに許可する権限を選択します。
- 会話リストを編成するには、会話を選択して、リスト順序の位置を変更する場合は 「上に移動」または「下に移動」をクリックし、リストから会話を削除する場合は 「削除」をクリックします。
- 8. カラー・スキームを選択し、追加の表示オプションを選択します。選択した表示オ プションが、リストの会話の名前の下に表示されます。
- 9. 終了したら、ウィンドウを閉じます。
- **10.**「一般」タブを使用して、間隔、位置合せおよびその他の表示オプションの変更を 行います。
- **11.**「スタイル」タブを使用して、事前定義済スタイルまたは独自のカスタム選択でコンポーネントを含むフレームをフォーマットします。

次のイメージは、すべての表示オプションが選択された Session Feedback というタイトルの会話リストを示しています。



Session	Feedback	<b>^</b>
Session_1 Posts 1	Unread 0	Ξ
Updated 10/21/	2016 by tenant1.admina	
Session_2		
Posts 0	Unread 0	-

# プロセス・コンポーネント

プロセス・コンポーネントを使用すると、サイト・ユーザーは事前定義済プロセスのタスク を開始および管理できます。

🎤 ノート:

プロセス・コンポーネントを使用するには、Oracle Process Cloud Service リリース 17.1.3 以降を使用する必要があります。

- プロセス開始フォーム
- プロセス・タスク・リスト
- タスク詳細フォーム

### プロセス開始フォーム

プロセス開始フォームを使用して、Oracle Process Cloud Service で定義されているプロセス を開始できます。

サイトでプロセス開始フォームが機能するには、次の条件に該当する必要があります。

- 関連付けられたプロセスおよびプロセス開始フォームをこのコンポーネントで表示する 前に、それらが Oracle Process Cloud Service で定義されている必要があります。Oracle Integration でのプロセスの使用の構造化プロセスの開発を参照してください。
- プロセス開始フォームを使用するには、開始フォームが含まれるプロセス・スイムレーン に関連付けられたロールがユーザーに割り当てられている必要があります。
- 管理者は、Oracle Process Cloud Service と Oracle Content Management の間で統合を設定する必要があります。Administering Oracle Content Management の Oracle Process Cloud Service との統合を参照してください。この2つのサービス間の統合では SSO を行うため、どちらのサービスも同じアイデンティティ・ドメイン内に存在している必要があります。

プロセス開始フォーム・コンポーネントをページに追加するには:

- 1. 編集するページに移動し、 ばい 「編集」に設定されていることを確認します。
- 2. ページにコンポーネントを追加します。



- 3. コンポーネントとその外観を編集するには、そのコンポーネントのメニュー・アイ コン をクリックして、「設定」を選択します。
- 4. 「カスタム設定」を選択して、プロセス開始フォームを選択し、フォームのデフォ ルトを設定します。
  - a. サイトがパブリック・サイトの場合、プロキシ・サービスを選択します。
  - b. 開始フォームのパーティションを選択します。テスト・パーティションを使用して、プロセスが計画どおり機能しているかどうかを検証できます。または、本番パーティションを使用して、汎用のプロセスをデプロイできます。
  - c. デフォルトとして選択されているプロセスのバージョンを常に使用する場合、 「デフォルトのプロセス・バージョンを使用」を選択します。このオプション を選択しない場合は、特定のバージョンを選択します。プロセスが更新された ら、これらの設定の選択を更新する必要があります。
  - d. プロセスを選択します。
  - e. 開始フォームを選択します。フォームの構文は、プロセス・タイプ:バージョン:プロセス名:開始です。たとえば、Basic
     Approval:1.0:Process:Start Basic Approvalのようになります。

ウィンドウ上部のメッセージに「Process Cloud Service 接続がありません」 と示されている場合、Oracle Process Cloud Service と Oracle Content Management の間の統合が構成されていない可能性があります。管理者に連 絡してください。

プロセスがリストに表示されるには、プロセス作成者によってあなたがプロセ スの開始者として追加される必要があります。プロセス作成者によってすべ てのサイト・ビジターがプロセスの開始者として追加される必要があります。 そうでない場合、ビジターはフォームを完成できますが、プロセスを開始でき ません。

- f. 次から選択して、フォームの追加のデフォルトを設定します。
  - フォーム・タイトル:オプションで、デフォルトのフォーム・タイトルを 独自のタイトルで置換します。
  - 送信ボタン名: オプションで、「送信」ボタンの名前を、指定した値に変更 します。
  - 「送信」ボタンの表示:オプションで、コンポーネントの「送信」ボタンを 表示または非表示にします(デフォルトでは表示されます)。類似するボタ ンがフォーム自体に提供されている場合は、コンポーネントの「送信」ボ タンを非表示にできます。
  - 送信確認: オプションで、デフォルトの確認メッセージを独自のメッセージで置換します。
  - 送信確認の表示:オプションで、「送信」ボタンがクリックされたときの確認メッセージを表示または非表示にします。デフォルトでは表示され、プロセス開始フォーム・コンポーネント内に表示されます。
  - 「保存」ボタンの表示:オプションで、コンポーネントの「保存」ボタンを 表示または非表示にします(デフォルトでは非表示になります)。関連付け られたプロセスにより、作業を保存して後で戻ることが許可されている場 合は、コンポーネントの「保存」ボタンを表示できます。
  - 「破棄」ボタンの表示: オプションで、コンポーネントの「破棄」ボタンを 表示または非表示にします(デフォルトでは非表示になります)。ユーザー

がフォーム・コンテンツを破棄して最初からやり直せるようにする場合は、**「破 棄」**ボタンを表示できます。

- 添付の表示: オプションで、1つまたは複数のファイルをアップロードできるフォ ーム上の「添付」領域を表示または非表示にします(デフォルトでは非表示になり ます)。アップロードされたファイルは、Oracle Process Cloud Service にプロセ スの一部として格納されます。
- デフォルト値のカスタマイズ:開始フォームに事前移入する名前と値のペアを設定できます。「フィールドの追加」をクリックし、フィールドの名前と値を入力します。名前は、フォームで使用されるラベルではなくフォーム・フィールドの1つであり、値はそのフィールドで許容される値です。現在ログインしているユーザーのデータを挿入するには、特殊な値%%username%%および%%userid%%を使用します。
- 5. 「一般」タブを使用して、間隔、位置合せおよびその他の表示オプションの変更を行いま す。
- 6. 「スタイル」タブを使用して、事前定義済スタイルまたは独自のカスタム選択でコンポー ネントを含むフレームをフォーマットします。
- 7. 「リンク」タブを使用して、コンポーネントによって提供されるトリガーにアクションを 割り当てます。
  - 送信された開始フォーム: このトリガーは、ユーザーが「送信」ボタンをクリックしたときに発生します。
  - **保存された開始フォーム**: このトリガーは、ユーザーが「保存」ボタンをクリックしたときに発生します。
  - **破棄された開始フォーム**: このトリガーは、ユーザーが「**破棄」**ボタンをクリックしたときに発生します。

トリガーをクリックしてアクションを割り当てます。トリガーとアクションの割当ての 詳細は、トリガーおよびアクションの使用を参照してください。

### プロセス・タスク・リスト

プロセス・タスク・リストを使用して、Oracle Process Cloud Service で定義されているプロ セスを選択的にリストできます。タスク・リスト・コンポーネントでタスクの詳細情報を表 示できます。あるいは、プロセス・タスク・リスト・コンポーネントをタスク詳細コンポー ネントと組み合せて使用し、リストを簡略化したり、選択したタスクのみの詳細情報を表示 したりできます。

サイトでプロセス・タスク・リストが機能するには、次の条件に該当する必要があります。

- 関連付けられたプロセスをこのコンポーネントとともに表示できるようにするには、 Oracle Process Cloud Service でそれらを定義しておく必要があります。Oracle Integration でのプロセスの使用の構造化プロセスの開発を参照してください。
- サイトは、指定されたユーザーまたは Oracle Content Management ユーザー・ロールを 持つユーザーに限られたセキュアなサイトである必要があります。サイト・セキュリティ の変更を参照してください。
- 管理者は、Oracle Process Cloud Service と Oracle Content Management の間で統合を設定する必要があります。Administering Oracle Content Management の Oracle Process Cloud Service との統合を参照してください。この2つのサービス間の統合では SSO を行うため、どちらのサービスも同じアイデンティティ・ドメイン内に存在している必要があります。


プロセス・タスク・リスト・コンポーネントをページに追加するには:

- 1. 編集するページに移動し、 ばい 「編集」に設定されていることを確認します。
- 2. ページにコンポーネントを追加します。
- 3. コンポーネントとその外観を編集するには、そのコンポーネントのメニュー・アイ コン をクリックして、「設定」を選択します。
- 4. 表示オプションの設定および使用可能なタスクのフィルタを行う場合は、「カスタ ム設定」を選択します。
- 5. 次のいずれかを選択して、タスク・リストの表示オプションを設定します。

🖋 ノート:

コンポーネントの表示オプションを設定できますが、タスクそのものの 表示は Oracle Process Cloud Service の設計によって決定されます。

- 詳細の表示: このオプションを選択すると、タスク・リストそのものにタスク 詳細情報が含められます。プロセス・タスク・リスト・コンポーネントをタス ク詳細コンポーネントと組み合せて使用して、リストを簡略化したり、選択し たタスクのみの詳細情報を表示したりできます。
- 検索の表示: このオプションを選択すると、タスク・リストの上部に検索バーが表示されます。「検索の表示」を選択すると、検索バーにフィルタ・オプションを表示するためのオプション「「フィルタ」を表示」が追加されます。次のフィルタ・オプションを使用して、デフォルトのフィルタ値を設定します。
- **すべて選択を表示**: このオプションを選択すると、ユーザーは表示されている すべてのタスクを処理のために選択できます。
- ページ・サイズ: これを調整して、表示するタスクの最大数を指定します。指定した最大数よりもタスクの数が少ない場合、コンポーネントは自動的に小さい方の数に調整されます。指定した最大数よりもタスクの数が多い場合、コンポーネントには最大数が表示され、その他のページへのリンクが追加されます。
- 6. 次のいずれかを選択して、リスト内の使用可能なタスクをフィルタします。

適用したフィルタによって、タスクの最初のリストが決定されます。前述の「検索 の表示」オプションと「「フィルタ」を表示」オプションを選択した場合、設定を 調整して、「元ユーザー」を除くすべての設定のフィルタを変更できます。

- 検索キーワード:オプションで、1つ以上の検索語を指定します。検索結果には、指定したキーワードのすべてがタイトルに含まれるタスクのみが表示されます。「検索の表示」を選択した場合、これらのキーワードが検索バーに表示され、ユーザーはこれらのキーワードを削除したり変更したりできます。
- ステータス: 使用可能なステータス値のいずれかを選択します。デフォルトは 「割当て済」です。
- 割当て先:使用可能な割当て先の値のいずれかを選択します。デフォルトは 「自分と自分のグループすべて」であり、ユーザーとそのグループが使用可能 なすべてのタスク(要求したが処理不可能なタスクを含む)が含まれます。

- 元ユーザー:オプションで、タスクを開始する1人以上のユーザーを選択します。ユ ーザー名の入力を開始すると、関連付けられた Oracle Process Cloud Service で使用 可能なユーザーの検索が始まります。指定した値はエンド・ユーザーに表示されず、 削除できません。ユーザーは、ユーザー名を追加してユーザーのリストをさらに拡張 できます。
- **期日**: オプションで、カレンダから期日(「指定日」、「次より前」、「次より後」または 「次の間」)を選択します。
- アプリケーション:オプションで、特定のアプリケーションの名前を選択します。ユ ーザーに使用可能なアプリケーションのリストから、1つ以上のアプリケーションを 選択できます。
- 7. 「一般」タブを使用して、間隔、位置合せおよびその他の表示オプションの変更を行いま す。
- 8. 「スタイル」タブを使用して、事前定義済スタイルまたは独自のカスタム選択でコンポー ネントを含むフレームをフォーマットします。

### / ノート:

ページ上にプロセス・タスク・リストとタスク詳細フォームがある場合、タスク詳細コンポーネントにより、リストから選択したタスクの詳細が自動的に表示されます。トリガーおよびアクションの詳細は、「トリガーおよびアクションの使用」を参照してください。

# タスク詳細フォーム

タスク詳細フォームをプロセス・タスク・リストと組み合せて使用すると、選択したタスク の詳細を表示できます。ページ上にプロセス・タスク・リストとプロセス詳細フォームがあ る場合、プロセス・タスク詳細コンポーネントにより、リストから選択したタスクの詳細が 自動的に表示されます。

サイトでタスク詳細フォームが機能するには、次の条件に該当する必要があります。

- 関連付けられたプロセスをこのコンポーネントとともに表示できるようにするには、
   Oracle Process Cloud Service でそれらを定義しておく必要があります。Using
   Processes in Oracle Integration の構造化プロセスの開発を参照してください。
- サイトは、指定されたユーザーまたは Oracle Content Management ユーザー・ロールを 持つユーザーに限られたセキュアなサイトである必要があります。サイト・セキュリティ の変更を参照してください。
- 管理者は、Oracle Process Cloud Service と Oracle Content Management の間で統合を設定する必要があります。Administering Oracle Content Management の Oracle Process Cloud Service との統合を参照してください。この2つのサービス間の統合では SSO を行うため、どちらのサービスも同じアイデンティティ・ドメイン内に存在している必要があります。

タスク詳細コンポーネントをページに追加するには:

- 1. 編集するページに移動し、 しが「編集」に設定されていることを確認します。
- 2. ページにコンポーネントを追加します。



- 3. コンポーネントとその外観を編集するには、そのコンポーネントのメニュー・アイ コン=をクリックして、「設定」を選択します。
- 4. 「カスタム設定」を選択して、表示オプションを設定します。
- 5. 次のいずれかを選択して、タスク詳細フォームの表示オプションを設定します。

✓ ノート:

コンポーネントの表示オプションを設定できますが、タスク詳細の表示 は Oracle Process Cloud Service の設計によって決定されます。

- 「アクション」の表示: このオプションを選択すると、ユーザーが選択可能なアクション(「承認」、「却下」など)が表示されます。
- 「保存」の表示: このオプションを選択すると、「保存」ボタンが表示されます。
- 「閉じる」の表示: このオプションを選択すると、「閉じる」ボタンが表示されます。
- 添付の表示: このオプションを選択すると、詳細フォームに「添付」セクションが表示されます。
- コメントの表示: このオプションを選択すると、詳細フォームに「コメント」 セクションが表示されます。
- **履歴の表示**: このオプションを選択すると、詳細フォームに「**履歴」**セクションが表示されます。
- 詳細情報の表示: このオプションを選択すると、詳細フォームに「詳細情報」 セクションが表示されます。
- リンクの表示: このオプションを選択すると、詳細フォームに「リンク」セクションが表示されます。
- 6. 「一般」タブを使用して、間隔、位置合せおよびその他の表示オプションの変更を 行います。

デフォルトでは、タスク詳細コンポーネントにより、指定したすべての詳細が表示 されるように展開されます。特定の高さを設定するには、「高さの設定」をクリッ クして高さの値を調整します。

- 7. 「スタイル」タブを使用して、事前定義済スタイルまたは独自のカスタム選択でコ ンポーネントを含むフレームをフォーマットします。
- 8. 「リンク」タブを使用して、コンポーネントによって提供されるトリガーにアクションを割り当てます。
  - タスク詳細が送信されました: ユーザーが「送信」ボタンをクリックすると、 このトリガーが発生します。
  - **タスクが承認されました**: ユーザーが「承認」ボタンをクリックすると、この トリガーが発生します。
  - タスクが却下されました: ユーザーが「却下」ボタンをクリックすると、この トリガーが発生します。
  - タスクが閉じられました: ユーザーが「閉じる」ボタンをクリックすると、このトリガーが発生します。



- タスクが保存されました: ユーザーが「保存」ボタンをクリックすると、このトリガ ーが発生します。
- タスク・コメントが追加されました: ユーザーがコメント・テキストを追加して「コ メントの投稿」ボタンをクリックすると、このトリガーが発生します。

トリガーをクリックしてアクションを割り当てます。

#### ♪ ノート:

プロセス・タスク・リスト・コンポーネントでは手動のトリガーやアクションをサ ポートしていませんが、タスク詳細コンポーネントをページに追加すると、リスト から選択したタスクの詳細が自動的に表示されます。トリガーおよびアクションの 詳細は、「トリガーおよびアクションの使用」を参照してください。

Oracle Process Cloud Service の使用の詳細は、プロセスの開始を参照してください。

# コンテンツ・アイテム

エンタープライズ・ユーザーの場合、サイトごとに、サイトに関連付けられたデジタル・ア セットおよびコンテンツ・アイテムが含まれているコレクションがあります。コンテンツ・ アイテム・コンポーネントを使用すると、アイテムをサイトに簡単に追加できます。

エディタの「コンテンツ」パネルからデジタル・アセットおよびコンテンツ・アイテムを直 接ドラッグ・アンド・ドロップでき、デジタル・アセット・イメージについても構造化コン テンツ・アイテムについても 、適切なコンポーネントが自動的に使用されます。

「コンポーネント」パネルで選択したその他のコンポーネントを使用すると、コンテンツ・タ イプに基づいてコンテンツ・アイテムを動的に表示できます。

- コンテンツ・アイテム・コンポーネント
- コンテンツ・プレースホルダ
- コンテンツ・リスト
- コンテンツ検索
- レコメンデーション

# コンテンツ・アイテム・コンポーネント

エンタープライズ・ユーザーは、コンテンツ・アイテム自体を追加する準備ができるまで、 ページのレイアウトやページ・インタラクションの設定に役立つコンテンツ・アイテム・コ ンポーネントを使用できます。

「コンテンツ」パネルからページにコンテンツ・アイテムをドラッグ・アンド・ドロップする と、コンテンツ・アイテムがカスタム・デジタル・アセット・タイプでないかぎり、アイテ ムを保持するコンテンツ・アイテム・コンポーネントが自動的に挿入されます。カスタム・ デジタル・アセットをページにドラッグすると、アセットがイメージまたはビデオ・コンポ ーネントとして挿入されます。これを削除して、まずコンテンツ・アイテム・コンポーネン トを追加してから、カスタム・デジタル・アセットまたは標準コンテンツ・アイテムを「コ ンテンツ」パネルからコンポーネントにドラッグし、適切に表示する必要があります。

ページにコンテンツ・アイテム・コンポーネントを追加するには:



- 1. 編集するページに移動し、 「編集」に設定されていることを確認します。
- 2. ページにコンポーネントを追加します。
- コンテンツ・アイテムのコンポーネント・コンテンツを編集するには、メニュー・ アイコンをクリックして「編集」を選択するか、「設定」を選択して設定パネ ルを開き、「編集」アイコンをクリックします。

### /ノート:

コンテンツ・アイテム・コンポーネントで使用されるコンテンツ・タイ プが標準コンテンツ・タイプではなく、カスタム・デジタル・アセット である場合、カスタム・デジタル・アセット・タイプを編集することは できません。「編集」オプションはコンポーネント・メニューにありませ ん。また「編集」アイコンも設定パネルにありません。

- **4.** コンテンツ・アイテムを変更して、「保存」をクリックします。コンテンツ・アイ テムは、新規バージョンとしてコンテンツ・リポジトリに保存されます。
- コンポーネントの外観を編集するには、そのコンポーネントのメニュー・アイコン
   をクリックして、「設定」を選択します。
- 6. 「一般」タブを使用して、間隔、位置合せおよびその他の表示オプションの変更を 行います。

コンテンツ・アイテムがコンポーネントに割り当てられている場合、コンテンツ・ アイテムのサムネイル表示が示されます。コンテンツ・アイテムがまだ割り当てら れていない場合は、プレースホルダー・イメージとメッセージが表示されます。

- 7. 次から選択して、表示されるコンテンツの追加のデフォルトを設定します。
  - 使用するバージョン:「最新バージョンのアセットを使用」を選択すると、公開されていない新しいバージョンのコンテンツ・アイテムが存在する場合、現在のサイト更新の公開時に新しいバージョンが自動的に公開されます。特にリクエストされないかぎり、すべてのアイテムはドラフトまたは最新のバージョンになります。
  - アイテムの表示: コンテンツ・アイテムの表示に使用するレイアウトを選択します。指定されている「デフォルト」レイアウトでは、コンテンツ・アイテム内のすべてのフィールドが表示されます。コンテンツ・アイテムに、そのアイテム用に設計されたカスタム・レイアウトが他にも存在する場合は、使用可能な任意のレイアウトを選択できます。サイトで使用される場合にカスタム設定のサポートを追加するが有効になっているカスタム・レイアウトを選択した

場合、 をクリックして**カスタム・データ**・フィールドにデータを入力す ることでレイアウトにカスタム・データを追加してから、**「戻る」**をクリック して「一般」タブに戻ることができます。

個々のアイテムを表示するページ:1つ以上のページを詳細ページとして設計した場合は、ここにリストされます。ユーザーが詳細情報を表示するためにコンテンツ・アイテムのリンクをクリックしたときに詳細情報を表示するページを選択します。

/ ノート:

詳細ページを作成しない場合、デフォルト・レイアウトでは、コンテンツ・ アイテムに詳細を表示するリンクは表示されません。

8. 「スタイル」タブを使用して、事前定義済スタイルまたは独自のカスタム選択でコンポー ネントを含むフレームをフォーマットします。

# コンテンツ・プレースホルダ

エンタープライズ・ユーザーは、コンテンツ・プレースホルダー・コンポーネントを使用して、1つ以上のタイプのコンテンツ・アイテムを動的に表示できます。

たとえば、指定した詳細ページにコンテンツ・アイテム・プレースホルダーを使用でき、こ れにより、ユーザーが特定のコンテンツ・アイテムに関する詳細情報を取得するためにリン クをクリックしたときに、関連付けられるコンテンツ・アイテムの詳細ビューが自動的にロ ードされます。たとえば、複数の記事を含むページがあり、それぞれの記事にヘッドライン とイメージが表示されている場合、ユーザーが特定の記事をクリックすると、どの記事を選 択しても、詳細ページにはその記事全体が表示されます。

ページにコンテンツ・プレースホルダー・コンポーネントを追加するには:

- 1. 編集するページに移動し、 **し**が「**編集」**に設定されていることを確認します。
- 2. ページにコンポーネントを追加します。
- 3. コンポーネントとその外観を編集するには、メニュー・アイコン をクリックして、「設定」を選択します。
- **4.** 「一般」タブを使用して、間隔、位置合せおよびその他の表示オプションの変更を行います。
- 5. 次から選択して、表示されるコンテンツの追加のデフォルトを設定します。
  - コンテンツ・タイプ:使用可能な1つ以上のコンテンツ・タイプを選択します。コン テンツ・タイプは、サイト・コレクション内のコンテンツ・アイテムのもので、カス タム・デジタル・アセット・タイプが含まれています。
  - アイテムの表示: コンテンツ・アイテムの表示に使用するレイアウトを選択します。 指定されている「デフォルト」レイアウトでは、コンテンツ・アイテム内のすべての フィールドが表示されます。コンテンツ・アイテムに、そのアイテム用に設計された カスタム・レイアウトが他にも存在する場合は、使用可能な任意のレイアウトを選択 できます。
  - 個々のアイテムを表示するページ:1つ以上のページを詳細ページとして設計した場合は、ここにリストされます。ユーザーが詳細情報を表示するためにコンテンツ・ア イテムのリンクをクリックしたときに詳細情報を表示するページを選択します。

#### ▲ ノート:

詳細ページを作成しない場合、デフォルト・レイアウトでは、コンテンツ・ アイテムに詳細を表示するリンクは表示されません。



6. 「スタイル」タブを使用して、事前定義済スタイルまたは独自のカスタム選択でコンポーネントを含むフレームをフォーマットします。

### コンテンツ・リスト

エンタープライズ・ユーザーは、コンテンツ・リスト・コンポーネントを使用して、 特定のタイプのコンテンツ・アイテムを動的に表示できます。

たとえば、指定した詳細ページにコンテンツ・アイテム・リストを使用でき、これに より、ユーザーが特定のコンテンツ・アイテムに関する詳細情報を取得するためにリ ンクをクリックしたときに、関連付けられるコンテンツ・アイテムの詳細ビューが自 動的にロードされます。たとえば、複数の記事を含むページがあり、それぞれの記事 にヘッドラインとイメージが表示されている場合、ユーザーが特定の記事をクリック すると、どの記事を選択しても、詳細ページにはその記事全体が表示されます。

ページにコンテンツ・リスト・コンポーネントを追加するには:

- 1. 編集するページに移動し、 **し**が「**編集」**に設定されていることを確認します。
- 2. ページにコンポーネントを追加します。
- 3. コンポーネントとその外観を編集するには、メニュー・アイコン こを
  クリックして、
  「設定」を選択します。
- 「一般」タブを使用して、間隔、位置合せおよびその他の表示オプションの変更を 行います。
- 5. 次から選択して、表示されるコンテンツの追加のデフォルトを設定します。
  - コンテンツ・タイプ:使用可能ないずれかのコンテンツ・タイプを選択します。 コンテンツ・タイプは、サイト・コレクション内のコンテンツ・アイテムのもので、カスタム・デジタル・アセット・タイプが含まれています。
  - 最大アイテム数とアイテムで開始:表示する最大アイテム数と表示の開始位置 を指定します。これを超える数のアイテムは表示されません。
  - ページ区切り: リストにページ区切りオプションを使用するかどうかを指定します。これを選択した場合、ボタンやページ番号、各種ラベル(必要な場合)を使用して、ページ区切りの表示方法を調整できます。
  - 日付: 提供されたオプションを使用し、アイテム作成日に基づいて、選択した 日付や日付範囲の前後、間、範囲内で表示するコンテンツ・アイテムを選択し ます。
  - カテゴリ: カテゴリを選択してコンテンツ・アイテムのリストをフィルタ処理 し、特定の1つまたは複数のカテゴリ内のアイテムのみを表示します。「カテ ゴリの選択」をクリックして、サイト・リポジトリからカテゴリを選択するた めに使用できるスライドアウト・パネルを開きます。同じタクソノミ内の複数 のカテゴリを選択する場合、表示されるアイテムは、それらに割り当てられて いる選択したカテゴリの1つを持つ必要のみあります。異なるタクソノミか ら複数のカテゴリを選択する場合、表示されるアイテムは、それらに割り当て られている選択したすべてのカテゴリを持つ必要があります。たとえば、目的 地タクソノミから市区町村と公園のカテゴリを選択した場合、市区町村または 公園であるアイテムが表示されます。目的地タクソノミから市区町村カテゴ リを選択し、リージョン・タクソノミからヨーロッパ・カテゴリを選択した場 合、ヨーロッパの市区町村であるアイテムが表示されます。



デフォルトでは、選択したカテゴリのすべての子が使用可能になります。リストを特定のカテゴリ・ノードのアイテムに限定する場合は、ノードを選択し、**「子カテゴリを含める」**を選択解除します。

- 言語: リポジトリに関連付けられた言語から選択します。
- 追加問合セ文字列(オプション): 追加の問合セパラメータを指定し、field.dept eq "Finance"のような構文を使用して、表示するアイテムのリストをさらに絞り込みま す。使用可能な演算子のリストは、下の表を参照してください。
- 並替え基準:名前または日付で、昇順または降順にアイテムをソートします。コンテンツ・タイプに他の日付、数値または小数フィールドが含まれる場合、それらのフィールドでもソートできます。

「カスタム」を選択してからカスタム式を入力することもできます。たとえば、昇順 の部門番号でソートするには、fields.deptno:ascと入力します。ソート用の URL パラメータを定義することもできます。たとえば、{{URLParams.sortBy}}と入力し、 サイト URL の末尾に?sortBy=fields.deptno:ascというパラメータを追加できま す。

 アイテムの表示: コンテンツ・アイテムの表示に使用するレイアウトを選択します。 指定されている「デフォルト」レイアウトでは、コンテンツ・アイテム内のすべての フィールドが表示されます。コンテンツ・アイテムに、そのアイテム用に設計された カスタム・レイアウトが他にも存在する場合は、使用可能な任意のレイアウトを選択 できます。サイトで使用される場合にカスタム設定のサポートを追加するが有効に

なっているカスタム・レイアウトを選択した場合、 をクリックして**カスタム・ データ**・フィールドにデータを入力することでレイアウトにカスタム・データを追加 してから、**「戻る」**をクリックして「一般」タブに戻ることができます。

個々のアイテムを表示するページ:1つ以上のページを詳細ページとして設計した場合は、ここにリストされます。ユーザーが詳細情報を表示するためにコンテンツ・アイテムのリンクをクリックしたときに詳細情報を表示するページを選択します。

#### /ノート:

詳細ページを作成しない場合、デフォルト・レイアウトでは、コンテンツ・ アイテムに詳細を表示するリンクは表示されません。

- リスト表示: セクション・レイアウトを選択して、アイテムを配置します。このリストにはカスタム・レイアウトを含めることができます。次のレイアウトが提供されています。
  - 水平方向: アイテムを水平線上に配置します。
  - 垂直方向: アイテムを垂直線上に配置します。これは、レイアウトが選択されていない場合のデフォルトです。
  - 2列:アイテムを複数の行に2つずつ配置します。
  - 3列: アイテムを複数の行に3つずつ配置します。

選択したレイアウトの横の右矢印をクリックして、レイアウトの設定を変更します。

- 空のリストの表示: 空のリストの表示方法を選択します。
- オプション: コンテンツ検索コンポーネントで指定した検索問合せの自動リフレッシュを行うかどうかを選択します。



- 間隔: アイテムの表示方法を選択します。
- 6. 「スタイル」タブを使用して、事前定義済スタイルまたは独自のカスタム選択でコンポーネントを含むフレームをフォーマットします。

#### 表 10-1 問合せ演算子

演算 子	例	サポートされている データ型	説明
eq	?q=name eq "John" ?q=type eq "DigitalAsset"	テキスト、参照、数 値、10 進数、ブー ル、日時	「等しい」演算子(eq)は、問合せ で指定した正確な値に一致しま す。この演算子は、複数値のデ
	<pre>?q=type eq "Employee" and fields.DOB eq "1994/09/26T16:23:45. 208"</pre>		の演算子とともに指定した値 は、標準フィールドの場合を除 き、大///文字が区別されませ ん。この演算子は、値の特殊文
	?q=type eq "Employee" and fields.DOB eq "1994/09/26T16:23:45. 208"		字も考慮します。
СО	<pre>?q=(type eq "Employee" AND name co "john alex") ?q=(type eq "Car" AND fields.features co "manual")</pre>	テキスト、参照、数 値、10 進数、日時、 大きいテキスト	「次を含む」演算子(co)は、基準 に指定したすべての語に一致し ます。語は、値を特殊文字で分 割した形式になります。指定し た部のうち1つ以上(この例で は、john またはalex、あるいは その両方)を含む結果を返しま す。この演算子は、検索中に値 の特殊文字を考慮しません。こ の演算子は、ストップ・ワード に対して検索を実行しません。 ストップ・ワードの詳細は、 Apache Luceneのドキュメン トを参照してください。 この演算子は、単一値属性の場 はテキスト、大きいテキスト に適用され、複数値属性の場合 はテキスト、参照、数値、10 進 数、日時、大きいテキストに適 用されます。指定可能な日時書 式を理解するには、下のサポー トされている日時書式(24 時間) の表を参照してください。この 演算子とともに指定した値は、 大小文字が区別されません。
SW	?q=type eq "Employee" AND name sw "Joh" ?q=type eq "Employee" AND fields.city sw "Los"	テキスト	「次で始まる」演算子(sw)は、フ ィールド条件に指定した先頭文 字の値にのみ一致します。この 演算子は、複数値のデータ型に は適用されません。この演算子 とともに指定した値は、大/小文 字が区別されません。

表 10-1 (続き) 問合せ演算子

演算 子	例	サポートされている データ型	説明
ge	<pre>?q=(type eq "Employee" AND fields.age ge "40") ?q=type eq "DigitalAsset" AND updatedDate ge "20171026"</pre>	数値、10 進数、日₿	「次以上」演算子(ge)は、数値および日時の値にのみ一致します。指定可能な日時書式を理解するには、下のサポートされている日時書式(24 時間)の表を参照してください。この演算子は、複数値のデータ型には適用されません。
le	<pre>?q=(type eq "Employee" AND fields.weight le "60.6")</pre>	数値、10進数、日間	「次以下」演算子(le)は、数値および日時の値にのみ一致します。指定可能な日時書式を理解するには、下のサポートされている日時書式(24 時間)の表を参照してください。この演算子は、複数値のデータ型には適用されません。
gt	?q=(type eq "Employee" AND fields.age gt "20")	数値、10 進数、日₿	「次より大きい」演算子(gt)は、 数値および日時の値にのみ一致 します。指定可能な日時書式を 理解するには、下のサポートされている日時書式(24 時間)の表 を参照してください。指定可能 な日時書式を理解するには、下 のサポートされている日時書式 (24 時間)の表を参照してください。この演算子は、複数値のデ ータ型には適用されません。
lt	<pre>?q=(type eq "Employee" AND fields.age lt "20") ?q=type eq "Employee" AND createdDate lt "1994/09/26T16:23:45. 208"</pre>	数値、10 進数、日₿	「次より小さい」演算子(lt)は、 数値および日時の値にのみ一致 します。指定可能な日時書式を 理解するには、「サポートされて いる日時書式(24時間)」の項を 参照してください。この演算子 は、複数値のデータ型には適用 されません。

表 10-1 (続き) 問合せ演算子

演算 子	例	サポートされている データ型	説明
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>	テキスト、大きいテ キスト	句問合せまたは近接検索(一致) 演算子(mt)を使用すると、相互 の間隔が特定の距離内にある語 を検索できます。結果は一致率 の高い順にソートされます。こ れは、基準"petrol 20kmpl"に指 定した値によって、" <i>petrol</i> fuel mileage runs 20KMPL in the speed way"を含む実際のコン テンツを検出する必要がある場 合に、するのに便利です。 一致演算子では、指定した値の 中でワイルドカードを使用する ことつ文字と複数文字の切っともできと複数文字の切っとしてはできした。 中でワイルドカードを使用します。 "John"と"Joan"のどちらも、単 一文字の場合は"Jo?n"、複数文 字の場合は"Jo?n"、複数文 字の場合は"Jo?n"、複数文 字の場合は"Jo?n"、複数文 字の場合は"Jo?n"、複数文 字の場合は"Jo?n"、複数文 字の場合は"Jo?n"、複数文 字の場合は"Jo?n"、複数文 字の場合は"Jo?n"、複数文 字の場合は"Jo?n"、複数文 字の場合は"Jo?n"で検索できま す。
sm	?q=(type eq "Employee" And fields.city sm "Rome")	テキスト、大きいテ キスト	類似度問合せ演算子。この演算 子を使用すると、指定した基準 に似た値を検索できます(これ はファジー検索とも呼ばれま す)。デフォルトでは、最大2回 の編集を使用して結果を一致さ せます。"Rome"は"Dome"に似 ています。この演算子は、単一 値と複数値両方のデータ型に適 用されます。この演算子ととも に指定した値は、大/小文字が区 別されません。
AND	?q=(type eq "Employee" AND name eq "John" AND fields.age ge "40")	N/A	AND 演算子を使用すると、複数 の問合せ条件の間に AND 条件 を挿入できます。これは OR よ り優先されます。
OR	type eq "Employee" AND name eq "John" OR fields.age ge "40"	N/A	OR 演算子を使用すると、複数の 問合せ条件の間に OR 条件を挿 入できます。

表 10-1 (続き) 問合せ演算子

演算 子	例	サポートされている データ型	説明
()	<pre>?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)</pre>	N/A	基準内の条件をグループ化する カッコ(包含)演算子。これが最 も優先度が高く、次が AND、そ の次が OR になります。

# コンテンツ検索

エンタープライズ・ユーザーは、コンテンツ検索コンポーネントを使用し、検索で返される アクションを指定できます。

カスタマイズした検索バーを挿入して、ページに表示されるコンテンツを変更またはリフレ ッシュできます。また、検索結果ページを開く、アラートを表示するなど、別のアクション を選択することもできます。

コンテンツ検索コンポーネントをページに追加するには:

- 1. 編集するページに移動し、 **し**が「**編集」**に設定されていることを確認します。
- 2. ページにコンテンツ検索コンポーネントを追加します。
- 3. コンポーネントとその外観を編集するには、メニュー・アイコンをクリックして、「設 定」を選択します。
- 4. 「一般」タブを使用して、プレースホルダ・テキスト、間隔、位置合せおよびその他の表 示オプションを変更します。
- 5. 「スタイル」タブを使用して、事前定義済スタイル、またはフォント、枠線、背景色など の独自のカスタム選択でコンポーネントを含むフレームをフォーマットします。
- 6. 検索コンポーネントを使用して現在のページのデータをリフレッシュする場合は、コンテンツ・リスト・コンポーネントをページに追加します。「設定」に移動し、コンテンツ・タイプと、表示するアイテムの数、結果のページ番号を付けるか、スクロール時の遅延ロード(ページをスクロールするとコンテンツをロードする)など、その他の問合せ情報を選択します。これで、表示モードになり、検索を試すことができます。
- 7. 検索コンポーネントを使用して現在のページのデータをリフレッシュし、ページに複数の コンテンツ・リスト・コンポーネントがある場合は、結果を表示するリストを除くすべて のコンテンツ・リストの自動問合せをオフにする必要があります。「コンテンツ・リスト」 設定に移動し、「コンテンツ検索コンポーネントでの検索問合せ時に自動リフレッシュ」 を選択解除します。
- 8. 1つのページに複数のコンテンツ検索コンポーネント(各検索コンポーネントが結果の特定のコンテンツ・リストを駆動する)を指定する場合は、コンテンツ・リスト・コンポーネントのすべての自動リフレッシュ・オプションを選択解除し、トリガー/アクションを



使用して、コンテンツ検索コンポーネントをそれぞれのコンテンツ・リストに関連 付ける必要があります。

- 9. 検索コンポーネントで、「リンク」タブを使用して、アクションとコンポーネント を関連付けます。「リンク・タイプの選択」ドロップダウン・リストからオプショ ンを選択します。使用可能なトリガーを参照する場合や、新規のトリガーを作成す る場合は、「トリガー・アクション」を選択して、「検索問合せ時」をクリックしま す。検索するコンテンツ・リストを見つけて展開し、「コンテンツの検索」をアク ション・リストにドラッグします。「検索」で、「検索文字列」を選択します。選択 したコンテンツ・リストは、「このアクションの実行場所」リストで選択されます。
- 10. コンテンツ検索コンポーネントを使用して、問合せを検索結果ページに渡すこともできます。検索コンポーネントの「リンク」タブで、検索結果ページを表示する場合は、「検索ページ」を選択します。デフォルト・ページを使用することも、作成した検索結果ページにリンクすることもできます。「ページ」ドロップダウン・リストからページを選択し、ページの表示アクションを選択します。結果を特定のコンテンツ・タイプに合せて調整したり、新規のウィンドウで結果を開くことなどができます。

次の指示は、効果的な検索結果ページを設計する際に役立ちます。

- ページを作成し、検索ページとして指定します。デフォルトでは非表示とマークさ れていますが、ページ・プロパティの設定を変更できます。
- ページを編集し、ページにコンテンツ・リストを追加します。コンテンツ・タイプ を指定して、コンテンツ・リストの設定を編集できます。または、コンテンツ・タ イプは、検索コンポーネント・リンク設定(前述)から指定できます。リンク設定を 使用すると、検索を開始するために使用されるコンポーネントに応じて、異なるコ ンテンツ・タイプの結果を表示できる検索結果ページを使用できます。
- 表示を編集するには、遅延ロード・ページ区切りを選択するなど、設定を変更します(ページが検索結果にのみ使用される可能性があるため)。
- 検索結果ページで、コンテンツ検索コンポーネントに格納することもできます。これにより、ページの起動に使用された検索文字列がエコーされ、ユーザーは必要に応じて検索を絞り込むことができます。

# レコメンデーション

エンタープライズ・ユーザーは、**レコメンデーション**・コンポーネントを使用して、 場所や関心のある分野に基づいてアセットを表示することで、Web サイトのビジター にパーソナライズされたエクスペリエンスを提供できです。リポジトリ・コントリビ ュータがレコメンデーションを作成する際は、サイト・ビジターのジオロケーション 情報など、オーディエンス属性に一致するアセットを検出するルール・セットを定義 します。たとえば、ヨーロッパの IP アドレスのサイト・ビジターには、まず、サイト のホーム・ページにヨーロッパのイベント告知が表示され、北アメリカからのサイト・ ビジターには、米国とカナダのイベントが表示されます。

デジタル・アセット同様、レコメンデーションはリポジトリに関連付けられており、 レビューと公開のワークフローを通じて、そのリポジトリのコンテンツ・コントリビ ュータであるユーザーなら誰でも、編集したり移動したりできます。ただし、コント リビュータ権限がない場合でも、サイトやヘッドレス・エクスペリエンスにレコメン デーションを使用する前に、レコメンデーションの表示やテストを実行して、どのよ うに機能するかを確認することが可能です。

ページにレコメンデーションを追加するには、次のようにします:



- サイド・メニューで「サイト」をクリックし、レコメンデーションを追加するサイトを選択して、右クリック・メニューの「開く」を選択するか、アクション・バーのをクリックします。
- 2. サイトを「編集」に切り替え、使用する更新を選択するか、新しく作成します。
- 3. サイド・メニューから「コンポーネント」を選択します。
- 4. 「シード」を開きます。
- 5. 「レコメンデーション」をクリックしてコンテンツ・セクションにドラッグし、ページに 配置します。
- 6. 「レコメンデーション」メニュー()から「設定」を選択します。

0	Content and Experience	0 (
=	Customer Events demo 💌 en (Default)	No Test Profile * Fit to Window * 1 View Letit Z Commit Save R O L
e) m	< Seeded * Find a component	Starter Template
\$	Basic	Upcoming Customer Events Around the World! Set addicated Recommendation Upcoming Customer Events 1
t⊒ ₽	Paragraph	Collegare l'Italia ai mercati scalabili
٥	T Text	Milan, italy October 18, 2020, 11:00 PM Recommendation S ×
	Structure	La versina sara beccura agii mesumenti nami no botopo Later Stage Cire decuono di espano businesi "Oversea", specificamente dedicta da la stratupe a is donaver digitali filadian che cercan mercato americano a San Francisco, New York, San Diego e esternalizzare la propria produzione a BIG E E Latal inclusa in un gruppo di tavole rotonde come Regno Unito, Germania, Francia e Spage Recommendation o
	1 Spacer	CEO Summit and Executive Workshop of Modern Fina Country name (Session)
	Content	Milan, Italy City name (Session)
	Content Item	July 18, 2019, 08:00 AM
	Content Placeholder	For all the challenges faining our nation and our world, solutions exist. With the growing adoption c Maximum Items     machine learning and artificial intelligence, to 3D printing and the internet of Things, we stand on     accelerate     growth and improve living standards in the U.S. and worldwide. Attend this CPD Summit and work
	Content List	taking advantage of energing technologies to unlock new idensis of productivity, both within triants finance best protectices by industry, the CPO-CPO subminishing, and the own intelligent finance organizar success in fodar's digital economy, CPO summit and Executive Workshop attendees also receive
	Recommendation	Pagnation 5
Ð	Media	F in V COPYRIGHT 6 2018 Content List Default OUD

- 7. レコメンデーションの設定の「一般」タブで、使用するレコメンデーションを選択しま す。ページがリフレッシュされ、推奨されるアセットが表示されます。
- 8. オプションで、選択したレコメンデーションの横にある矢印をクリックして、このレコメ ンデーションで使用するオーディエンス属性のデフォルト値を入力します。値の追加が 完了したら、「**戻る」**をクリックします。
- 9. 使用するバージョンや表示するアイテムの最大数、デフォルトのコンテンツ・レイアウト など、必要なプロパティを追加で変更します。終了したらレコメンデーションの設定を閉 じ、「保存」をクリックします。



### / ノート:

事前定義されたオーディエンス属性値を含むテスト・プロファイルを作 成済の場合、メニュー・バーのテスト・プロファイル・メニューからプ ロファイルを選択すると、レコメンデーションによるそれらの入力への 応答方法をプレビューできます。テスト・プロファイルの値は、コンポ ーネント設定に指定されているデフォルト値をオーバーライドします。

 レコメンデーションがそのルールで現在の日付(システム)を使用しているが、それ とは別の日付でレコメンデーションをテストする場合、テスト・プロファイル・メ ニューからシステム日付を選択し、テストで使用する日付を選択します。

# その他のコンポーネント

タイプの異なるコンテンツを結合するコンポーネントをいくつか見てみましょう。

- ・ マップ
- ヘッドライン
- 記事
- テキスト付きイメージ
- コンポーネント・グループ
- ページでの共有ブラウズの使用
- Oracle Intelligent Advisor
- Oracle Visual Builder



ユーザーが場所の周囲の領域をインタラクティブに検索できるように、サイトにマッ プを追加します。



ページにマップを追加するには:

- 1. 編集するページに移動し、 **し**が「**編集」**に設定されていることを確認します。
- 2. ページにコンポーネントを追加します。Oracle マップ・コンポーネントには、デ フォルトの場所が表示されます。



- マップの場所の変更およびその表示プロパティの調整を行うには、メニュー・アイコン
   をクリックして「設定」を選択します。
- 4. マップの開始する「場所」を入力します。住所、郵便番号またはカンマ区切りの緯度と経 度(40.5,-57.6 など)を使用できます

住所を使用する場合、1つの場所に一致させるための十分な情報を指定する必要がありま す。住所が複数の場所と一致する場合、マップは空白のままです。

- 5. マップの初期の「ズーム」レベルを指定します。必要に応じて、ユーザーがマウス、トラ ック・パッドまたは画面上のコントロールを使用してズーム・レベルを調整できるように することができます。
- 6. 任意のスタイル、位置合せおよび表示オプションを指定します。
  - ズーム: 画面上およびデバイスのオプションを有効にして、ユーザーがマップのズーム・レベルを調整できるようにします。
  - **移動**: 画面上およびデバイスのオプションを有効にして、ユーザーがマップのフォー カスの領域を移動できるようにします。
  - **マーカーの表示**: ピン・アイコンで開始場所をマークします。
  - **スケール**:現在のズーム・レベルのスケールを示すインジケータが含まれます。
  - 概観:より大きなコンテキストで現在の表示を示すインセット・マップが含まれます。

# ヘッドライン

ヘッドラインを追加して、タイトル・テキストおよびサポートしているイメージと段落を使 用して、サイトの特定のスポットに対する注意を促すことができます。

- 1. 編集するページに移動し、 **し**が「**編集」**に設定されていることを確認します。
- 2. ページにコンポーネントを追加します。
- コンポーネントの様々な領域をクリックしてテキストを追加します。メイン・ヘッドラインおよびメイン・ヘッドラインの下にある段落にコンテンツを追加します。テキストは、コンポーネントのデフォルトのスタイルにフォーマットされます。[Enter]を押して、テキストの追加の行を含めます。
- テキストの任意の部分に対してデフォルトのフォーマットを変更する場合は、フォーマットするテキストを選択し、フォーマット・ツール・バーの任意のオプション(フォント、 色、位置合せなど)を選択します。
- 5. これらのオプションにより適用されたフォーマットを削除するには、テキストを選択し、
   ▲をクリックします。
- 6. コンポーネントとその外観を編集するには、メニュー・アイコン = をクリックして、「設 定」を選択します。
- 7. 「コンポーネント」を選択して、表示されるコンテンツに関する詳細を設定します。
  - イメージ:
    - 「選択」をクリックして、ヘッドラインのイメージを使用します。サイト・リポジトリからイメージ(ユーザーと共有されているドキュメント・フォルダのイメージ)を選択するか、イメージをドキュメント・フォルダにアップロードします。使用するイメージを検索して選択します。デジタル・アセットが表示されない場合、

をクリックして、コレクション・フィルタを「すべて」に変更します。 ドキュメント・フォルダからイメージを選択した場合、ファイルをサイトにコピーせずに、そのファイルにリンクできます。ファイルにリンクするには、「サイトにファイルをコピーするかわりに元のファイルへの参照を使用します」を選択します。このオプションを選択しない場合、ファイルの コピーがサイトで格納され、サイトから参照されます。元のファイルにリンクすることで、コンテンツを複製せずに済みます。リンクを使用すれば、ファイルの権限が変更されたり、表示が制限されるような場合でも、サイトのビジターがコンテンツを表示できます。「OK」をクリックします。デジタル・アセットを選択した場合、特定のレンディションを選択できます。レンディションを選択しない場合、元のサイズが使用されます。サイトの公開時に最新バージョンのアセットを公開する場合、「最新バージョンのアセットを使用」を選択します。

イメージを編集するには、 をクリックし、次のいずれかのアクション でイメージを編集します。

- * イメージを切り取るには、[□]「切取り」をクリックします。切取りツ ールバーで事前定義されたイメージ比率の1つを選択するか、イメー ジの切取りハンドルを目的の位置までドラッグします。完了したら、 切取りツールバーで「切取り」をクリックします。
- * イメージを回転または反転するには、 「 「回転」をクリックします。回転ツールバーで、カスタム回転度を入力する、ボタンを使用してイメージを左または右に回転する、あるいはイメージを水平方向または垂直方向のいずれに反転するかを選択します。
- * イメージにウォーターマークを追加するには、 「ウォーターマー ク」をクリックします。テキストをイメージに追加し、テキスト・サ イズ、スタイル、色および不透明度を、ウォーターマーク・ツールを 使用して必要に応じて変更します。
- イメージ・フォーマットを変更するには、 「オプション」をクリックし、「フォーマット」ドロップダウン・リストから新しいフォーマットを選択します。
- * 背景色を変更するには、 ⁽²⁾ 「オプション」</mark>をクリックし、「背景色」 ドロップダウン・メニューからオプションを選択します。
- * .jpg または.webp (Google Chrome ブラウザに対応)を編集中の場合、
   イメージ品質を変更してより小さなファイル・サイズを作成できます。
   (アオプション)をクリックし、「品質」ボックスに新しいパーセン
   テージを入力します。
- * 変更を元に戻すか、やりなおすには、 ^{*}⊃ または ^{C*} をクリックしま す。行ったすべての変更を削除するには、「**リセット」**をクリックしま す。
- * イメージの拡大率を変更するには、ズーム・コントロールを使用します(⁻ ○ + )。
- **タイトル**: ツールチップに表示するテキストを入力します。

- 代替テキスト: アクセシビリティのために表示される代替テキストを入力します。
- **キャプション**: イメージの下に表示されるキャプションを入力します。
- 位置合せ、幅および間隔:必要に応じて、イメージのレイアウトを変更します。
- **タイトル**: ヘッドラインのタイトルの間隔を変更します。
- 段落: ヘッドラインの下に表示されるテキストの間隔を変更します。
- 8. 「一般」タブを使用して、間隔、位置合せおよびその他の表示オプションの変更を行いま す。
- 9. 「スタイル」タブを使用して、テキストの書式の追加、および背景色、フォントと枠線の カスタマイズを行います。

# 記事

サイトに記事を追加することができ、これによりヘッドライン、段落およびイメージのコン ポーネントを1つの使用しやすいコンポーネントに集約することができます。

- 1. 編集するページに移動し、 **し**が「**編集」**に設定されていることを確認します。
- 2. ページにコンポーネントを追加します。
- コンポーネントの様々な領域をクリックしてテキストを追加します。記事のヘッドラインおよびその下のサブヘッダーにコンテンツを追加します。その後、サブヘッダーの下に記事のテキストを追加できます。すべてのテキストはコンポーネントのデフォルトのスタイルのフォーマットになります。[Enter]を押して、テキストの追加の行を含めます。
- テキストの任意の部分に対してデフォルトのフォーマットを変更する場合は、フォーマットするテキストを選択し、フォーマット・ツール・バーの任意のオプション(フォント、 色、位置合せなど)を選択します。
- これらのオプションにより適用されたフォーマットを削除するには、テキストを選択し、
   をクリックします。
- 6. コンポーネントとその外観を編集するには、メニュー・アイコンをクリックして、「設 **定」**を選択します。
- 7. 「コンポーネント」を選択して、表示されるコンテンツに関する詳細を設定します。
  - イメージ:
    - 「選択」をクリックして、記事でイメージを使用します。サイト・リポジトリから イメージ(ユーザーと共有されているドキュメント・フォルダのイメージ)を選択 するか、イメージをドキュメント・フォルダにアップロードします。使用するイ

メージを検索して選択します。デジタル・アセットが表示されない場合、 かん クリックして、コレクション・フィルタを「すべて」に変更します。ドキュメント・フォルダからイメージを選択した場合、ファイルをサイトにコピーせずに、そのファイルにリンクできます。ファイルにリンクするには、「サイトにファイルをコピーするかわりに元のファイルへの参照を使用します」を選択します。このオプションを選択しない場合、ファイルのコピーがサイトで格納され、サイトから参照されます。元のファイルにリンクすることで、コンテンツを複製せずに済みます。リンクを使用すれば、ファイルの権限が変更されたり、表示が制限されるような場合でも、サイトのビジターがコンテンツを表示できます。「OK」をクリックします。デジタル・アセットを選択した場合、特定のレンディションを選

択できます。レンディションを選択しない場合、元のサイズが使用されま す。サイトの公開時に最新バージョンのアセットを公開する場合、「最新バ ージョンのアセットを使用」を選択します。

イメージを編集するには、 をクリックし、次のいずれかのアクションでイメージを編集します。

- * イメージを切り取るには、^[4]「切取り」をクリックします。切取りツ ールバーで事前定義されたイメージ比率の1つを選択するか、イメー ジの切取りハンドルを目的の位置までドラッグします。完了したら、 切取りツールバーで「切取り」をクリックします。
- * イメージを回転または反転するには、 ⁽⁵⁾ 「回転」をクリックしま す。回転ツールバーで、カスタム回転度を入力する、ボタンを使用し てイメージを左または右に回転する、あるいはイメージを水平方向ま たは垂直方向のいずれに反転するかを選択します。
- * イメージにウォーターマークを追加するには、 ^〇 「ウォーターマー ク」をクリックします。テキストをイメージに追加し、テキスト・サ イズ、スタイル、色および不透明度を、ウォーターマーク・ツールを 使用して必要に応じて変更します。
- イメージ・フォーマットを変更するには、 「オプション」をクリックし、「フォーマット」ドロップダウン・リストから新しいフォーマットを選択します。
- * 背景色を変更するには、 「オプション」をクリックし、「背景色」 ドロップダウン・メニューからオプションを選択します。
- * .jpg または.webp (Google Chrome ブラウザに対応)を編集中の場合、
   イメージ品質を変更してより小さなファイル・サイズを作成できます。
   「オプション」をクリックし、「品質」ボックスに新しいパーセン
   テージを入力します。
- * 変更を元に戻すか、やりなおすには、 [•]⊃ または [•] をクリックしま す。行ったすべての変更を削除するには、「**リセット」**をクリックしま す。
- * イメージの拡大率を変更するには、ズーム・コントロールを使用します(- - +)。
- **タイトル**: ツールチップに表示するテキストを入力します。
- 代替テキスト: アクセシビリティのために表示される代替テキストを入力 します。
- **キャプション**: イメージの下に表示されるキャプションを入力します。
- 位置合せ、幅および間隔: 必要に応じて、イメージのレイアウトを変更し ます。
- **タイトル**: 記事のタイトルの間隔を変更します。
- 「段落」(サブヘッダー)および「段落」: 記事のタイトルの下に表示されるテキ ストの間隔を変更します。

- 8. 「一般」タブを使用して、コンポーネントの間隔、位置合せおよび他の表示オプションを 変更します。
- 9. 「スタイル」タブを使用して、テキストの書式の追加、および背景色、フォントと枠線の カスタマイズを行います。

# テキスト付きイメージ

イメージおよびテキスト・コンポーネントを追加すると、段落のコンポーネントとイメージ を使用しやすい1つのコンポーネントに結合できます。

- 1. 編集するページに移動し、 **し**が「**編集」**に設定されていることを確認します。
- 2. ページにコンポーネントを追加します。
- コンポーネントの段落領域をクリックして、テキストを追加します。すべてのテキストは コンポーネントのデフォルトのスタイルのフォーマットになります。[Enter]を押して、テ キストの追加の行を含めます。
- テキストの任意の部分に対してデフォルトのフォーマットを変更する場合は、フォーマットするテキストを選択し、フォーマット・ツール・バーの任意のオプション(フォント、 色、位置合せなど)を選択します。
- これらのオプションにより適用されたフォーマットを削除するには、テキストを選択し、
   をクリックします。
- 6. コンポーネントとその外観を編集するには、メニュー・アイコン <br/>
  をクリックして、「設定」を選択します。
- 7. 「コンポーネント」を選択して、表示されるコンテンツに関する詳細を設定します。
  - イメージ:
    - 「選択」をクリックしてイメージを使用します。サイト・リポジトリからイメージ (ユーザーと共有されているドキュメント・フォルダのイメージ)を選択するか、 イメージをドキュメント・フォルダにアップロードします。使用するイメージを

検索して選択します。デジタル・アセットが表示されない場合、 して、コレクション・フィルタを「すべて」に変更します。ドキュメント・フォ ルダからイメージを選択した場合、ファイルをサイトにコピーせずに、そのファ イルにリンクできます。ファイルにリンクするには、「サイトにファイルをコピー するかわりに元のファイルへの参照を使用します」を選択します。このオプショ ンを選択しない場合、ファイルのコピーがサイトで格納され、サイトから参照さ れます。元のファイルにリンクすることで、コンテンツを複製せずに済みます。 リンクを使用すれば、ファイルの権限が変更されたり、表示が制限されるような 場合でも、サイトのビジターがコンテンツを表示できます。「OK」をクリックし ます。デジタル・アセットを選択した場合、特定のレンディションを選択できま す。レンディションを選択しない場合、元のサイズが使用されます。サイトの公 開時に最新バージョンのアセットを公開する場合、「最新バージョンのアセットを 使用」を選択します。

イメージを編集するには、 をクリックし、次のいずれかのアクションでイメージを編集します。

* イメージを切り取るには、[□]「切取り」をクリックします。切取りツールバ ーで事前定義されたイメージ比率の1つを選択するか、イメージの切取りハ



ンドルを目的の位置までドラッグします。完了したら、切取りツール バーで**「切取り」**をクリックします。

- * イメージを回転または反転するには、 「**回転」**をクリックしま す。回転ツールバーで、カスタム回転度を入力する、ボタンを使用し てイメージを左または右に回転する、あるいはイメージを水平方向ま たは垂直方向のいずれに反転するかを選択します。
- * イメージにウォーターマークを追加するには、 〇 「ウォーターマー ク」をクリックします。テキストをイメージに追加し、テキスト・サ イズ、スタイル、色および不透明度を、ウォーターマーク・ツールを 使用して必要に応じて変更します。
- * イメージ・フォーマットを変更するには、 「オプション」をクリックし、「フォーマット」ドロップダウン・リストから新しいフォーマットを選択します。
- * 背景色を変更するには、 「オプション」をクリックし、「背景色」 ドロップダウン・メニューからオプションを選択します。
- * .jpg または.webp (Google Chrome ブラウザに対応)を編集中の場合、
   イメージ品質を変更してより小さなファイル・サイズを作成できます。
   「オプション」をクリックし、「品質」ボックスに新しいパーセン
   テージを入力します。
- * 変更を元に戻すか、やりなおすには、 [•] または [•] をクリックしま す。行ったすべての変更を削除するには、「リセット」をクリックしま す。
- **タイトル**: ツールチップに表示するテキストを入力します。
- 代替テキスト: アクセシビリティのために表示される代替テキストを入力 します。
- **キャプション**: イメージの下に表示されるキャプションを入力します。
- 位置合せ、幅および間隔:必要に応じて、イメージのレイアウトを変更します。
- 段落: テキストの間隔を変更します。
- 8. 「一般」タブを使用して、コンポーネントの間隔、位置合せおよび他の表示オプションを変更します。
- 9. 「スタイル」タブを使用して、テキストの書式の追加、および背景色、フォントと 枠線のカスタマイズを行います。



# コンポーネント・グループ

1つ以上のコンポーネントを結合して、名前を付けて再使用できるコンポーネント・グループ を作成できます。

コンポーネント・グループを保存する場合、コンポーネント・グループは指定した名前のカ スタム・コンポーネントとして保存され、エディタのカスタム・コンポーネントのリストに 表示されます。

- 1. 編集するページに移動し、 **し**が「**編集」**に設定されていることを確認します。
- 2. ページにコンポーネント・グループを追加します。コンポーネント・グループは で識 別されます。
- 1つ以上のコンポーネントをカスタム・コンポーネントのフレームにドラッグ・アンド・ ドロップします。
- スロットのコンポーネントの場合と同じ方法で、コンポーネントをコンポーネント・グル ープに配置し、サイズ変更します。
- コンポーネントとその外観を編集するには、そのコンポーネントのメニュー・アイコン
   をクリックして、「設定」を選択します。メニュー・アイコンではなくコンポーネント名をクリックした場合は、コンポーネント・グループ(またはスロット)のメニュー・アイコンが表示され、選択できます。



コンポーネント・グループの「設定」タブでは、コンポーネント・グループの位置、背景 イメージおよびコンポーネント・グループ全体に適用されるその他の設定を指定できま す。

- 変更をコンポーネント・グループに保存する準備ができたら、コンポーネント・グループ をクリックし、そのコンポーネントのメニュー・アイコン をクリックして、「保存」 を選択します。
  - a. ダイアログで、コンポーネント・グループの名前を入力します。文字、数字、アンダ ースコア()およびハイフン(-)を使用できます。スペースを入力した場合、ハイフンに 自動的に置換されます。

これが新規コンポーネント・グループの場合、既存のカスタム・コンポーネントの名 前は使用できません。 既存のコンポーネント・グループをページに追加し、コンポーネント・グループを変 更し、変更を保存しようとした場合、名前を付けて新規コンポーネント・グループを 作成し、「既存のコンポーネント・グループの上書き」を選択して既存のコンポーネ ント・グループを変更内容で更新するためのオプションが表示されます。

**b. 「保存」**をクリックします。

コンポーネント・グループは、カスタム・コンポーネントとして指定した名前 で保存されます。コンポーネント・マネージャの固有のフォルダと、ユーザー が所有者であるか他のユーザーがコンポーネントを共有した場合はエディタ のカスタム・コンポーネントのリストに表示されます。コンポーネント・グル ープは、カスタム・コンポーネントの場合と同様に共有できます。

# ページでの共有ブラウズの使用

Oracle Cobrowse Cloud Service は、画面の共有や他のユーザーとの共有ブラウズ・セッションの開始を可能にするコラボレーション・ツールです。たとえば、これを注文フォームに含めることにより、顧客が注文する際に担当者が顧客の画面を表示することが可能になります。

この機能を使用するには、先にサイトで有効化しておく必要があります。有効化でき るランチャ・スクリプトは2種類あります。1つは、カスタマイズされたボタンを使用 するもの(起動ポイント2)で、もう1つは、デフォルトの「共有ブラウズ」ボタンを使 用するもの(起動ポイント1)です。サイトでこの機能を有効化する際は、どちらのラン チャを使用するかを決定し、必要なスクリプトを追加してください。詳細は、共有ブ ラウズ統合の有効化を参照してください。

サイトで共有ブラウズを有効化したら、共有ブラウズを使用できるよう任意のページ を構成できます。

- 1. 編集するページに移動し、 **し**が「**編集」**に設定されていることを確認します。
- 2. 
   をクリックします。「共有ブラウズ」セクションのチェック・ボックスを選択します。
- 3. 「閉じる」をクリックします。

起動ポイント1のスクリプトを有効化した場合、次にページを表示する際、またはホット・キーを構成している場合にはホット・キーを押した後に、デフォルトの「共有 ブラウズ」ボタンが表示されます。

起動ポイント**2**のスクリプトを有効化した場合は、ページにカスタム・ボタンを追加 する必要があります。

- 1. **し**が「編集」に設定されていることを確認します。
- ページにコンポーネントを追加します。コンポーネントの「統合」セクションに、 「共有ブラウズ・ランチャ」コンポーネントがリストされます。
- 3. コンポーネントとその外観を編集するには、そのコンポーネントのメニュー・アイ

コン この この この この この この に、 この この に、 たま、 の この に、 たま、 の たの して、 たま、 たる し、 の よりの にの ら にの 追加、 およ び 背景色、 フォントと 枠線のカスタマイズを行います。 より細かくスタイルを設定 する 場合は、現在の サイト・ テンプレートの テーマ・ デザイン・ フォルダにある design.json および design.css ファイルのスタイル・ クラスを編集するか追 加してください。 スタイル・ クラスの接頭辞は scs-cobrowse です。

サイトを公開したら、サイトのビジターが「共有ブラウズ」ボタンを使用して、お客様の組織の担当者とセッションを開始できます。ビジターが「共有ブラウズ」ボタン をクリックすると、セキュアなセッション ID が表示されます。ビジターはその ID を、 共有ブラウズ・エージェント・コンソールへのアクセス権を持っているお客様の組織 の担当者に電話で伝えます。エージェントがコンソールを使用してセッションを開始



すると、ビジターかエージェントがそのセッションを停止するまで存続します。起動ポイント2の設定を使用する場合、ランチャ・コンポーネントには、cec-start-cobrowseの固定 ID があります。共有ブラウズ・コンソールでは、その ID を使用します。

#### 使用上のノート

共有ブラウズのインスタント・モード(ICB)を使用する場合は、同じ共有ブラウズ・サイト ID で iFrames コンテンツが有効化されていないと、ビデオや組込みの iFrame をページで表示で きません。その結果、Oracle Content Management の一部のコンポーネントが、ICB モード の共有ブラウズ・エージェント・コンソールにレンダリングされなくなります。次のコンポ ーネントをレンダリングするには、共有ブラウズ拡張モード(ACB)を使用してください。

- ビデオ
- Youtube
- ドキュメント・マネージャ
- Facebook のいいね
- Twitter のフォロー
- Twitter の共有
- Facebook のおすすめ
- 会話

コンテンツの取得に iFrame を使用するカスタム・コンポーネントも、インスタント・モード ではレンダリングされません。

Oracle Cobrowse Cloud Service の詳細は、*共有ブラウズ・デプロイメントおよび使用ガイド* の共有ブラウズの概要を参照してください。セキュア・サイトまたは開発中のサイトで共有 ブラウズを使用する際の詳細は、共有ブラウズ統合の有効化を参照してください。

### Oracle Intelligent Advisor

Oracle Intelligent Advisor (以前の Oracle Policy Automation)は、トラブルシューティングのフィードバックやサービスに対する適格性の評価など、オンラインのインタビュー・シナリオの実装に使用します。自然言語で記述された Microsoft Word や Excel 文書内のルールを取得し、インタビューと呼ばれる、それらのルールに関するインタラクティブな顧客サービス・エクスペリエンスを確立することで、複数のチャネルにわたってアドバイスを提供します。

Intelligent Advisor 機能を使用する前に、それを構成して有効にする必要があります。使用中の Intelligent Advisor ハブのホスト名、URL、ユーザー名およびパスワードの追加を含め、サ ービス管理者が、サービスに対して機能を有効化します。この2つのサービス間の統合では SSO を行うため、どちらのサービスも同じアイデンティティ・ドメイン内に存在している必 要があります。Oracle Content Management の統合および拡張の Intelligent Advisor との統 合を参照してください。

Intelligent Advisor 側で、インタビューを作成し、ホスト・サイトに格納する必要があります。 また、Intelligent Advisor ホストで使用するように Oracle Content Management サービスを認 可する必要があります。

**Intelligent Advisor** が構成されて有効化されたら、**Intelligent Advisor** コンポーネントをサイト 上のページに追加できます。

1. 編集するページに移動し、 **し**が「**編集」**に設定されていることを確認します。



- ページにコンポーネントを追加します。コンポーネントが、「コンポーネント」リ ストの「統合」セクションに表示されます。
- 3. コンポーネントとその外観を編集するには、そのコンポーネントのメニュー・アイ

サイトを公開すると、サイトのビジターには、Intelligent Advisor コンポーネントで選 択されたインタラクティブなインタビューが表示されます。Intelligent Advisor の詳細 は、Intelligent Advisor ドキュメント・ライブラリを参照してください。

### **Oracle Visual Builder**

Oracle Visual Builder は、アプリケーション開発インフラストラクチャのためのホスト 環境です。オープンソースの標準をベースとするソリューションで、Oracle Cloud 内 でのアプリケーションの開発やコラボレート、デプロイが可能です。

#### 最初のステップ

Oracle Visual Builder を使用するには、有効化して構成する必要があります。アプリケ ーションを作成して格納するホストの名前の追加を含め、サービス管理者が、サービ スに対して機能を有効化します。*Integrating and Extending Oracle Content Management* の Oracle Visual Builder との統合を参照してください。この2つのサー ビス間の統合では SSO を行うため、どちらのサービスも同じアイデンティティ・ドメ イン内に存在している必要があります。

**Oracle Content Management** でこの機能を使用するには、**Oracle Visual Builder** 側で次の内容を実行しておく必要があります:

- Oracle Visual Cloud Service のサイトで、Cross-Origin Resource Sharing (CORS) が有効化されている必要があります。
- アプリケーションを作成し、埋込みできるようにして、Oracle Content Management で使用できるよう構成する必要があります。
- Web アプリケーションを作成し、iframe に埋込みできるようにする必要があります。サイト SDK をインポートし、Web アプリケーションで参照する必要があります。"id"というページ URL パラメータを Web アプリケーションに追加する必要があります。

#### Oracle Visual Builder コンポーネントの作成

統合を有効化し、アプリケーションと Web アプリケーションを作成して使用する準備 ができたら、サイト・ページに追加する各アプリケーションの新しいコンポーネント を作成する必要があります。

Oracle Visual Builder で、公開された Web アプリケーションの URL を取得します。Web アプリケーションを含むライブ・プロジェクトをクリックし、Web アプリケーションをクリックします。アドレス・バーから URL をコピーします。



- 2. Oracle Content Management で、「開発者」をクリックし、「すべてのコンポーネントの表示」をクリックします。登録されているリモート・コンポーネントおよびレイアウトが表示されます。
- 3. 「作成」をクリックし、「Visual Builder コンポーネントの作成」を選択する必要がありま す。
- **4.** コンポーネントの名前を入力します。別のコンポーネントまたはレイアウトで使用されている名前は使用できません。

文字、数字、アンダースコア()およびハイフン(-)を使用できます。スペースを入力した 場合、アンダースコアに自動的に置換されます。

テンプレート、テーマ、コンポーネント、サイトまたはサイト・ページについて、 authsite、content、pages、scstemplate_*、_comps、_components、_compsdelivery、 _idcservice、_sitescloud、_sitesclouddelivery、_themes、_themesdelivery といった名前 を使用しないでください。documents および sites は、サイト・ページには使用できます が、テンプレート、テーマ、コンポーネントまたはサイトには使用しないでください。

- 5. オプションで、コンポーネントの説明を入力します。
- 6. ライブ Web アプリケーションまたはライブ・クラシック・アプリケーションへの URL を 貼り付けます。
- 7. 「作成」をクリックします。コンポーネントが作成されると、名前がコンポーネントのリ ストに表示されます。リストでコンポーネント名をクリックすると、コンポーネントまた はレイアウトを構成するフォルダとファイルを確認できます。
- 8. コンポーネントに割り当てられているデフォルト・アイコン以外のアイコンを選択するに は:
  - a. リストからコンポーネントを選択します。
  - **b. 「プロパティ」**をクリックします
  - c. 「コンポーネント・ロゴ」タブをクリックします。
  - d. ロゴのギャラリでロゴをクリックし、「完了」をクリックします。

#### サイト・ページへのコンポーネントの追加

アプリケーションのコンポーネントをサイト・ページの**1**つに追加できるようになりました。 コンポーネントは、パブリック・ページまたはセキュア・ページに追加できます。

- 1. 編集するページに移動して、 **し**が「**編集」**に設定されていることを確認します。
- ページにコンポーネントを追加します。「カスタム」セクションにコンポーネントが表示 されます。
- コンポーネントとその外観を編集するには、そのコンポーネントのメニュー・アイコン
   をクリックして、「設定」を選択します。コンポーネントのラベルや外観、サイズ、

位置を調整できます。Oracle Visual Builder ホストのコンポーネントに関連付けられてい るデフォルトのスタイルを使用するには、「スタイル」タブを使用します。より細かくス タイルを設定する場合は、現在のサイト・テンプレートのテーマ・デザイン・フォルダに ある design.css ファイルのスタイル・クラスを編集するか追加してください。スタイル・ クラスの接頭辞は scs-component です。

サイトの編集中は、プレビュー・モードでコンポーネントを表示できます。サイトを公開すると、サイトのビジターには、そのコンポーネントで使用するものとして Oracle Visual



Builder アプリケーションが選択された状態で表示されます(コンポーネントは、そのページの iFrame で実行されています)。



# 第 III 部 サイトの公開と管理

このパートでは、サイトの操作、セキュリティ保護、改善および公開の方法を詳細に説明し ます。次の章が含まれます。

- サイトの管理
- サイトの公開
- セキュア・サイト
- 多言語サイトの操作
- サイト・リダイレクトまたは URL マッピングの使用
- サイト・パフォーマンスの向上



# 11 サイトの管理

サイト・レイアウトまたはコンテンツを変更するには、エディタで更新を作成し、開きます。 サイト自体とサイトのプロパティを作成および管理するには、サイト・マネージャのオプシ ョンを使用します。

- 「サイト」ページについて
- サイトおよびサイト設定の管理
- サイトのオンライン化またはオフライン化
- サイトの説明、ロゴまたは埋込みプロパティの変更
- 検索エンジンのプロパティの設定
- サイト設定のカスタマイズ
- 共有ブラウズ統合の有効化
- 分析追跡の追加

# 「サイト」ページについて

「サイト」ページは、Web サイトを操作するための開始点です。



コールア ウト	説明
1	<b>パネル・トグル</b> では、ナビゲーション・メニューの表示と非表示を切り替えます。
2	<b>フィルタ・メニュー</b> では、サイト・リストをフィルタリングして、アクセス可能な すべてのサイトまたはサイトの一部を表示することができます。サイト・ガバナン スが有効になっている場合、保留中のサイト・リクエストを表示できます。
3	リストのサイトを選択した後に、 <b>アクション・バー</b> が使用できるようになります。 アクション・バーのオプションを使用して、サイトのオープン、表示、ステータス の変更、またはサイト名の変更を行い、その他のタスクを実行します。
	表示されるオプションは、選択したサイトに対するロールによって異なります。た とえば、サイトを作成した場合、そのサイトに対してマネージャ・ロールを持ち、 リストされたすべてのタスクを実行できます。ただし、他のユーザーがあなたとサ イトを共有し、参照者ロールをあなたに割り当てた場合、サイト・プロパティを参 照できますが、何も変更することはできません。
	アクション・バー・オプションは、サイトの現在のステータスによっても異なりま す。 <b>「名前の変更」</b> および <b>「削除」</b> オプションは、サイトがオフラインである場合の み使用できます。
	<b>ヒント:</b> ショートカットを探していますか。サイトを右クリックしてコンテキスト を開き、オプションを選択してください。
4	<b>サイト・リスト</b> には、あなたが所有しているサイトと、あなたと共有されているサ イトがすべて表示されます。リストには、サイト名、更新数、サイトがセキュアか どうか、およびサイトのステータス(オンライン、オフラインまたは更新保留)が含ま れます。
5	コンテンツ管理者としてログインしている場合、 <b>「管理」メニュー</b> が表示されます。 これらのオプションにより、サービス設定の変更、統合の設定およびリポジトリや 他のアセット関連機能の設定を行えます。
6	<b>ユーザー・メニュー</b> には、プリファレンスの設定、フィードバックの提供、ヘルプ へのアクセスおよびサインアウトのためのオプションがあります。ユーザー・ピク チャをクリックして、メニュー・オプションを表示します。
7	「作成」をクリックして、Web サイトの作成プロセスを開始します。サイトを作成 する前に、管理者がサイトの作成を有効にし、1 つ以上のテンプレートを提供する必 要があります。サイト・ページの「作成」オプションまたはテンプレート・ページ のテンプレートが表示されない場合は、サービス管理者に連絡してください。
8	<b>ビュー・アイコン</b> をクリックして、サイトがグリッド、リストまたは表ビューに表示されるかどうかを選択します。
9	<b>ソート・オプション</b> を使用して、サイトの表示順序を変更します。
10	<b>番号の更新</b> をクリックして、サイトの既存の更新を編集したり、サイトの新規更新 を作成します。サイトに更新がない場合、サイトを開いて、サイト・ビルダーを編 集モードに変更することで新しい更新を作成できます。
11	「オフライン」、「オンライン」または「再公開」アイコンをクリックして、サイトの ステータスの変更、またはコミット済だがライブ・サイトへは公開されていない変 更の公開を行います。
12	<b>セキュリティ・</b> アイコンは、サイトにログインが必要かどうかを示します。
13	<b>言語リスト</b> を使用して、サイトに関連付けられたローカリゼーション・ポリシーで 指定されている言語を表示します。ローカライズ・バージョンをプレビューするに は、言語を選択してサイトを開きます。

# サイトおよびサイト設定の管理

サイト管理には、サイトの作成および編集と、サイトで使用されるコンテンツの管理が含ま れます。*サイト・ビルダー*を使用して、サイト・ページとページ・コンテンツを作成、コピ ーおよび削除します。フォルダおよびファイル・マネージャを使用して、サイト全体を作成、 コピー、共有および削除します。

「サイト」ページおよびサイト・ビルダーのメニュー・バーまたは右クリック・メニューのオ プションを使用して、次のタスクを実行します。

タスク	説明
サイト・ビューのフ ィルタ	サイト・リストをフィルタリングするには、フィルタ・メニューの次のオプショ ンのいずれかを選択します。 <b>オペイ</b> - アクセス可能かせイトがすべて表示されます。白分がサイト管理者
	で、サイト・ガバナンスが有効になっている場合、環境内のすべてのサイト にアクセスできます。
	<ul> <li>目分か所有—目分が所有しているすべてのサイトが表示されます。</li> <li>自分と共有—自分が共有しているすべてのサイトが表示されます。</li> </ul>
	<ul> <li>オンライン—オンライン(ライブ)のサイトが表示されます。</li> </ul>
	• <b>オフライン</b> —オフラインのサイトが表示されます。
	<ul> <li>リクエスト(サイト・ガバナンスが有効になっている場合にのみ使用可能)—</li> <li>保留中のサイト・リクエストが表示されます。自分がサイト管理者である場合、システム内のサイト・リクエストがすべて表示されます。それ以外の場合は、自分で送信したサイト・リクエストのみがまでされます。</li> </ul>
	<ul> <li>ごみ箱 — 自分が所有している削除済サイトを表示します。自分がサイト管理者で、ガバナンスが有効になっている場合は、すべての削除済サイトも表示されます。</li> </ul>
サイトの作成	サイトを作成するには、「作成」をクリックします。サイトを作成する場合、テン プレートから開始します。テンプレートには、ユーザーがサイトの使用を開始す るにあたり必要なもの、たとえば、サイト・コード・フレームワーク、サンプル・ ページとコンテンツを含むデフォルト・サイト、スタイルが設定されたテーマ、
	イメージなどのリソース、カスタム・コンポーネントなどがすべて含まれます。サイトの作成を参照してください。
	サイト・カハナンスか有効になっているかとっかは、「サイト」ページのノイル タ・メニューで確認できます。「リクエスト」オプションが表示されている場合、 サイト・ガバナンスは有効になっています。サイト・ガバナンスの理解を参照し てください。
サイト・リクエスト の表示および管理	保留中のサイト・リクエストを表示するには(サイト・ガバナンスが有効になって いる場合にのみ可能)、フィルタ・メニューで「 <b>リクエスト」</b> を選択します。自分
	かサイト管理者である場合、システム内のサイト・リクエストかすべて表示され ます。それ以外の場合は、自分で送信したサイト・リクエストのみが表示されま す。サイト・リクエストの管理を参照してください。
サイトの編集	サイトを編集するには、右クリック・メニューの「開く」を選択するか、アクシ
	ョン・バーの 🗗 をクリックします。サイトの編集を参照してください。
サイトについての ディスカッション または注釈付け	サイト・ビルダーでサイトについてディスカッションし、注釈を付けることができます。サイトを選択して、右クリック・メニューの「開く」を選択するか、ア
	クション・バーの 2 をクリックします。サイトについてディスカッションする
	には、 Ш をクリックして会話パネルを開きます。サイトに注釈を付けるには、
	■ をクリックします。サイト・ビルダー・ページについてを参照してください。

タスク	説明
サイトの表示	サイトがライブ(オンライン)の場合にどのように表示されるかを確認するには、サ イトを選択し、右クリック・メニューの「 <b>表示」</b> を選択するか、アクション・バ
	の♥をクリックします。
サイトの公開	サイトを公開するには、これを選択し、右クリック・メニューまたはアクション・ バーの <b>「公開」</b> を選択します。サイト変更の公開を参照してください。
再公開	以前にサイトを公開してから、関連付けられたアイテムまたはポリシーが変更された(例: 関連付けられたローカリゼーション・ポリシーが更新されて言語の数が 増えた)場合、サイトを「再公開」してその変更をリフレッシュすることができま す。サイトを選択して、右クリック・メニューまたはアクション・バーの「再公 開」を選択します。
非公開	以前にサイトを公開してから、 <b>Oracle Cloud</b> 内のホスティングの場所からサイト・ファイルを削除する場合、サイトを選択して右クリック・メニューまたはアクション・バーの「非公開」を選択します。サイトを非公開にするにはオフラインにする必要があります。
サイトのオンライ ン化またはオフラ イン化	サイトをオンラインにするには、その前にサイトを公開する必要があります。サ イトをオンライン化またはオフライン化するには、サイトを選択して、右クリッ ク・メニューの <b>「オンラインにする」</b> または <b>「オフラインに設定」</b> を選択します。
	サイト・タイルの をクリックしてサイトをオンラインにすることも、サイト・
	タイルの をクリックしてサイトをオフラインにすることもできます。サイト が <i>オンライン</i> である場合、ユーザーは Web ブラウザを使用して指定したアドレス (URL)でサイトを表示できます。サイトが <i>オフライン</i> である場合、一般公開されま せん。サイトは Oracle Content Management でのみ表示できます。サイトのオ ンライン化またはオフライン化を参照してください。
サイト名の変更	サイトの名前を変更するには、これを右クリックして、右クリック・メニューの
	「名前変更」を選択するか、アクション・バーの ・ をクリックします。同じサー バー上の既存のサイト名とは異なる名前をサイトに対して入力します。
	サイトがオンラインの場合、名前の変更はできません。サイトをオフラインにす るには、サイトの所有者であるか、サイトに対してマネージャ・ロールを持って いる必要があります。
サイトのコピー	サイトをコピーして、サイトを迅速に構築できます。テーマ、すべての未処理の 更新内容、ページ、ページ・コンテンツおよびその他のすべてのアセット(イメー ジなど)を含む元のサイトのすべてが、指定した新しい名前の新規サイトにコピー されます。新規サイトはオフラインで、編集の準備ができています。
	ノート: サイト・カバナンスか有効になっている場合、コピーを作成する前にサイトで承認を得ることが必要な場合があります。
	複数のリポジトリからのコンテンツを使用するサイトをコピーしている場合、 OCE ツールキットを使用して、これを行う必要があります。OCE ツールキットで の開発および cec コマンドライン・ユーティリティの使用を参照してください。
	コピーするサイトを選択して、右クリック・メニューの <b>「コピー」</b> を選択するか、
	アクション・バーの



# タスク 説明 サイトからのテン プレートの作成 他のサイトの開始点として使用するサイトがある場合は、そのサイトからテンプ レートを作成できます。サイトを選択して、右クリック・メニューの「テンプレ ートの作成」を選択するか、アクション・バーの画をクリックします。「サイトか らのテンプレートの作成」を参照してください。

複数のリポジトリからのコンテンツを使用するサイ トからテンプレートを作成している場合、OCE ツー ルキットを使用して、これを行う必要があります。 OCE ツールキットでの開発および cec コマンドラ イン・ユーティリティの使用を参照してください。

サイトの削除また 適切な権限を持っている場合、サイトとそのコンテンツを削除できます。サイト は復元 を削除すると、すべての未処理の更新、サイト・ページ、ページ・コンテンツお よびアセット(ページに追加したイメージなど)を含む、サイト・フォルダ内のすべ てがごみ箱に移動されます。

> サイトを削除または復元できるのは、サイトを作成した(サイトの所有者である)場 合か、他のユーザーがあなたとサイトを共有していて、あなたにコントリビュー タまたはマネージャ・ロールを付与している場合です。

> オンラインの場合、サイトを削除できません。サイトをオフラインにするには、 サイトの所有者であるか、サイトに対してマネージャ・ロールを持っている必要 があります。

> また、公開されているサイトは削除できません。先にコンテンツを非公開にする 必要があります。サイトに複数のリポジトリからのコンテンツが含まれる場合、 各リポジトリ内の関連付けられているコンテンツを非公開にする必要がありま す。

> サイトを削除するには、これを選択して、右クリック・メニューの**「削除」**を選

択するか、アクション・バーの ^{●●} をクリックします。サイトとすべてのサイト 更新をごみ箱に移動するよう求められます。削除したサイトは、次までごみ箱に あります。

- サイトを復元します。
- サイトを完全に削除します。
- ごみ箱の割当て制限に達しました。
- ごみ箱は、サービス管理者が設定した間隔に基づいて自動的に空になります。
   デフォルト値は 90 日です。

サイトを復元するには、「サイト」メニューで「ごみ箱」を選択します。リストか らサイトを選択して「復元」を選択します。

タスク	説明
サイトへのメンバ ーの追加	管理者が共有を有効にしている場合、他の Oracle Content Management ユーザ ーとサイトを共有し、そのユーザーに付与した権限に基づいて Oracle Content Management でのサイトの表示、変更または管理を許可することができます。
	サービスにアクセスできるすべてのユーザーがメンバーとみなされます。サイト を共有する場合、メンバーがサイトで実行できることまたは実行できないことを 定義するロールを割り当てます。メンバーは、割り当てるロールに従ってのみサ イトを使用できます。
	共有するサイトを選択し、右クリック・メニューの「メンバー」を選択するか、 アクション・バーの「サイドバー」をクリックし、「メンバー」を選択してメンバ ー・サイドバーを開きます。メンバー・サイドバーが開いたら、「メンバーの追 加」をクリックします。
	<ul> <li>1つ以上のユーザー名または電子メール・アドレスを入力し、次のロールのいず れかを割り当てます。</li> <li>参照者:参照者はエディタでサイトを表示できますが、変更することはできま</li> </ul>
	せん。 <ul> <li>ダウンロード実行者: サイトでは、ダウンロード実行者ロールは参照者ロール</li> <li>と同じ権限を提供し、ユーザーはサイトから新しいテンプレートを作成できます。</li> </ul>
	<ul> <li>コントリビュータ:参照者と同じ権限に加え、オフラインの場合、サイトの編集、サイト・ページの削除およびサイトの削除を行うこともできます。</li> <li>マネージャ:コントリビュータと同じ権限に加え、ユーザーの追加、ロールの割当て、オンライン・サイトへの変更の公開、およびオンラインとオフラインのサイトの切替えを行うこともできます。サイトの作成者(所有者)は自動的にマネージャ・ロールを割り当てられます。</li> </ul>
	ノート:     複数のリポジトリにアクセスできるサイトを共有している場合、デフォルトのリポジトリのみが共有されます。追加のリポジトリは個別に共有する必要があります。
サイト・プロパティ (説明、ロゴなど)の 変更	サイトの説明の変更、ロゴの追加、サイトの埋込みの許可、バニティ URL の追加、およびサイトの重要な情報(サイトの URL、サイトの所有者、その他の詳細) の表示を行うには、サイトを選択して右クリック・メニューの「プロパティ」を 選択するか、アクション・バーの「サイドバー」をクリックし「プロパティ」を 選択します。「サイトの説明、ロゴまたは埋込みプロパティの変更」を参照してく ださい。
サイトの翻訳	関連付けられたリポジトリで許可されている場合、サイト・ファイルをエクスポートして翻訳した後、翻訳済ファイルをインポートすることにより、エンタープライズ・サイトを複数の言語に翻訳できます。サイトの翻訳を参照してください。
検索エンジンのプ ロパティの設定	サイト・ビルダーで、検索エンジンがサイトのコンテンツを識別するためのキー ワードおよびテキストを指定できます。検索エンジンのプロパティの設定を参照 してください
サイト設定(お気に 入りアイコン、コン トローラ・ファイル など)のカスタマイ ズ	サイト・ビルダーで、様々なブラウザおよびプラットフォームで使用するサイト・ アイコンを指定したり、リンク動作を処理するコントローラ・ファイルを追加で きます。サイト設定のカスタマイズを参照してください



タスク	説明
共有ブラウズの有 効化	サイト・ビルダーで、画面の共有や他のユーザーとの共有ブラウズ・セッション の開始を可能にするコラボレーション・ツールである Oracle Cobrowse Cloud Service を有効化できます。たとえば、これを注文フォームに含めることにより、 顧客が注文する際に担当者が顧客の画面を表示することが可能になります。共有 ブラウズ統合の有効化を参照してください
サイトへの分析追 跡の追加	サイト・ビルダーで、Web 分析追跡用の JavaScript 追跡コードのスニペットをサイトに追加し、Google、Adobe または Oracle Infinity などの外部のアナリティクス・プロバイダとの統合を容易にできます。分析追跡の追加を参照してください。
ライブ・サイトの保 護	サイトを公開してオンラインで使用できるようにすると、すべてのユーザーに対 して公的に利用可能になります。ただし、サイトの所有者であるか、マネージャ・ ロールを持っている場合、登録済ユーザーまたはユーザーの一部がライブ・サイ トを使用できるように制限できます。サイトのステータスを変更するには、サイ トの所有者であるか、マネージャ・ロールを持っている必要があります。
	保護するサイトを選択し、右クリック・メニューの「プロパティ」を選択するか、 アクション・バーの「サイドバー」をクリックし、「プロパティ」を選択して「サ イト・セキュリティ」タブを選択します。サイト・セキュリティの変更を参照し てください。
サイト・リダイレク トの追加	サイト URL が変更された場合、リダイレクトによって、ある URL (ソース)が別の URL (ターゲット)に転送されます。これにより、ユーザーのブックマークと検索 エンジンのランキングが維持されるため、便利です。サイト・リダイレクトの追 加を参照してください。

# サイトの説明、ロゴまたは埋込みプロパティの変更

ほとんどのサイト・プロパティは、サイトの作成または更新時に設定されます。プロパティ を表示すると、サイト URL、サイト所有者およびその他の詳細など、サイトに関する有益な 情報を参照できます。

サイトの説明やロゴを変更し、バニティ・サイトを追加して、サイトを埋込み可能にするに は:

- サイトを選択し、右クリック・メニューの「プロパティ」を選択するか、アクション・バ ーの「サイドバー」をクリックし、「プロパティ」を選択します。
- 2. サイトのオプションの説明を変更するには、「サイトのプロパティ」タブをクリックして、 説明を入力または変更します。
- サイトの URL は直接変更できませんが、サイトにアクセスできるように、サイトの URL を選択してコピーし、これをドキュメント、プレゼンテーションおよび電子メールに貼り 付けることができます。

サイトがオンラインの場合は、[●]をクリックしてオンライン・サイトに直接移動します。

- 4. サイトを埋込みサイトとして使用できるようにするには、「埋込み可能なサイト」の横の 「はい」を選択します。
- サイトのイメージを変更するには、「サイト・ロゴ」タブをクリックし、「変更」をクリックします。使用するイメージを検索して選択します。これは、.png、.jpeg または.jpg ファイルで、4:3 (長方形)の比率である必要があります。最適なサイズは、300x225 ピクセルです(イメージを小さくすると、ゆがんで見えることがあり、イメージを大きくすると、パフォーマンスに影響する可能性があるためです)。



6. 完了後、「閉じる」をクリックします。

# 検索エンジンのプロパティの設定

検索エンジンがサイトのコンテンツを識別するためのキーワードおよびテキストを指 定できます。

検索エンジン最適化(SEO)設定をサイト・レベルおよびページ・レベルで定義できま す。次の表に示すように、サイト・レベルの設定は、個々のページの同様の設定に付 加またはオーバーライドされます。

オプション	サイト・レベル	ページ・レベル
説明またはページの説 明	サイト自体には含まれない、サイ トに関する一般的な情報を指定し ます。サイトの説明は、サイトの 各ページに含まれます ホーム・ページに「ページ設定」 を使用して設定された値がない場 合のみ、ホーム・ページのページ・ レベルの説明として、この説明も 使用されます。	ページ自体には含まれない、ペー ジに関する一般的な情報を指定し ます。ページの説明は、サイトの 各ページに含まれるサイトの説明 に付加されます。
キーワード	サイトのすべてのページに適用さ れる用語または概念を識別しま す。 これらの値は個々のページに指定 されたキーワードに追加(付加)さ れます。	個々のページに適用される用語ま たは概念を識別します。 ページ・キーワードは、ページの テキストに表示されないまたはイ メージに表示される用語または概 念を識別するのに有用です。
ヘッダーまたはペー ジ・ヘッダー	サイトにアナリティクスまたはト ラッキング用のスクリプト記述ま たはタグを追加します。サイト・ ヘッダー・コンテンツは、サイト の各ページに含まれます。	サイトにアナリティクスまたはト ラッキング用のスクリプト記述ま たはタグを追加します。ページ・ ヘッダー・コンテンツは、各ペー ジに含まれるサイト・ヘッダー・ コンテンツに付加されます。
フッターまたはペー ジ・フッター	サイトにアナリティクスまたはト ラッキング用のスクリプト記述ま たはタグを追加します。サイト・ フッター・コンテンツは、サイト の各ページに含まれます。	サイトにアナリティクスまたはト ラッキング用のスクリプト記述ま たはタグを追加します。ページ・ フッター・コンテンツは、各ペー ジに含まれるサイト・フッター・ コンテンツに付加されます。

サイト・レベルで次の検索除外オプションを選択した場合、設定はすべてのページに 適用され、個々のページの設定をオーバーライドします。サイト・レベルでオプショ ンを選択しない場合、オプションを個々に指定したページのみでこのオプションが使 用されます。

オプション	サイト・レベル	ページ・レベル
検索エンジンから非表示	選択した場合、NOINDEX メタ・タ グがすべてのページに追加され、 検索エンジンでは、サイトのどの ページでもコンテンツの索引が作 成されなくなります。この場合、 サイトとそのすべてのページが Web 検索結果に表示されません。	選択した場合、NOINDEX メタ・タ グが現在のページに追加され、検 索エンジンでは、ページのコンテ ンツの索引が作成されなくなりま す。この場合、個々のページが Web 検索結果に表示されません


オプション	サイト・レベル	ページ・レベル
ページ・リンクを検索 エンジンから隠す	選択した場合、NOFOLLOW メタ・ タグがすべてのページに追加さ れ、検索エンジンでは、サイトの どのページでもリンクに従ってリ ンク先の索引が作成されなくなり ます。	選択した場合、NOFOLLOW メタ・ タグが現在のページに追加され、 検索エンジンでは、ページでリン クに従ってリンク先の索引が作成 されなくなります。
ページの説明を検索エ ンジンから隠す	選択した場合、NOSNIPPET メタ・ タグがすべてのページに追加さ れ、検索エンジンでは、検索結果 にページの後の(上で指定した)説 明を含めなくなります。	選択した場合、NOSNIPPET メタ・ タグが現在のページに追加され、 検索エンジンでは、検索結果にペ ージの後の(上で指定した)説明を 含めなくなります。

検索エンジン最適化(SEO)の設定を変更するには:

1. 編集するサイトを開きます。

2. サイドバーの <sup>
②</sup>をクリックし、 **SEO** をクリックします。

サイトの説明(オプション)を入力します。サイトの説明は、サイトの各ページに含まれます。

ホーム・ページに「**ページ設定」**を使用して設定された値がない場合のみ、ホーム・ページのページ・レベルの説明として、この説明も使用されます。

4. 必要に応じて、検索エンジンがサイトのコンテンツを識別するためのカンマ区切りのキー ワードを指定します。

サイト・キーワードは、サイトのすべてのページに適用される用語または概念を識別しま す。これらの値は個々のページの「ページ設定」を使用して指定されたキーワードに追加 (付加)されます。

- 必要に応じて、サイトにアナリティクスまたはトラッキング用のヘッダーのスクリプト記述またはタグを追加します。ヘッダー・コンテンツは、サイトの各ページに含まれます。ヘッダーで使用するすべてのコードを検証して、適切に動作し、サイトにセキュリティ・リスクをもたらさないことを確認してください。
- 6. 必要に応じて、サイトにアナリティクスまたはトラッキング用のフッターのスクリプト記述またはタグを追加します。フッター・コンテンツは、サイトの各ページに含まれます。フッターで使用するすべてのコードを検証して、適切に動作し、サイトにセキュリティ・リスクをもたらさないことを確認してください。
- 7. 前の表で説明されているとおり、必要に応じてオプションを1つ以上選択して、検索結果 の情報を非表示にできます。
- 8. 現在の更新における保留中のすべての変更を保存するには、「保存」をクリックします。

# 検索エンジン最適化のためのプリレンダー・サービ スの有効化

検索クローラ・リクエストに対応するために、Oracle Content Management で作成されたサイトで、静的 HTML ページのプリレンダーを有効化できます。

プリレンダー・サービスが有効化されていて、検索クローラからのリクエストが来た場合、 ページはキャッシュ内で検索されます。



- キャッシュにページが見つかり、15日を経過していない場合は、キャッシュされたページでリクエストに対応します。
- キャッシュのページが 15 日を経過しているか、キャッシュに見つからない場合 は、Oracle Content Management から新しいバージョンがフェッチされ、キャッシ ュが更新されて、新しくプリレンダーされたページで対応します。
- キャッシュにも Oracle Content Management にもページがない場合は、エラー・ページが戻されます。

パブリック・サイトの有効期限が切れたページや、まだキャッシュされていない新し いサイトは、1日に1回レンダリングされます。ページがすでにキャッシュされている 場合、キャッシュは更新されません。プリレンダー・サービスが有効化されている場 合、更新や再公開が行われたサイトでは、プリレンダーされたキャッシュは1時間ご とに更新されます。サイト・ページの変更を行っていて、それらの変更が最適化に与 える影響をテストする場合は、サイト・プロパティでプリレンダーされたキャッシュ を手動で更新できます。

- 1. サイド・ナビゲーションで、「サイト」をクリックします。
- サイトを選択し、右クリック・メニューの「プロパティ」を選択するか、アクション・バーの「サイドバー」をクリックし、「プロパティ」を選択します。
- 3. プロパティ・サイドバーの「SEO」をクリックします。
- キャッシュが最後にリフレッシュされた日時がリストされます。キャッシュを更 新するには、今すぐリフレッシュをクリックします。進行状況バーにキャッシュ・ ステータスが表示されます。

### 💉 ノート:

サイト SEO の設定で、ページが検索エンジンに非表示になるよう設定されている場合、そのページのプリレンダーやキャッシュは行われません。

**Oracle Content Management** でプリレンダー・サービスを有効にするには:

- 1. サイド・ナビゲーションの「管理」領域で、「システム」をクリックします。
- 2. システム設定メニューで、「サイトの SEO」を選択します。
- 3. 「有効」をクリックします。
- 4. 必要な user-agent を追加で定義します。

#### 表 11-1 Oracle Content Management のハードコードされた User-agent

User-agent	User-agent	User-agent	User-agent
baiduspider	facebookexternalhit	twitterbot	rogerbot
linkedinbot	embedly	quora link preview	showyoubot
outbrain	pinterest/0.	developers.google.co m/+/web/snippet	slackbot
vkShare	W3C_Validator	redditbot	Applebot
WhatsApp	flipboard	tumblr	bitlybot
SkypeUriPreview	nuzzel	Discordbot	Google Page Speed



User-agent	User-agent	User-agent	User-agent
Qwantify	pinterestbot		

#### 表 11-1 (続き) Oracle Content Management のハードコードされた User-agent

### 表 11-2 config.cfg ファイルの PrerenderUserAgents プロパティに事前構成されている追加 の User-agent

User-agent	User-agent	User-agent	User-agent
AddSearchBot	AdIdxBot	AdsBot-Google	AdsBot-Google-Mobile- Apps
AppEngine-Google	Baidu-YunGuanCe	Bingbot	BingPreview
DuckDuckBot	DuckDuckGo- Favicons-Bot	endeca webcrawler	Exabot
Facebot	Feedfetcher-Google	FeedValidator	Fetch
FlipboardProxy	Google Favicon	Google Web Preview	Google-Adwords- Instant
Googlebot	Googlebot-Image	Googlebot-Mobile	Googlebot-News
Googlebot-Video	Google-PhysicalWeb	Google-Structured- Data-Testing-Tool	HubSpot,ia_archiver
Mediapartners-Google	MSNBot	NetcraftSurveyAgent	nutch
Oracle Secure Enterprise Search	pinterest.com	PIs-Google	SEOkicks
seoscanners	Siteimprove.com	Slurp	Sogou web spider
VSE/1.0	W3C_CSS_Validator	W3C_I18n-Checker	W3C_Unicorn
W3C-checklink	W3C-mobileOK	Y!J	Yahoo Link Preview
Yahoo! Slurp	Yandex	YandexBot	YunGuanCe

# 静的サイト配信オプションの設定

会社でコンパイル済サイトが使用されている場合、静的サイトがキャッシュされる期間、お よびどのモバイル user-agent がサイト・コンパイルによってサポートされているアダプティ ブ・モバイル・レイアウトを要求するかを制御できます。

- 公開時の自動コンパイルの有効化
- コンパイル済サイトのデフォルト・キャッシュ制御ヘッダーのオーバーライド
- コンパイル済アダプティブ・レイアウトをサポートするモバイル User-Agent の指定

### 公開時の自動コンパイルの有効化

Oracle Content Management には、追加の構成なしでサイトを簡単にコンパイルできるよう にする組込みのコンパイル・サービスがあります。または、サービス管理者は、テストの目 的でコンパイル・サービスを手動で設定してサイト・コンパイルを検証するか、カスタム・ ライブラリを使用できます。使用されているコンパイル・サービスに関係なく、サイトが公 開または再公開されるときに、サイトのコンパイルを選択できます。



- 1. Oracle Content Management にログインした後、サイド・ナビゲーション・メニュ ーで、「サイト」をクリックします。
- 2. 変更するサイトを選択し、「プロパティ」をクリックします。
- 3. プロパティ・ダイアログで「静的配信」をクリックします。
- 4. 「自動コンパイルの有効化」セクションの公開後にサイトをコンパイルを有効にします。
- 5. 終了したら、「完了」をクリックします。

自動コンパイルを有効にしてサイトを公開または再公開すると、公開ステータスが追 跡され、「サイト」ページのサイト・タイルに表示されます。公開が完了すると、コン パイル・プロセスが「サイトのプロパティ」ダイアログの「静的配信」セクションで 追跡されます。コンパイルが完了すると、「サイトのプロパティ」の「静的配信」セク ションに最後のコンパイルの日時がリストされ、コンパイル・ログをダウンロードす るためのリンクが表示されます。

### コンパイル済サイトのデフォルト・キャッシュ制御ヘッダ ーのオーバーライド

コンパイル済のサイトは、デフォルトで 300 秒(5 分間)ユーザーのブラウザにキャッシ ュされます。サービス管理者はこのデフォルトを変更できますが、サイト開発者とし てサイト・プロパティで特定のサイトのデフォルトをオーバーライドできます。

- 1. Oracle Content Management にログインした後、サイド・ナビゲーション・メニュ ーで、「サイト」をクリックします。
- 2. 変更するサイトを選択し、「プロパティ」をクリックします。
- 3. プロパティ・ダイアログで「静的配信」をクリックします。
- 4. レスポンス・ヘッダーのキャッシュセクションで、Cache-control: max-age=を入力し、次にページがユーザーのブラウザにキャッシュされる秒数の数値を入力します。たとえば、Cache-control: max-age=600の場合、ページが10分間キャッシュされます。
- 5. 終了したら、「完了」をクリックします。

インスタンスで Akamai が使用されている場合、既存の Akamai 設定を保持するには、 レスポンス・ヘッダーのキャッシュセクションを空白のままにします。既存の Akamai 設定をオーバーライドするには、Edge-Control: !no-store,maxage=1800,downstream-ttl=1800を入力します(太字のアイテムは秒単位のデフォルト 設定です)。

- !no-store は、この設定により、対応するプロパティの Akamai キャッシュ構成が オーバーライドされることを示しています。
- max-age により、Akamai がこのページをキャッシュする長さが決定されます。デフォルトは 1800 秒(30 分)です。この間、Akamai は、Oracle Content Managementのページをリクエストすることなく、ページのリクエストを実行します。
- downstream-tt1 は、Akamai に"Cache-Control: max-age"ヘッダーをそのレスポン スとともにクライアント・ブラウザに送信するように伝え、これらのブラウザに、 割り当てられた時間の間、ページをキャッシュするように指示します。デフォルト は 1800 秒(30 分)です。



### コンパイル済アダプティブ・レイアウトをサポートするモバ イル User-Agent の指定

サイトをコンパイルする際、モバイル・ページを作成してアダプティブ・レイアウトをサポ ートできます。サイト用のコンパイルされた標準ページのかわりに、サーバーがモバイル・ ページを配信できる user-agents を指定できます。ここで入力された値は、ブラウザによって 送信された user-agent ヘッダーに対して照合時に、大文字と小文字が区別されない部分文字 列として処理されます。キーワード Mobile は、一般的にモバイル・デバイス上のブラウザの user-agents 文字列で使用されることに注意してください。

- Oracle Content Management にログインした後、サイド・ナビゲーション・メニューで、 「サイト」をクリックします。
- 2. 変更するサイトを選択し、「プロパティ」をクリックします。
- 3. プロパティ・ダイアログで「静的配信」をクリックします。
- モバイル User-Agent セクションで、モバイル・ページを提供するユーザー・エージェントの user-agent 部分文字列のカンマ区切りリストを入力します。部分文字列のいずれかの部分が、ブラウザの user-agent 文字列と一致した場合、モバイル・ページが提供されます。

## バニティ URL の指定および構成

サイトにバニティ URL を設定すると、ユーザーは、Oracle Content Management インスタン スのドメインではなく、指定されたドメイン上の Oracle Content Management サイトにアク セスできます。たとえば、インスタンス URL が https://

myinstance.cec.ocp.oraclecloud.com/site/MyCustomerSite/であっても、https:// www.example.comのようなわかりやすい URL のほうが、覚えやすく、ブランディングに適し ていて、一般的には使いやすいです。また、必要とするものに応じて、Oracle Content Management サイトを https://www.example.com/store/のようなカスタム・パスでホストす ることもできます。

バニティ・ドメインを利用するには、いくつかのステップが必要です。

- コンテンツ配信ネットワーク(CDN)を使用します。
- Domain Name System (DNS)を使用してバニティ・ドメインを管理し、ドメインの正規名 (CNAME)レコードが CDN にマップされるようにします。
- バニティ・ドメインを保護する CDN に有効な証明書をデプロイします。
- バニティ・ドメインでリクエストを受け入れるように Oracle Content Management サイトを構成します。
- リクエストを Oracle Content Management に適切にルーティングするように CDN を構成します。

#### コンテンツ配信ネットワークの使用

コンテンツ配信ネットワーク(CDN)とは、Web サイトのパフォーマンスとセキュリティの向 上を目的とした、グローバルに分散したサーバーのプラットフォームです。CDN は、コンテ ンツのリクエストのパフォーマンスを最適化しつつ、ユーザーとサーバー間の距離を最小限 に抑えます。CDN の主な目的はユーザー・エクスペリエンスの向上ですが、CDN は転送中の



リクエストを変更するために使用することもでき、裏のプロセスがクリーンでなくて も、ユーザーにはクリーンなものが表示されるようにできます。

バニティ・ドメインでの Oracle Content Management サイトのホスティングをサポー トするには、CDN と連携して、構成済のバニティ・ドメインからのすべてのリクエス トを処理し、それらが適切に Oracle Content Management に戻るように構成し、リク エストに変更を加えて、Oracle Content Management で適切かつ安全に処理されるよ うにする必要があります。

#### Domain Name System (DNS)を使用したドメインの管理

Oracle Content Management サイトのバニティ・ドメインとして任意のドメインを使用できます。バニティ・ドメインとして使用するドメインは、Oracle Content Management サイトで使用するように構成する前に、制御する必要があります。

DNS の制限により、*example.com* のようなルート・ドメインを、*www* や store.example.com のような別のサブドメインなしで使用することは、おそらく不可能 です。DNS および CDN プロバイダに問い合せて、ルート・ドメインの使用が可能か どうかを確認してください。

DNS はパス・レベルではなくドメイン・レベルで機能するため、Oracle Content Management がドメインの一部のパスをホストし、別のサービスが他のパスをホスト するためには、ルーティングを CDN で処理する必要があります。DNS は、ドメイン およびサブドメイン・レベルでトラフィックを分離するためにのみ使用できます。

#### 証明書のデプロイ

バニティ・ドメインを保護する証明書は、CDN で作成およびホストする必要がありま す。証明書は、単一のドメイン、複数のドメインとサブドメイン、または*.example.com などのワイルドカードのサブドメインを保護できます。バニティ・ドメインには、任 意の組合せを使用できます。保護されているすべてのドメインが証明書の詳細に表示 されるため、これらの詳細を公開することが意図されていない場合は、個別の証明書 を使用する必要があります。

### 🎤 ノート:

証明書を作成してホストするプロセスは、多くの場合 CDN に固有であり、 これを行うための最善の方法を指定する必要があります。

#### バニティ・ドメイン用の Oracle Content Management サイトの構成

バニティ・ドメインを使用しているときに Oracle Content Management サイトを正し くロードするためには、最初に Oracle Content Management にバニティ・ドメインを 使用しているサイトの URL を通知する必要があります。これは、サイトのプロパテ ィ・タブの Oracle Content Management で行います。

- Oracle Content Management のサイド・ナビゲーションで、「サイト」をクリック します。
- バニティ・ドメインを使用するサイトを選択し、右クリック・メニューの「プロパ ティ」を選択するか、アクション・バーの「サイドバー」を選択します。
- **3.** 「バニティ・ドメイン」フィールドにバニティ・ドメインを入力し、**「保存」**をクリックします。



### 🔊 ノート:

Oracle Content Management がドメインでリクエストを受け入れる準備ができるまでに、最大1時間以上かかることがあります。進行状況は、サイトのプロ パティ・パネルでいつでも確認できます。

### リクエストをルーティングするためのコンテンツ配信ネットワークの構成

Oracle Content Management が適切に構成され、受け入れる準備ができると、バニティ・ド メインに対して行われたリクエストは、DNS を介して CDN にルーティングされ、CDN はそ のリクエストを Oracle Content Management に適切に転送する必要があります。たとえば、 サイト URL が https://myinstance.cec.ocp.oraclecloud.com/site/MyCustomerSite/であ る Oracle Content Management サイトが、バニティ・ドメイン https://www.example.com/ store で構成されている場合、CDN は次のように構成する必要があります:

- バニティ・ドメイン https://www.example.com/store を認識します
- バニティ・ドメイン https://myinstance.cec.ocp.oraclecloud.com/を使用して、オリジン Oracle Content Management インスタンスを識別します
- 特定のサイトのサイト・パス site/MyCustomerSite/を追加します
- そして、完全なサイト URL https://myinstance.cec.ocp.oraclecloud.com/site/ MyCustomerSite/をオリジン Oracle Content Management インスタンスに送信します。
- Oracle Content Management はリクエストを受け取り、CDN に応答します。CDN はユー ザーのブラウザへのリクエストを満たし、ユーザーにはバニティ・ドメイン https:// www.example.com/store のみを表示します

CDN の構成ステップは CDN に固有であることが多いため、CDN プロバイダと連携して、オリジン定義(オリジン Oracle Content Management インスタンスへのアクセス方法を CDN に 通知する)と、サイト・パス(正しいサイト・パスを付加する)を適切に構成します。

### 🎤 ノート:

リクエストに次のパスを含めることはできません:

- /documents*
- /content/published*
- /osn*
- /pxysvc*
- /site*

これらのパスへのリクエストは、サイト・パスを含めることを意図していません。 適切に処理されるように、オリジン Oracle Content Management インスタンスのル ートに解決される必要があります。



# 共有ブラウズ統合の有効化

共有ブラウズ機能は、Oracle Cobrowse Cloud Service で使用されるコラボレーション・ツールです。

**Cobrowse Cloud Service** との統合は、まずサービス管理者が、承認された統合として 追加する必要があります。*Oracle Content Management の統合および拡張*の Oracle **Cobrowse Cloud Service** との統合を参照してください。

**Oracle Cobrowse Cloud Service** 統合が有効化されたら、機能をサイト用に構成し、使用する特定のサイト・ページに追加できます。

サイトで共有ブラウズを有効化するには:

- 1. 編集するサイトを開きます。
- 2. サイドバーの 😳 をクリックし、 🛄 をクリックします。
- 3. 「共有ブラウズ」セクションで、「このサイトで Oracle 共有ブラウズを使用可能に します」を選択します。
- サイトにランチャ・スクリプトを入力します。Oracle Cobrowse Cloud Service 管 理者は、共有ブラウズ管理コンソールにアクセスし、適切なランチャ Javascript ス ニペットを取得できます。ランチャには、2 つの異なる種類があります。
  - 起動ポイント 1: 共有ブラウズ・ボタンがページに自動的に追加されます。
  - 起動ポイント 2:ページに追加されたボタンとインタフェースをカスタマイズ できます。
- 5. 「閉じる」をクリックします。

サイトで共有ブラウズを有効化したら、ページへの追加やページでの使用方法のカス タマイズが可能になります。詳細は、ページでの共有ブラウズの使用を参照してくだ さい。

### セキュア・サイトおよびサイト・ビルダー・テストでの共有ブラウズの使用

共有ブラウズには、インスタント(ICB)と拡張(ACB)の2つのモードがあります。保護 されたサイトで共有ブラウズをICBモードで使用したり、開発中のサイトをプレビュ ーしたりするには、追加の構成が必要です。この構成は、共有ブラウズ管理コンソー ルで行います。スタンドアロンの共有ブラウズの使用のエージェント・コンソールへ のログインに関する項を参照してください。

公開されたセキュア・サイトの、パスワードで保護されたリソースをエージェントが 表示できるようにするには、カスタム機能フィールドに機能を追加します。

```
function () {
return {
passwordProtectedPatterns: [
"<PROTOCOL>://<DOMAIN>/authsite/*?*",
"<PROTOCOL>://<DOMAIN>/documents/*?*",
"<PROTOCOL>://<DOMAIN>/content/*?*#*"
]
```



} } 公開された両方のセキュア・サイトでこの機能を使用する場合、および開発中のサイトを表示/プレビューする場合は、次のコードを追加します。

```
function () {
return {
passwordProtectedPatterns: [
"<PROTOCOL>://<DOMAIN>/authsite/*?*",
"<PROTOCOL>://<DOMAIN>/documents/*?*",
"<PROTOCOL>://<DOMAIN>/content/*?*#*"
"<PROTOCOL>://<DOMAIN>/_sites/*?*#*",
"<PROTOCOL>://<DOMAIN>/_themes/*?*",
"<PROTOCOL>://<DOMAIN>/_sitescloud/*?*",
"<PROTOCOL>://<DOMAIN>/_compdelivery/*?*"
]
}
```

保護されたリソースの構成は、共有ブラウズの新機能です。共有ブラウズ・ページ・マスキ ングと同じワイルドカードの URL パターンが使用されます。 詳細は、*共有ブラウズ・デプロ イメントおよび使用ガイド*のページ・マスキングの構成を参照してください。

### 分析追跡の追加

サイト・ビルダーで、Web 分析追跡用の JavaScript 追跡コードのスニペットをサイトまたは ページに追加し、Google、Adobe または Oracle Infinity などの外部のアナリティクス・プロ バイダとの統合を容易にできます。

サイトに分析追跡を追加するには:

- 1. 編集するサイトを開きます。
- 3. スイッチをクリックして分析追跡を有効にします。
- 「JavaScript トラッキング・スニペット」ボックスで、新しいスニペットを追加するか、 既存のスクリプトを編集します。
   管理者が環境用のコード・スニペットを提供している場合があります。その場合、それが ボックスに表示されます。スクリプトをカスタマイズするか、独自のスクリプトを追加で きます。管理者が提供したスニペットを編集すると、スクリプトが変更されたことを示す メッセージが表示されます。カスタマイズ内容を削除するには:最新のテナント・スクリ プトにリストアします。

Google Analytics 追跡スニペットの例を次に示します:

```
<!-- 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>
```

サイトの分析を収集するには、この変更を保存して公開し、必要に応じてサイトをオ ンラインにする必要があります。

### 分析データの表示

サイトを公開してオンラインにすると、追跡された分析データを Google Analytics な どベンダーのサイト上で表示できます。Oracle Infinity 分析追跡用のスニペットを使用 した場合、Oracle Infinity ホーム・ページに移動し「分析」をクリックしてデータを表 示し、レポートを選択または作成します。



# 12 サイトの公開

サイトのオンライン化、サイトのオフライン化、サイトの変更の公開について学習しましょ う。

- サイトのオンライン化またはオフライン化
- サイト変更の公開

# サイトのオンライン化またはオフライン化

サイトが*オンライン*である場合、適切なアクセス権を持つユーザーは、標準の Web ブラウザ を使用して、指定したアドレス(URL)でサイトを表示できます。サイトが*オフライン*である場 合、一般公開されません。サイトは Oracle Content Management でのみ表示できます。

サイトをオンラインにする前に、サイトが公開されている必要があります。サイトを公開す るには、「サイト変更の公開」を参照してください。

右側のステータス・アイコンは、サイトがオンラインかオフラインかを示します。

- サイトがこれまで一度も公開されていない場合は、ダッシュ(-)が表示されます。
- サイトがオンラインの場合は、
  のが表示されます。
- サイトがオフラインの場合は、 が表示されます。

オンラインまたはオフラインのサイトに誰がアクセスできるかについては、「サイト・セキュ リティの理解」を参照してください。

サイトのステータスを変更するには、サイトの所有者であるか、マネージャ・ロールを持っ ていることが必要です。サイト・ガバナンスが有効になっている場合は、サイトがサイト管 理者と共有されているかどうかにかかわらず、サイト管理者が任意のサイトのステータスを 変更できます。

- 1. 「サイト」ページでリストからサイトを選択します。
- サイトをオンライン化またはオフライン化するには、右クリック・メニューの「オンラインにする」または「オフラインに設定」を選択します。サイト・タイルの をクリック

してサイトをオンラインにすることも、サイト・タイルの^Oをクリックしてサイトをオ フラインにすることもできます。

選択内容を確認するよう求められます。

サイトをオンラインにすると、完全にレンダリングされた HTML バージョンのサイトが作成 され、Oracle Cloud のホスティング場所にコピーされます。オンライン・サイトで、サイト 名の下にその URL が表示されます。デフォルトの URL の形式は次のとおりです。

https://service_name.identity_domain.sites.oraclecloud.com/site_name



サイトをオフラインにすると、サイトおよびそのフォルダとファイルは、Oracle Cloud のホスティング場所から削除されます。

### サイト変更の公開

サイトの変更を公開するには、サイトの所有者であるか、マネージャ・ロールを持っ ている必要があります。

サイト変更を公開する際は、次の点を考慮してください。

- オンラインのサイトに変更を公開すると、コミットされた変更は、サイトへのアク セス権を持つすべてのユーザーにすぐに表示されます。
- サイトに対する変更を公開する際は、サイトのチャネルをターゲットとするすべてのアセットを公開することも、サイトのページで使用されるアセット(レコメンデーションを含む)のみを公開することも可能です。たとえば、サイトのチャネルをターゲットとするすべてのアセットを公開する場合に、サイトに直接属していないコンテンツ・アイテムを参照するリストがサイトに含まれていると、参照されるアセットも公開されます。
- 多言語サイトに変更を公開すると、サイトに関連付けられたローカリゼーション・ ポリシーで定義されている言語の翻訳のみが公開されます。
- サイトに関連付けられたローカリゼーション・ポリシーですでに定義されていない 言語の翻訳が以前公開されている場合、翻訳済コンテンツは公開されたサイトから 削除されます。
- 多言語サイトからページを削除する場合、サイトが公開される際に翻訳済ページも 削除されます。
- 1. サイトの変更を公開するには、これを選択し、右クリック・メニューまたはアクション・バーの「**公開」**を選択します。
- サイトとサイトの公開チャネルをターゲットとするすべてのアセットを公開する か、サイトとサイトのページに追加されたすべてのアセットを公開するかを選択し ます。
- 3. Oracle Content Management では、すべてのサイト文字列およびアセットに必要な 承認および翻訳があることが検証されます。サイトに関連付けられたローカリゼ ーション・ポリシーで必要とされている翻訳がない場合、アセットが翻訳済とマー ク付けされていない場合、またはアセットに承認が必須であるにもかかわらず承認 済とマーク付けされていない場合は、サイトを公開することができません。アイテ ムを展開して詳細を表示します。問題がある場合はその問題を修正して、公開を再 試行してください。すべてのアイテムが有効である場合は「公開」をクリックしま す。選択内容を確認するよう求められます。
- **4.** サイトがまだオンライン(**○**)ではない場合、サイトをオンライン化して、ユーザ ーに対してサイトを使用可能にする必要があります。



# 13 セキュア・サイト

サイトを公開してオンラインで使用可能にする場合、誰がサイトにアクセスできるかを制御 する必要があります。

サイトを保護する場合、割り当てられているロールに基づいて、公開された(オンラインの)サ イトにアクセスできるユーザーのグループを指定します。これらのロールは、サービス・イ ンスタンスの管理者によって割り当てられているサービス全体のロールです。

- サイト・セキュリティの理解
- サイト・セキュリティの変更

管理者の場合は、カスタム・サインインの有効化や、サイトおよびテーマの共有許可、サイトやテンプレート、コンポーネントの作成制限、ガバナンスの有効化、その他のアクションなど、サイト・セキュリティに関連して実行できる追加のアクションがあります。『Oracle Content Management の管理』のサイトおよびアセット設定の構成に関する項を参照してください。

### サイト・セキュリティの理解

セキュリティを適用して、誰が公開された(オンラインの)サイトを表示できるか、誰がサイト でセキュアなコンテンツを表示したり、操作できるか、および誰が公開されていない(オフラ インの)サイトを表示したり、編集できるかを制御できます。

#### サイト・セキュリティ

サイトを公開してオンラインで使用可能にする場合、誰がサイトにアクセスできるかを制御 する必要があります。システム管理者およびサイト管理者が環境をどのように構成したかに 応じて、サイトをすべてのユーザーに公開するか、登録済ユーザーに制限するか、特定のユ ーザーに制限することができます。

サイト・セキュリティまたはその他の設定を変更するには、サイトの所有者であるか、マネ ージャ・ロールを持っていることが必要です。サインイン要件を変更するには、サイトをオ フラインにする必要があります。ただし、指定されたユーザーまたはユーザー・ロールを変 更する場合は、サイトをオンラインにしておくことができます。サイトをオフラインにする と、サイトおよびそのフォルダとファイルは、Oracle Cloud のホスティング場所から削除さ れます。

使用可能なセキュリティ・オプションは、サイト・ガバナンスが有効になっている場合はテ ンプレート・ポリシー、サイト・ガバナンスが無効になっている場合はテナント・ポリシー によって制限されることがあります。サイト・ガバナンスの理解を参照してください。

サイトを保護する場合、割り当てられているロールに基づいて、公開された(オンラインの)サ イトにアクセスできるユーザーのグループを指定します。これらのロールは、サービス・イ ンスタンスの管理者によって割り当てられているサービス全体のロールです。

クラウド・ユーザー: 認証されたユーザーは、ユーザー名とパスワードを使用してサービス・インスタンスにサインインします。これには、の「訪問者」ロールまたは「ユーザー」ロールの有無に関係なく、すべての認証済ユーザーが含まれます。



 訪問者: このロールを持つユーザーのみがサイトにアクセスできます。たとえば、 このロールは、公開されたサイトを参照できるが、Oracle Content Management の このインスタンスのフォルダとファイルにはアクセスできないユーザーに付与さ れます。

🖉 ノート: これには、「ユーザー」ロールを持つユーザーは含まれません(サイトの所 有者である場合、またはサイトがそうしたユーザーに対して明示的に共 有された場合を除く)。

- サービス・ユーザー: このロールを持つユーザーのみがサイトにアクセスできます。 たとえば、このロールは、公開されたサイトを参照でき、かつ、Oracle Content Management のこのインスタンスのフォルダとファイルにアクセスできるユーザ ーに付与されます。
- **特定のユーザー**: サイトのメンバーとして追加したユーザーのみが、公開されたサ イトを表示できます。

#### サイトの共有

「サイトの共有」を使用して、未公開(オフライン)のサイトにアクセス可能な個々のユ ーザーを指定し、付与している権限に基づいて、そのユーザーにサイトを表示、変更 または管理できるようにします。サイトの所有者である場合、またはサイトを共有し ており、マネージャ・ロールが付与されている場合、サイトを共有できます。

/ ノート:

ユーザーに共有ロールを割り当てると、セキュリティ・ロールが付加されま す。たとえば、ユーザーが「訪問者」ロールを持っていても、サイトを共有 してこのユーザーにコントリビュータ・ロールを付与すると、「訪問者」ロー ルを持つ他のユーザーはオンライン・サイトの参照のみが可能であるのに対 し、このユーザーはオフライン・サイトを変更することができます。

- 参照者: 参照者はエディタでサイトを表示できますが、変更することはできません。
- ダウンロード実行者: サイトでは、これは参照者ロールと同じ権限を持ちます。
- コントリビュータ:参照者と同じ権限に加え、オフラインの場合、サイトの編集、 サイト・ページの削除およびサイトの削除を行うこともできます。
- マネージャ:コントリビュータと同じ権限に加え、ユーザーの追加、ロールの割当 て、オンライン・サイトへの変更の公開、およびオンラインとオフラインのサイト の切替えを行うこともできます。サイトの作成者(所有者)は自動的にマネージャ・ ロールを割り当てられます。

サイトを作成すると、そのサイト名のチャネルが作成されます。他のユーザーとチャ ネルを共有するには、サイトを共有して、最小限のコントリビュータ・ロールを付与 し、そのユーザーがアセットの公開にチャネルを使用できるようにする必要がありま す。サイトを公開するには、ユーザーにマネージャ・ロールが必要です。



#### コンポーネントの共有

ー部のコンポーネントでは、フォルダ、ファイルまたは会話などの共有リソースへのアクセ スが提供されます。*コンポーネントの共有*では、サイトのセキュリティ(誰が公開されたサイ トを表示できるか)とリソースの共有(誰がフォルダ、ファイルおよび会話を表示および操作で きるか)の両方が考慮されます。

たとえば、ドキュメント・マネージャ・コンポーネントをサイトに追加した場合、サイトの すべてのビジターがフォルダのコンテンツを表示でき、ロールおよび他の権限に基づいてフ ォルダ内にあるものを追加、変更または削除できます。

一般的な注意点:

- サイト作成者は、フォルダに対して自分が持っているアクセス権より上位のアクセス権を 付与することはできません。たとえば、作成者がフォルダのダウンロード実行者アクセス 権を持っている場合、サイトのビジターにコントリビュータ権限を付与することはできま せん。
- コンポーネントに設定された権限によって、ビジターの権限を引き上げることができます。たとえば、ビジターがフォルダに対する参照者権限を持っている(または権限を持っていない)場合、ドキュメント・マネージャ・コンポーネントがコンポーネントで選択されたロールに基づいて上位の権限を付与できます。これらの引き上げられた権限は、そのコンポーネント内でのみ有効です。
- サイトのビジターがコンポーネントに指定された権限より上位の権限を持っている場合、
   その権限でコンポーネントに設定された権限がオーバーライドされます。
- フォルダに付与された権限は、そのフォルダ内にネストしているフォルダおよびファイル に適用されます。

パブリック・サイトの場合:

- 会話コンポーネントは、セキュアなサイトでのみサポートされます
- ドキュメント・マネージャ・コンポーネントでは、関連付けられているフォルダに対する ダウンロード実行者権限がデフォルトですべてのビジターに付与されます。前述のガイ ドラインの規定内でロールを変更したり、ユーザーに提示されるオプションをコンポーネ ント自体の設定で制限できます。
- フォルダ・リストおよびファイル・リスト・コンポーネントでは、すべてのユーザーにダウンロード実行者アクセス権が付与されます。ユーザーは、自分のロールに関係なくファイルを表示およびダウンロードできます。

#### セキュアなサイト URL

サイトをオンラインにすると、完全にレンダリングされた HTML バージョンのサイトが作成 され、Oracle Cloud のホスティング場所にコピーされます。オンライン・サイトで、サイト 名の下にその URL が表示されます。

セキュアではないサイトのデフォルトの URL の形式は次のとおりです。

https://service_name.identity_domain.sites.oraclecloud.com/site_name

セキュアなサイトのデフォルトの URL の形式は次のとおりです。

https://service_name.identity_domain.sites.oraclecloud.com/authsite/site_name



**URL** に authsite が追加されていることに注意してください。

### サイト・セキュリティの変更

サイトを公開してオンラインで使用可能にする場合、誰がサイトにアクセスできるか を制御する必要があります。システム管理者およびサイト管理者が環境をどのように 構成したかに応じて、サイトをすべてのユーザーに公開するか、登録済ユーザーに制 限するか、特定のユーザーに制限することができます。

サイト・セキュリティまたはその他の設定を変更するには、サイトの所有者であるか、 マネージャ・ロールを持っていることが必要です。サインイン要件を変更するには、 サイトをオフラインにする必要があります。ただし、指定されたユーザーまたはユー ザー・ロールを変更する場合は、サイトをオンラインにしておくことができます。サ イトをオフラインにすると、サイトおよびそのフォルダとファイルは、Oracle Cloud のホスティング場所から削除されます。サイトをオフラインにするには、「サイト」ペ ージからサイトを選択し、右クリック・メニューの「**オフラインに設定」**を選択する

か、アクション・バーの**∪**をクリックします。選択内容を確認するよう求められま す。

サイト・セキュリティを変更するには:

- 「サイト」ページで、サイトを選択して右クリック・メニューの「プロパティ」を 選択するか、アクション・バーでクリックします。
- 2. 「サイト・セキュリティ」タブをクリックします。

「サイト・セキュリティ」タブで使用可能なオプションは、システム管理者がサイト・セキュリティ設定をどのように構成しているかと、サイト・ガバナンスが有効になっている場合は、このサイトが基づいているテンプレートのセキュリティ設定をサイト管理者がどのように構成しているかによって異なります。サイトのスタート・ガイドおよびサイト・ガバナンスの理解を参照してください。

- オンライン時にサイトを表示するために登録済ユーザーがサインインする必要が あるようにするには、「ログインが必要です」の横にある「はい」をクリックしま す。要件を削除し、オンライン時にサイトを公的に利用可能にするには、「ログイ ンが必要です」の横にある「いいえ」をクリックします。
- オンラインのサイトにアクセス可能な登録済ユーザーのグループを選択します。 個々のグループを選択するには、まず「クラウド・ユーザー」の選択を解除しま す。
  - クラウド・ユーザー:認証済ユーザーのみがサイトにアクセスできます。認証 済ユーザーは、ユーザー名とパスワードを使用してドメインにサインインしま す。これには、「Oracle Content Management Cloud 訪問者」ロールまたは 「Oracle Content Management Cloud ユーザー」ロールを持つユーザーが含 まれます。
  - 訪問者: このロールを持つユーザーのみがサイトにアクセスできます。これには、「Oracle Content Management Cloud ユーザー」ロールを持つユーザーは含まれません。
  - サービス・ユーザー: Oracle Content Management のこのインスタンスへのサインインが可能なユーザーのみが、サイトにアクセスできます。
  - 特定のユーザー: サイトにアクセスできる個々のユーザーを指定します。「メンバーの追加」をクリックします。検索フィールドにユーザー名またはユーザー名の一部を入力します。表示されたリストからユーザーを選択し、操作を繰り返して他のユーザーを追加します。終了したら、「追加」をクリックします。

ユーザーを削除するには、ユーザー名の下のメニューから**「削除」**をクリックしま す。

**5. 「保存」**をクリックして変更内容を保存し、ウィンドウを閉じます。

サイトに、そのサイトがオフラインであり、ログインが必要であることが示されます。

サイトをオンラインにするには、右クリック・メニューの「オンラインにする」を選択するか、アクション・バーののをクリックします。「確認して続行」をクリックし、「OK」をクリックします。

サイトをオンラインにすると、完全にレンダリングされた HTML バージョンのサイトが作成 され、Oracle Cloud のホスティング場所にコピーされます。オンライン・サイトで、サイト 名の下にその URL が表示されます。

セキュアではないサイトのデフォルトの URL の形式は次のとおりです。

https://service_name-identity_domain.cec.ocp.oraclecloud.com/site/site_name

セキュアなサイトのデフォルトの URL の形式は次のとおりです。

https://service_name-identity_domain.cec.ocp.oraclecloud.com/site/authsite/
site_name

URL に authsite が追加されていることに注意してください。

ログアウト用の URL を追加し、これをメニューに表示されるリンク、ボタンまたはページとして実装できます。段落、ボタンおよびページの追加を参照してください。

ログアウト用の URL の形式は次のとおりです。

https://service_name-identity_domain.cec.ocp.oraclecloud.com/cloudgate/ logout.html?postlogouturl=%2Fsite%2Fauthsite%2Fsite_name

### / ノート:

postlogouturl は、上に示すように、エンコード形式にする必要があります。



# 14 多言語サイトの操作

対象が絞られた多言語体験に関するサイト翻訳について学習するには、次を参照してください。

- 多言語サイトの概要
- サイトの翻訳
- サイト翻訳ジョブの管理

## 多言語サイトの概要

サイトは、ローカリゼーション・ポリシーで指定されている任意の言語に翻訳できます。サ イトに関連付けられたローカリゼーション・ポリシーで指定されている言語を確認するには、 サイトのタイトルで言語の横にある下矢印をクリックします。ローカライズ・バージョンを プレビューするには、言語を選択してサイトを開きます。

en Default 🔻
en Default
de
fr

🎤 ノート:

Oracle Content Management Starter Edition では、多言語サイトを作成することはできません。完全な機能セットおよび無制限のサイトが必要な場合は、Oracle Content Management Premium Edition にアップグレードします。

翻訳するサイトを選択すると、サイト・ファイルの.zip ファイルが作成されます。

✓ ノート: サイトに複数のリポジトリのアセットが含まれる場合、デフォルト・リポジトリの これらのアセットのみが.zip に含まれます。

.zip には、次のファイルとフォルダが含まれます。



- assets フォルダーこのフォルダは、サイト全体またはターゲット対象アセットの みを翻訳する場合にのみ配置されます。
  - root フォルダ
    - * 各コンテンツ・アイテムの<*contentItem_ID*>.jsonファイル―コンテ ンツ・アイテムのフィールドに基づく名前、説明および翻訳可能文字列に 加え、編集できないコンテンツ・アイテムに関する追加情報が含まれます。
  - job.json—翻訳ジョブが記述されているファイル。このファイルは編集しな いでください。
- site フォルダ—このフォルダは、サイト全体またはサイト・コンテンツのみを翻訳する場合にのみ配置されます。
  - root フォルダ
    - * 各サイト・ページのzpage_ID>.json ファイル―ページのコンポーネン トに基づく名前、タイトル、説明、キーワード、ヘッダー、フッターおよ び翻訳可能文字列に加え、編集できないページに関する追加情報が含まれ ます。カスタム・コンポーネントがある場合、翻訳可能な文字列を使用す るように構成されている可能性があります。多言語サイト用の翻訳可能コ ンポーネントの開発を参照してください。
    - * siteinfo.json—説明、キーワード、ヘッダーおよびフッターが含まれ ます。
    - * structure.json—ナビゲーションおよびサイト構造が含まれます。
  - job.json—翻訳ジョブが記述されているファイル。このファイルは編集しな いでください。

### サイトの翻訳

翻訳済サイトのページに、直接配置されるかコンテンツ・リストに含まれる形でアセットが存在し、それらのアセットが翻訳されている場合、アセットはサイトと同じ言 語で表示されます。サイトとは別にコンテンツ・アイテムを翻訳することもできます。 コンテンツ・アイテムのローカライズを参照してください。

複数のリポジトリのアセットを含むサイトを翻訳する場合、デフォルト・リポジトリ のアセットのみが翻訳ジョブに含まれます

### ♪ ノート:

Oracle Content Management Starter Edition では、翻訳を使用することはで きません。完全な機能セットおよび無制限のサイトが必要な場合は、Oracle Content Management Premium Edition にアップグレードします。

### 翻訳ジョブの作成

手動翻訳用として翻訳パッケージをエクスポートするか、翻訳コネクタを使用する場 合、翻訳ジョブを作成する必要があります。

1. 翻訳するサイトを選択し、「翻訳」をクリックします。「翻訳」オプションを表示す るには、「その他」をクリックする必要があります。



- 2. エクスポートするサイト・ファイル、つまり翻訳ジョブの情報を入力し、次に「**作成」**を クリックします。
  - 翻訳ジョブの名前を入力します。
  - サイトの翻訳後のターゲット言語を選択します。
  - サイト全体、つまりサイト・コンテンツ(ページ、構造およびサイト情報)とターゲット対象アセットをエクスポートするか、サイト・コンテンツのみをエクスポートするか、またはターゲット対象アセットのみをエクスポートするかを選択します。
  - 翻訳コネクタを選択するか、手動翻訳用の翻訳パッケージをエクスポートすることを 選択します。
  - 完了したら、「作成」をクリックします。

翻訳ジョブが完了したら、翻訳されたパッケージが Oracle Content Management にインポートされます。

#### ソース言語ファイルの手動翻訳

コンテンツを手動で翻訳する場合、翻訳ジョブが終了したら、サイト・ファイルの.zipファイルをダウンロードする必要があります。

- 1. バナーの「翻訳ジョブ」をクリックして翻訳ジョブを選択し、「ダウンロード」をクリックします。
- ターゲット言語として選択した各言語について、.zip ファイル内の root フォルダと同じ レベルに de、es、fr などのフォルダを作成します。選択した言語の一部を翻訳できま す。たとえば、今ドイツ語(de)に翻訳し、後でスペイン語(es)およびフランス語(fr)に翻 訳できます。
- 3. *すべての*.json ファイルを root フォルダから各言語フォルダにコピーします。
- 4. すべての . json ファイルの文字列を、適切な言語に翻訳します。.json ファイルから文字 列を削除したり、ファイルの名前を変更したりしないでください。
- 5. assets (翻訳ジョブにアセットが含まれている場合)フォルダと site (翻訳ジョブにサ イト・コンテンツが含まれている場合)フォルダを、job.json、root、および翻訳済フ ァイルに伴うすべての言語フォルダとともに圧縮します。

#### 手動翻訳されたパッケージのインポート

翻訳が手動で行われるか自動的に行われるかとは関係なく、終了した翻訳パッケージをイン ポートする必要があります。

- 1. 「翻訳ジョブ」ページで、「インポート」をクリックします。
- 「アップロード」をクリックし、翻訳済サイト・ファイルの.zip ファイルを選択してから、 「開く」をクリックします。
- 3. アップロードの終了後、「OK」をクリックします
- Oracle Content Management は、ジョブで定義されている翻訳が.zip ファイルで使用可能 であることを検証します。どのページとアセットが翻訳ジョブに含まれているかを確認 するには、ダイアログのリンクをクリックします。
- 翻訳をインポートする準備ができたら、「インポート」をクリックします。 インポートのステータスがバナーの上に表示されます。「詳細」をクリックすると、ジョ ブの詳細を表示できます。



### 翻訳コネクタからの翻訳済パッケージのインポート

翻訳が手動で行われるか自動的に行われるかとは関係なく、終了した翻訳パッケージ をインポートする必要があります。

- 1. •••をクリックし、「翻訳ジョブ」を選択します。
- 2. 終了した翻訳ジョブを右クリックし、「インポート」を選択します。

TranslateSiteByLingotek Trains	
Updated just now by you   LINGOTEK Translation Ready (Just now)	Open
	Refresh
	Import
	Delete

- Oracle Content Management は、ジョブで定義されている翻訳を検証します。どの ページとアセットが翻訳ジョブに含まれているかを確認するには、ダイアログのリ ンクをクリックします。
- 翻訳をインポートする準備ができたら、「インポート」をクリックします。 インポートのステータスがバナーの上に表示されます。「詳細」をクリックすると、 ジョブの詳細を表示できます。

翻訳をインポートした後、その翻訳がサイト上に適用されるようにサイトを公開して オンラインにする必要があります。

翻訳パッケージをインポートした後に翻訳サービスで翻訳が変更された場合、「リフレ ッシュ」をクリックして翻訳を更新できます。

翻訳後にサイトを変更した場合、新しい文字列や編集された文字列があれば翻訳する 必要があります。サイト・ビルダーでサイトを編集する際は、サイトのデフォルト言 語バージョンを編集します。コンポーネントの追加やページの再編成など、サイト構 成に加えた変更はサイトのローカライズ・バージョンにすべてレプリケートされます。 その後、新しい翻訳ジョブを作成して、更新された文字列を翻訳できます。

### サイト翻訳ジョブの管理

翻訳するサイトを選択すると、*翻訳ジョブ*が作成されます。その後、翻訳対象のファ イルをダウンロードし、翻訳してから、翻訳ファイルをインポートすることができま す。

翻訳ジョブを作成するには、「サイトの翻訳」を参照してください。

「翻訳ジョブ」ページには、すべての翻訳ジョブとそのステータスがリストされます。

- **小** (準備完了) サイト・ファイルの.zip ファイルは、ダウンロードする準備がで きています。
- **(**進行中) .zip ファイルはダウンロードされました。すべてのターゲット言語のすべての翻訳が正常にインポートされるまで、ステータスは進行中のままです。



- A (完了) このジョブのすべてのターゲット言語の翻訳は、正常にインポートされました。
- 【 (失敗) 翻訳ジョブが失敗しました。ジョブが失敗した理由に関する失敗メッセージ がバナーの上に表示されたはずです。このメッセージを再度表示する必要がある場合は、 翻訳ジョブを**再送信**できます。問題を修正した後、ジョブを**再送信**してください。

実行できるアクションは、次のとおりです:

- 翻訳ジョブの詳細を表示するには、そのジョブを開きます。詳細には、ソース言語と選択 されたすべてのターゲット言語、およびそれらの翻訳のステータスが含まれます。
- サイト・ファイルの.zip ファイルをダウンロードするには、ジョブを選択して「ダウンロード」をクリックします。
- ジョブを削除するには、ジョブを選択して「削除」をクリックします。
- 翻訳をインポートするには、「インポート」、「アップロード」の順にクリックし、翻訳済 サイト・ファイルの.zip ファイルを選択してから、「OK」をクリックします。
   Oracle Content Management は、ジョブで定義されているすべての翻訳が.zip ファイルで 使用可能であることを検証します。どのサイト・ページとアセットが翻訳ジョブに含まれ ているかを確認するには、ダイアログのリンクをクリックします。翻訳をインポートする 準備ができたら、「インポート」をクリックします。

### 翻訳のロケール

翻訳用のアイテムを送信すると、ターゲット言語がコードで識別され、翻訳サービス・プロ バイダは、そのアイテムを何語に翻訳して返すかを判断します。たとえば、fr はフランス語 を、de はドイツ語を表します。

これらのコードは、その他にも地域の方言に拡張できます。たとえば、de-LI はリヒテンシュ タインで話されるドイツ語のコードで、de-LU はルクセンブルグで話されるドイツ語のコー ドです。ただし、言語サービス・プロバイダが地域の方言に対応していない場合は、コード を指定しても、2 文字のベース言語に切り捨てられます。たとえば、de-LI と de-LU の場合、 コードは de に切り捨てられます。

言語サービス・プロバイダがサポートする地域の方言がすべてではなく、1 つの場合は、代用 が行われます。たとえば、ms-BN はブルネイで話されるマレー語ですが、言語サービス・プ ロバイダがこの方言をサポートしていない場合には、マレーシアで話されるマレー語のコー ドである ms-MY などに切り替えられます。ベリーズの英語である en-BZ とジャマイカの英 語である en-JM など、言語サービス・プロバイダが方言を区別していない場合は、ベース言 語に切り捨てられ、このケースでは英語の en になります。

### 翻訳のカスタム・ロケール

カスタム・ロケールは、組織のニーズに基づいて、開発者が作成します。カスタム・ロケー ル・コードは、ベース言語と、任意の地域の方言コード(該当する場合)、カスタム・ロケール であることを示す x、カスタマイズを識別するためにお客様の組織が必要とするその他のもの で構成します。たとえば、英語のカスタム・ロケールは、en-JM-x-customのようにします。

カスタム・ロケールはお客様の組織に固有であるため、翻訳用にカスタム・ロケール・コードを送信すると、ベース言語と地域の方言(言語サービス・プロバイダがサポートしている場合)に切り捨てられます。前述した en-JM-x-custom の例では、カスタマイズに固有のコード部分がなくなり、en-JM に切り捨てられます。また、言語サービス・プロバイダがジャマイ



カ(JM)の地域の方言コードをサポートしていない場合には、ベース言語である en のみ に切り捨てられます。

### URL リダイレクトへのロケール・エイリアスの設定

実行時およびランタイム・プレビューでサイト URL に使用されるロケールのエイリア スを簡単に設定できます。これは、URL を長くして複雑さを追加できるカスタム・ロ ケールを組織で利用している場合に特に便利です。

たとえば、en-GB-x-cornish として定義されているカスタム・ロケールがあり、URL 内では次のようになっているとします。

https://example.com/site/BlogSite/en-GB-x-cornish/home.html

別名を追加して、より単純な URL にリダイレクトできます。

https://example.com/site/mysite/en/home.html

- 1. 編集するサイトを開きます。
- 2. サイドバーの ^②をクリックし、「ロケール」をクリックします。

サイトで使用されているすべてのロケールのリストが、対応する URL エイリア ス・フィールドの横に表示されます。

0	Content and E	xperience			
=	Product-Marketin	ng Update 🔻	en (Default) 🔻 🖒 🔇	No Test Profile 🔻 Fit to Window	•
Ð	Settir	ngs Ø	Site Locales		
	~~7	050	Provide aliases for each lo the alias is used in the lang will render in the correspo	cale specified in the Site's localization policy. Whe guage segment of the URL to a site page, the page nding locale.	n B
$\Theta$	SEO	Site	Site Locale	URL Alias	
ţ,		B	de-DE	de	
20			en-CA	са	
Ą	Redirects	Analytics	en-GB	uk	
070			en-GB-x-cornish	en	
<b>с</b> э			en-US	en-US	
	Properties	Locales	es-ES	es-ES	
			fr-FR	fr-FR	
			it-IT	it-IT	
ē			zh-CN	zh-CN	

3. エイリアスが必要な各サイト・ロケールの横にエイリアスを入力し、「閉じる」を クリックしてから「保存」をクリックします。



- 4. 実行時に URL で使用されるエイリアスをプレビューするには、 2をクリックします
- 5. 更新を公開すると、変更も公開されて使用されます。



# 15

# サイト・リダイレクトまたは URL マッピング の使用

Web サイトを再構築または移動する場合、ユーザー・リクエストを古い URL から現在の URL にリダイレクトできます。URL の 30x リダイレクトを指定すると、サイトを再設計した後も ブックマークまたは公開リンクを維持できます。

他のインフラストラクチャ・テクノロジから Oracle Content Management でホストされてい るサイトに移動する場合、検索エンジンで評価ランキングが高いページを別の URL に移動で きます。リダイレクトは、サイトの URL 構造を再編成し、検索エンジンのランキングを保持 する上で役に立ちます。

- リダイレクトのプラン
- サイト・リダイレクトの追加
- JSON ファイル内のリダイレクト・ルールの指定
- サイトへのリダイレクト・ルール・ファイルのアップロード
- サイト URL のマッピング

# リダイレクトのプラン

指定した URL の HTTP 30x レスポンスを送信するリダイレクトを指定できます。指定した リダイレクトの1つとリクエストが一致しない場合、URL の正規の処理が行われ、ページは 通常の方法で返されます。

リダイレクトを指定して JSON ファイルを作成し、そのファイルをサーバーにアップロード できます。サーバーは、受信リクエストの URL を処理するときに JSON ファイルを使用しま す。

2種類のリダイレクト・ルールを使用して、受信 URL を新しい場所にリダイレクトできます:

- 単純な文字列間の一致
- 簡易ワイルドカード一致

### 単純な文字列間の一致

単純な文字列の一致および置換の場合、明示的な URL を指定してから、各 URL をターゲット URL に直接マッピングすることによってリダイレクトできます。

次の表は、サンプルの文字列間の一致を示しています。

ソース URL	ターゲット場所の URL
/index.html	/home.htm
/products/widget	/items/knickknack



ソース URL	ターゲット場所の URL
/index?page=widgets	/items/widgets

文字列間のマッピングは、簡単に理解およびテストできます。ルールは、単純な文字 列一致およびマップ参照を使用して迅速に評価します。

ただし、URL の問合せパラメータに関してはほとんど柔軟性がありません。これらは 正確に一致する必要があります。余計な URL があったり、パラメータの順序が異なっ たりすると、ルールが一致しなくなります。

### 簡易ワイルドカード一致

簡易ワイルドカード一致により、ルールを多くの URL に一致させながら、結果を取得 するために必要な正規表現のバックトラッキングの量を制限することもできます。

正規表現は作成が複雑になる場合があるため、また、作成が不十分な正規表現では評価に要する時間が不明であるため(ReDoS)、第2タイプのルールには簡易一致メカニズムを使用できます。この場合、ワイルドカード文字("*")を受信 URL 内のゼロ(0)個以上の文字と一致させ、キーワード'wildcard'を索引値とともに使用して URL の受信部分をリダイレクト済 URL にコピーします。

次の表は、サンプルの簡易ワイルドカード一致を示しています。

ソース URL	ターゲット場所の URL
/old/*	/new/<\$wildcard(1)\$>
/dispatch.asp?page=*&facet=Lang*	<pre>/page&lt;\$wildcard(1)\$&gt;/&lt;\$wildcard(2)\$&gt;</pre>

簡易ワイルドカードー致は、単純な文字列間の一致より強力な機能を URL の一致に提 供しますが、バインド解除された正規表現処理なしにその処理を実行します。構文は 単純で、URL 上のパターン一致を使用して、1 つのパターンのみで多数の URL に対応 します。

これらは正規表現に基づいているため、ワイルドカード・ルールの方が単純な文字列 一致より評価が若干遅くなります。ルールの数が多くなると、一般的なページ配信の パフォーマンスが低下する可能性があります。

### サイト・リダイレクトの追加

サイト URL が変更された場合、リダイレクトによって、ある URL (ソース)が別の URL (ターゲット)に転送されます。これにより、ユーザーのブックマークと検索エンジンの ランキングが維持されるため、便利です。

次の2つのタイプのリダイレクトを使用できます。

- 永続リダイレクト: 301 HTTP サービス・レスポンス・コードを使用します
- 一時リダイレクト: 302 HTTP サービス・レスポンス・コードを使用します

redirect.jsonファイルをアップロードするには:

1. 編集するサイトを開きます。



- 2. サイドバーの[◎]をクリックし、 **「リダイレクト」**をクリックします。
- 3. 「アップロードするファイルの選択」をクリックし、使用するファイルに移動してそれを 選択し、「OK」をクリックします。
- 4. 更新を公開すると、変更も公開されて使用されます。

## JSON ファイル内のリダイレクト・ルールの指定

JSON ファイル内の URL のリダイレクト・ルールを指定できます。

URL のリダイレクト・ルールを指定するには、JSON ファイルで次のフォーマットを使用します。

```
{
     "redirectRules":
     [
        ł
            "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
        }
    ]
}
```

JSON ファイルの外部格納構造は配列です。配列にはルール・インスタンスが含まれています。

"string"ルールが最初に評価され、続いて"wildcard"ルールが順番に評価されます。ルールの1つが一致すると、後続のルールの評価は放棄され、対応するリダイレクトが生成されます。

各ルールには次のプロパティがあります:

- The "comment" プロパティは、ルールの評価に影響を与えないオプションのストリングです。注釈や解説が含まれています。
- "expression"プロパティは、受信サイトの相対 URL と一致する必須文字列です。ワイル ドカード・ルールでは、アスタリスク(*)トークンは0個以上の文字と一致します。
- "location"プロパティは、リダイレクトの場所または宛先を示す必須文字列です。リダイレクトには、完全 URL または相対 URL を使用できます。
- "code"プロパティは、リダイレクトを発行するときに使用する HTTP レスポンス・コードを提供するオプションの整数です。値は次の整数のいずれかである必要があります:
  - 301: リソースが永久に移動したことを示します。これは、"code"プロパティが省略 された場合のデフォルト値です。



- 302: リソースが一時的に移動したことを示します。
- "type"プロパティは、リダイレクト・ルールのタイプを示すオプションの文字列です。値は次のいずれかの文字列である必要があります:
  - "string"は、式が入力 URL 全体と正確に一致する、より高速なルールを指定 します。
  - "wildcard"は、複数の URL に一致するワイルドカード・ルールを指定します。 プロパティが省略されている場合、これがデフォルト値です。

#### 場所トークン

場所トークンを使用して、リダイレクト場所の作成を支援できます。次の各場所トー クンは、リダイレクトの指定に役立ちます:

- <\$urlPath\$>: 一致する URL のパス部分。
- <\$urlQueryString\$>: 一致する URL の URL 問合せ文字列全体。
- <\$urlQueryStringExcept(name1,name2)\$>: 一致する URL からの URL 問合せ文 字列全体から、指定されたパラメータを引いたものです。
- <\$wildcard(N)\$>: 一致する URL の一致するワイルドカードの1から始まる索引。 (これは正規表現の\1..\9 に似ています。)
- <\$name\$>:名前付き問合せ文字列パラメータの値。たとえば、入力に問合せ文字列 msmith: ?page=42 がある場合、<\$page\$>を使用してその場所に'42'を配置できま す。

#### 制限

redirects.json ファイル全体およびそれに含まれるルールには、次の制限が適用されます:

- Oracle Content Management で受け入れられるファイル全体の最大サイズは 250KB です。
- redirects.json ファイル内のルールの最大数は 1,000 です。
- ルールの"expression"の最大長は 1,000 文字です。
- ルールの"location"の最大長は 2,000 文字です。
- ワイルドカード・ルール式の'*'トークンの最大数は10です。

#### 文字列一致の例

```
ルール:
```

```
{
    "type": "string",
    "expression": "/old/page.jsp?id=material&type=glass",
    "location": "/new/<$id$>.htm"
}
```

次の URL はルールに一致します:

/old/page.jsp?id=material&type=glass



- 結果の場所:/new/material.htm
- 問合せ文字列を含む URL 全体が一致します。
- <\$id\$>がその場所で使用されていますが、使用可能な問合せ文字列が1つしか一致できないため、この例では必要ありません。場所は/new/material.htmと書かれていた可能性があります。

次の URL はルールに一致しません:

/old/page.jsp

(ルールの式は、一致する必要がある問合せ文字列を与えます。)

- /old/page.jsp?id=material&type=glass&index=2
   (候補 URL 内の余分な&index=2 がルール式と正確に一致しません。)
- /old/page.jsp?type=glass&id=material

```
(問合せ文字列パラメータの順序は、「文字列」ルールで一致する必要があります。)
```

#### ワイルドカード一致の例:

ルール:

```
{
    "type": "wildcard",
    "expression": "/old/*/pages/*?id=*&item=sheet-*",
    "location": "/new/<$id$>/<$wildcard(4)$>.html"
}
```

次の URL はルールに一致します:

- /old/phones/android/pages/info.asp?id=XT1045&item=sheet-specs
  - 結果の場所:/new/XT1045/specs.html
  - URLのパス部分が一致するので、問合せ文字列も一致条件について調査されます。
  - この例のパラメータは、ルール式のパラメータの順序と一致しますが、これは必須で はありません。
- /old/phones/android/pages/info.asp?item=sheet-specs&id=XT1045
  - 結果の場所:/new/XT1045/specs.html
  - URL のパス部分は疑問符(?)の前のルール式と一致するため、パラメータも一致する かチェックされます。
  - パラメータはルール式内で異なる順序でリストされますが、パラメータは個別に一致します。
- /old/phones/android/pages/info.asp?id=XT1045&item=sheetspecs&unrelated=thing
  - 結果の場所:/new/XT1045/specs.html
  - URLのパス部分が一致するので、問合せ文字列も一致条件について調査されます。
  - 候補 URL には追加の&unrelated=thing パラメータがありますが、ルール式の名前付 き問合せパラメータが一致するため、ルールは一致すると見なされます。



unrelated パラメータは、<\$unrelated\$>のように場所内でトークンとして使用でき、ルールの一致には寄与しなかったとしても、thing という値を持ちます。

次の URL は一致しません:

/old/pages/info.jsp

(URL のパス部分がルール式のパス部分と一致しません。)

/old/phones/android/pages/info.asp

(URL のパス部分はルール式のパス部分と一致しますが、ルール式の問合せパラメ ータは一致しません。)

/old/phones/android/pages/info.asp?id=cellular

(URL のパス部分はルール式のパス部分と一致しますが、ルール式のすべての問合 せパラメータが一致するわけではありません。)

#### トークン配列の定義

redirects.json ファイル内のトークン定義の配列を作成して、複数のバニティ URL をサポートするリダイレクトを構成する際に役立てることもできます。これにより、 受信 URL の特性に基づいて適切にリダイレクトできます。

redirects.json ファイル内で次の形式を使用してリダイレクト・ルール URL で使用 するトークンを定義します。

```
{
     "tokenDefinitions":
     [
        {
            "token": "sitePrefix",
            "type": "hostmatch",
            "expresion": "example.com"
            "value": ""
        },
            "token": "sitePrefix",
            "type": "hostmatch",
            "expresion": "*.com"
            "value": "/site/Starter-Site"
        },
            "token": "gotoRedirect",
            "type": "pathmatch",
            "expresion": "*oracle*"
            "value": "https://www.oracle.com"
            "flags": "caseinsensitive"
        },
    ]
}
```

tokenDefinitions には次のプロパティがあります。

• "token": 定義するトークンの名前。



- "type": 次のいずれか。
  - 受信 URL のホスト値を照合する "hostmatch"。
  - 受信 URL のパス名値を照合する "pathmatch"。
  - 受信 URL の問合せ値を照合する "querymatch"。
- "expression": 照合に使用する必要のある式。ワイルドカード文字がサポートされます。
- "value": トークンに使用する必要のある値。
- "flags": デフォルトでは、flags 値が caseinsensitive に設定されていないかぎり、式の照合では大文字と小文字が区別されます

トークンの値を計算する場合、tokenDefinitions 配列が順番に列挙されます。最初に一致す る定義が使用されます。トークンを満たすトークン定義がない場合は、かわりに空の文字列 が使用されます。便宜とパフォーマンスのために、よく使用されるトークンは tokenDefinitions リストの先頭に配置する必要があります。

tokenDefinitions には次の制約があります。

- 作成できるトークン定義は 250 個までです。
- token 名は 100 文字未満にする必要があります。
- expression で使用できるワイルドカードは 10 個までです。
- expression は **1000** 文字未満にする必要があります。
- value は **1000** 文字未満にする必要があります。

### 例

たとえば、次の redirects.json ファイルがあるとします。

```
{
     "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"
        },
    ]
}
```

この場合、location プロパティには<\$pathPrefix\$>トークンがあります。pathPrefix トー クンは tokenDefinitions セクションで定義されます。受信 URL が"vanity.com"と一致する



場合、pathPrefix 値は/fashion に設定されます。これは location レスポンスで使用 され、/fashion/about/new-privacy-policy.html になります。

最初のバニティ・ドメイン URL が http://example.com/legacy-privacypolicy.html であると仮定しましょう。これは最初のリダイレクト・ルールとのみー 致します。

このルールに対して宣言された location は<\$pathPrefix\$>/about/new-privacypolicy.html です。この状況では、<\$pathPrefix\$>トークンを評価する必要がありま す。そのために、tokenDefinitions 配列が列挙されて一致が検索されます。

最初のトークン定義が検討されます。その token は目的のものであるため、さらに評価されます。式 vanity.com は受信 URL の example.com と一致しないため、この定義は要件を満たさず、列挙が続行されます。

この時点で、これ以上トークン定義はないため、<\$pathPrefix\$>トークンの値には空の文字列が使用されます。このリダイレクトに対して返される最終的な場所は/ about/new-privacy-policy.htmlです。

2番目のバニティ・ドメイン URL が http://vanity.com/legacy-privacypolicy.html であると仮定しましょう。最初の URL の場合と同様に、このルールに対 して宣言された location は<\$pathPrefix\$>/about/new-privacy-policy.html です。 この状況では、<\$pathPrefix\$>トークンを評価する必要があります。そのために、 tokenDefinitions 配列が列挙されて一致が検索されます。

最初のトークン定義が検討されます。前と同様に、その token は目的のものであるため、さらに評価されます。式 vanity.com は受信 URL の vanity.com と一致するため、この定義は要件を満たし、値/fashion がトークンの値として使用されます。

トークンとの一致が見つからなかったため、トークン定義配列の列挙が停止し、最終 的な場所は/fashion/about/new-privacy-policy.html として計算されます。

#### サイト・リダイレクトのテスト

「設定」パネルを開き、「リダイレクト」をクリックすることにより、サイトを編集す るときにサイト・リダイレクトをテストできます。テストする URL を入力し、「テス ト」をクリックします。





# サイトへのリダイレクト・ルール・ファイルのアッ プロード

**Oracle Content Management** 内のサイトにリダイレクト・ルールをアップロードできます。 サイトに redirect.json ファイルをアップロードするには:

- 1. 編集するサイトを開きます。
- サイドバーの[◎]をクリックし、³をクリックします。
- 3. 「アップロードするファイルの選択」をクリックし、使用するファイルに移動してそれを 選択し、「OK」をクリックします。
- 4. 更新を公開すると、変更も公開されて使用されます。

# サイト URL のマッピング

Oracle Content Management を使用してサイトを作成して公開したら、登録済ドメイン名 (www.mysite.com など)でこのサイトにアクセスできるように Domain Name System (DNS) を構成できます

Domain Name System (DNS)では、ドメイン名をサイトの場所にマッピングすることでユー ザーが Web ページを見つけられる場所、つまり別名(CNAME)を指定します。

ドメイン名をマップするには、次のものが必要です。

Oracle Content Management インスタンスの URL。一般に、次のような形式になります。



service-tenant.documents.datacenter.oraclecloud.com

- ドメイン名レジストラによって登録されたドメイン名。 たとえば、www.example.com などです。www.example.com/subdomain などのサブ ドメインでもかまいません。
- コンテンツ配信ネットワーク(CDN)プロバイダのアカウント。Oracle Content Management は Akamai との統合を提供します。使用しているインスタンス用に Akamai を構成するには、Oracle サポートに連絡してください。

**Oracle Content Management** で提供されている Akamai ではなく独自の CDN を使用 する場合は、次のステップを実行します。

様々な Domain Name System プロバイダには、CNAME レコードを更新する様々な Web インタフェースと様々なステップがあります。次のステップでは、必要な情報と 実行する一般的なステップを示します。

ドメイン名にサイト URL をマップするには:

- コンテンツ配信ネットワーク・プロバイダからドメインの Secure Sockets Layer (SSL)証明書を要求します。たとえば、https://www.example.com です。
- 2. コンテンツ配信ネットワークを構成し、次のことが可能になるようにします。
  - a. コンテンツ配信ネットワークがドメインへのすべての受信リクエストを受け 入れ、セキュアなプロトコル(https)を使用して転送します。
  - b. オリジンが Oracle Content Management からドメインを指し示します:

service-tenant.documents.datacenter.oraclecloud.com

3. DNS サーバー・ゾーン・ファイルを変更し、ドメイン名をコンテンツ配信ネット ワーク・プロバイダによって提供されたエッジ・サーバーにマップします。

#### domain CNAME CDN Server

 更新が伝播するのを待機します。DNS サービスによっては、2 時間から 48 時間か かることがあります。
 変更が伝播されたら、ドメイン名を使用してサイトにアクセスできます。例:

https://www.mysite.com/site_name

デフォルトでは、サイトに指定された標準の URL を使用している場合、Oracle Cloud REST API for Content Management のエンドポイントは使用可能です。たとえば、フ ォルダおよびファイル・リスト・コンポーネントは REST API を使用してフォルダお よびファイル操作を実行します。カスタム URL を使用する場合、ドメイン名でエンド ポイントにアクセスできることを確認します。例:

https://www.mysite.com/documents



# 16 サイト・パフォーマンスの向上

ブラウザのキャッシュを活用することにより、ブラウザでのコンテンツ配信およびレンダリングのパフォーマンスを向上させることができます。Above the Fold (ATF)レンダリングに よって Web サイトのレンダリングを向上させることもできます。

- キャッシュの利用によるパフォーマンスの向上
- Above the Fold (ATF)レンダリング

### キャッシュの利用によるパフォーマンスの向上

コンテンツ・アイテム、デジタル・アセットおよびサイトの配信では、ビジターのブラウザ・ キャッシュを完全に活用し、ブラウザでのコンテンツ配信とレンダリングのパフォーマンス を改善する必要があります。

サイト、テーマ、コンテンツ・アイテムおよびデジタル・アセットは、ビジターのブラウザ・ キャッシュ内に一定時間キャッシュされます。サイト、テーマ、コンテンツ・アイテムまた はデジタル・アセットが更新された後、URL 内のキャッシュバスター・キーが変更され、ブ ラウザが異なる URL をフェッチし、新しいアイテムを取得しなくてはならないようになりま す。

キャッシュ・キーは、現在のソースのみを参照することによってブラウザ・キャッシュの使 用を管理する上で役に立ちます。キャッシュ・キーは URL に含まれていますが、たいていの 場合、物理的な場所(フォルダ)ではなく、論理要素です。キャッシュ・キーの変更が、リソー スを検索するために異なる物理的な場所を指し示すことはありません。この変更は単に、リ ソースの現在のバージョンをフェッチするようサーバーに通知します。

リソースには、CSS ファイル、JS ファイルおよびイメージ・ファイルなどの静的なもの、またはページ・データ、サイト・データおよびコンテンツ・アイテム・データなどの動的なものがあります。Web サイトを構築するためのリソースには 5 つのカテゴリがあります:

- 製品リソース 製品の新規バージョンがリソースまたはパッチ適用されるたびに更新される製品の一部であるリソース。
- サイト・リソース structure.json・ページ・データおよびイメージなど、サイトの一 部であるリソース。これらは、サイトが公開されるときに更新されます。コントローラに ついては、後に続く本文を参照してください。
- テーマ・リソース レイアウト、CSS およびイメージなど、テーマの一部であるリソース。これらは、テーマが公開されるときに更新されます。
- コンポーネント・リソース カスタム・コンポーネントの一部であるリソース。これらには、コンポーネントを構成する HTML ファイル、JS ファイル、CSS ファイルおよびイメージ・ファイルが含まれます。これらは、コンポーネントが公開されるときに更新されます。1つのコンポーネントが変更されて再公開される場合、すべてのコンポーネントのキャッシュ・キーが変更されますが、これは、キャッシュ・キーがすべてのコンポーネントの単一キーであるからです。



 CaaS リソース – コンテンツ・アイテムおよびデジタル・アイテムを提供するリソ ース。これらは、コンテンツ・アイテムが公開または再公開される場合、またはコ レクション・ターゲットが変更されるときに更新されます。

次のトピックでは、Oracle Content Management のランタイムおよびサイト・ビルダ ーのキャッシュについて説明します:

- ランタイム・キャッシュ
- サイト・ビルダーのキャッシュ

### ランタイム・キャッシュ

ランタイムの場合、Oracle Content Management の Cache-Control ヘッダーは 15 日間 に設定されます。すべてのリソースについて URL にキャッシュ・キーが追加されま す。

URL が同じである限り、ブラウザでは、ローカル・キャッシュ(使用可能である場合) 内のリソースを提供します。リソースが更新されると、URL 内のキャッシュ・キーが 更新され、その結果、ブラウザは、サーバーに対する新しいリクエストを行い、ロー カル・キャッシュを更新するよう強制されます。

キャッシュ・キーが含まれるコントローラも1分間キャッシュされます。このため、 更新されたキャッシュ・キーが最大1分間表示されなくなります。

ランタイム時に、サーバーは、製品、サイト、テーマ、コンポーネントおよび CaaS リソース用の最新のキャッシュ・キーを使用して controller.html を返します。キー が含まれるスクリプトが controller.html に追加されます。次に例を示します:

```
<script type="text/javascript">
    var SCSCacheKeys = {
        product: '123',
        site: '456',
        theme: '789',
        component: '012',
        caas: '345'
```

};

</script>

これらのキーが controller.js によって使用され、次の表内のような URL が構築され ます。

リソース・タイプ	例
製品リソース	
	/sitePrefix/productCacheKey/_sitesclouddelivery/

/mySite/_cache_947d/_sitesclouddelivery/


リソース・タイプ	例
テーマ・リソース	/sitePrefix/themeCacheKey/_themesdelivery/themeName/
コンポーネント・リソース	/sitePrefix/compCacheKey/_compdelivery/compName/
サイト・リソース	/sitePrefix/siteCacheKey/content/ /sitePrefix/siteCacheKey/structure.json /sitePrefix/siteCacheKey/pages/100.json
CaaS リソース	RegularCaaSUrl?cacheKey=caasCacheKey

このようにキャッシュ・キーを URL に挿入することにより、Oracle Content Management は、URL を効率的に変更し、ブラウザが URL を実際に新しいリリースであると見なすように することにより、更新済リソースをロードするようブラウザに強制できます。

NJ-h:

セキュアなサイトの場合、製品、テーマおよびコンポーネント・リソースのみがキャッシュされ、サイトまたは CaaS コンテンツはキャッシュされません。

### サイト・ビルダーのキャッシュ

サイト・ビルダーでは、静的リソースは15日間キャッシュされます。

サイト・ビルダーを使用する場合、製品リソース、テーマ・リソースおよびコンポーネント・ リソースのキャッシュが行われます。(サイト・リソースおよび CaaS リソースでは行われま せん。)サイト・ビルダーの起動またはリフレッシュ時にテーマおよびコンポーネントのキャッ シュ・キーが再生成されます。

テーマまたはコンポーネントを変更し、その変更がサイト・ビルダーで表示されるようにするには、サイト・ビルダーをリフレッシュする必要があります(**F5**)。

# Above the Fold (ATF)レンダリング

ATF レンダリングは、実際に行うよりも速く見える web サイトのロードを可能にします。目的は、表示されるページのすべての部分を最初にレンダリングしてから、ユーザーがスクロール・ダウンする前に、最初には表示されていない残りのページをレンダリングすることです。

スロットには、タブ上にアイコンを表示する「折りたたみ」の指定があります。



スロットをこの新しい方法でレンダリングするには、次のように scs-atf でマークす る必要があります:

<div class="scs-slot scs-atf" id="headline"></div>

コンポーネントは、レンダリングが完了したらレンダラに通知する必要があります。 すぐに使用可能なコンポーネントは、デフォルトでこれを行います。カスタム・コン ポーネントは、追加のコールを実行でき、次を実行する必要があります:

1. レンダリングが完了するまでレンダラを待機するようレンダラに通知します。

2. 完了したらレンダラに通知します。

#1 の場合は、カスタム・コンポーネントの appinfo.json ファイルに対して次のプロ パティを追加します:

```
"initialData": {
    . . .
    "customRenderComplete": true,
    . . .
```

#2 の場合は、コンポーネントの render.js ファイルで、次のコールを行い、完了した ことをレンダラに必ず通知します:

SitesSDK.setProperty('renderComplete', true);

ATF スロットの一部のコンポーネントがタイムリに完了したことを報告していない場合、レンダラは残りのページを続行する前に2秒待機します。これが十分な長さではないことが分かっている場合は、ページ・テンプレートで次のグローバル変数を宣言することで時間を延長できます:

var SCSAtfPassTimeout = 3000;

/ ノート:

時間はミリ秒単位であるため、この例ではタイムアウトを3秒に設定しています。

API は、ATF プロセスの診断データを提供します。デバッグ・コンソールで次のメソッドを呼び出すことも、必要に応じてページからアクセスすることもできます:

SCSRenderAPI.getRenderMetrics();

例:

```
{currentTime: 16243.40000000001, renderStartTime: 264.36,
atfPassEndTime: 306.535, mainPassStartTime: 316.475, mainPassEndTime:
331.3850000000005, ...}
```



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-aa59-5c925bbbebee", time: 283.54}

3. 2:{atf: true, componentId: "c1c3aec8-e52f-406c-8c29-ab69c05877ed", time: 283.5600000000006}

4. 3:{atf: true, componentId: "b3a31dc6-62a1-44d9-9c80-bdb2c5bedaaa", time: 284.1300000000005}

5. 4:{atf: true, componentId: "c05aa1a2-c11c-4ef5-9051-4799c5bee24a", time: 284.1550000000003}

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: 299.7800000000003}

9. 8:{atf: true, componentId: "ccfedc98-1dbd-440e-b867-5e683cea2ec5", time: 301.1950000000005}

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: "delfa2ce-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: 320.4550000000004}

16. 15:{atf: false, componentId: "a48b5abb-49b2-4456-90bd-a3de998150c8", time: 320.48}



```
17. 16:{atf: false, componentId: "a9650e6d-7e7e-42a2-
b758-58f2aeab18a2", time: 322.615000000007}
18. 17:{atf: false, componentId: "aca9836a-f955-4aa7-8db2-
fd3cf1189dea", time: 324.2350000000007}
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: 330.090000000003}
23. 22:{atf: false, componentId: "fd603b96-2beb-4e87-
a54f-12d0e264cd0a", time: 331.385000000005}
24. length:23
25. __proto__:Array(0)
5.
    componentCount:23
     currentTime:16243.40000000001
6.
```

- 7. mainPassEndTime:331.3850000000005
- 8. mainPassStartTime:316.475
- 9. renderStartTime:264.36
- 10. __proto__:Object



# 第 Ⅳ 部 サイトの開発

このパートでは、テンプレート、テーマ、スタイル、コンポーネントおよびその他の便利な アイテムの理解と開発を支援することで、サイト開発を開始する方法を詳細に説明します。 次の章が含まれます。

- デザインおよびスタイルのカスタマイズ
- 背景の使用の理解
- トリガーおよびアクションの設定
- テンプレートの開発
- テーマの開発
- レイアウトの開発
- コンポーネントの開発



17

# デザインおよびスタイルのカスタマイズ

**Oracle Content Management** の各テーマには、テーマのデフォルト・サイトで使用されるコ ンポーネントの、ページのルック・アンド・フィール、およびスタイル設定を指定するデザ インが必要です。

- デザインについて
- デザイン・ファイル
- 会話リストのスタイルのカスタマイズ
- フォルダ・リストおよびファイル・リストのスタイルのカスタマイズ
- ソーシャル・バーのアイコンのカスタマイズ
- Oracle Intelligent Advisor のインタビュー・スタイル設定拡張機能の構成

### デザインについて

Oracle Content Management の各テーマには、テーマで使用されるコンポーネントの、ページのルック・アンド・フィール、およびスタイル設定を指定するデザインが必要です。

ユーザーがサイトを作成する場合、デフォルトでテーマが含まれるテンプレートを選択する 必要があります。サイトに対して異なるテーマを作成することも、異なるテーマを使用する ことを選択することもできます。

テーマの作成時は、ページ・レイアウトのデザインと同様に、ユーザーが使用できる各コン ポーネント・タイプ(段落、タイトル、イメージなど)の「設定」パネルに表示される、使用可 能なスタイルを指定する必要があります。様々なコンポーネントのデフォルト設定を指定す るために、1つのデザイン内で複数のスタイルを使用できます。テキストのフォントとフォン ト・サイズ、イメージのフレーミングなどを選択してから、それらを名前付きのデザインと して保存します。デザイン内のアイテムは、レンダリング時にページ・レイアウトの外観に 影響します。

Oracle Content Management でのサイトの構築のサイト・ページの操作を参照してください。

# デザイン・ファイル

design.json および design.css の 2 つのファイルが、テーマのデフォルトのデザイン・ファイルであると見なされます。

- design.json は、コンポーネントのスタイルを指定します
- design.css は、クラス値(色やフォントなど)の定義を提供します

これらのファイルは、テーマ構造内の/designs/default/ディレクトリにあります。



#### design.json ファイル

```
design.json ファイルには、次の構造があります:
{
    "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": []
        }
    }
}
```

"styles":[]の各エントリには、特定のコンポーネントのスタイルのリストが含まれま す。たとえば、タイトル・コンポーネントには、次のデフォルトのスタイルが用意さ れています:



#### 🥒 ノート:

]

コンポーネントのカスタム・スタイルを作成し、design.json内のスタイルにマッ プする場合、システムに用意されている名前のようなフルネームを使用する必要は ありません。使用する文字列のみを指定します。たとえば、名前に "COMP_STYLE_BOX"を使用するかわりに、単純に"Box"を使用します。つまり、その コンポーネントの「設定」パネルの「スタイル」タブのリストには、"COMP_STYLE_BOX" のかわりに"Box"という名前のみが表示されます。

名前値は、次のように、ユーザー・インタフェースに表示する実際の単語にマップされます:

```
"COMP_STYLE_FLAT": "Flat",
"COMP_STYLE_HIGHLIGHT": "Highlight",
"COMP_STYLE_DIVIDER": "Divider",
```

#### design.css ファイル

design.css ファイルは、クラス値の定義を提供します。次にいつくか例を示します。

```
.scs-title-default-style {
  color: #33333;
  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; }
```

# レスポンシブ表の設計

**Oracle Content Management** では、段落コンポーネント内にレスポンシブ表のサンプ ル **CSS** が用意されており、これを使用することで、モバイル・デバイスでは行データ が積み上げられて表示されます。

レスポンシブ表は、画面のサイズに応じて、コンテンツが効果的に表示されるよう表 を調整します。たとえば、5列の表は、Webページでは横方向で問題なく表示されま すが、スマートフォンで表示した場合には、積み上げた方がデータは見やすくなりま す。レスポンシブ表が適切に動作するには、ヘッダー行が必要なことに注意してくだ さい。



生成された次の HTML では、表の各セルに、列ヘッダーのテキストと同じ値の data-label 属 性が追加されています。

```
<thead>
ACCOUNT
```

各セルにその属性があると、次の CSS ルールを適用した場合には、TD がそれぞれの上に積み上げられます。

```
.scs-paragraph:not(.scs-paragraph-edit) table td {
   border-bottom: lpx 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;
}
```

次のメディア・ルールにより、画面サイズの問合せが行われ、画面サイズが 767 ピクセル未 満の場合にのみ有効になります:

@media screen and (max-width: 767px) {

css を使用して、挿入された表をレスポンシブにすることができるコード・サンプルは、提供 されている StarterTheme のデフォルトの design.css にあります。StarterTheme からサイト を作成する場合、段落スロットに挿入した表は、デフォルトでレスポンシブになります。表 を挿入するには:

- 1. Oracle Content Management でサイトを開き、「編集」モードに切り替えます。
- 2. 新しい更新を作成するか、既存の更新を選択して編集します。
- 新しい段落コンポーネントをページにドラッグして、表を挿入する場所をクリックする か、表を挿入する既存の段落をクリックします。



リッチ・テキスト・ツールバーで、 ●●をクリックして、表プロパティを設定します。ヘッダーには必ず1行目を選択し、使用が予測される最小の画面で効果的に機能するよう、表の幅を調整します。たとえば、スマートフォンでのサイトの表示が予想される場合、デフォルトの幅である767ピクセルは、スマートフォンの画面で完全に表示するには、積み上げられたとしても幅が広すぎます。表の幅を300ピクセルなど、より小さいサイズに設定するか、幅を100%に設定する必要があります。

完了したら、**「表示」**モードに切り替えて、画面の幅が **767** ピクセルより狭い表示オプ ションを選択し、結果をプレビューします。編集モードでは表がレスポンシブに動作 しないため、表示モードにする必要があります。

Fit to Window 🔻 🖳 🞸	View DEdit
Fit to Window	
Desktop	eloping Templates
768 x 1024	
Mobile	
360 x 740 (Galaxy S8, S9)	
375 x 667 (iPhone 8)	
375 x 812 (iPhone X)	
414 x 736 (iPhone 8 Plus)	
Create a Device Preset	

別のテーマからサイトを作成する際にレスポンシブ表を使用する場合は、 StarterTheme の design.css から、使用するテーマの design.css ファイルにコードをコ ピーする必要があります。

- **1.** StarterTheme の design.css からコードをコピーするには、Oracle Content Management のサイド・ナビゲーションで「開発者」をクリックします。
- 2. 「すべてのテーマの表示」をクリックします。
- 3. StarterTheme を選択し、「開く」をクリックします。
- 4. フォルダを開く設計をクリックし、「デフォルト」をクリックします。
- 5. StarterTheme から design.css ファイルを選択し、「ダウンロード」 をクリックしま す。
- **6.** テキスト・エディタでファイルを開き、「*An example CSS of how to render a table responsively*」というコメントで始まるファイルのセクションを探します。
- 7. 次のコメントまでのコードを選択してコピーします。

/**

* 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. 編集するテンプレートの design.css ファイルをダウンロードするステップを繰り返し、
```

- テンプレートを開いて、コピーしたコードをファイルに貼り付けます。 本面を保存し 編集した design css ファイルを 変更するテーマに新しいリビジョンと
- 9. 変更を保存し、編集した design.css ファイルを、変更するテーマに新しいリビジョンと してアップロードします。



# 会話リストのスタイルのカスタマイズ

design.css ファイルにセレクタを追加することにより、会話リスト・コンポーネン トのスタイルをカスタマイズできます。

次の **CSS** セレクタを使用して、会話リスト・コンポーネントのスタイルをカスタマイ ズします。

セレクタ名	説明
.scs-convo-list-cust .scs-convo-list-container	コンポーネントの最も外部の DIV
.scs-convo-list-cust .scs-convo-list-title	リスト内の会話のタイトル <b>(</b> 選択 されている場合 <b>)</b>
.scs-convo-list-cust .scs-convo-list-line- separator	リスト・タイトルとリスト間の セパレータ
.scs-convo-list-cust .scs-convo-list-convo- title	リスト内の会話のタイトル
.scs-convo-list-cust .scs-convo-list-convo- line-separator	各会話間のセパレータ
.scs-convo-list-cust .scs-convo-list-active	リスト内の会話のタイトル <b>(</b> 選択 されている場合 <b>)</b>
.scs-convo-list-cust .scs-convo-list-convo- posts	会話の投稿数
.scs-convo-list-cust .scs-convo-list-convo- unread	会話の未読メッセージの数
.scs-convo-list-cust .scs-convo-list-convo- updated	会話の最終更新日
.scs-convo-list-cust .scs-convo-list-no-convo- msg	メッセージ <b>(</b> リストが空である場 合 <b>)</b>
.scs-convo-list-cust .scs-convo-list-no-auth- msg	メッセージ(ユーザーの認可なし で会話リストがパブリック・サ イトでレンダリングされている 場合)

*Oracle Content Management でのサイトの構築*のスタイルおよびフォーマットの使用 を参照してください。

#### 例

このサンプルは、会話リストに対するタイプ・フォント、スタイルおよび色の変更に 関するカスタマイズ済 **CSS** の使用について説明しています。



次のコードは、サンプルを作成するために使用したカスタマイズ済 CSS を示しています:

```
.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;
}
```

# フォルダ・リストおよびファイル・リストのスタ イルのカスタマイズ

design.css ファイルにセレクタを追加することにより、フォルダ・リストおよびフ ァイル・リストのコンポーネントのスタイルをカスタマイズできます。

フォルダ・リスト・コンポーネントを使用して、Oracle Content Management アカウ ントから、指定したフォルダ内のフォルダをリストできます。フォルダ・リストは自 動的にページでファイル・リスト・コンポーネントやドキュメント・マネージャとや り取りして、フォルダ・リスト内で選択したフォルダのファイルを表示します。

ファイル・リスト・コンポーネントを使用して、Oracle Content Management アカウ ントの指定したフォルダからファイルのビューを表示できます。ファイル・リストは、 ページ上で自動的にフォルダ・リスト・コンポーネントとやりとりして、フォルダ・ リストで選択したフォルダ内のファイルを表示します。

#### フォルダ・リストの CSS セレクタ

次の **CSS** セレクタを使用して、フォルダ・リスト・コンポーネントのスタイルをカス タマイズします。

セレクタ名	説明
.scs-folder-list-cust .scs-folder-list-	コンポーネントの最も外部の
container	DIV
.scs-folder-list-cust .scs-folder-list-folder- title	フォルダ名
.scs-folder-list-cust .scs-folder-list-line-	フォルダ名とサブフォルダのリ
separator	ストの間のセパレータ
.scs-folder-list-cust .scs-folder-list-sub- folder-title	サブフォルダ名
.scs-folder-list-cust .scs-folder-list-sub-	サブフォルダ名(選択されている
folder-title-active	場合 <b>)</b>
.scs-folder-list-cust .scs-folder-list-no-	表示するサブフォルダがない場
folder-msg	合のメッセージ

#### ファイル・リストの CSS セレクタ

次の **CSS** セレクタを使用して、ファイル・リスト・コンポーネントのスタイルをカス タマイズします。

セレクタ名	説明
.scs-file-list-cust .scs-file-list-container	コンポーネントの最も外部の DIV



セレクタ名	説明
.scs-file-list-cust .scs-file-list-folder-title	フォルダ名
.scs-file-list-cust .scs-file-list-line- separator	フォルダ名とファイルのリスト の間のセパレータ
.scs-file-list-cust .scs-file-list-row	ファイルに関する情報が含まれ る行
.scs-file-list-cust .scs-file-list-left-col	コンポーネントの左セクション にあるファイルのサムネイル
.scs-file-list-cust .scs-file-list-mid-col	ファイルの名前、説明、最終変 更およびサイズが含まれるコン ポーネントの中央セクション
.scs-file-list-cust .scs-file-list-file-title	アプリケーションの中央セクシ ョンにあるファイル名
.scs-file-list-cust .scs-file-list-file-desc	アプリケーションの中央セクシ ョンにあるファイルの説明
.scs-file-list-cust .scs-file-list-file- lastModified	ファイルの最終変更
.scs-file-list-cust .scs-file-list-file-size	最終変更との垂直セパレータが あるファイルのサイズ
.scs-file-list-cust .scs-file-list-file-size- no-sep	垂直セパレータがないファイル のサイズ <b>(</b> 最終変更は非表示 <b>)</b>
.scs-file-list-cust .scs-file-list-right-col	アプリケーションの右セクショ ン
.scs-file-list-cust .scs-file-list-file- download-icon	アプリケーションの右セクショ ンにあるダウンロード・アイコ ン
.scs-file-list-cust .scs-file-list-no-file-msg	表示するファイルがない場合の メッセージ

『Oracle Content Management でのサイトの構築』のファイル・リストおよびフォルダ・リストに関する項を参照してください。

# ソーシャル・バーのアイコンのカスタマイズ

カスタム・ソーシャル・アイコンを作成して、テーマのデフォルト・サイト内のソーシャル・ バーで使用できます。

サイト内のソーシャル・バーに表示されるソーシャル・アイコンは、サイトのテーマのデザ インによって決まります。サイトのテーマを変更すると、テーマとともにソーシャル・アイ コンが変更されます。共通のソーシャル・アイコンは、Facebook、Twitter、LinkedIn、 Google+および YouTube 用の Oracle Content Management テーマに含まれています。

design.json および design.css ファイルを編集することで、カスタム・ソーシャル・アイコ ンをソーシャル・アイコン・コンポーネントに追加できます。



#### design.json ファイル

design.json ファイル内で、このサンプル・コードで示すように名前およびクラス構造を使用して新しいアイコンを指定できます:

```
"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"
        ]
    }
```

#### design.css ファイル

design.css ファイル内で、このサンプル・コードで示すように名前および url 指定を 使用して新しいアイコンを追加できます:

```
.scs-facebook-icon {
   background-image: url("facebook.png"); }
.scs-twitter-icon {
   background-image: url("twitter.png"); }
.scs-linkedin-icon {
   background-image: url("linkedin.png"); }
```

# Oracle Intelligent Advisor のインタビュー・スタイル設 定拡張機能の構成

**Oracle Intelligent Advisor (OIA) (**以前の **Oracle Policy Automation)**インタビューをスタイル設定して、企業のルック・アンド・フィールで外観を統一できます。

design.css ファイルで次の CSS クラス・セレクタを構成できます。これらのクラス・ セレクタはすべて、接頭辞 scs-opainterview-を持ちます。



各クラスは、指定されたコンポーネントのすべてのスタイルを定義します。クラスが完全に 制御し、既存の OIA スタイルは使用されません。

セレクタ	適用先	
interview	interview	インタビュー・リージョンは、ヘッダー、フッタ ー、ナビゲーション・リージョンを含むインタビ ュー・コンテンツ全体で構成されます。
interviewContent	インタビューのコンテンツ	インタビュー・コンテンツ・リージョンには、画 面タイトルとコントロールが含まれますが、ヘッ ダー、フッター、およびナビゲーション領域は除 外されます。
screenTitleBlock	画面タイトル・ブロック	画面タイトル・ブロックは、画面タイトルと、その行に含まれる他のウィジェット(たとえば、画面ドロップ・ダウン・リストおよび/または次および後ボタン)を含むリージョンで構成されます。
screenTitle	画面タイトル	画面タイトル・リージョンは、画面タイトルを含 む単なるリージョンです。
nextButton	次ボタン	次へボタン。
backButton	戻るボタン	戻るボタン。
restartButton	再起動ボタン	再起動ボタン。
exitButton	終了ボタン	終了ボタン。
header	header	ヘッダー・リージョン。
footer	footer	フッター・リージョン。
question	質問テキスト	質問テキスト用のスタイル。
control	コントロールのコンテナ	コントロールを含む要素のスタイル。
label	ラベル・コントロール	ラベル・コントロール用のスタイル。
controlError	すべてのコントロール	エラー・テキスト・コンテナのスタイル。
controlErrorText	すべてのコントロール	エラー・テキスト範囲のスタイル。
textInput	1 行のテキスト・ボックス、 パスワードおよびマスク	テキスト入力コントロール用のスタイル。
textAreaInput	複数行テキスト・ボックス	複数の行のテキスト入力コントロール用のスタイ ル。
calendarInput	カレンダ	カレンダ入力コントロール用のスタイル。カレン ダ・アイコンの色を変更できるようにする iconColor フィールドと、カレンダ・アイコンを 表示するかどうかを示す keepIcon フィールドを サポートします。
dropDownInput	ドロップダウン・リスト	ドロップダウン・リスト入力コントロール用のス タイル。
filterDropDownIn put	フィルタリングされたドロッ プダウン・リスト	フィルタリングされたドロップダウン・リスト入 カコントロール用のスタイル。ドロップダウン矢 印の色を変更できる iconColor フィールドをサ ポートしています。
listInput	固定リスト	固定リスト入力コントロール用のスタイル。
radioInput	ラジオ・ボタン	ラジオ・ボタン入力コントロール用のスタイル。 borderColor および fillColor オプションを使 用してスタイルを設定できます。iconType プロ パティは、使用されるアイコンのタイプを変更で きます。現在、tick および fill が唯一サポート されている代替オプションです。



セレクタ	適用先	説明
checkboxInput	チェック・ボックス	チェック・ボックス入力コントロール用のスタイ ル。borderColor および fillColor オプション を使用してスタイルを設定できます。iconType プロパティは、使用されるアイコンのタイプを変 更できます。現在、square および fill が唯一サ ポートされている代替オプションです。
autoCompleteInp ut	カスタム検索	customSearch 拡張子が使用されている場合の autocomplete フィールドのスタイル。
captchaInput	<b>CAPTCHA</b> 入力フィールド	ユーザーが <b>CAPTCHA</b> 情報を入力する入力フィー ルドのスタイル。
signatureInput	署名コントロール	署名コントロール用のスタイル。追加の inkColor フィールドをサポートしているため、署 名用のペンのインクを変更できます。
explanationHead er	説明コントロール	説明コントロールのための最上位の拡張可能なへ ッダーのスタイル。
explanationText	説明コントロール	拡張された説明コントロール・テキストのスタイ ル。
signatureClearBut ton	署名コントロール	署名コントロール上のクリア・ボタンのスタイル。
uploadAddButton	アップロード・コントロール	アップロー ^ド ・コントロールの追加ボタンのスタ イル。
entityRemoveButt on	エンティティ 収集コントロー ル	エンティティ 収集コントロール。

#### design.css で定義されたスタイル拡張の例

```
.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;
}
```



# 18 背景の使用の理解

サイト・ページおよびページ上の個々のスロットの背景色およびイメージを指定できます。 ページ、スロットおよびコンポーネントの背景は積み重なっています。たとえば、スロット の背景を指定すると、ページに指定されている背景の上に積み重ねられます。

- 背景およびテーマについて
- 背景の実装方法
- 設定の格納場所

### 背景およびテーマについて

背景機能は、主に、サイトのページおよびスロットでの使用が対象となっています。ページ およびスロットに対して設定される背景は、開発者ではなくユーザーによってテーマ・レベ ルで構成されるよう設計されています。

背景効果は、テーマのスロットで構成できますが、これらの効果により、サイト編集時に、 ページに対して設定されたすべての背景効果がオーバーライドされる可能性があります。

背景機能では、テーマ、または構成要素であるテーマ・ファイルは変更されません。テーマ はページの背景設定では変更できず、サイト内の同一レイアウトに基づく別ページは、ソー ス・ページの背景設定を受け継ぎません。テーマによる背景の追加または変更には、テーマ の更新が必要となります。

ユーザーは、サイトのすべてのページ上の様々な背景のみでなく、同一レイアウトを使用す る様々なページ上の背景も構成できます。これは、テーマ内では不可能です。テーマではペ ージの背景を指定できますが、ユーザーの背景設定ではこの指定をオーバーライドできます。 (背景設定では、特定のページに対してオーバーライドが適用されるのみであり、テーマ自体 が変更されるわけではありません)。

ユーザーがサイトのページおよびスロットで設定する背景スタイルがテーマによってオーバ ーライドされないように注意してください。これは、次のいくつかの方法で起こる可能性が あります:

- <body>とスロットで要素ベースの"style"属性を使用すると、サイトのページ背景設定を テーマで上書きすることができます。
- テーマの背景スタイルをテーマのカスケード・スタイル・シート(CSS)ファイルの"! important"としてマークすると、サイトのページ背景設定をテーマで上書きすることが できます。

**Oracle Content Management** でのサイトの構築の背景またはテーマの変更を参照してください。

### 背景の実装方法

ページおよびスロットの背景機能を実装するために、Oracle Content Management では、ページの<head>に CSS スタイルシートを動的に作成します。



スタイルのセレクタは、ページの背景設定のためのタグベースのセレクタ(body)です。 スロットの背景設定および ID ベースのセレクタの場合は、スロット ID が使用されま す。

たとえば、ページの背景色を設定すると、ページの<head>内に次のような CSS マーク アップが生成される場合があります:

```
body
{
    background-color: #fa7c9d;
}
```

同様に、スロット上に背景イメージを設定すると、ページの<head>内に次のような CSS マークアップが生成される場合があります:

```
#PageFooter
{
    background-image: url("footer_image.png");
```

この実装は、body タグまたは slot 要素の"style"属性で直接指定されたスタイルが、<head>コードのスタイルシートで構成された設定をオーバーライドできることを意味します。

#### **日** 重要:

}

テーマの開発者は、要素ベースのスタイルで背景設定をオーバーライドしな いように注意する必要があります。

*Oracle Content Management でのサイトの構築*の背景またはテーマの変更を参照して ください。

### 設定の格納場所

ページがレンダリングされる場合、背景設定は、ページの<head>コード内の"style"タ グに動的に書き込まれます。

背景設定は、ページ・モデル・ファイル(*<pageid*>.json など)内に永続化されます。特 に、ページの背景設定は properties.styles セクションに格納され、スロットの背景 設定は slots[*<slot_id>*].styles セクションに格納されます。

次の典型的なサンプルに示されるとおり、背景設定はページの JSON ファイル(特に "styles"内)に格納されます。

```
{
    "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"
                }
          }
     }
}
```

## 19

# トリガーおよびアクションの設定

コンポーネント内のトリガーが別のコンポーネント上のアクションをコールするように、コ ンポーネント(インライン・フレームでレンダリングされたコンポーネントを含む)間の通信を 構成できます。

- トリガーおよびアクションについて
- トリガーの設定
- アクションの設定

### トリガーおよびアクションについて

コンポーネント内の**トリガー**が別のコンポーネント上の**アクション**をコールするように、コ ンポーネント(インライン・フレームでレンダリングされたコンポーネントを含む)間の通信を 構成できます。

トリガーは、Oracle Content Management のコンポーネント間の通信の一部です。任意のコ ンポーネントが任意の数のトリガーを呼び出すことができます。コンポーネントは、トリガ ーにペイロードを提供できます。このペイロードは、トリガーが呼び出されたときに実行さ れる任意のアクションに渡されます。各トリガーに対して実行されるアクションを選択でき ます。連携して動作するよう構築されたコンポーネントは、ユーザー操作なしで他のコンポ ーネントに対してアクションを実行するためにトリガーを自動的に呼び出すことができま す。

基本的な手順は次のとおりです:

- 1. トリガーの登録
- 2. アクションの呼出し
- 3. アクションの登録
- 4. 設定を確認するためのアクションの実行

たとえば、ボタン・コンポーネントを使用して、ページ・コンポーネントの表示または非表 示とメッセージの表示など1つ以上のアクションを実行できます。営業所の場所のリストが 1つのコンポーネント内にあるときに、リスト内の1つの場所がクリックされた場合、その場 所に関する詳細が別のコンポーネントに表示されます。

カスタマイズするコンポーネントについては、トリガーおよびアクションは、コンポーネン ト登録データの一部であり、コンポーネント実装の一部ではありません。登録データ内には、 コンポーネントがサポートしているトリガーおよびアクションのリストが含まれる "triggers": [],および"actions":[],エントリがあります。実際の構文は、ローカル・コン ポーネントとリモート・コンポーネントも同じであり、その場所と取得方法のみが異なりま す。

**Oracle Content Management** でのサイトの構築のトリガーおよびアクションの使用を参照してください。



### トリガーの設定

コンポーネントには、他のコンポーネントでアクションを実行するトリガーを含める ことができます。コンポーネントによって呼び出されるトリガーを登録する必要があ ります。

コンポーネントは、トリガーにペイロードを提供します。このペイロードは、トリガ ーが呼び出されたときに実行される任意のアクションに渡されます。各トリガーに対 して実行されるアクションを選択できます。連携して動作するよう構築されたコンポ ーネントは、ユーザー操作なしでアクションを実行するためにトリガーを自動的に呼 び出すことができます。

#### トリガーの登録

カスタム・コンポーネントの場合、トリガーは、コンポーネントの登録データの一部 として登録されます。トリガーを追加するには、コンポーネントによってサポートさ れている各トリガーを使用して"triggers"プロパティ配列を更新します。また、トリ ガーがサポートしているペイロードを指定し、アクションによってサポートされるプ ロパティにユーザーがペイロード内の値をマップできるようユーザー・インタフェー スを作成できるようにする必要があります。

1. appinfo.json ファイルを編集し、"triggers":[],エントリをレビューします:

```
"triggers": [{
    "triggerName": "helloWorldWhoAreYou",
    "triggerDescription": "Show Who I Am",
    "triggerPayload": [{
        "name": "whoAreYou",
        "displayName": "Who I Am"
    }]
}],
```

2. ファイルをサイト・サーバーに同期化します。

このサンプルのトリガー・エントリでは、triggerName ("helloWorldWhoAreYou")を定 義しました。名前値は一意である必要があります。次に、トリガーに説明("Show Who I Am")を付けました。これは、トリガーを表示するためにユーザー・インタフェース・ ダイアログで使用されます。最後に、トリガーの単一値のペイロードを定義しました。 ユーザーは、このペイロード内のエントリを選択し、これらをアクション内のフィー ルドにマップします。

トリガーが登録されたら、コンポーネントの「設定」パネルの**「リンク」**タブに移動 したときにトリガーを表示して選択できるようになります。

#### トリガーの呼出し

トリガーは、コンポーネントが任意の時点で呼び出すことができます。通常、トリガ ーは、ボタンのクリックや表内の行の選択などのユーザー操作によって呼び出されま す。REST API コールが原因でデータが変更された場合など、コンポーネントが任意の 基準に基づいてトリガーを呼び出すことができます。トリガーが呼び出された場合、 任意の数のアクションを実行できます。

トリガーを呼び出す方法の例は、次のとおりです:



 render.js ファイルを編集し、トリガーを呼び出すためにサイト SDK をコールする JavaScript 関数を viewModel オブジェクトに追加します。

```
self.raiseTrigger = function (triggerName) {
   SitesSDK.publish(SitesSDK.MESSAGE_TYPES.TRIGGER_ACTIONS, {
    'triggerName': 'helloWorldWhoAreYou',
    'triggerPayload': { "whoAreYou": "This is " + self.whoAreYou() + "!"}
  });
};
```

2. トリガーを呼び出すための関数をコールするユーザー・インタフェース内のエントリ(edit template.html)およびボタンを</div>の前に追加します。

<button data-bind="click raiseTrigger">Who Am I?</button>

**3.** render.js ファイルを Oracle Content Management インスタンス・サーバーに同期また はアップロードします。

ViewModel オブジェクトで、ボタンがクリックされたときに呼び出される JavaScript 関数を 作成しました。この関数は、サイト SDK を呼び出し、このトリガー"helloWorldWhoAreYou" に定義されているすべてのアクションをトリガーするよう命じます。これはまた、単一のフ ィールド"whoAreYou"を持つ triggerPayload を経ます。これらの値"helloWorldWhoAreYou" および"whoAreYou"は、前の手順でトリガーを登録したときに入力した値と一致します。

### / ノート:

アクションが実行されるタイミングに関して事前定義済の順序はありません。各ア クションはリストされている順序でコールされますが、現在のアクションが完了す るまで待機しないと次のアクションがコールされないわけではありません。アクシ ョンによって非同期コールが行われる場合、次のアクションが実行される前に完了 しない可能性があります。

# アクションの設定

コンポーネントを設定してアクション登録を活用することにより、コンポーネント内でアク ションを実行するページにコンポーネントをドロップできるようになります。

#### アクションの登録

トリガーが呼び出されると、コンポーネントでアクションがコールされます。コンポーネン トは、任意の数のアクションを登録できるほか、アクションがサポートしているペイロード を定義することもできます。ユーザーは、アクションを選択するときに、アクションに渡す ペイロードを移入できます。

トリガーを登録する場合と同じように、コンポーネントがサポートしているアクションをテ ーマの appinfo.json 登録データに登録できます。

アクションを登録する方法の例は、次のとおりです:



 コンポーネントの appinfo.json ファイルを編集し、"actions":[],エントリを 更新します。

 アクションは、登録された後、コンポーネントの「設定」パネル上の「リンク」タ ブでトリガーをクリックするときに呼び出されるアクション・ダイアログに表示さ れます。

#### アクションの実行

アクションを登録したら、コンポーネント内のアクションを実行するページにコンポ ーネントをドロップできるようになります。コンポーネントがアクションを実行する には、EXECUTE_ACTION メッセージをリスニングする必要があります。このメッセージ には、アクションに渡されるペイロードが含まれ、ここから想定される値を抽出する 必要があります。

例として、EXECUTE_ACTION メッセージをリスニングするには、render.jsファイル を編集し、次のエントリを使用して ViewModel オブジェクトを更新します:

```
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);
    }
};
```

これにより、アクションを実行するための JavaScript 関数が作成され、EXECUTE_ACTION メッセージが呼び出されるたびにサイト SDK を使用して関数がコールされます。

このアクションは、EXECUTE_ACTION メッセージが呼び出されるたびにコールされ、処 理対象として設計されているアクションのみを処理するかどうかはコンポーネント次 第です。これを行うには、アクションの名前を確認し、それが処理可能なアクション であることを確認する必要があります。



アクションのペイロードは値の配列です。この例では、値は配列内の最初のエントリである ことが前提です。通常、関心のあるペイロード値を配列から探し出す必要があります。

### ✓ ノート: アクション・リスナーはコールバックであるため、JavaScript クロージャを使用す るか、関数を適切にバインドし、関数の実行時に ViewModel にアクセスできること を確認する必要があります。



# 20 テンプレートの開発

テンプレート・パッケージには、サイトの開発バージョン、ページ・レイアウトが含まれる テーマ、スタイルとナビゲーション、およびサイト内で使用される関連コンポーネントが含 まれます。Oracle Content Management は、サイトの作成に使用したり、カスタム・テンプ レートを作成するための開始点として使用するテンプレートのセットをを提供します。

- テンプレートについて
- 基本テンプレート構造
- テンプレートの作成
- テンプレートのエクスポート
- テンプレートのインポート
- スタータ・テンプレートの操作
- Bootstrap または Web サイト・デザイン・テンプレートからのサイト・テンプレートの作成
- Developer Cloud Service によるテンプレートの開発

# テンプレートについて

テンプレートには、ユーザーが Web サイトの作成を開始するために必要なすべての要素が含まれます。これには、サンプル・ページとコンテンツを含むサイト、スタイルが設定されたテーマ、ナビゲーション、イメージなどのアセットおよび関連付けられたコンポーネントが含まれます。

Oracle Content Management には、サイトの作成に使用するための多数のテンプレートが用 意されています。これらのテンプレートは、通常、サービスが初期化される際に管理者によ ってインストールされます。Oracle Content Management の管理のサイト設定の構成を参照 してください。

新しいサイトを作成するたびに、テンプレートを選択する必要があります。テンプレートは、 パートナ・ポータルやマーケティング・キャンペーンなど、テーマをサイトおよびコンポー ネントと組み合せて機能またはソリューションを推進します。

すぐに使用可能なテンプレートのリストは、「テンプレートの理解」を参照してください。

開発者とユーザーの両方が、新しいテンプレート作成し、既存のテンプレートを変更および 置換できますが、開発者の主要作業の1つは、新しいテンプレートを設計することです。こ のプロセスは基本的に、次の手順で構成されます。

既存のテンプレート(JET スタータ・テンプレートなど)をコピーして、新しいテンプレートを作成します。また、これにより、テーマがテンプレートに関連付けられます。
 例:

cec create-template My_JET_Template -f JETStarterTemplate



- 2. .zip ファイルのテンプレートを開発環境にエクスポートします。
- 3. テンプレート・パッケージ内のファイルを開き、変更を加えます。
- 4. .zip ファイル内に改訂されたテンプレート・パッケージを作成します。
- 5. Oracle Content Management インタフェースを使用して、新しいテンプレートをイ ンスタンスにインポートします。
- 6. テンプレートを共有して他のユーザーが使用できるようにします。

あるいは、Oracle Content Management インタフェースを使用して、ページ・レイア ウトおよびアセットの追加と変更、サイト構造の拡張、サイト・ページへのコンポー ネントの追加、およびこのテンプレートのテーマを使用するすべてのサイトに表示す るシード・コンテンツの追加を行うことによって、テンプレートを変更できます。

テンプレートの管理に関する項も参照してください。

### 基本テンプレート構造

テンプレートの基本構造には、サイト(アセット、レイアウト、ページ、およびコンテンツを含む)、関連するテーマ、およびカスタム・コンポーネントが含まれます。

サイトの作成時は、サイトの構造と初期コンテンツを提供するテンプレート、デザイ ンとレイアウトの指定が含まれたテーマ、および任意のカスタム・コンポーネントを 選択する必要があります。

テンプレートは、基本的なフォルダおよびファイルを示すこの例のとおり、特定の構 造で編成されます:

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
```

コンポーネントとテーマのフォルダおよびファイルについては、このガイド内の別の場所で 説明します。「コンポーネントの開発について」および「基本テーマ構造」を参照してくださ い。

/ ノート:

- テーマのコピーを使用する JET スタータ・テンプレートまたはスタータ・テン プレートを使用しない限り、一般にテーマはテンプレート間で共有されます。 スタータ・テンプレートの操作を参照してください。
- テーマには、もはやサイトが含まれていません。サイトのフォルダおよびファ イルは、/template フォルダ内にあります。

*template_name*/templateフォルダには、サイトのフォルダおよびファイルが含まれています。

- assets: ユーザー・インタフェースのテンプレート詳細ページに表示されるイメージが含まれています。
- content: サイトで使用される管理コンテンツが含まれています。
- layouts: この時点では使用されません。
- pages: データとともにすべてのページ JSON ファイルが含まれています。nnn.json と いうフォーマットを使用します。ここでの nnn はページ ID です。
- variants: サイトに対するすべての更新の詳細が含まれています。
- _folder.json: サイト作成者、サイト名、アイテム GUID、サイトの簡易説明および詳細 説明など、テンプレートのメタデータが含まれています。
- _folder_icon.png: ユーザー・インタフェースでのサイトを表します。



- componentsused.json:(非推奨。)サイト内で使用されるカスタム・コンポーネント がある場合は、それらを記録します。下位互換性のためにのみ維持されています。
- controller.html: ブラウザでサイトを表示する、キー・コードが含まれています。 このファイルを変更する場合は、インタフェースのサイト設定でこれを行うことを お薦めします。オフラインでファイルを変更できます。コントローラ・ファイルの カスタマイズを参照してください。
- siteinfo.json: サイト名、関連付けられているテーマの名前、およびサイトのその他のメタデータを識別します。このファイルは変更しないでください。
- structure.json: ページ(親ページおよび子ページ)に関してサイトの階層を定義します。サイトのナビゲーションを設定する場合は、ツリー構造を引き出すためにレンダリング API を使用できます。「サイト・ナビゲーション」および「レンダリング API リファレンス」を参照してください。

## テンプレートの作成

他のサイトの開始点として使用するサイトがある場合は、そのサイトからテンプレートを作成できます。既存のテンプレートをコピーし、コピーに変更を加えることによって、新しいテンプレートを作成することもできます。

既存のサイトからテンプレートを作成する場合、新規テンプレートでは、サイトのコ ピーがデフォルト・サイトとして使用されます。テンプレートは、サイトによって使 用されるテーマ、およびサイトのページで使用されるカスタム・コンポーネントを参 照します。テーマおよびカスタム・コンポーネントは、テンプレートにコピーされま せんが、サイトによる参照と同じ方法で参照されます。テンプレートは、テンプレー トが作成されるときに、それを作成するために使用されたサイトを反映しています。 テンプレートの作成に使用されたサイトへのその後の変更は、そのテンプレートとと もに格納されているサイトでは反映されません。





既存のテンプレートをコピーし、コピーの名前を変更を変更することによって新しいテンプ レートを作成する場合、コピーを変更します。テンプレートをコピーする場合、テンプレー トの共有情報はコピーされないことに注意してください。

テンプレート、テーマ、コンポーネント、サイトまたはサイト・ページについて、authsite、 content、pages、scstemplate_*、_comps、_components、_compsdelivery、_idcservice、 _sitescloud、_sitesclouddelivery、_themes、_themesdelivery といった名前を使用しないでく ださい。documents および sites は、サイト・ページには使用できますが、テンプレート、テ ーマ、コンポーネントまたはサイトには使用しないでください。

独自のカスタム・テンプレートを(サイトおよびテーマを使用して)作成する場合、サイトおよ び関連テーマの基本要素が含まれる、JET スタータ・テンプレートまたは Oracle Content Management によって提供されるスタータ・テンプレートを使用することをお薦めします。 スタータ・テンプレートには、カスタム・テンプレートでサイトとテーマをレイアウトおよ び設計する方法を調べる上で役に立つ、サイト・ページに書き込まれた情報および手順が含 まれています。

テンプレートの管理およびスタータ・テンプレートの操作を参照してください。

## テンプレートのエクスポート

テンプレートをエクスポートしオフラインで変更して、その後新規テンプレートとしてイン ポートするか、既存テンプレートを置き換えることができます。また、テンプレートをエク



スポートして別の Oracle Content Management インスタンスに移動し、そこでインポートすることもできます。

テンプレートをエクスポートする際、基本的に、テンプレートを Oracle Content Management のフォルダに単一の.zip ファイルとしてコピーします。テンプレー ト・パッケージは、フォルダから直接ダウンロードして解凍し、個別のファイルを操 作できます。テンプレート・ファイルの操作が完了したら、テンプレート・パッケー ジを含む.zip ファイルを作成し、Oracle Content Management にインポートして、元 のテンプレートを上書きするか、新規に作成します。

ノート:
 テンプレートをエクスポートした場合、テンプレートの共有情報は含められません。

テンプレートをエクスポートするには:

**1.** Oracle Content Management サイド・ナビゲーションで、「開発者」をクリックします。

「開発者」ページが表示されます。

- 2. 「すべてのテンプレートの表示」をクリックします。 既存のテンプレートのリストが表示されます。
- テンプレートを選択して、右クリック・メニューの「エクスポート」を選択する
   か、アクション・バーの
   たクリックします。
- フォルダに移動するか、「作成」をクリックして名前およびオプションの説明を指定し、「作成」をクリックして新規フォルダを作成します。
   フォルダを開くには、フォルダ・アイコンまたはフォルダ名をクリックします。
- 5. 関連付けられているフォルダのチェック・ボックスをクリックしてフォルダを選択 し、「OK」をクリックします。

テンプレート・パッケージ・ファイルは、テンプレート名および.zip 拡張子で、 選択されたフォルダに作成されます。

## テンプレートのインポート

テンプレートをエクスポートしオフラインで変更して、その後新規テンプレートとし てインポートするか、既存テンプレートを置き換えることができます。また、テンプ レートをエクスポートして別の Oracle Content Management インスタンスに移動し、 そこでインポートすることもできます。

テンプレートをエクスポートする際、基本的に、テンプレートを Oracle Content Management のフォルダに単一の.zip ファイルとしてコピーします。テンプレー ト・パッケージは、フォルダから直接ダウンロードして解凍し、個別のファイルを操 作できます。テンプレート・ファイルの操作が完了したら、テンプレート・パッケー ジを含む.zip ファイルを作成し、Oracle Content Management にインポートして、元 のテンプレートを上書きするか、新規に作成します。

テンプレート・パッケージをインポートするには:



- Oracle Content Management サイド・ナビゲーションで、「開発者」をクリックします 「開発者」ページが表示されます。
- 2. 「すべてのテンプレートの表示」をクリックします。 既存のテンプレートのリストが表示されます。
- 3. 「作成」をクリックして、「テンプレート・パッケージのインポート」を選択します。
- テンプレート・パッケージがアップロード済の場合、テンプレート・パッケージが含まれるフォルダに移動します。フォルダを開くには、フォルダ・アイコンまたはフォルダ名をクリックします。

テンプレート・パッケージをまだアップロードしていない場合:

- a. テンプレート・パッケージをアップロードするフォルダに移動するか、「新規」をク リックしてフォルダを作成し、名前およびオプションの説明を指定して、「作成」を クリックします。
- b. 「アップロード」をクリックします。
- c. テンプレート・パッケージを検索して選択し、「開く」をクリックします。 進行状況バーに、ファイル名とアップロード・ステータスが表示されます。
- 5. ファイル名の横にあるチェック・ボックスをクリックしてテンプレート・パッケージを選 択し、「OK」をクリックします。

インポートされたテンプレートと既存のテンプレート、テーマまたはカスタム・コンポー ネントのコンテンツで競合がない場合、新規の Oracle Content Management フォルダが、 テンプレート、関連付けられているテーマおよびカスタム・コンポーネント用に作成され ます。

6. テンプレート、テーマまたはカスタム・コンポーネントの名前または ID が存在する場合、 競合を解決するよう求められます。

競合の性質に応じて、新規テンプレート、テーマまたはカスタム・コンポーネントを作成 するオプションが表示されるか、場合によっては、既存のテンプレート、テーマまたはカ スタム・コンポーネントを、インポートしたバージョンで上書きできます。

# スタータ・テンプレートの操作

**Oracle Content Management** に用意されているスタータ・テンプレートのコピーを使用して、 テーマ・コンポーネントおよびカスタム・コンポーネントで新しいサイトを作成します。

スタータ・テンプレートは、カスタム Web サイトを構築するために必要な部分をすべて 1 つ のパッケージに収集します。これには、デフォルト・サイト、レイアウト、ナビゲーション、 サンプル・コンテンツ、テーマ、関連コンテンツ・アイテムなどが含まれます。Oracle Content Management に用意されている他のテンプレートとは異なり、スタータ・テンプレートには、 新しいテーマを含む新しいサイトを作成するために使いやすい基本フレームワークが用意さ れています。

/ ノート:

通常、テーマはテンプレート間で共有されますが、スタータ・テンプレートではテ ーマのコピーが使用されます。


スタータ・テンプレート StarterTemplate および JETStarterTemplate は、管理者によっ てサービスのインストールおよび構成時にテンプレートが有効化されている場合に Oracle Content Management で他のテンプレートとともに提供されます。Oracle JavaScript Extension Toolkit (JET)スタータ・テンプレートには、テンプレート用の最 新の JET スタイルが含まれ、ページ・コンテンツが(JET コンポーネントとして)実装 され、JET ベースのテンプレートおよびサイトを構築するためのスタータ・コンポー ネントが用意されています。

#### 基本プロセス

スタータ・テンプレートを操作するための基本プロセスを次に示します:

**1.** 新しいサイトを作成し、スタータ・テンプレートを選択します。新しいサイトとと もに新しいスタータ・テーマが作成されます。

✓ ノート: サイト名;

サイト名を慎重に選択します。サイトに付ける名前は新しいテーマの名 前と複製され、このテーマは、サイトを公開した後にユーザーに対して 表示されます。テーマの名前は作成後に変更できません。

スタータ・サイトおよびテーマには、サイトとテーマの開発を開始するために必要 な一連のフォルダとファイルが含まれます。

- 新しいテーマをデスクトップと同期化します。デスクトップを操作して、レイアウトおよび静的アセットを使用してテーマを拡張およびカスタマイズできます。これはテーマのコピーであり、テンプレートに固有の名前を使用するため、ソース・テーマに影響を及ぼさずにテーマを変更できます。
- サイト・ビルダーでサイトを開きます。サイト・ビルダーを使用して、サイト構造 に追加し、コンポーネントおよびインタラクションをページに追加し、新しいテン プレートのデフォルト・サイトの一部になるコンテンツを追加できます。スター タ・テンプレートに用意されているサイト・コンテンツを再使用または削除できま す。
- **4.** サイトおよびテーマの準備が整ったら、Oracle Content Management インスタンス と変更を同期化してから、選択したサイトから新しいテンプレートを作成します。
- 5. テンプレートを共有して他のユーザーが使用できるようにします。

#### スタータ・テンプレートを使用したテンプレートの作成

スタータ・テンプレートを使用するには:

Oracle Content Management サイド・ナビゲーションで、「サイト」をクリックします。

既存のサイトのリストが表示されます。

- 2. 「作成」をクリックします。
- 「サイトの作成」ダイアログで、サイトの基準として使用するスタータ・テンプレ ートを選択します。

新しいサイトでは、スタータ・テンプレートに用意されているテーマが使用され、 新しいサイト名に合せて名前が変更されます。



4. ダイアログで、サイトの名前を入力します。この名前はサイト URL に使用されます。文字、数字、アンダースコア()およびハイフン(-)を使用できます。スペースを入力した場合、アンダースコアに自動的に置換されます。

テンプレート、テーマ、コンポーネント、サイトまたはサイト・ページについて、 authsite、content、pages、scstemplate_*、_comps、_components、_compsdelivery、 _idcservice、_sitescloud、_sitesclouddelivery、_themes、_themesdelivery といった名前 を使用しないでください。documents および sites は、サイト・ページには使用できます が、テンプレート、テーマ、コンポーネントまたはサイトには使用しないでください。

### ♪ ノート:

Oracle Content Management のサイト URL のパスでは、大文字と小文字が区別 されます。問合せまたはフラグメント文字列の大文字と小文字の区別は、開発 者によってカスタム・コードで管理されます。

https:[//host[	:port]][/]path[?quer	ry][#fragment]
\	/\	/
Location	Data	

- 5. オプションで、サイトの説明を入力します。
- 6. 準備ができたら、「作成」をクリックします。

進行状況バーに、新規サイト名と作成ステータスが表示されます。サイトが作成される と、サイトのリストに名前が表示されます。その初期ステータスはオフラインです。

リストから新規に作成したサイトをすばやく見つけるには、**「最終更新」**でリストをソートします。直前に作成したサイトがリストの上位に表示されます。

ノート:

作成したサイトのマネージャのロールが自動的に割り当てられます。

- デスクトップ・アプリケーションを使用して、テーマをデスクトップと同期化します。
   これで、テーマのフォルダ階層およびファイルが表示されるようになります。
- 8. スタータ・テーマには、次のようなフォルダおよびファイルの最小限のセットが含まれます:

```
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
resonsepages
404.html
_folder.json
_folder_icon.jpg
components.json
viewports.json
```

**x-close.png** ファイルには、**Cookie** 承諾ポップアップのデフォルトの閉じるアイコンが含まれます。

/layouts フォルダには、次のコンテンツが含まれるスタータ・ページ・レイアウト・ファイル(index.html)があります:

- ファイルをページ・レイアウトとして使用することを可能にする一連の HTML タグ。
- シード済テキストおよび手順が含まれる単一のスロット。これには、テーマを デスクトップと同期化する方法、新しいページ・レイアウトを追加する方法、 ページ・レイアウトにコンポーネントを追加する方法、新しいページ・レイア ウトを使用してサイト階層を構築する方法などがあります。
- レンダラ API JavaScript 関数およびオブジェクトを使用する方法の例を提供 する単純な JavaScript ナビゲーション・ファイル。サイト階層内を横断し、必 要な HTML マークアップを生成し、サイト内のナビゲーションを可能にするに は、レンダリング API が必要です。

「基本テーマ構造」も参照してください。

- テーマの変更が終了したら、テーマのフォルダおよびファイルを Oracle Content Management インスタンスと同期化します。テーマに対する変更がサイト内でど のように表示されて動作するかを確認するには、テーマをサイト・ビルダーで開き ます。これは、反復的プロセスになる場合があります。
- **10.** サイトを表示または変更するには、サイトを選択して、右クリック・メニューの 「聞く」をクリックするか、アクション・バーの ひをクリックします。
- 11. エディタ・モードを「編集」に切り替え、サイトを変更できるようにします。スタ ータ・テーマに用意されているページ・レイアウトを使用して、既存のページを変 更したり、新しいページを追加したりできます。

💉 ノート:

ページ・レイアウトの構造と用途に精通している場合、スタータ・テン プレートに用意されている不要なセクションを削除し、必要なセクショ ンのレイアウトを新しいレイアウトの1つに切り替えることができま す。



- **12.** サイトの変更が終了したら、サイトを「**保存」**し、「公開」をクリックして更新をベース サイトにマージします。
- **13.** サイトを選択し、サイトに基づいてテンプレートを作成します。これにより、新しいテンプレートのアセットとテーマが取り込まれます。
- 14. テンプレートの使用を許可するメンバーとテンプレートを共有します。

# Bootstrap または Web サイト・デザイン・テンプレートからのサイト・テンプレートの作成

Oracle Content Management のオープン・アーキテクチャとは、Foundation または Bootstrap などの他のコーディング・フレームワークで行われた作業内容を使用できることを意味しま す。いくつかの変更を行うと、Bootstrap テンプレートをテーマに変更し、Oracle Content Management テンプレートの一部にすることができます。

#### 基本プロセス

各ステップの概要は、後続のセクションで詳細に説明します:

- **1.** 前提条件
- 2. サイトの作成
- 3. テーマ・フォルダの同期化
- **4.** 基本テーマの設定
- 5. サイト・ページの更新
- 6. ナビゲーションの更新
- 7. サイト・レイアウトの更新
- 8. サイトの公開
- 9. 新規テンプレートの作成

#### 前提条件

- フォルダおよびファイルをローカル・コンピュータに同期化するための Oracle Content Management デスクトップ・アプリケーションを設定して起動します。
- Bootstrap テンプレート・テーマのフォルダ、ファイルおよびコンテンツをローカル・コンピュータにダウンロードし、使用できるようにします。

#### サイトの作成

**Oracle Content Management** スタータ・テンプレートからサイトを作成します:

- サイド・ナビゲーションで、「サイト」をクリックします。
   既存のサイトのリストが表示されます。
- 2. 「作成」をクリックします。
- 3. 「サイトの作成」ダイアログで、サイトの基準として使用する JET スタータ・テンプレー トまたはスタータ・テンプレートを選択します。



4. ダイアログで、サイトの名前を入力します。この名前はサイト URL に使用されま す。文字、数字、アンダースコア()およびハイフン(-)を使用できます。スペース を入力した場合、ハイフンに自動的に置換されます。

テンプレート、テーマ、コンポーネント、サイトまたはサイト・ページについて、 authsite、content、pages、scstemplate_*、_comps、_components、 _compsdelivery、_idcservice、_sitescloud、_sitesclouddelivery、_themes、 _themesdelivery といった名前を使用しないでください。documents および sites は、サイト・ページには使用できますが、テンプレート、テーマ、コンポーネント またはサイトには使用しないでください。

### N-ト:

Oracle Content Management のサイト URL のパスでは、大文字と小文字 が区別されます。問合せまたはフラグメント文字列の大文字と小文字の 区別は、開発者によってカスタム・コードで管理されます。



- 5. オプションで、サイトの説明を入力します。
- 6. 準備ができたら、「作成」をクリックします。

進行状況バーに、新規サイト名と作成ステータスが表示されます。サイトが作成されると、サイトのリストに名前が表示されます。その初期ステータスはオフラインです。

リストから新規に作成したサイトをすばやく見つけるには、**「最終更新」**でリスト をソートします。直前に作成したサイトがリストの上位に表示されます。

#### テーマ・フォルダの同期化

スタータ・テンプレートからサイトを作成する場合、スタータ・テンプレートのテーマのコピーが作成され、サイト名の後ろにテーマ名を加えた名前が付けられます。たとえば、My_New_Site のテーマは My_New_SiteTheme です。

デスクトップ・アプリケーションを使用して、サイトのテーマ・フォルダおよびファ イルをローカル・コンピュータに同期化します。『Oracle Content Management でのド *キュメントのコラボレート*』の同期化の開始に関する項を参照してください。

これで、テンプレート・テーマのフォルダ階層およびファイルがローカル・コンピュ ータ上に表示されるようになります。次に例を示します:

```
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

#### 基本テーマの設定

- 1. ローカル・コンピュータ上の同期テーマ・フォルダに次の Bootstrap ファイルをコピー し、既存のファイルにオーバーレイします。
  - *theme_name*/layouts フォルダに入る html ファイル
  - *theme_name*/assets/cssフォルダに入る cssファイル
  - *theme_name*/assets/css フォルダに入る js ファイル
  - theme_name/assets/images フォルダに入る image ファイル(背景、フッター、人 などのイメージとともにサブフォルダ内にグループ化できます)
- layout フォルダ内の html ファイルを変更し、相対パスおよび必須要素を更新します。 一般的な Bootstrap テーマの場合、/assets フォルダに対する相対パスが多数存在する ため、テーマ・フォルダを指し示すようこれらを変更する必要があります。

css、js および images フォルダのパスを使用するために修正します:

```
_scs_theme_root_/assets/css/
_scs_theme_root_/assets/js/
_scs_theme_root_/assets/images/
```

### / ノート:

このステップが完了したら、_scs_theme_root 部分は、テーマが使用されている環境に合わせて自動的に調整されます。

- 3. Oracle Content Management のレイアウトごとに 3 つの要件があります:
  - a. html ファイルの<head>に次のタグを含めます:
    - <!--\$SCS_RENDER_INFO-->
    - <!--\$SCS_SITE_HEADER-->
    - <!--\$SCS_PAGE_HEADER-->



b. レイアウト・ファイルの末尾、<body>タグの内側にレンダラ・スクリプトを含めます。これらのパスは両方とも、サイト・ビルダーおよび実行時環境内で自動的に調整されます。

<script data-main="/_sitescloud/renderer/renderer.js" src="/
_sitescloud/renderer/require.js"></script>

- c. renderer.js ファイルのインクルードの後ろに次のタグを含めます:
  - <!--\$SCS_PAGE_FOOTER-->
  - <!--\$SCS_SITE_FOOTER-->

変更したテーマ・ファイルが Oracle Content Management のサイトと同期化されてい ることを確認します。

#### サイト・ページの更新

サイト・ビルダーでサイトを開くと、デフォルトのサイト内にあるページが最初に表示されます。不要なページを削除し、保持するページ上のレイアウトを新しいレイアウトの1つに切り替えることができます。

1. サイド・ナビゲーションで、「サイト」をクリックします。

既存のサイトのリストが表示されます。

- サイトを選択して、右クリック・メニューの「開く」を選択するか、アクション・ バーの をクリックします。
- 3. 更新の名前および説明(オプション)を入力し、「作成」をクリックします。

更新の名前には、文字、数字、アンダースコア()およびハイフン(-)を使用できま す。スペースを入力した場合、ハイフンに自動的に置換されます。

すでにサイトに対する更新がある場合、リストから更新を選択して します。

- サイト・ビルダーがプレビュー・モードで開きます。変更を加えるか、サイドバーのナビゲーション・オプションを使用する場合、「編集」スイッチがの「編集」に設定されていることを確認してください。
- 5. 特定のページを編集する場合は、サイドバーのサイト・ツリーを使用するか、サイト独自のナビゲーションを使用して、ページを選択します。
- 6. 不要なページを削除するには、ページを選択し、 をクリックします。
- 新しいページを追加するには、「ページの追加」をクリックします。ページをドラ ッグ・アンド・ドロップすることによってサイト・ツリー内でページの位置を変更 できます。
- ページに関連付けられたレイアウトを変更するには、サイト・ツリー内でページを 選択し、●をクリックし、ページ設定を表示します。

「ページ・レイアウト」フィールドに移動し、メニューから異なるレイアウトを選択します。ページ・レイアウトの数とタイプは、サイトに関連付けられたテーマによって異なります。

9. 現在の更新に変更を保存するには、「保存」をクリックします。現在の更新で作業 を続行するか、必要に応じて新しい更新を作成できます。



#### ナビゲーションの更新

サイト・プレビュー内のナビゲーションを参照すると、このナビゲーションは Bootstrap テーマのレイアウト内のハードコードされたナビゲーションが原因で現在の階層と一致していません。

ナビゲーションを更新し、Bootstrap テーマ内のハードコードされたコードを、サイト階層から動的に生成されたコードで置き換えます。

テーマ・ファイルのローカル同期コピーを編集します。

 レイアウトからハードコードされたナビゲーション・コードを削除します。一般的な Bootstrap テーマから削除する必要があるハードコードされたナビゲーションの例を次に 示します。次に示すのは一般的なヘッダー・セクションで、ロゴ、ページが狭すぎるとき の「Hamburger」メニューの「Toggle Navigation」部分(レスポンシブ部分)、および他の ページのハードコードされたページ・ナビゲーションがあります。

```
<header id="header" class="header navbar-fixed-top">
       <div class="container">
           <h1 class="logo">
              <a href="index.html"><span class="text">Velocity</
span></a>
          </hl><!--//logo-->
           <nav class="main-nav navbar-right" role="navigation">
              <div class="navbar-header">
                  <button class="navbar-toggle" type="button" data-</pre>
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</pre>
collapse">
                  class="active nav-item"><a</li>
href="index.html">Home</a>
                     <a
href="features.html">Features</a>
                     <a
href="pricing.html">Pricing</a>
                     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>
                         <a href="download.html">Download
Apps</a>
                             <a href="blog.html">Blog</a>
                             <a href="blog-single.html">Blog</a>
Single</a>
                             <a href="blog-category.html">Blog
Category</a>
                             <a href="blog-archive.html">Blog
```



```
Archive</a>
                           <a href="about.html">About
Us</a>
                           <a
href="contact.html">Contact</a>
                        <!--//dropdown--
>
                    <a
href="login.html">Log in</a>
                    class="nav-item nav-item-cta last"><a</li>
class="btn btn-cta btn-cta-secondary" href="signup.html">Sign Up
Free</a>
                 <!--//nav-->
             </div><!--//navabr-collapse-->
          </nav><!--//main-nav-->
       </div><!--//container-->
   </header><!--//header-->
```

JavaScript コードを作成することにより、サイト構造情報を横断し、ナビゲーション・コードを生成し、レイアウト上の JavaScript コードを含めます。次に例を示します:

<script type="text/javascript" src="_scs_theme_root_/assets/js/
navbar.js"></script>

3. 特定の CSS 内で想定されるマークアップと一致するように topnav.js ファイル の正確な出力を変更します。

#### サイト・レイアウトの更新

この時点で、サイトは機能しますが、レイアウト内に編集可能な領域(スロット)はあり ません。

1. レイアウト内に DIV 要素を配置または追加し、スロットとして指定します。

**スロット**は、クラス属性内に値 scs-slot を持つレイアウトの **DIV** 要素です。スロットごとに一意の id 属性を設定する必要があります。例:

<div id="slot-content1" class="scs-slot scs-responsive"></div>

スロットがブラウザ・ビューポイントのサイズに自動的に調整されるようにするに は、クラス属性 scs-responsive を含めます。

 必要に応じてこのステップを繰り返し、レイアウトまたは他のレイアウト内に追加 スロットを作成します。

#### サイトの公開

サイトに対するすべての変更を完了して保存したら、サイトを「**公開」**する必要があ ります。

サイトを公開すると、現在の更新内のすべての変更が採用され、ベース・サイトにマ ージされ、更新内の内容が新しいベース・サイトに格納されます。



第20章

#### 新規テンプレートの作成

これで、テーマが関連付けられた機能サイトが用意されたので、これをテンプレートにして 他のユーザーと共有し、これらのユーザーがテンプレートからサイトを作成できるようにし ます。

- 1. Oracle Content Management サイド・ナビゲーションで、「開発者」をクリックします。
- 2. 「すべてのテンプレートの表示」をクリックします。
- 3. 「作成」をクリックし、「既存のサイトから」を選択します。
- 4. スタータ・テンプレートおよび Bootstrap テンプレートを使用して作成した新しいサイト を選択します。
- 5. 新しいテンプレートの名前を入力し、「作成」をクリックします。
- 他の Oracle Content Management インスタンスと使用するためにテンプレートをパッケ ージ化するには、テンプレートを選択し、「エクスポート」メニュー・オプションを選択 し、ダウンロード可能な.zip ファイルを作成します。

### Developer Cloud Service によるテンプレートの開発

**Developer Cloud Service** を使用して、**Oracle Content Management** のテンプレートを開発できます。

次のステップを実行して、Developer Cloud Service 内でテンプレートを開発し、ローカルで テストし、Oracle Content Management にエクスポートします:

- 1. ローカル・マシンでの OCE ツールキットの設定。
- 2. Oracle Content Management 用の Developer Cloud Service コンソールへのサインイン.
- 3. Developer Cloud Service でのプロジェクトの作成。
- 4. 新規 GIT リポジトリ内のプロジェクト・コードへの OCE ツールキットの追加.
- 5. Developer Cloud Service でのテンプレートの作成.

開発する新しいテンプレートの作成、Developer Cloud Service での既存のテンプレート のコピーまたは Oracle Content Management からのテンプレートのインポートを行うこ とができます。

- 6. ローカル・テスト・ハーネスでのテンプレートのテスト.
- 7. 変更のマージ.
- **8.** Developer Cloud Service から Oracle Content Management へのテンプレートのエクスポート.

### Oracle Content Management 用の Developer Cloud Service コンソー ルへのサインイン

**Developer Cloud Service** コンソールで **Oracle Content Management** のカスタム・コンポーネ ントの開発を開始します。

**Oracle Cloud** サービスの管理者として、My Service Administration を使用してクラウド・サービスを作成および管理できます。Oracle Content Management のサービス・インスタンス管



理者および **Standard Developer Service** のサービス管理者の場合、設定して使用を開 始できます:

- 1. アカウントに提供された情報を使用して、Oracle Cloud にサインインします。
- 2. My Service Administration にサインインして、Oracle Content Management インス タンスと Standard Developer Service を作成および管理します。



- 3. 要望どおりに Oracle Developer Cloud Service 電子メールを確認します。
- サービスのサブスクリプション詳細を使用して Oracle Content Management イン スタンスを設定し、インスタンスの Oracle Content Management URL に移動しま す。
- 5. Standard Developer Service の URL にアクセスします。
- 6. Oracle Developer Cloud Service アカウントにサインインします。

Developer Cloud Service URL にアクセスし、コンソールにサインインします。

### Developer Cloud Service でのプロジェクトの作成

Content Experience Cloud プロジェクト・テンプレートを使用して Developer Cloud Service でプロジェクトを作成できます。または、空の Git リポジトリを使用してプロ ジェクトを作成し、OCE ツールキットを Oracle Content Management インスタンスか らインポートできます。

- Oracle Content Management テンプレートを使用した Developer Cloud Service プロジェクトの作成
- Oracle Content Management からの OCE ツールキットのダウンロードを使用した Developer Cloud Service でのプロジェクトの作成
- 新規 GIT リポジトリ内のプロジェクト・コードへの OCE ツールキットの追加

### Developer Cloud Service でのテンプレートの作成

cec コマンドライン・ユーティリティを使用して、使用可能なソース・テンプレートから Oracle Content Management テンプレートを作成できます。



cec create-template を使用して、使用可能なソース・テンプレートの1つからテンプレートを作成します。コマンドラインに cec create-template -h と入力すると、使用可能なソース・テンプレートが提供されます。

テンプレートの作成の例を次に示します:

cec create-template CafeSupremoLite_yourname -f CafeSupremoLite

Windows: このコマンドは、テンプレートの作成時に、外部の HTML WYSIWYG エディタで レンダリングするためにテーマのシンボリック・リンク(_scs_theme_root_など)を作成しま す。Windows でシンボリック・リンクを作成するには、通常、管理者権限でコマンドライン・ ユーティリティを実行する必要があります。テーマを編集するために WYSIWYG エディタ を使用していない場合は、管理者権限で実行する必要はなく、シンボリック・リンクの作成 エラーを無視できます。

上記の例では、CafeSupremoLite_yournameTheme テンプレートを作成し、ceccomponents/src/main/でソース・コードを使用できるようにしています。次の表に、 Developer Cloud Service でサイト・テンプレートを作成した後のソース・コードの場所を示 します。

テンプレート・ソース・コード	テーマ	テンプレートのコンポーネン	۲

cec-components/src/main/	cec-components/src/main/	cec-components/src/main/
templates	themes	components

ローカルのテスト・ハーネスは、テンプレートとタイプでフィルタリングする機能を備えた コンポーネントも示しています。

テキストやコード・エディタを使用して、テーマ・ファイルとコンポーネント・ファイルを 編集できます。ローカル・テスト・ハーネスを使用したテストを参照してください。テーマ やコンポーネントを編集した後、ブラウザをリフレッシュして変更を確認します。

### **日** 重要:

テンプレート、テーマ、およびコンポーネントのソース・コードは、src/main/に あります。Oracle Content Management ローカル・サーバーの機能に必要であるた め、src/main の外部のファイルは変更しないでください。

### Developer Cloud Service でのテンプレートのコピー

**Developer Cloud Service** に既存の Oracle Content Management テンプレートをコピーできます。

既存のテンプレートの1つを src/main/templates からコピーするには、cec copy-template コマンドを使用します。テンプレートに他のリポジトリのアセットが含まれる場合、オプシ ョンで、リポジトリ・マッピングを指定しますが、そうでない場合、これらのアセットはコ ピーしません。



次の例では、Temp1 テンプレートを Temp2 という名前の新しいテンプレートにコピー しています:

cec copy-template Temp1 -n Temp2

### Developer Cloud Service へのテンプレートのインポート

さらなる開発のためにテンプレートを Oracle Content Management から Developer Cloud Service にインポートできます。

Oracle Content Management サーバーから作成されたテンプレートの zip ファイルが ある場合、テーマまたはコンポーネントの編集など、さらなる開発のためにそのファ イルを Developer Cloud Service にインポートできます。次のコマンドを使用します。

cec import-template <location of the template zip file>

**Oracle Content Management**内に zip ファイルを含むフォルダを指定します。

「テンプレートについて」および「テンプレートのエクスポート」を参照してくださ い。

### 変更のマージ

コンポーネント、テンプレートまたはコンテンツ・レイアウトを作成した後、または ローカル・マシン上のソース・コードを編集した後、新規および変更済のコンポーネ ントとテンプレートをプロジェクトの GIT リポジトリにマージする必要があります。

変更を GIT リポジトリにマージするには、ターミナル・ウィンドウで次のコマンドを 順に入力します。

```
cd cec-components git pull
git add .
git status
git commit -a -m "Your comments" git pull
git push
```

### Developer Cloud Service からのテンプレートのエクスポート

**Developer Cloud Service** からテンプレートの zip ファイルをエクスポートし、このファイルを使用して Oracle Content Management 内でサイトを作成できます。

テンプレートの開発が完了したら、次のコマンドを実行してテンプレートをエクスポートできます。コマンド・レスポンスから、テンプレートの zip ファイルが Oracle Content Management 内のどこに作成されたかが分かります。

cec export-template CafeSupremoLite_yourname

「テンプレートについて」および「テンプレートのインポート」を参照してください。



## 21 テーマの開発

テーマでは、色スキーム、フォント・サイズ、フォント・タイプ、ページ背景など、サイト の一般的なルック・アンド・フィール(全体スタイル)を定義します。テーマは、サイト内のペ ージ間に視覚的な統一感を与えます。デザインおよびサンプル・コンテンツを指定して、一 意のテーマや様々なテーマを作成できます。これらを使用して、ブランドやビジョンを促進 するサイトを作成できます。

- テーマについて
- 基本テーマ構造
- サイト・ナビゲーション
- テーマの作成
- コンポーネントとテーマの関連付け
- サイト・レンダリング API

### テーマについて

テーマでは、コンテンツ、外観および動作を含む、サイトの一般的なルック・アンド・フィ ールを定義します。テーマは、サイト内のページ間に視覚的な統一感を与えます。

新しいテーマを定義することは、レイアウト、スタイル、サンプル・コンテンツ、ナビゲー ション、および新しいサイトの開始ポイントとして役に立つすべての基本情報を指定するこ とを意味します。テーマ・デザイナは、サイトの外観と動作方法に関する想定事項を設定し ます。テーマを設計する際は、たとえば、ほとんどまたはすべてのユーザーがモバイル・デ バイス内のサイトにアクセスすることが想定されるなど、テーマの用途を念頭に置く必要が あります。ユーザーが多くの類似サイトを作成することを求めている場合、カスタム・テー マを設計すると役に立ちます。テーマは、テーマ間で共有できるカスタム・パターンのペー ジ・レイアウトを使用して設計できます。



		Home	Products -	About	Contact	Privacy Policy
	ADD C	OMPON	ENT OR APP			
	Home Products - About	Contact	Privacy Poli	су		
About Us						
a . Mar 2 3	Our most important asset. Ou	r peop	ole.			
	We are proud of our employees and of the w commitment and dedication fuel our success	ork they . Our em	do. Their ployees are th	e		f У in 📴 🗖
	reason we have the highest customer satisfa	ction rati	ng in the indus	stry.		
	Opportunity is the key. We offer a flexible, ch that provides equal opportunity for all our en	allenging ployees	work environ to succeed.	ment		
	We invite you to join us and help us create p changing needs of a technology-driven work	roducts ti I.	o meet the			
			f У in	G+ 🖸		

テーマには、サイトのデザイン・コンテンツ、外観および動作を設計するために使用 できるページ・レイアウトが含まれています。デザインおよび設定を変更し、コンテ ンツを追加して、スタイル、ブランドおよびビジョンを売り込むサイトを作成します。

テーマの内容は次のとおりです。

- 背景イメージまたはページ・レイアウトの一部である他のコンテンツのアセット (イメージ、JavaScript ファイルなど)
- サイトのスタイル設定(CSS)
- 複数のページ・レイアウト(HTML ファイル)
- サイトのナビゲーションを作成するためのコード(JavaScript ファイル)
- コンポーネントとともに使用できる基本スタイルのリスト(design.css および design.json ファイルに指定されています)

テーマには、ページ・テンプレートの1つから作成される新しいページを移入するために使用されるシード・データを含めることもできます。たとえば、ユーザーが、製品セクション用の新しいページを作成し、new_product.htmlと呼ばれるページ・レイアウトを選択できます。new_product-pageseed.jsonと呼ばれるファイルがテーマに含まれる場合、新しいページには、初めて作成されるときにページ・シード・ファイルのコンテンツが移入されます。サンプル・コンテンツの場合と同様、このシード・データは、変更が可能で、ページを構築するための開始ポイントの役割を果たすことのみを目的として存在します。

テーマと連携して機能することを目的としたコンポーネントのサブセットを使用する テーマを作成できます。ユーザーがサイト用としてこのテーマを選択した場合、ユー ザーには、そのテーマに対して指定されたコンポーネントのみが表示されます。コン ポーネントとテーマの関連付けを参照してください。

Web サイトごとに1つのテーマを使用します。テンプレートからサイトを作成する と、テーマがテンプレートから継承されます。サイトのテーマはいつでも変更できま す。Oracle Content Management には、使用開始時に使用できるテーマを含む多数の テンプレートが用意されています。



サイトで新規の非公開テーマを使用している場合、サイトを初めてオンラインにした際に、 テーマはサイトとともに自動的に公開されます。テーマに変更を加え、オンライン・サイト を更新して変更を表示する場合は、テーマを明示的に公開する必要があります。テーマを明 示的に公開できるのは、テーマ所有者またはマネージャ権限を持つユーザーのみです。

🖉 ノート:

テーマへの変更を公開すると、テーマを使用するすべてのオンライン・サイトに変 更が反映されます。たとえば、テーマに指定したデフォルトのフォントを変更して テーマを公開すると、そのテーマを使用するすべてのサイトで新しいデフォルトの フォントが使用されます。

「テーマの管理」も参照してください。

### 基本テーマ構造

テーマの基本構造には、Oracle Content Management に格納されたフォルダに指定されたデ ザイン、ナビゲーションおよびスタイルが含まれます。テーマは、サイトのテンプレートの 一部です。

ユーザーが新しいサイトを作成するためにテンプレートを選択すると、関連付けられている テーマ・データが自動的にロードされます。JET スタータ・テンプレートまたはスタータ・ テンプレートを使用する場合、テーマは参照されるかわりに自動的にコピーされます。開発 者は、スタータ・テンプレートを使用している場合は、独自のテーマのコピーが必要です。

テーマは、基本的なフォルダおよびファイルを示すこの例のとおり、特定のフォルダおよび ファイル構造で編成されます:

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
```

特定のフォルダには、次のフォルダを含む特定のタイプの情報が含まれています:

 assets: JavaScript、カスケーディング・スタイル・シート(CSS)、イメージ、およびレイ アウトで参照されるその他のサポート・ファイル。



- designs: design.css および design.json ファイル。これらは、コンポーネントのスタイル・オプションを指定するために使用されます。
- layouts: ページ・テンプレートのための HTML ファイル。これらは、サイトのペ ージを表示するために使用されます。

```
✓ ノート:
★ベブの
```

すべての HTML ファイルで、<!DOCTYPE html>のような DOCTYPE 要素 がファイルの先頭にある必要があります

- publish: 公開されたサイト内でテーマが使用されると、このディレクトリがリストされ、このディレクトリにファイルのコピーが含まれます。このディレクトリは、デスクトップ・アプリケーションを使用してテーマを同期した場合に表示されますが、テンプレートの一部としてテーマをエクスポートした場合には含まれません。
- responsepages: エラー(404)を処理するための特別なページ。

```
✓ ノート:
サイト内のページにエラー・ページのフラグが付けられている場合、テ
ーマからの 404 エラー・メッセージは無視され、指定されたエラー・ペ
ージがかわりに使用されます。
```

- viewport.json: テーマのビューポート設定を指定します。
- _folder.json: テーマの名前と GUID を指定します。例:

```
{
    "themeName":"MarketingCampaignTheme",
    "itemGUID":"TB79D65F699B022AC4E11F4D4EE870070A1ADD86BBBB"
}
```

GUID は、テーマが最初にインポートされたとき、またはコピーされたときに、 Oracle Content Management によって作成されます。テーマ名は、テーマ作成時 に、テーマの開発者によって割り当てられます。

components.json:(非推奨。)テーマ内で使用されるカスタム・コンポーネントを記録します。下位互換性のためにのみ維持されています。

新しいテーマを作成する際には、2つの重要なファイルがあります。これらのファイルは、コンポーネントのスタイルを設定します:

- design.css
- design.json

ナビゲーション情報は1つの JavaScript ファイル(nav.js という名前のファイルなど)に格納することをお薦めします。テーマの/assets/js/フォルダは、そのようなファイルに適した場所です。



### サイト・ナビゲーション

サイトの階層は、サイトに関連付けられた structure.json ファイルに格納されます。階層 は、メモリーにロードされ、SCS.structureMap オブジェクトとしてページ・コンテキストで 使用可能になります。

サイト・ビルダーは、サイト・ビルダーでサイト・ツリーを描画するために structure.json ファイルを読み取ります。structure.json ファイルには、サイト・ページのコードが含ま れます。例:

```
"pages": [ {
    "id": 100,
     "name": "Home",
    "parentId": null,
     "pageUrl": "index.html",
     "hideInNavigation": false,
     "linkUrl": "",
    "linkTarget": "",
     "children": [ 200,
                   300,
                   400.
                   500],
     "overrideUrl":false
     }
     {
     "id":200,
     "name": "Products"
     "parentId":100,
     "hideInNavigation":false,
     "LinkUrl":"",
         "linkTarget":"",
         "children": [ 204, 205],
         "overrideUrl":false
     }
     "id":204,
     "name": "Hiking Boots",
     "parentId":200,
         "pageUrl": "products/hiking_boots.html",
         "hideInNavigation":false,
         "linkUrl":"",
     "linkTarger":"",
     "children":[],
     "overrideUrl":false
    }
```

また、この構造を読み取り、サイトのナビゲーション・リンクを描画するには、サイト・ペ ージ内にナビゲーション JavaScript コードも必要です。Oracle Content Management に用意 されているテンプレートには、この仕組みを説明するサンプルのナビゲーション JavaScript ファイルが含まれます。 Oracle Content Management に用意されているいくつかのテーマで使用される topnav.jsファイルは、SCS.structureMap オブジェクトをレンダリング API コール (SCSRenderAPI.getPageLinkData など)とともに使用して、サイト構造を横断する方法 や、ページ内のナビゲーション・リンクをレンダリングするために必要な HTML マー クアップを描画する方法の例です。サンプルの topnav.js ファイルのコードは次の とおりです:

```
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++)</pre>
                {
                    renderNode(navNode.children[c], navSub);
                }
                navItem.appendChild(navSub);
            }
            navBar.appendChild(navItem);
        }
    }
}
function renderNav()
ł
    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)
ł
    document.addEventListener('scsrenderstart', renderNav, false);
}
else if (document.attachEvent)
{
    document.documentElement.scsrenderstart = 0;
    document.documentElement.attachEvent("onpropertychange",
        function(event)
        {
            if (event && (event.propertyName == "scsrenderstart"))
            ł
                renderNav();
            }
        }
    );
}
```

レンダリング API コールを使用して、編集およびプレビューモードのサイト内および公開済 オンライン・サイトで機能するナビゲーション・リンクを生成できます。レンダリング API リファレンスを参照してください。

ナビゲーション情報は1つの JavaScript ファイル(topnav.js など)に格納することをお薦めします。JavaScript ファイルは通常、Oracle Content Management に用意されているサン プル・テーマの場合と同じように、テーマの/assets/js/フォルダに格納されます。

### テーマの作成

既存のテーマをコピーし、コピーに変更を加えることによって、新しいテーマを作成できま す。新しいテーマにブートストラップ・コンテンツをインポートして使用することもできま す。

### / ノート:

新しいテーマを作成するか既存のテーマを更新するかとは関係なく、常にテーマの コピーを作成し、そのコピーを操作してください。これをサンプル・サイトまたは 実際のサイトのコピーとともにテストし、正常に機能することを確認してください。 既存のテーマに対する変更は改訂されたテーマが公開された直後に実装されること に注意してください。



テーマのコピー

- ホーム・ページで、「開発者」をクリックします。
   「開発者」ページが表示されます。
- 2. 「すべてのテーマの表示」をクリックします。

既存のテーマのリストが表示されます。「表示」アイコンをクリックしてリストか らオプションを選択することにより、テーマがどのように表示されるかを制御でき ます。

3. テーマを選択して、右クリック・メニューの「**コピー」**を選択するか、アクショ ン・バーの をクリックします。

サンプル・ページやコンテンツを含め、テーマのフォルダとファイルがすべてコピ ーされます。

✓ ノート: テーマをコピーする場合、共有情報はコピーされません。

4. コピーしたテーマの名前を入力します。別のテーマで使用されている名前は使用 できません。

名前には、文字、数字、アンダースコア()およびハイフン(-)を使用できます。スペースを入力した場合、ハイフンに自動的に置換されます。

テンプレート、テーマ、コンポーネント、サイトまたはサイト・ページについて、 authsite、content、pages、scstemplate_*、_comps、_components、 _compsdelivery、_idcservice、_sitescloud、_sitesclouddelivery、_themes、 _themesdelivery といった名前を使用しないでください。documents および sites は、サイト・ページには使用できますが、テンプレート、テーマ、コンポーネント またはサイトには使用しないでください。

- 5. オプションで、テーマの説明を入力します。
- **6. 「コピー」**をクリックします。

進行状況バーに、新しいテーマ名とコピー・ステータスが表示されます。テーマが コピーされると、名前がテーマのリストに表示されます。テーマのリストのテーマ 名をクリックすると、テーマを構成するフォルダとファイルを検索できます。

Oracle Content Management デスクトップ・アプリケーションを使用して、テーマのフォルダとファイルをローカル・システムに同期化します。これにより、ローカル・フォルダを検索し、ファイルを直接操作できるようになります。テーマに対する変更は自動的に同期化されます。お気に入りの HTML、コードまたはテキスト編集ツールを使用して変更を行うことができます。

#### サイトでカスタム・スタイルを使用できるかどうかの設定

テーマ管理者は、テーマに伴うスタイルを指定したり、スタイルをカスタマイズした りできます。

テーマの components.json ファイル内の設定で、サイト・ビルダーでカスタム・スタ イル設定が行えるかどうかを指定します。これにより、サイト・コントリビュータが サイトの構築時にサイトのスタイルの範囲内にとどまり、他のフォントや色などを使 用しないようにするかどうかを管理します。



サイト内でカスタム・スタイルの使用を禁止するには:

 コンポーネントの「設定」パネルの「カスタマイズ」オプションを非表示にします。 テーマの components.json ファイルで、次のオブジェクトとともに他のコンポーネント 定義を追加します:

```
[
    {
        showStyleClassOnly": "true"
    }
]
```

2. CKEditor のツールバー・グループおよびボタンをカスタマイズし、カスタム・スタイル を削除します:

テーマの components.json ファイルで、**CKEditor** の toolbarGroups および removeButtons プロパティを指定し、Title および Paragraph コンポーネントのグループ およびグループ内のボタンをカスタマイズできます。

たとえば、次のオブジェクトとともに他のコンポーネント定義を追加し、Styles グルー プからスタイル、フォントおよび色を削除します:

```
[
    {
        "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, F
ont,TextColor,BGColor"
                 }
            },
             {
                 "type": "scs-paragraph",
                 "id": "scs-paragraph",
                 "config": {
                     "toolbarGroups": [
                         {
                              "name": "basicstyles",
                              "groups": ["basicstyles"]
                          },
                          ł
                              "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,Te
xtColor,BGColor"
}
}
]
```

3. スタイルが CKEditor に貼り付けられないようにします。 貼付けは、構成されているスタイルおよびボタンに従います。前の例では、貼り付けられ たデータから Font が削除されています。

カスタム・スタイルを禁止すると、次のような影響があります:

- すべての組込みコンポーネントで、style タブのカスタマイズ・オプションが非表示になります。
- CKEditor で、スタイルを設定するためのツールバー・ボタンが非表示になり、フォント・ ファミリ、フォントの色などの仕様がオーバーライドされます。
- また、CKEditor で、貼り付けたテキストを使用してスタイルを設定できなくなります。
   太字または斜体は使用できます。

#### テーマのコンポーネントおよびセクション・レイアウトの非表示

テーマの component.json ファイルを編集して、コンポーネントやセクション・レイアウトを 非表示にし、サイト開発者が使用しないようにすることができます。一貫性のあるルック・ アンド・フィールを奨励するために、この実行が必要になる場合があります。テーマのコン ポーネントおよびセクション・レイアウトの非表示を参照してください。

#### サイトでのテーマの確認

テーマの編集が完了した後、すべての部分がサイトで計画どおりに機能することを確認する 必要があります。

- コピーしたフォルダおよびファイルが Oracle Content Management デスクトップ・アプ リケーションに同期化されていること、および変更がすべて保存されていることを確認し ます。
- 2. サイト・ビルダーでサイト(テスト・サイトである場合や既存のオフライン・サイトであ る場合があります)を開き、新しいテーマを使用するようサイトを切り替えます。
- **3.** ページを追加し、設定を使用し、イメージを表示し、ナビゲーション、および新しいテー マで変更したすべての内容を確認することにより、このテーマでサイトをテストします。
- すべてのファイルおよびリンクを含むすべてが適切にレンダリングされることを確認します。
- 5. サイトにテーマの変更が実装されるようテーマを公開し、ランタイム環境で確認します。

テーマの公開を参照してください。

#### ブートストラップ・テーマの使用

Oracle Content Management のテーマとブートストラップ・テーマには類似点があるため、 既存のブートストラップ・テーマのページおよびコンテンツを変換して Oracle Content Management のテーマで使用可能にすることができます。



ブートストラップは、Web サイトや Web アプリケーションを作成するための無料かつ オープンソースのツール・コレクションです。これには、インタフェース・コンポー ネントと JavaScript 拡張機能用の HTML および CSS ベースのデザイン・テンプレー トが含まれます。

テーマを作成するために同じ手順に従いますが、(どのようなエディタを使用しようと も)コードを操作して、必要な任意のページまたはコンテンツをブートストラップ・テ ーマからインポートして編集します。

すべてのブートストラップ・テーマをそのすべてのページおよびコンテンツとともに 使用することはめったにありません。通常、Oracle Content Management のテーマで 使用するテーマの部分を選択します(おそらく、いくつかのページ・テンプレートのみ、 およびこれらのいくつかの部分のみ)。たとえば、ブートストラップ・テーマには、ヘ ッダー、ナビゲーション・ブロック、本文ブロックおよびフッター・ブロックを含む テンプレート内の複数のブロックが含まれます。ブートストラップ・テーマがファイ ル・システムから(Web サーバーなしで)直接表示できるファイルのコレクションとし て出荷される方法のため、ブートストラップ・テーマには、各ページ内に多数の複製 が含まれます(ページ・アセンブリ・エンジンが関連していないため、このように多数 の複製が含まれる必要があります)。Oracle Content Management のテーマでこの情 報を使用する場合、これらの部分を採用し、再使用可能な Oracle Content Management のページ・テンプレートに追加してから、Oracle Content Management を使用してこ れらのテンプレートから複数のページを動的にアセンブルする必要があります。

再使用可能なテンプレートに情報を追加する例の1つに、ナビゲーション・セクショ ンがあります。通常のブートストラップ・テーマ内のナビゲーションはすべてのペー ジで複製されますが、これを Oracle Content Management のテーマに移動する場合、 JavaScript コードを使用して、サイトの階層内を動的に移動し、ナビゲーション構造 を生成する必要があります。これにより、このスクリプトをすべてのページに含める だけで、すべてのページにナビゲーションが使用され、サイトでのページの追加また は削除にそのナビゲーションが適応します。

Bootstrap または Web サイト・デザイン・テンプレートからのサイト・テンプレート の作成を参照してください。

### テーマのコンポーネントおよびセクション・レイ アウトの非表示

デフォルトでは、すべてのコンポーネントとセクション・レイアウトが、サイトを作成するユーザーに表示されます。特定のテーマを使用する際に、コンポーネントやセクション・レイアウトを非表示にして、使用できないようにすることが必要な場合もあります。たとえば、ボタン・コンポーネントやスライダ・セクション・レイアウトを使用しない、一貫性のあるルック・アンド・フィールを奨励する場合には、テーマの components.json ファイルを編集して、テーマにそれらが表示されないようにすることができます。

テーマにコンポーネントやセクション・レイアウトが表示されないようにするには:

- **1.** Oracle Content Management で、左側のナビゲーション・メニューで「開発者」を クリックします。
- 2. 「開発者」ページで、「すべてのテーマの表示」をクリックします。
- 3. 「テーマ」ページで、変更するテーマを選択して、右クリック・メニューの「開く」 をクリックするか、アクション・バーの をクリックします。



- 4. components.json ファイルを選択して、「ダウンロード」をクリックします。
- 5. components.json ファイルをテキスト・エディタで開きます。
- コンポーネント・タイプと ID を指定するオブジェクトを追加し、プロパティ hidden を true に設定します。
   たとえば、次に示すのは、ボタン・コンポーネントを非表示にするサンプル入力です:

```
"type":"scs-button",
"id":"scs-button",
"hidden":true
```

同様に、次に示すのは、スライダ・セクション・レイアウトを非表示にするサンプル入力 です:

```
"type":"scs-sectionlayout",
"id":"scs-sl-slider",
"hidden":true
```

ボタン・コンポーネントとスライダ・セクション・レイアウトの両方を非表示にするため に components.json ファイルに入力する完全な内容は、次のようになります:

```
ſ
    {
        "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
            }
        1
```



1

}

**7.** 必要な変更を行ったら、components.json ファイルを新しいバージョンとしてテーマにアップロードします。

### テーマのコンポーネントの位置合せ、幅または間 隔オプションの非表示

デフォルトでは、ほとんどのコンポーネントではサイトの構築者がコンポーネントの 設定ダイアログで位置合せ、幅および間隔のオプションを指定できます。ただし、開 発者は、カスタム・コンポーネントでこれらのオプションを非表示にして、サイト・ テーマの一貫性がなくなるような方法でコントリビュータがコンテンツをレンダリン グするのを防ぐことが必要な場合があります。たとえば、ボタンを中央揃えにし、間 隔を 30 ピクセルに設定した一貫性のあるルック・アンド・フィールを促進する場合 は、オプションを非表示にして、テーマの components.json ファイルを編集すること で設定のテーマを変更するオプションを非表示にできます。

カスタム・コンポーネントの初期デフォルト値をオーバーライドしたい場合もありま すが、これは components.json ファイルを編集することで実行できます。初期値に対 する変更は、編集された components.json ファイルが新規バージョンとしてアップロ ードされた後でページに追加されたコンポーネントにのみ適用されます。

テーマのコンポーネントの位置合せ、幅または間隔の設定オプションを非表示にする には:

- **1.** Oracle Content Management で、左側のナビゲーション・メニューで「開発者」を クリックします。
- 2. 「開発者」ページで、「すべてのテーマの表示」をクリックします。
- 3. 「テーマ」ページで、変更するテーマを選択して、右クリック・メニューの「開く」 をクリックするか、アクション・バーの をクリックします。
- 4. components.json ファイルを選択して、「ダウンロード」をクリックします。
- 5. components.json ファイルをテキスト・エディタで開きます。
- コンポーネント・タイプと ID を指定するオブジェクトを追加し、非表示にするプロパティを true に設定します。
   たとえば、設定ダイアログのボタン・コンポーネントの位置合せ、幅および間隔のプロパティを非表示にするエントリを次に示します:

```
"type":"scs-button",
"id":"scs-button",
"hideAlignmentAndWidth": true,
"hideSpacing": true,
```

ボタン・コンポーネントのデフォルトの位置合せおよび間隔プロパティをオーバー ライドする場合、初期値の指定方法の例を次に示します:

```
"initialData": {
    "alignment": "center",
```

**ORACLE**^{*}

```
"marginTop": 30,
"marginBottom": 30
```

デフォルト値を指定し、テーマで使用されるボタン・コンポーネントの設定ダイアログで 位置合せ、幅および間隔オプションを非表示する components.json ファイルへの完全な入 力は次のようになります:

```
[
    {
        "name": "",
        "list": [
             {
                 "type": "scs-button",
                 "id": "scs-button",
                 "hideAlignmentAndWidth": true,
                 "hideSpacing": true,
                 "initialData": {
                     "alignment": "center",
                     "marginTop": 30,
                     "marginBottom": 30
                 }
            }
        ]
    },
        "name": "Starter",
        "list": [
             {
                 "type": "component",
                 "id": "StarterComponent",
                 "themed": true
             },
             {
                 "type": "component",
                 "id": "StarterFooter",
                 "themed": true
             }
        ]
    }
]
```

7. 必要な変更を行ったら、components.json ファイルを新しいバージョンとしてテーマにア ップロードします。

### コンポーネントとテーマの関連付け

コンポーネント、セクション・レイアウトおよびコンポーネント・グループをテーマに関連 付けることにより、テーマとともにコンポーネントの特定のサブセットを使用できます。

開発者として、マーケティング担当者が組織の製品を販売促進するサイトを作成するために 使用するテンプレートのコンポーネントおよびテーマを作成できます。コンポーネントをテ ーマに関連付けることにより、ユーザーがサイト・ビルダーで「テーマ・コンポーネント」 を選択するときにコンポーネントを使用できるようにすることができます。「テーマ・コンポ ーネント」タブでは、テーマに関連付けたコンポーネントのみがユーザーに表示されます。 テーマにコンポーネントを関連付けることにより、このコンポーネントが、サイトで 使用されていなくても、サイト・テンプレートとともにエクスポートされるようにし ます。

テーマにコンポーネントを関連付けるには:

- **1.** Oracle Content Management で、左側のナビゲーション・メニューで「開発者」を クリックします。
- 2. 「開発者」ページで、「すべてのテーマの表示」をクリックします。
- 3. 「テーマ」ページで、テーマを選択して、右クリック・メニューの「プロパティ」 をクリックするか、アクション・バーでクリックします。
- 「テーマ・コンポーネントをクリックし、「テーマ・コンポーネント」タブを開きます。
- 5. テーマと関連付ける1つ以上のコンポーネントを選択し、コンポーネントをカテゴ リ化します:
  - a. 「コンポーネントの選択」ドロップダウン・リストから、カスタム・コンポー ネントを選択します。
  - b. 選択する最初のコンポーネントについては、カテゴリの名前を入力し、コンポ ーネントのカテゴリを作成します。
  - c. 選択する追加コンポーネントごとに、既存のカテゴリを選択するか、新しいカ テゴリを作成します。
  - d. 「コンポーネントの追加」をクリックします。

Theme Properties	Theme Components List of components to be used with this	theme	
Theme Components	Buttons (new category)	AnchorButtons	•
	Buttons (new category)	Button-Headline	•
	Select a category or enter new 🕶	Select a component	•
	Add Component	1	
		AnchorButtons	^
		AnchorButtons_Light	
		AnchorLink	

- 6. 「保存」をクリックし、選択したコンポーネントをテーマに関連付けます。
- 「テーマ」ページで、テーマを選択し、「プロパティ」をクリックし、「テーマ・コンポーネント」タブを再度開きます。



8. 選択したコンポーネントが保存されたことを確認します。

テーマから関連付けられたコンポーネントを削除するには、「テーマ・コンポーネント」 タブの横の「X」をクリックし、「コンポーネントの削除」ダイアログの「はい」をクリ ックします。

テンプレートの「詳細」ページに、テンプレートが使用しているテーマに関連付けられたコ ンポーネントのリストが表示されます。テーマに関連付けられたコンポーネントに対する参 照は、テーマの components.json ファイルに格納されています。



テーマの関連付けを構成した後、Oracle Content Management にテンプレートをデプロイし、 マーケティング・チームと共有できます。マーケティング担当者がサイト用としてそのテー マを選択すると、あなたがサイト・ビルダー内の「テーマ・コンポーネント」タブでそのテ ーマに指定したコンポーネントのみが表示されます。「カスタム」および「すべて」コンポー ネントのリストにも、関連付けられたテーマのコンポーネントが含まれています。

#### OCE ツールキットでのテーマへのコンポーネントの関連付け。

OCE ツールキットでテーマにコンポーネントを関連付けるには、次のコマンドを使用できま す。コンポーネントは、サイト・ビルダー内の「テーマ・コンポーネント」タブのほか、「カ スタム」および「すべて」コンポーネントのリストにも表示されます。

cec add-component-to-theme <component>



次の **cec** コマンドを使用すると、テーマからコンポーネントの関連付けが削除されます。

cec remove-component-from-theme <component>

OCE ツールキットの詳細は、「OCE ツールキットでの開発」を参照してください。

### サイト・レンダリング API

Oracle Content Management のサイト・レンダリング API (SCSRenderAPI)は、Oracle Content Management のすべての Web ページに存在する window-global オブジェクト です。主に、ページのスロットやコンポーネントのレンダリングを行い、テーマ・レ イアウトに存在する JavaScript コードのインタフェースとなります。開発者としてテ ーマまたはコンポーネントの作業に当たっている場合に、これらのイベントや関数を 見かける可能性があります。runtime SCSRenderAPI は、ビューやプレビューの表示 モードをレンダリングし、design time SCSRenderAPI は、ナビゲーションや編集、 注釈の表示モードをレンダリングします。

実行時、サイト・レンダリング API には、次に示す3つの汎用的な目的があります:

- ページのスロットに、コンポーネントおよびコンテンツを移入する。
- ページ・レイアウトにある JavaScript コードの情報リクエストに対応する。
- ページレンダリング・プロセスのライフサイクル中にイベントを呼び出す。

設計時、サイト・レンダリング API には、次に示す 4 つの汎用的な目的があります:

- ページのスロットに、コンポーネントおよびコンテンツを移入する。
- ページ・レイアウトにある JavaScript コードの情報リクエストに対応する。
- ページレンダリングおよび編集プロセスのライフサイクル中にイベントを呼び出す。
- ページの編集や注釈を可能にするために、サイト・ビルダーとやり取りする。

サイト・レンダリング API は、テーマ・レイアウトに次のスクリプト・タグを配置することで、Oracle Content Management の Web ページにロードされます:

<script data-main="/_sitescloud/renderer/renderer.js" src="/_sitescloud/ renderer/require.js"></script>

これは通常、レイアウトの下部に配置されます。(このタグの URL は、設計時または実 行時の適切な環境に合せて自動的に調整されます。)SCSRenderAPI オブジェクトは非 同期でロードされます。カスタムの JavaScript コードは、scsrenderstart イベントを処 理することで、サイト・レンダリング API の可用性をリスニングできます。



# 22 レイアウトの開発

レイアウトは、ページ上にコンテンツを配置する方法を定義し、Oracle Content Management サイトで使用されるページの HTML を作成するために使用されます。

- レイアウトについて
- 検索エンジン最適化(SEO)
- components.json ファイルおよびフォーマットの理解
- サイト・ビルダーでのツールバー・グループのカスタマイズ
- スロット内のコンポーネントの制限
- レイアウト・コンテンツの編集可能化
- セクション・レイアウトの作成
- 遅延ロードをサポートするセクション・レイアウトの作成
- APIを使用したカスタム・セクション・レイアウトの開発
- コンテンツ・レイアウトの開発

### レイアウトについて

レイアウトは、どのようにコンテンツをサイト・ページに配列するかを定義します。異なる レイアウトに、異なる数の名前付きスロットを含めることができます。スロットはページの 幅に広がる領域で、1つ以上のタイプのコンテンツを含めることができます。

各テーマには複数のページ・レイアウトがあります。テーマについてを参照してください。

サイトにページを追加する場合、そのページに使用するレイアウトを選択します。各レイア ウトにはページ上の領域(スロットと呼ばれる)があり、コンテンツをドラッグ・アンド・ドロ ップできます。これらのスロットにどんなコンテンツを配置するかは、すべて自由に選択で きます。タイトル、テキストおよびドライバからマルチメディア、ギャラリおよびソーシャ ル・メディアまで何でも使用できます。

レイアウトには、有効な HTML コンストラクトに加え、Oracle Content Management レンダ ラによって認識される特別なマークアップが含まれます。レイアウトは、標準に準拠したモ ードでページをレンダリングするようブラウザを構成するために、DOCTYPE 文で始まる必 要があります。たとえば、<!DOCTYPE html>です。この文は、最良の結果を得るために特定の コンポーネントに必要となります。

次のサンプル・コードは、最小限のレイアウトを示しています:



```
7
            <link rel="stylesheet" type="text/css"</pre>
8
                    href="/_themes/[!--$SCS_THEME_NAME--]/assets/css/
main.css">
9
            <link rel="stylesheet" type="text/css"</pre>
10
                    href="/_themes/[!--$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
            <script data-main="/_sitescloud/renderer/renderer.js"
19
20
                    src="/_sitescloud/renderer/require.js"></script></script></script></script></script></script></script>
21
            <!--$SCS SITE FOOTER-->
22 </body>
23 </html>
```

ブラウザでページがレンダリングされるときには、様々なトークンが拡張されます。

• [!--\$SCS_THEME_NAME--]

これは、現在サイトのために選択されているテーマの名前を拡張します。このトー クンを使用すると、このトークンを使用する URL では現在のテーマが参照される ため、テーマをコピーできます。

• [!--\$SCS_DESIGN_NAME--]

これは、現在サイトのために選択されているデザインの名前を拡張します。これに より、レイアウトをテーマ内の複数のデザインによって使用できるようになりま す。

• [!--\$SCS_RENDER_INFO--]

これは、ページ階層、およびページの情報をレンダリングするコンポーネントを保 持する、スクリプト・タグを拡張します。これは、レイアウトの<head>セクション に配置する必要があります。

• [!--\$SCS_SITE_HEADER--]

これは、検索エンジン最適化(SEO)プロパティ内のヘッダー・フィールドに指定される、サイト・ヘッダー値を拡張します。すべてのページに配置する必要がある、 サイト全体に及ぶマークアップは、ここで入力できます。検索エンジンのプロパティの設定を参照してください。

• [!--\$SCS_PAGE_HEADER--]

これは、サイト・ビルダーの「ページ設定」プロパティの「ページ・ヘッダー」フ ィールドで見つかったページ・ヘッダー値に拡張されます。このページに配置する 必要があるページ固有のマークアップは、ここで入力できます。

• [!--\$SCS_SITE_FOOTER--]

これは、検索エンジン最適化(SEO)プロパティ内のフッター・フィールドにある、 サイト・フッター値を拡張します。すべてのページに配置する必要がある、サイト 全体に及ぶマークアップは、ここで入力できます。検索エンジンのプロパティの設定を参照してください。

✓ ノート: トークンでは、[!--\$および--]のかわりにデリミタとして<!--\$接頭辞および-->接 尾辞を使用することもできます。

サイトがオンラインの場合は、リンク内の次のトークンが、使用されているコンテキストを 認識している実際の値と置き換えられます。これにより、サイトが編集されている場合、お よびオンライン時に公開済サイト内にある場合に、リンクが機能できます。

- /_sitescloud/は、/_sitesclouddelivery/と置き換えられます
- /_themes/は、/_themesdelivery/と置き換えられます

スロットは、クラス属性内に値 scs-slot を持つレイアウトの DIV 要素です。スロットは、ユ ーザーがサイト・コンテンツへの入力のためにコンポーネントを追加できる場所です。"scsslot"クラス属性を割り当てることで、複数の DIV 要素をスロットとして指定できます。スロ ットごとに一意の id 属性を設定する必要があります。

🔊 ノート:

スロットはネストできませんが、**CSS**を使用してそれらをオーバーレイすることに よって、それらの提案を示すことができます。これを行うには、コンポーネント・ グループまたはセクション・レイアウトを使用します。セクション・レイアウトの 作成を参照してください。

"scs-responsive"というクラス属性値もあるスロットでは、ブラウザのビューポートで解像 度が変更されると、それらのコンテンツがリフレッシュされます。これにより、大きいデス クトップ・ディスプレイまたは小さいモバイル・デバイスに対応してスロットをレンダリン グできます。

最後の<script>タグ(例では 19 行目)は、Oracle Content Management レンダリング・コード をロードします。このコードは、ページでのコンポーネントの描画を担当するほか、カスタ ム・コードによるレンダリング API へのアクセスを可能にします。この<script>タグを使用 しない場合、レイアウトに基づくページをサイト・ビルダーで変更できません。

### 検索エンジン最適化(SEO)

検索エンジンがサイトのコンテンツを識別するためのキーワードを指定できます。

#### SEO 設定

検索エンジン最適化(SEO)設定は、サイト・レベルおよびページ・レベルで定義されます。 SEO のテキストは、すぐに使用可能なすべてのテンプレートで、フッターに入れられます。

検索エンジンのプロパティの設定を参照してください。

#### サイト・ビジターの Cookie

サイト・ビジターは、SEO のために Cookie を使用します。各ブラウザおよび各サイトで1つの Cookie が、請求目的で使用されます。 各 Cookie は毎時間または 24 時間 後に更新する必要があります。

各サイトには、ビジターに **Cookie** について通知するポップアップが含まれている必要 があります。

Footer	
	/
Text entered above will be displayed in the Cookie consent message.	
Oracle Content and Experience uses cookies for the purpose of identifying un user visists. To comply with EU Cookie Regulations your site should display the Cookie consent message and include a separate page explaining use of cooki the site to visitors. For more information about use of cookies in Oracle Conte Experience, see your Service Agreement.	ique e es on ent and

ポップアップには、次の情報が含まれています:

- SEO のヘッダーおよびフッター・テキスト
- テキストが選択されてリンクされる、特定の ID があるフッター内の Div

<div class=[iscs-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>

#### プライバシ・ポリシー・ページ

テキストは、忠告のテキストが含まれているプライバシ・ポリシー・ページにリンクされます。





#### 検索エンジン最適化の内部 HTML

コンポーネント・データをサイト・ビルダーに保存するときに、実行時にコンポーネントが レンダリングされた場合に生成される内部 HTML を保存できます。この内部 HTML はペー ジ・データに格納されるため、ページをレンダリングするときに、コンポーネントのかわり にデータをページに挿入できます。これは、ページ・レンディションの非常に早期に起こる 可能性があるため、検索エンジンがページ・コンテンツを正常にクロールする機会が増えま す。

### components.json ファイルおよびフォーマットの理解

テーマには、テーマで使用されるコンポーネントを指定する、/ThemeName/ components.json にある components.json ファイルが必要です。

このファイルには有効な JSON を含める必要があり、ファイルに最小限含める必要があるの は空の empty JSON 配列[]です。

components.json ファイル構文は、すべてのローカル・コンポーネントをリストし、コン ポーネントのカテゴリ化を完全にサポートしています。(リモート・コンポーネントはコンポ ーネント・カタログに登録されます。)

テーマレベルの components.json ファイルにどのようなコンポーネントを追加しようと も(none を含む)、Oracle Content Management では、ユーザーに使用可能なコンポーネント のデフォルト・セットが移入されます。このデフォルト・セットは、ソース・コードに定義 されます。次のリストは、コンポーネント、およびインライン・フレームでレンダリングさ れる(シード済)コンポーネントを示しています。また、サービス・レベルで登録され、インス タンス内でユーザーが使用できるリモート・コンポーネントが、サイト・ビルダーで使用可 能になります。

次のローカル・コンポーネントは、Oracle Content Management に含まれています。

名前	タイプ	ID
タイトル	scs-title	scs-title


名前	タイプ	ID
段落	scs-paragragh	scs-paragragh
イメージ	scs-image	scs-image
ギャラリ	scs-gallery	scs-gallery
ギャラリ・グリッド	scs-gallerygrid	scs-gallerygrid
ドキュメント	scs-document	scs-document
ボタン	scs-button	scs-button
マップ	scs-map	scs-map
区切り	scs-divider	scs-divider
スペーサ	scs-spacer	scs-spacer
YouTube	scs-youtube	scs-youtube
ソーシャル・バー	scs-socialbar	scs-socialbar
ビデオ	scs-video	scs-video
記事(カスタム・コンポーネント)	scs-component	scs-comp-article
ヘッドライン(カスタム・コンポ ーネント)	scs-component	scs-comp-headline
イメージおよびテキスト(カスタ ム・コンポーネント)	scs-component	scs-comp-image-text

インライン・フレームでレンダリングされるこれらのコンポーネントは、Oracle Content Management に含まれています。これらには、登録されたリモート・コンポー ネントは含まれません。

タイプ	ID
scs-app	会話
scs-app	ドキュメント・マネージャ
scs-app	フォルダ・リスト
scs-app	ファイル・リスト
scs-app	Facebook のいいね
scs-app	Facebook のおすすめ
scs-app	Twitter のフォロー
scs-app	Twitter の共有
	タイプ scs-app scs-app scs-app scs-app scs-app scs-app scs-app

#### 一般フォーマット

components.json ファイルの一般フォーマットは、次のとおりです:

- コンポーネントのプロパティは、各コンポーネント内に指定されます。上位レベルの"components"または"apps"プロパティは、非推奨です。
- 各コンポーネントには、"type"プロパティがあります。コンポーネントには、特定の値のみを使用できます(使用可能な値はすべて、デフォルト・コンポーネントの表に示されています)。

**ORACLE**[®]

- 各コンポーネントには"id"プロパティがあり、これは一意である必要があります。このプロパティを使用して、同じ"type"のコンポーネントを区別します。以前は、アプリケーションには"appName"プロパティがありました。"id"プロパティが使用可能でない場合、"appName"は依然として機能しますが、"appName"プロパティは非推奨です。
- 各コンポーネントには、ユーザー・インタフェース内の表示名である"name"プロパティがあります。フォールバック値が指定されていないときに、コンポーネントの場合、値は対応するデフォルト・コンポーネントの名前であり、リモート・コンポーネントの場合、値は ID です。

components.json ファイルの例を次に示します:

```
[
    {
        "name": "COMP CONFIG TEXT CATEGORY NAME",
        "list": [
             {
                 "type": "scs-title",
                 "id": "my-headline",
                 "name": "My Headline",
                 . . .
             },
             {
                 . . .
             },...
        1
    },
    {
        "name": "My own category name",
        "list": [ ... ]
    }
1
```

ー般構造は、カテゴリ・オブジェクトの JSON 配列です。各カテゴリ・オブジェクトには、 "name"プロパティおよび"list"プロパティがあります。"name"プロパティは、ローカライズ された文字列にマップするキーである場合があります。これらのデフォルトのカテゴリでは 十分でない場合、ローカライズされない独自のカテゴリ名を指定できます。次の表は、使用 可能なデフォルトのカテゴリおよび対応するキーを示しています。

+-	カテゴリ名(英語)
COMP_CONFIG_CONTENT_CATEGORY_NAME	コンテンツ
COMP_CONFIG_CUSTOM_CATEGORY_NAME	カスタム
COMP_CONFIG_MEDIA_CATEGORY_NAME	メディア
COMP_CONFIG_SOCIAL_CATEGORY_NAME	ソーシャル
COMP_CONFIG_TEXT_CATEGORY_NAME	テキスト

各カテゴリ・オブジェクトの"list"プロパティには、コンポーネント・オブジェクトの配列 が含まれます。各コンポーネントまたはオブジェクトには、"type"および"id"プロパティが 必要です。その他のプロパティはオプションです。

 "type"プロパティは、デフォルト・コンポーネント内にあるタイプの1つと等しい必要 があります。"type"がまだ存在しない場合、コンポーネントは表示されません。



- "id"プロパティは、コンポーネント全体にわたって一意である必要があります。
   "id"がすでに存在することが判明している場合、コンポーネントは表示されません。
- "name"プロパティは、ユーザー・インタフェース内のコンポーネントの表示名です。これにより、アプリケーション(現在はリモート・コンポーネント)の前の "appName"プロパティが置き換えられます。
- その他すべてのプロパティは、前のリリースの場合と同じように処理されます。

#### components.json への新規コンポーネントの追加

デフォルト・コンポーネントを変更することはできません。ただし、既存のデフォルト・コンポーネントに基づいて新しいコンポーネントを作成できます。たとえば、 "scs-title"コンポーネントに基づいて、デフォルト・テキストを設定する新しいコン ポーネントを作成できます。新しいコンポーネントを追加する最小限の要件は、 "type"および"id"プロパティを指定することです。

- "type"は、デフォルト・コンポーネント内にあるタイプの1つと等しい必要があります。"type"がまだ存在しない場合、コンポーネントは表示されません。
- "id"は、コンポーネント全体にわたって一意である必要があります。"id"がすで に存在することが判明している場合、コンポーネントは表示されません。

新しいタイトル・コンポーネントを追加するコードの例は、次のとおりです。このコ ンポーネントは、デフォルトのタイトル・コンポーネントとともに表示されます。

表示名とデフォルト・テキストを使用して新しいタイトル・コンポーネントを追加す るコードの例は、次のとおりです。

```
[
{
    "name": "COMP_CONFIG_TEXT_CATEGORY_NAME",
    "list": [
        {
            "type": "scs-title",
            "id": "my-headline",
            "name": "My Headline",
            "initialData": {
                "userText": "This is a second title component"
                }
        }
     ]
```



}

タイトル・コンポーネントはデフォルトのタイトル・コンポーネントのすべてのプロパティ をベースとして使用し、その上部にテーマレベルの変更を適用して新しいコンポーネントを 作成することに注意してください。

#### 下位互換性

前のフォーマットの components.json ファイルを依然として読み取ることができます。

- 上位レベルの"components"または"apps"プロパティを持つファイル。
- ファイルに"apps"プロパティが含まれる場合、このプロパティの下位のユーザー定義済リ モート・コンポーネントが依然としてロードされます。
- ファイルに上位レベルの"apps"プロパティが含まれる場合、その下位にリストされている 任意のリモート・コンポーネントのタイプが"scs-app"であることを想定してください。
- "appName"プロパティが存在する場合、"id"を"appName"値に設定します。表示名は "name"と同じになるか(指定されている場合)、"id"値にフォールバックします。

# サイト・ビルダーでのツールバー・グループのカス タマイズ

カスタム・コンポーネントの場合、サイト・ビルダー・ツールバーに表示されるすぐに使用 可能なツールバー・グループのコピーをカスタマイズできます。

限定セットを備え、カスタム・タグで使用できる、すぐに使用可能なコンポーネントの独自 のバージョンを作成できます。すぐに使用可能なコンポーネントにはどのようなカスタマイ ズも影響しません。

限定セットを備え、カスタム・タグで使用できる、すぐに使用可能なコンポーネントの独自 のバージョンを作成できますが、すぐに使用可能なコンポーネントには影響しません。

ツールバー・グループにより、タイトル・コンポーネントまたは段落コンポーネントを編集 するためにクリックしたときにツールバー内に表示される内容を定義します。タイトル・コ ンポーネントまたは段落コンポーネントによってサポートされる内容は削除および順序変更 できますが、追加プラグインは許可されていません。

サイト・ビルダー内のツールバー・グループをカスタマイズするには、CKEditor で toolbarGroups 構成に使用される構文と同じ構文を使用する必要があります。https:// ckeditor.com/latest/samples/toolbarconfigurator/index.html#basic を参照してください。

タイトル・コンポーネントまたは段落コンポーネントは、次のグループをサポートしていま す:

- "basicstyles" 太字/イタリック/下線に制限されます https://ckeditor.com/cke4/addon/ basicstyles
- "styles" フォント・スタイル https://docs.ckeditor.com/ckeditor4/latest/guide/ dev_styles.html
- "colors" テキストおよび背景色 https://docs.ckeditor.com/ckeditor4/latest/guide/ dev_colorbutton.html
- "undo" CKEditor 内の現在のインスタンスを元に戻す/やり直す



- "links" ダイアログをリンクするためのカスタム・プラグイン
- "paragraph" 箇条書き/番号付きリストおよびインデントのサポート
  - "list"
  - "indent"
- "align" 左揃え/右揃え/中央揃え
- "cleanup" 選択したテキストのスタイルの削除

段落コンポーネントは、image および table insert プラグインもサポートしています

- "insert"
  - "image"
  - "table"

また、行セパレータ・エントリを使用できます:

• "/"

🖋 ノート:

ツールバー・グループ構成に他の値を設定した場合、サイト・ビルダー・ツ ールバーが作成される前に値が削除されます。"extraPlugins"は指定できま せん。"name"/"groups"構成のみサポートされます。"items"エントリは無視 されます。

たとえば、ユーザーがフォント、色、スタイルまたはサイズを定義することを禁止す る場合、次のようにツールバー構成を更新できます。"id"値の場合、すぐに使用可能な 値とは異なるカスタム値を指定する必要があります。

```
[ {
    "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;"
        }
    }]
}]
```

#### 検証

components.json ファイルを前のコードに置き換え、サイトを編集します(すでに編集してい る場合はブラウザをリフレッシュします)。この時点では、タイトル・コンポーネントを編集 する場合、選択対象のフォント・スタイルまたは色は表示されなくなります。段落コンポー ネントでは引き続きこれらが表示され、使用可能なフォント・サイズのリストは 16、24 およ び 48 に制限されます。

#### デフォルトのツールバー・グループ

タイトルおよび段落のデフォルトのツールバー・グループは、次のとおりです:

• タイトル

```
[ {
        "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"]
    }
]
段落
[ {
        "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"]
    }
]
```

# スロット内のコンポーネントの制限

任意のレイアウト・スロットに対して、スロットで許可されるコンポーネントに特定 の制限を指定できます。

スロット内のコンポーネントを制限すると、許可されていないコンポーネントをドラ ッグしたユーザーに警告メッセージが表示され、そのスロットにコンポーネントを追 加または移動できなくなります。

この制限を構成するには、テーマ内のレイアウト(たとえば、レイアウト・ファイル themes*theme_name*\layouts\oneslot.htm)を編集し、カスタム・データ属性を スロットの div に追加します。



これは、カスタム属性のフォーマットです。前のバージョンからの主な相違点は、ユーザー がコンポーネントの id のみを指定する必要があることです。前はコンポーネントを制限す るために、ユーザーは冗長な"<type> <id>"構文(まだサポートされています)を使用する必要 がありました。

```
data-allowed-items='["<id>:","<type>,"<type>:<id>",...]'
data-disallowed-items='["<id>":"<type>","<type>:<id>",...]
```

### ♪ ノート:

data-allowed-items および data-disallowed-items の値には、二重引用符(')また は一重引用符(')を使用できます。次の例では、data-allowed-items では一重引用符 で JSON 配列が囲まれていますが、data-disallowed-items では二重引用符で JSON 配列が囲まれています:

```
<div id="slot101"
    class="scs-slot"
    data-allowed-items='["scs-app","scs-title"]'
    data-disallowed-items="['File List', 'scs-map']">
</div>
```

次の表は、Oracle Content Management で提供されているコンポーネントとそれぞれの ID を 示しています。ネーミングの競合を防ぐために、カスタマイズした(ローカルまたはリモート) コンポーネントの ID に scs-という接頭辞を*付けない*ようにするか、この表で示されている任 意のタイプまたは ID を使用してください。

名前	タイプ	ID
ドキュメント・マネージャ	scs-app	ドキュメント・マネージャ
Facebook のいいね	scs-app	Facebook のいいね
Facebook のおすすめ	scs-app	Facebook のおすすめ
ファイル・リスト	scs-app	ファイル・リスト
フォルダ・リスト	scs-app	フォルダ・リスト
Twitter のフォロー	scs-app	Twitter のフォロー
<b>Twitter</b> の共有	scs-app	<b>Twitter</b> の共有
ボタン	scs-button	scs-button
記事(カスタム・コンポーネント)	scs-component	scs-comp-article
ヘッドライン(カスタム・コンポ ーネント)	scs-component	scs-comp-headline
イメージおよびテキスト(カスタ ム・コンポーネント)	scs-component	scs-comp-image-text
コンポーネント・グループ	scs-componentgroup	scs-componentgroup
コンテンツ検索	scs-contentsearch	scs-contentsearch



名前	タイプ	ID	
コンテンツ・リスト	scs-contentlist	scs-contentlist	
コンテンツ・プレースホルダ	scs-component	scs-contentplaceholder	
コンテンツ・アイテム	scs-component	scs-contentitem	
区切り	scs-divider	scs-divider	
ドキュメント	scs-document	scs-document	
ギャラリ	scs-gallery	scs-gallery	
ギャラリ・グリッド	scs-gallerygrid	scs-gallerygrid	
イメージ	scs-image	scs-image	
マップ	scs-map	scs-map	
段落	scs-paragraph	scs-paragraph	
ソーシャル・バー	scs-socialbar	scs-socialbar	
スペーサ	scs-spacer	scs-spacer	
タイトル	scs-title	scs-title	
YouTube	scs-youtube	scs-youtube	

ユーザーは、ローカル・コンポーネントまたはリモート・コンポーネントを作成でき ます。この例で提供されている名前(My_Local_Component)は、スロット内で制限する ようこのコンポーネントを指定するために使用できる ID です。

Provide a name fo	r your component	
My_Local_Compo	nent	
Use only letters, numbers	, hyphens, and underscores in component names.	
Provide a descript	on for your component (optional)	
Add an optional c	escription for this component	
Component Type		
Default		•

# レイアウト・コンテンツの編集可能化

レイアウトに基づいてページを処理しているユーザーが編集できるように、レイアウト内の 特定のテキストまたはイメージ・コンテンツを構成できます。

この機能は、Bootstrapのテーマを含む任意のテーマで使用できます。変更では、元のページのスタイルが自動的に前提とされます。

レイアウト内の HTML タグ(、<h1>から<h6>まで、<div>および<img>)に簡易マークアップ を追加できます。

テキストおよびイメージのフォーマット・オプションには、次が含まれます:

- テキスト: ユーザーが太字、イタリック体、アンダースコアおよびリンクを指定できます。
- イメージ: ユーザーがイメージ・ファイルへのリンクを指定できます。タイトルのプロパ ティ(ユーザーがイメージの上にカーソルを置いたときに表示される内容)を変更し、アク セシビリティのための代替テキストを追加します。

テキストまたはイメージを変更してユーザーが編集できるようにするには、次の2つの作業 が必要です:

1. class に scs-editable を追加します。

2. 一意の id 属性を追加します。

レイアウトが変更されたら、(新しいページであっても)そのレイアウトに基づいてすべてのペ ージに対してこの機能が使用可能になります。

ページをコピーして貼り付けると、変更が新しいページにコピーされます。

#### ヘッダー・タグの変更

レイアウト内のヘッダー・タグを変更してユーザーが編集できるようにする方法の例を次に 示します。

- レイアウト・ファイルをローカル・デスクトップに同期化するか、<h1>ヘッダーの HTML ソース・ファイルを編集します。
- scs-editable を class に追加し、ヘッダーのコード行に属性 id="test-heading"を追加 し、これにエディタをアタッチできるようにします。例:

<hl class="brand-heading scs-editable" id="test-heading">Sample Heading Value</hl>

- 3. ファイルを保存します。
- 4. Oracle Content Management と同期化し、ブラウザを再ロードします。

ユーザーがヘッダーの上にカーソルを置くと黒い枠線がヘッダーの周りに表示され、コン テンツを編集できることを示します。ユーザーがヘッダーをクリックすると、枠線が緑色 になり、ユーザーがヘッダーのコンテンツを編集できるようになったことを示します。

5. テーマは公開されていない場合、公開します。公開されている場合、ブラウザをリフレッシュすると、変更が表示されます。

テーマが公開されたら、サイト・ユーザーがヘッダーをクリックし、サイト・ビルダー内で編集できます。



scs-editable 要素に疑似コンポーネントが作成され、サイト・ビルダー内で変更 してページ・データとともに格納できるようになります。実行時には、ページがレ ンダリングされる前に、scs-editable タグが、サイト・ビルダーに設定した値に置 き換えられます。

#### イメージ・タグの変更

ユーザーが編集できるようにレイアウト内のイメージ・タグを変更する手順は、テキ ストの場合の手順と似ています。

- 1. class に scs-editable を追加します。
- 2. 一意のイメージ id を追加します。

ユーザーがイメージをクリックし、異なるイメージを使用するためにプロパティを変 更できます。

ユーザーが編集できるイメージのサンプル・コードを次に示します:

<img class="scs-editable" id="test-image" src="_scs_theme_root_/
assets/img/downloads-bg-small.jpg"/>

# セクション・レイアウトの作成

サイト・ページ上のスロット内にコンテンツを配置するためにセクション・レイアウトを作成します。

エンタープライズ・ユーザーは、開発者として提供するセクション・レイアウトに基 づいてサイト上にコンテンツ・アイテムを配置できます。デフォルトのレイアウトか ら新しいセクション・レイアウトを作成できます。

セクション・レイアウトをエクスポートしてオフラインで変更し、その後新規セクシ ョン・レイアウトとしてインポートするか、既存のセクション・レイアウトを置き換 えることができます。セクション・レイアウトを個々にエクスポートするか、カスタ ム・コンポーネントおよびレイアウトを含むテンプレート・パッケージの一部として エクスポートできます。

次のすぐに使用可能なセクション・レイアウトを使用できます:

- 水平
- 2列
- 3列
- 垂直
- タブ付き
- スライダ

これらは、何も作成する必要なくサイト・ビルダー内で即時使用できます。

これらのセクション・レイアウトのファイルには、セクション/レイアウト・ファイル の構造に関する詳細とともにコメントが含まれます。コメントを表示するには、次の 手順で説明されているとおり、すぐに使用可能セクション・レイアウトに新しいセク ション・レイアウトを作成してから、編集のために新しいレイアウトをエクスポート します。

セクション・レイアウトを作成するには:



- 1. ホーム・ページで、「開発者」をクリックします。
- 2. 「すべてのコンポーネントの表示」をクリックします。
- 3. 右側の「作成」ドロップダウン・メニューから、「セクション・レイアウトの作成」を選 択します。
- 4. 「セクション・レイアウトの作成」ダイアログ・ボックスで、セクション・レイアウト・ コンポーネントの名前および説明を入力します。
- 編集のためにセクション・レイアウトをエクスポートするには、これを選択し、右クリック・メニューの「エクスポート」をクリックするか、アクション・バーのをクリックします。
  - a. 既存のフォルダに移動するか、「作成」をクリックして新しいフォルダを作成し、名前、および必要に応じて説明を入力します。
  - b. フォルダの横にあるチェック・ボックスを選択し、「OK」をクリックします。
  - c. フォルダのアイコンまたは名前をクリックして開きます。

選択したフォルダ内に、レイアウト・パッケージ・ファイルがセクション・レイアウト名 および.zip 拡張子を使用して作成されます。ファイルを開発環境にダウンロードして編 集します。

セクション・レイアウト API の詳細は、「API を使用したカスタム・セクション・レイア ウトの開発」を参照してください。

- 6. 変更したファイルを新しいセクション・レイアウトとしてインポートするか、既存のセク ション・レイアウトを置き換えます。
  - a. ホーム・ページで、「コンテンツ」タブをクリックし、「ドキュメント」をクリックします。
  - **b.** 変更されたセクション・レイアウトを、**.zip** 拡張子を持つファイルで、エクスポート した同じフォルダ名およびファイル名が含まれるフォルダにアップロードします。
  - c. 「開発者」ページで、「すべてのコンポーネントの表示」をクリックします。
  - d. 「作成」メニューから、「コンポーネントのインポート」を選択します。
  - e. 変更したセクション・レイアウトが含まれるアップロード済 zip ファイルの横にある チェック・ボックスを選択し、「OK」をクリックします。

変更したセクション・レイアウトが、選択したフォルダにインポートされます。

また、セクション・レイアウトをエクスポートして別の Oracle Content Management インス タンスにコピーまたは移動し、そこでインポートすることもできます。

# 遅延ロードをサポートするセクション・レイアウト の作成

コンテンツ・リストは、問合せの実行時に、追加コンポーネントとともにセクション・レイ アウトを呼び出すことができます。

コンテンツ・リストをレンダリングする際、セクション・レイアウトを選択して、返された すべてのコンテンツ・アイテムをレンダリングするオプションがあります。これにより、表、 スライダ、8列レイアウトなど、コンテンツ・アイテムに対して様々なレイアウトを作成でき ます。これらのカスタム・セクション・レイアウトは、さらに高度なページ区切り機能で使 用することもできます。 コンテンツ・リストでは、次のページ区切りがサポートされています:

- ページ区切り
- スクロール時にロード
- クリック時にロード

標準のページ区切り機能では、セクション・レイアウトは何も実行する必要がありま せん。ユーザーが次のページをクリックすると、次のアイテム・セットを使用して再 度レンダリングされます。ただし、「スクロール時にロード」および「クリック時にロ ード」については、セクション・レイアウトを再度レンダリングするのではなく、セ クション・レイアウトに追加のコンポーネントが追加されます。これは、主に無限ス クロール・モデルに使用されます。この場合、最初の n アイテムをロードし、ユーザ ーによってページがスクロール・ダウンされたときに次のアイテム・セットをフェッ チおよびレンダリングします。「スクロール時にロード」および「クリック時にロー ド」をサポートするには、カスタム・セクション・レイアウトでは次を実行する必要 があります

 render.js: addComponent() API を実装します。これは、セクション・レイアウト に追加される新規コンポーネントごとに呼び出されます。

 appInfo.json:次を含めることにより、セクション・レイアウトが addComponent() apiをサポートしていることをコンテンツ・リストに伝えることができます。

```
"contentListData": {
   "addComponent": true
},
```

appInfo.jsonを更新した後、ユーザーが設定パネルでこのセクション・レイアウトを 選択し、ページ区切り画面に移動すると、Load on click および Load on scroll オプ ションが表示されます。

# API を使用したカスタム・セクション・レイアウ トの開発

レンダリング API および編集 API が含まれるセクション・レイアウト API を使用して Oracle Content Management でカスタム・セクション・レイアウトを開発できます。

検討すべきスタータ・ファイルについては、「セクション・レイアウトの作成」を参照 してください。セクション・レイアウトのスタータ・ファイルには、セクション/レイ アウト・ファイルの構造に関する詳細とともにコメントが含まれます。



#### レンダリング API

render.js モジュールからロードするレンダリング API は、サイト・ビルダ内および実行時 に使用されます。

レンダリング API	説明	入力パラメータ	戻り結果
(コンストラクタ)	セクション・レイア ウトのレンダリン グ・モジュールを初 期化します。	<ul> <li>次のプロパティが含まれる JavaScript オブジェクト:</li> <li>sectionlayoutData (オブジェク ト): ページ・モデル内にあるセクシ ョン・レイアウト・データ。</li> <li>componentId (文字列): セクショ ン・レイアウトの componentId 値 (通常は GUID)。</li> <li>renderMode (文字列、オプショ ン): レンダリング操作のレンダリ ング・モード。</li> <li>customSettingsData (オブジェ クト): sectionLayoutData にあ る customSettingsData のコピ ー</li> </ul>	セクション・レイア ウトのレンダリン グ API は初期化さ れます。
レンダリング	セクション・レイア ウトに適した DOM 要素をペー ジに発行します(子 コンポーネント用 のコンテナ DIV を 含む)。	。 <b>container (要素)</b> : セクション・レイア ウトのマークアップのレンダリング先 にする必要がある DOM 要素。	このメソッドが返 された後、子コンポ ーネントがレンダ リングされます。 div[id]子要素を 見つけることによ り、子コンポーネン トを特定できます
addComponent	子コンポーネント をセクション・レイ アウトに動的にコン テンツ・リスト・コ ンポーネントとと もに使用します。 この機能はオプシ ョンです。	<b>container (要素)</b> : 新しいコンポーネ ントのレンダリング先にする必要があ る DOM 要素。 <b>componentId (文字列)</b> : セクション・ レイアウトに追加する新しいコンポー ネントの ID。	このメタデータが 返された後、 <b>ID</b> が componentId 入力 と一致する要素が レンダリングされ ます。

#### 編集 API

セクション・レイアウトに関連付けられた appinfo.json ファイルで hasEditHandlers プロ パティが true に設定されている 場合、edit.js モジュールがロードされます。

編集 API はサイト・ビルダーで使用されます。

コンストラクタを除くこのモジュール内のすべての関数はオプションです。



編集 API	説明	入力パラメータ	戻り結果
(コンストラクタ)	セクション・レイ アウトの編集モ ジュールを初期 化します。	次のプロパティが含まれる JavaScript オブジェクト: • componentId (文字列): セクシ ョン・レイアウトの componentId 値 (通常は GUID)。	セクション・レイ アウトの編集 API は初期化されま す。

編集 API	説明	入力パラメータ	戻り結果
getCapabilities	セクション・レイ アウトの編集機 能を説明するオ ブジェクトを返 します。	セクション・レイアウトの編集機能 を説明する JavaScript オブジェクト。 入力時に、デフォルトの機能が関数 に提供されます。この関数は、必要	(オブジェクト): セクション・レイ アウトの機能。
		に応じて機能オブジェクトを変更で きます。	
			機能オブジェクトには、次の機能が 含まれます:
		<ul> <li>title (文字列): ユーザーに表示 するセクション・レイアウトの タイトル。</li> </ul>	
		<ul> <li>settingsTitle (文字列): 設定パ ネル・ダイアログに表示するタ イトル。</li> </ul>	
		<ul> <li>hasSettings (ブール): セクション・レイアウトが「設定」パネルをサポートするかどうかを示</li> </ul>	
		します。 ・ allowMove (ブール): セクショ ン・レイアウトで子アイテムを	
		移動できるかどうかを示しま す。 <b>allowDelete (ブール)</b> : セクシ	
		ョン・レイアウトで子アイテム を削除できるかどうかを示しま す。	
		<ul> <li>isHidden (ブール): セクショ ン・レイアウトがユーザー・オ プションに対応して現在非表示 であるかどうかを示します</li> </ul>	
		<ul> <li>dropTarget (ブール): セクション・レイアウトがドラッグ・アンド・ドロップ操作のターゲットボスカバミカホニー キキ</li> </ul>	
		ト であるかとうかを示します。 ・ customMenuOptions (配列): 「セクション・レイアウト」コン	
		テキスト・メニューに追加する カスタム・メニュー・オプショ ン。各メニュー・オプション・	
		オフシェクトには、火のフロハ ティがあります: - label (文字列): メニュー・	
		アイテムの表示テキスト。 - action (関数): メニュー・ア イテムがクリックされたと	
		きに呼び出される関数。 - <b>disabled (ブール):</b> メニュ ー・アイテムを無効状態で	
		表示する必要があることを 示します。 - icon (文字列)・メニュー・ア	
		イテムのラベルの横に表示 する URL。(このプロパテ	

編集 API	説明	入力パラメータ	戻り結果
		<ul> <li>ィは将来的な使用に備えて 確保されています。)</li> <li>checkmark (ブール): メニ ュー・アイテムのラベルの 横にチェックマークを表示 する必要があることを示し ます。</li> <li>subMenuItems (配列): サ ブメニューに表示するメニ ュー・オプション。</li> </ul>	
getCaptionCont ent	UI 要素に表示さ れるセクション・ レイアウトの表 示名を返します。	なし。	(文字列): セクシ ョン・レイアウト の表示名。
filterCapabiliti es	メニュー ボコー ボコー ににセイアジェク や た う や た す で し イ ブ ジ で ま 変 に し イ ブ ジ で ま 変 に し イ ブ ジ で ま 変 に し イ ブ ジ で ま 変 に し イ ブ ジ で ま 変 に し イ ブ ジ で ま る 。 こ の <b>API</b> を う 。 の <b>API</b> を う 。 の <b>API</b> を う 。 の <b>API</b> を も ー 一 調 し て シ た ち る 。 こ の <b>API</b> を も ー 一 調 書 、 、 、 当 は 読 で ま 。 を 一 一 調 ま き き 。 ま 、 、 、 当 は に 、 、 当 は に 。 、 ろ こ 、 こ の <b>API</b> を も 一 … ま を き 。 、 。 、 、 当 は 読 き き 。 、 、 、 の 、 、 う に 、 、 。 、 。 、 。 、 。 、 、 、 、 、 、 、 、 、 、 、 、 、	セクション・レイアウトの編集機能 を説明する JavaScript オブジェク ト。入力時に、デフォルトの機能が 関数に提供されます。	(オブジェクト)セ クション・レイア ウトの機能。
onDragOver	ドラッグ・アンド・ロッグ・アンド・ロールされ、 ド・コールさたアイテムをセクシー イテム・レップをセクアウト にいたがでいた。 します。	<b>eventObject (イベント・オブジェク</b> ト): ドラッグ・イベントに関する情 報を保持するイベント・オブジェク ト。 <b>dataTransfer (DataTransfer オブ ジェクト):</b> セクション・レイアウト にドラッグされるアイテムに関する 情報を保持する DataTransfer オブ ジェクト。	(ブール)ドラッム ブールンドラッム ドラテム・ レアショでとう アクトーンでしていた アウムンけででした アウンントローン アウンントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクントン アクント アクント
onDrop	ドラッグ・アプ・アク・アプ・ドロップのクロップ処理のクロークのたいので、アクシンをでした。 ドレコークにファインをした。 ドインのたいので、 ドインのたいので、 「ののので、 「のののので、 「ののののので、 「のののののので、 「のののののので、 「のののののので、 「のののののので、 「ののののののので、 「ののののののので、 「のののののので、 「ののののののので、 「のののののので、 「のののののので、 「のののののので、 「のののののので、 「のののののので、 「のののののので、 「のののののので、 「のののののので、 「のののののので、 「のののののので、 「ののののののので、 「のののののので、 「ののののののので、 「のののののので、 「ののののののので、 「ののののののので、 「のののののので、 「ののののののので、 「ののののので、 「のののののので、 「のののののので、 「のののののので、 「のののののので、 「ののののので、 「ののののので、 「ののののので、 「のののので、 「ののののので、 「ののののので、 「ののののので、 「ののののので、 「のののので、 「のののので、 」ののので、 「ののので、 「のののので、 」のので、 「ののので、 」のので、 「ののので、 」のので、 「ののので、 」のので、 」のので、 「ののので、 」のので、 「ののので、 」のので、 「のので、 「のので、 」のので、 「のので、 」のので、 「のので、 」のので、 「のので、 」のので、 「のので、 」のので、 「のので、 」のので、 「のので、 「のので、 「のので、 「のので、 「のので、 「のので、 「のので、 「のので、 「のので、 「のので、 「のので、 「のので、 「のので、 「のので、 「のので、 「のので、 「のので、 「のので、 」」、 「のので、 「のので、 「のので、 「のので、 」」、 「のので、 」、 「のので、 「のので、 」、 「のので、 」」、 「のので、 」のので、 「のので、 」のので、 」のので、 「のので、 」のので、 」のので、 「のので、 」のので、 」のので、 」のので、 」のので、 」のので、 」のので、 」のので、 」のので、 」のので、 」、 、 、 、 、 、 、 、 、 、 、 、 、 、 、 、 、 、	<b>eventObject (イベント・オブジェク</b> ト): ドロップ・イベントに関する情 報を保持するイベント・オブジェク ト。 <b>dataTransfer (DataTransfer オブ</b> <b>ジェクト)</b> : セクション・レイアウト にドロップされるアイテムに関する 情報を保持する DataTransfer オブ ジェクト。	(ブール)ドロップ 操作がセクショ ン・レイアウトに よって処うかを示 す値。 <i>true</i> が返される 場合、デックはバ イパスされます。

編集 API	説明	入力パラメータ	戻り結果
onAddCompon ent	ドラッグ・アン ド・ドロップ操作 によってセクシ ョン・レイアウト にアイテムが追 加されたことを セクション・レイ アウトに通知し ます。	<b>eventObject (イベント・オブジェク</b> ト): ドラッグ・イベントに関する情 報を保持するイベント・オブジェク ト。 <b>dataTransfer (DataTransfer オブ ジェクト)</b> : セクション・レイアウト にドロップされるアイテムに関する 情報を保持する DataTransfer オブ ジェクト。	セクション・レイ アウト通知。
onMoveCompo nent	ドラッグ・アン ド・ドロップ操作 によってセクシ ョン・レイアウト	<b>componentId (文字列):</b> 新しく追加 されたアイテムの componentId 値。 <b>eventObject (イベント・オブジェク</b> <b>ト):</b> ドラッグ・イベントに関する情 報を保持するイベント・オブジェク ト。	セクション・レイ アウト通知。
	のアイテムが移 動されたことを セクション・レイ アウトに通知し ます。	dataTransfer (DataTransfer オブ ジェクト): セクション・レイアウト にドロップされるアイテムに関する 情報を保持する DataTransfer オブ ジェクト。	
		<b>componentId (文字列)</b> : 移動された アイテムの componentId 値。	
getSettingsDat a	「設定」パネルが 表示される前に セクション・レイ アウトで設定デ ータを変更する ことを許可しま す。	<b>settingsData (オブジェクト)</b> : セク ション・レイアウト用として計算さ れたデフォルトの設定データ	(オブジェクト)セ クション・レイア ウトの設定デー タ。
updateSettings	「設 っ ん か い か か か か か か か か か か か か か か か か か	<b>parameters (オブジェクト):</b> 「設 定」パネルから返される RAW パラメ ータ・オブジェクト。 <b>sectionLayoutData (オブジェク</b> <b>ト):</b> 格納されるセクション・レイア ウト・データ。デフォルトのデータ は、parameters (オブジェクト)か ら生成され、このパラメータの関数 に渡されます。	(オブジェクト)ペ ージ・モデル内に 格納するセクシ ョン・レイアウ ト・データ。

編集 API	説明	入力パラメータ	戻り結果
dispose	編が放デハ付ーすし こルに操うン完る合 モモ、ッドうれ割を これすの、レに要シ マリイチラれ割を ひれすの、レに要 と、関リ解許 コ、すよっと必ず がは直のり合クア描ある ですよったす場	なし。	セクション・レイ アウトの再描画 は有効です。

# コンテンツ・レイアウトの開発

コンテンツ・レイアウトは、サイト・ページで使用されているコンテンツ・リストま たはコンテンツ・プレースホルダー・コンポーネントを使用してコンテンツ・アイテ ムのデータを表示するのに役立ちます。コンテンツ・タイプの複数のコンテンツ・レ イアウトを作成して、異なるビューを作成したり、コンテンツ・アイテムの異なる部 分を表すことができます。

たとえば、Blog-Post コンテンツ・タイプでは、Blog-Post コンテンツが使用される方 法および場所に応じて異なるコンテンツ・レイアウトが必要である場合があります。 サイトのホーム・ページに Blog-Post アイテムのリストが表示される場合があります が、ホーム・ページでブログ投稿をクリックすると、「詳細」ページでそのブログ投稿 の詳細を表示できます。

「ホーム」ページには、Blog-Post-Summary コンテンツ・レイアウトをアイテム・ビュ ーとして使用して、Blog-Post コンテンツ・タイプのアイテムをリストするように構成 されたコンテンツ・リストがあります。

「詳細」ページでは、コンテンツ・プレースホルダーの Blog-Post-Header コンテンツ・ レイアウトを使用して、ヘッダー・イメージとタイトルを表示します。2 列のセクショ ン・レイアウトでは、Blog-Post-Content と Blog-Post-Author のコンテンツ・レイアウ トを使用して、70%と 30%の幅で 2 つのコンテンツ・プレースホルダーが囲まれます。 ご覧のとおり、4 つの異なるコンテンツ・レイアウトを使用して同じコンテンツ・タイ プをビジュアル化しています。

次の2つの方法のいずれかでコンテンツ・レイアウトを作成できます:

- Oracle Content Management で、「開発者」→「すべてのコンポーネントの表示」
   →「作成」→「コンテンツ・レイアウトの作成」を選択します。
- OCE ツールキット・プロジェクトで、cec create-contentlayout コマンドを使用 します。



Oracle Content Management は、コンテンツ・タイプのデフォルトのコンテンツ・レイアウトを作成します。デフォルトのコンテンツ・レイアウトを変更するには、次のファイルを編集します:

assets/layout.html

このファイルを編集して HTML ビューを変更します。

• assets/design.css

このファイルを編集してコンテンツ・レイアウトのスタイルを設定します。

• assets/render.js

このファイルを編集して、layout.html で使用されるデータを変更するか、コンテンツ・ レイアウトに動的な動作を追加します。

コンテンツ・レイアウト・コンポーネントは、Oracle Content Management サーバーからコ ンテンツ・アイテムをレンダリングします。ほとんどのアセットは Oracle Content Management サーバーに格納されています。場合によっては、スタイル設定目的のバックグ ラウンド・イメージなど、コンテンツ・レイアウト自体でローカルに使用できる静的アセッ トを使用することもできます。たとえば、次のコンテンツ・レイアウトでは、images/ background.jpg の絶対 URL を render.js で生成し、layout.html で使用することができま す。

絶対 URL を生成する最も簡単な方法は、組込み requirejs ファイルを使用することです。次のコードが示すとおり、依存性として require を定義し、require.toURL()を使用して URL を 生成します:

```
define([
```

```
'require',
'jquery',
'mustache',
'text!./layout.html',
'css!./design.css'
], function (require, $, Mustache, templateHtml, css) {
```

'use strict';

var imageURL = req.toUrl('./images/background.jpg');

次のトピックでは、コンテンツ・レイアウトを開発する方法について説明します:

- Oracle Content Management を使用したコンテンツ・レイアウトの作成
- コンテンツ・レイアウトへのレイアウト・ビューの受け渡し
- APIを使用したサイト詳細ページ URL の生成
- Developer Cloud Service を使用したコンテンツ・レイアウトのローカル開発
- コンテンツ・リスト問合せのマクロの展開
- 強力なコンテンツ・レイアウトの開発
- サンプル・ブログ・テンプレートの作成
- テンプレートへのコンテンツ・レイアウト・マッピングの追加
- ローカル・テスト・ハーネスを使用したコンテンツ・レイアウトのテスト



Oracle Content Management へのコンテンツ・レイアウトを含むテンプレートのインポート

#### 関連トピック

コンテンツ・レイアウトの管理方法の詳細は、「カスタム・コンポーネントおよびレイ アウトの管理」を参照してください。

サイトでのデジタル・アセットおよびその他のコンテンツ・アイテムの使用の詳細は、 『Oracle Content Management でのアセットの管理』のアセットの使用に関する項およ び「デジタル・アセットの管理」を参照してください。

# Oracle Content Management を使用したコンテンツ・レイアウトの作成

サイト・ページ上に表示するときにコンテンツ・アイテム内のフィールドをレイアウ トするためのコンテンツ・レイアウトを作成します。ページにコンテンツ・アイテム を追加すると、選択したコンテンツ・レイアウトが使用されます。

エンタープライズ・ユーザーは、開発者として提供するコンテンツ・タイプとレイア ウトに基づいてコンテンツ・アイテムを作成および使用できます。デフォルトのレイ アウトから新しいコンテンツ・レイアウトを作成できます。コンテンツ・タイプに関 連付けられた複数のコンテンツ・レイアウトを使用すると、サイト・デザイナが、コ ンテンツを変更せずにコンテンツ・アイテムを様々なコンテキストで表示できるよう になります。

コンテンツ・リストコンポーネント内でコンテンツ・レイアウトを使用する場合、コ ンテンツ・レイアウトはコンテンツ・アイテムごとに 1 回繰り返されます。これによ り、コンテンツ・レイアウトはセクション・レイアウトによって配置されます。

コンテンツ・レイアウトをエクスポートしてオフラインで変更し、その後インポート して既存のコンテンツ・レイアウトを置き換えることができます。

コンテンツ・レイアウトを作成するには:

Oracle Oracle Content Management のホーム・ページで、サイド・ナビゲーションの「開発者」をクリックします。

「開発者」ページが表示されます。

- 2. 「すべてのコンポーネントの表示」をクリックします。
- 3. 右側の「作成」ドロップダウン・メニューから、「コンテンツ・レイアウトの作成」 を選択します。
- 「コンテンツ・レイアウトの作成」ダイアログ・ボックスで、レイアウトを使用するコンテンツ・タイプを選択し、表示するフィールドを選択します。また、コンテンツ・アイテムおよびコンテンツ・リスト・コンポーネントをページに追加する際にサイト作成者がカスタム設定およびスタイルを追加できるようにする場合は、サイトで使用される場合にカスタム設定のサポートを追加するを有効にします。



٦

choose a conten	t Туре		
Create a content l	ayout to display items of this	s type	
Choose a con	ent type		•
Choose Fields to	Display		
Overview			•
Add support	for custom settings when u	sed in Sites	
Provide a name f	or your content layout		
<b>Provide a name f</b> Use only letters, n	or your content layout Imbers, hyphens, and under	rscores in content layout na	mes.
<b>Provide a name f</b> Use only letters, n Add a name for t	or your content layout umbers, hyphens, and under his content layout	rscores in content layout na	mes.
Provide a name f Use only letters, n Add a name for t Provide a descrip	or your content layout umbers, hyphens, and under his content layout tion for your content layo	rscores in content layout na ut (optional)	mes.
Provide a name f Use only letters, n Add a name for t Provide a descrip Add an optional	or your content layout umbers, hyphens, and under his content layout tion for your content layo description for this content	rscores in content layout na ut (optional) layout	mes.
Provide a name f Use only letters, n Add a name for t Provide a descrip Add an optional	or your content layout umbers, hyphens, and under his content layout tion for your content layo description for this content	rscores in content layout na ut (optional) layout	mes.

5. コンテンツ・レイアウト・コンポーネントの名前と説明を指定し、「**作成」**をクリックします。

コンポーネントにコンテンツ・レイアウトが追加されます。

✓ ノート: コンテンツ・レイアウト・タイトルでは英数字、ハイフンおよびアンダースコ アのみが有効です。

- 編集のためにコンテンツ・レイアウトをエクスポートするには、これを選択し、右クリック・メニューの「エクスポート」をクリックするか、アクション・バーのをクリックします。
  - a. 既存のフォルダに移動するか、「作成」、「フォルダ」の順にクリックして新しいフォ ルダを作成し、名前、および必要に応じて説明を入力します。
  - b. フォルダの横にあるチェック・ボックスを選択し、「OK」をクリックします。
  - c. フォルダのアイコンまたは名前をクリックして開きます。



Г

選択したフォルダ内に、レイアウト・パッケージ・ファイルがコンテンツ・レイア ウト名および.zip 拡張子を使用して作成されます。ファイルを開発環境にダウン ロードして編集します。

=	0						© ¤	0
asse	ts					Upload	Create 🝷	
Developer	> Components > Employee	-Card > assets	1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1				Name •	=
	design.css	v2	Just now	You	730 B	CSS	Ł	
	ayout.html	v2	Just now	You	534 B	HTML	₹	
	render.js	v2	Just now	You	4 KB	JS	₹	
	settings.html	v1	Just now	You	3 KB	HTML	₹	

これらのファイルは、従業員カード・コンテンツ・レイアウトを使用するコンテン ツ・アイテムのフィールドのレイアウトを制御します。サイトで使用される場合に カスタム設定のサポートを追加するを有効にした場合は、単一のコンテンツ・アイ テムのデフォルト・レンダリングを提供する settings.html という名前の追加ファ イルも作成されるため、それを表示できます。



Default Layout

Card Layout

Marketing Analyst

Redwood Shores Phone: (415) 555-5555



(415) 555-5555

```
7. design.css、layout.html および render.js ファイルを編集して、必要なコンテンツ・
   レイアウトを取得します。
   たとえば、次のファイルは従業員カード・コンテンツ・レイアウトを指定します:
   a. design.css ファイルを編集します:
      .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. layout.html ファイルを編集します:
   { { #data.employee_profile_pictureURL } }
   <div class="scs-tile-layout-img-container">
       <img class="scs-tile-layout-img"
   src="{{data.employee_profile_pictureURL}}" />
   </div>
   {{/data.employee_profile_pictureURL}}
   >
          <b>{{name}}
       
       {{data.employee_job_title}}
          {{data.employee_location}}
          <b>Phone: </b>{{data.employee_phone}}
       {{#scsData.detailPageLink}}
       >
          <a href="{{scsData.detailPageLink}}" title="Go to detail
   page"><span class="detail-page">Profile</span></a>
      {{/scsData.detailPageLink}}
   c. render.js ファイルを編集します:
   /* 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.customSettiings = 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;
});
```

render.js ファイルおよびその他のファイルの編集の詳細は、「コンポーネントの開発」 を参照してください。

- 8. 変更されたファイルをインポートして、既存のコンテンツ・レイアウトを置き換えます。
  - a. ホーム・ページで、「ドキュメント」をクリックします。
  - b. 変更されたコンテンツ・レイアウトを、.zip 拡張子を持つファイルで、エクスポート した同じフォルダ名およびファイル名が含まれるフォルダにアップロードします。
     新しいコンテンツ・レイアウトとしてインポートする場合は、_folder.json内のコ ンテンツ・レイアウトの GUID を変更する必要があります。
  - c. ホーム・ページで、「開発者」をクリックします。



「開発者」ページが表示されます。

- d. 「すべてのコンポーネントの表示」をクリックします。
- e. 「作成」メニューから、「コンポーネントのインポート」を選択します。
- 変更したコンポーネントが含まれるアップロード済 zip ファイルの横にあるチェック・ボックスを選択し、「OK」をクリックします。

変更されたコンテンツ・レイアウトは「**コンポーネント」**にインポートされます。

また、コンテンツ・レイアウトをエクスポートして別の Oracle Content Management インスタンスにコピーまたは移動し、そこでインポートすることもできます。

# コンテンツ・レイアウトへのレイアウト・ビューの受け渡 し

コンテンツ・レイアウトを開発する際に、コンテンツ・レイアウトがサイトで使用さ れている場合、基礎となるコンポーネントのプロパティを取得できます。

この特定のユース・ケースでは、コンテンツ・リストのコンテンツ・タイプに対して 選択されたコンテンツ・レイアウト・ビューで取得できます。次に、カテゴリの内容 に応じて、コンポーネントのレンダリング方法を変更できます。このプロパティへの アクセス権がない場合、実質的に同じ2つのコンテンツ・レイアウトを作成する必要 があります。

contentLayoutCategory プロパティは、コンテンツ・アイテムとコンテンツ・リスト・ コンポーネントの両方に対してレンダリングされるコンテンツ・レイアウトに対して 使用可能です。このプロパティには、次のようにサイト SDK を使用してアクセスでき ます。

scsData.SitesSDK.getProperty('contentLayoutCategory', function
(layoutCategory)

{ console.log(layoutCategory);});

「Oracle Content Management SDK」を参照してください。

このプロパティは、コンテンツ・レイアウトが、Oracle Content Management サイト 内にあるコンテンツ・アイテムまたはコンテンツ・リストからレンダリングされる場 合にのみ使用できます。コンテンツ・レイアウトがサードパーティ・アプリケーショ ンからレンダリングされる場合は、使用できません。

### API を使用したサイト詳細ページ URL の生成

独自の問合せからコンテンツ・レイアウト内にコンテンツ・アイテムのリストをレン ダリングする場合は、コンテンツ・アイテムの詳細ページへのリンクを作成できます。 サイト SDK の SCSRenderAPI.getPageLinkData API を使用して、「サイト詳細」ページ URL を生成できます。

detailPageId プロパティは、コンテンツ・アイテムまたはコンテンツ・リストで選択 した詳細ページの ID です。値が設定されていない場合、返される値は、isDetailPage プロパティが設定されている SiteStructureMap の最初のページです。

このプロパティにアクセスするには、サイト SDK を使用できます。これは、コンテン ツ・レイアウトが Oracle Content Management サイトで使用されている場合にのみ使



用できます。これには、scsData を介してのみアクセスできます。scsData は、コンテンツ・ レイアウトの作成時に引数の1つとして渡されます。例:

```
scsData.SitesSDK.getProperty('detailPageId', function (detailPageId)
{    console.log(detailPageId);});
```

detailPageId を使用したら、これを使用して詳細ページへのリンクを構築できます。

SCSRenderAPI には、pageId および追加オプションを取り、オプションを通過するページに 必要な URL を構築する関数 getPageLinkData()があります。この関数の署名は次のとおり です:

次のパラメータがあります:

- pageld: サイト SDK の detailPageId プロパティから返される detailPageId と同じです。
- ・ オプション:
  - contentType
  - Content-ID
  - contentName

戻り値は、次のプロパティを持つオブジェクトです:

- hideInNavigation
- href
- href

```
次の例では、すべてをまとめています:
```

```
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);
});
```

これにより、次のように出力されます:"/sites/{site}/{detailPageName}/{contentType}/ {contentId}/{contentSlug}"

pageId が詳細ページでない場合、コンテンツ値は URL に追加されません。

「Oracle Content Management SDK」を参照してください。



# Developer Cloud Service を使用したコンテンツ・レイアウト のローカル開発

OCE ツールキットを通じて Developer Cloud Service を使用すると、Oracle Content Management 用のコンテンツ・レイアウトをローカルで作成、編集、構成およびテストできます。

**Developer Cloud Service** を使用してコンテンツ・レイアウトを開発するための準備を するには、次のステップを実行します:

- 1. ローカル・マシンでの OCE ツールキットの設定。
- 2. Oracle Content Management 用の Developer Cloud Service コンソールへのサイ ンイン
- 3. Developer Cloud Service でのプロジェクトの作成。
- 4. 新規 GIT リポジトリ内のプロジェクト・コードへの OCE ツールキットの追加.
- 5. コンテンツ・レイアウトの作成

次のトピックでは、OCE ツールキットを使用して Developer Cloud Service でコンテンツ・レイアウトをローカルで開発する方法について説明します:

- Developer Cloud Service によるコンテンツ・レイアウトの作成
- RequireJS モジュールの定義
- コンストラクタ関数パラメータの構成
- コンテンツ・レイアウトのレンダリング
- Mustache テンプレート内のコンテンツ・レイアウトの編集
- 動的 **DOM** 操作の追加
- design.css ファイル内のスタイルの定義
- 参照アイテムの取得
- メディア **URL** の取得
- トリガーの呼出し
- 検索問合せを使用した検索ページへの移動
- マクロの展開とリッチ・テキストのレンダリング
- 「詳細」ページへのリンク

**Developer Cloud Service** テンプレートを使用してコンテンツ・レイアウトを開発した 後、プロジェクトの **Git** リポジトリを使用して変更をマージします。

### Developer Cloud Service によるコンテンツ・レイアウトの作成

cec create-contentlayout コマンドを使用して、**Developer Cloud Service** プロジェ クト内のコンテンツ・タイプのコンテンツ・レイアウトを作成できます。

**Developer Cloud Service** プロジェクトでコンテンツ・レイアウトを作成するには:

1. ターミナル・ウィンドウで、cec-components ディレクトリに移動します。



```
2. cec create-contentlayout を入力し、コマンドのオプションおよび例を確認します:
   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> Content layout style
     --style, -s
     --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. 次のコマンドを入力し、サーバー上で使用可能なコンテンツ・タイプを確認します:

cec list-server-content-types

 「Usage」に示されているとおり、コンテンツ・レイアウトは、Oracle Content Management サーバー内のコンテンツ・タイプ用として作成することも、cec-components/src/main/ templatesの下のテンプレート内のコンテンツ・タイプ用として作成することもできま す。たとえば、次のコマンドを使用すると、サーバー内のタイプ用としてコンテンツ・レ イアウトが作成されます:

cec create-contentlayout Blog-Post-Overview-Layout -c Blog-Post -r

次のファイルを編集すると、コンテンツ・レイアウトを変更できます:

• assets/layout.html

このファイルは、HTML ビューを指定します。Mustache テンプレート内のコンテンツ・ レイアウトの編集を参照してください。



• assets/design.css

このファイルは、コンテンツ・レイアウトのスタイルを指定します。design.css フ ァイル内のスタイルの定義を参照してください。

• assets/render.js

このファイルでは、layout.html で使用されるデータを指定し、コンテンツ・レイ アウトに動的動作を追加できます。RequireJS モジュールの定義を参照してくだ さい。

(レイアウトの作成時に-a オプションを使用して)カスタム設定を許可するレイアウト を作成した場合は、単一のコンテンツ・アイテムのデフォルト・レンダリングを提供 する settings.html という名前の追加ファイルも作成されるため、それを表示できま す。

### RequireJS モジュールの定義

render.js ファイルに RequireJS モジュールを定義できます。サイトは、JQuery、 Mustache、RequireJS Text Plugin、RequireJS CSS プラグインなどの依存関係をロー ドします。

```
define([
```

```
'jquery',
'mustache',
'text!./layout.html',
'css!./design.css'
], function ($, Mustache, templateHtml, css) {
```

Mustache テンプレート・システムを使用してレイアウトをレンダリングできます。

コンテンツ・レイアウトの assets/render.js ファイルには、次のプロパティがあります:

- RequireJS モジュールである必要があります
- JavaScript コンストラクタ関数を返す必要があります。サイトは、パラメータ・オ ブジェクトを渡してコンストラクタ関数を呼び出します。パラメータ・オブジェク トには、コンテンツ・アイテム・データと、レイアウトのレンダリングに必要な API があります。
- このコンストラクタ関数には、コンテンツ・レイアウトのレンダリングを処理する render(parentObj)メソッドが必要です。render()method に渡される parentObj オブジェクトにコンテンツ・レイアウト DOM オブジェクトを追加する必要があり ます。
- RequireJS モジュールは、JQuery、Mustache、RequireJS Text Plugin、RequireJS CSS プラグインなどの依存関係を使用できます。これらの依存関係はサイトによって読み込まれます。他のライブラリも使用できます。

# コンストラクタ関数パラメータの構成

サイトでコンストラクタ関数の新しいインスタンスが作成される場合、コンテンツ・ レイアウトの開発を支援するために、contentItemData、scsData および contentClient が含まれるパラメータが渡されます。



```
コンストラクタ関数のコード例を次に示します:
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;
```

コンストラクタ関数パラメータには、次のオブジェクトが含まれます:

- params.contentItemData: 名前、説明、ID およびデータを含むコンテンツ・アイテムが 含まれます。たとえば、コンテンツ・アイテム内のフィールド blogpost_title には、 params.contentItemData.data['blogpost_title']を使用してアクセスできます。
- params.scsData:: このオブジェクトは、サイト内からコンストラクタがコールされると きに情報を渡します。このオブジェクトは、サードパーティ・アプリケーションでレンダ リングされたコンテンツ・レイアウトには存在しません。このオブジェクトには、サイト SDK オブジェクト、トリガーを呼び出すためのメソッド contentTriggerFunction、およ び「詳細」ページ・リンクが含まれます。
- params.contentClient: これは、コンテンツ SDK から作成され、コンテンツ・レイアウトを呼び出すために使用される contentClient オブジェクトです。したがって、これは、コンテンツ・サーバーに適したパラメータを使用して構成されます。コンテンツ・サーバーに対して追加のコールを実行する必要がある場合、独自のオブジェクトを作成するかわりに、この contentClient オブジェクトを使用できます。このオブジェクトには、コンテンツ用のクライアント API が含まれます。API を使用すると、コンテンツ・アイテムとそのコンテンツ・タイプを問合せ、検索および取得できます。他のヘルパー API も用意されています。たとえば、expandMacros()を使用すると、リッチ・テキストで使用されるマクロを拡張できます。

# コンテンツ・レイアウトのレンダリング

ContentLayout の render(params)メソッドは、コンテンツ・レイアウトをテンプレートから レンダリングします。Mustache テンプレートはコンテンツ・レイアウトにデフォルトで使用 されますが、必要な任意のテンプレート・テクノロジを使用できます。



ContentLayout の render(params)メソッドは、次のコードを使用してデータでテンプ レートをレンダリングできます:

```
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);
}
```

params.contentItemData から作成された content オブジェクトに必要なデータを追 加できます。params.scsData のプロパティをこのオブジェクトにマージすることを お薦めします。これにより、テンプレートでもこのテンプレートを使用できるように なります。レンダリングされたテンプレートは、render()メソッドに渡される親オブ ジェクトに追加する必要があります。

# Mustache テンプレート内のコンテンツ・レイアウトの編集

デフォルトのコンテンツ・レイアウトが含まれる assets/layout.html ファイル内のデ フォルトの Mustache テンプレートを編集できます。

ブログをレンダリングする簡易テンプレートは、次のようになります:

```
<hl>
{{blogTitle}}
</hl>
```

### 動的 DOM 操作の追加

Mustache.render()がコールされ、テンプレートが親オブジェクトに追加された後、動 的 DOM 操作を render.js に追加できます。

たとえば、スタイルを動的に変更して、リスナーを添付できます:

```
// Dynamic DOM Manipulation can be done here
$('h1').click(function (event) {
    alert('Title is : ' + $(this).text());
});
```

### design.css ファイル内のスタイルの定義

design.css ファイルで、コンテンツ・レイアウト・テンプレートで使用される任意の スタイルを定義できます。

design.css は、RequireJS CSS プラグインを使用してモジュール定義にロードされます。



### 参照アイテムの取得

別のコンテンツ・タイプを参照する参照データ・フィールドを使用してコンテンツ・タイプ の参照アイテムを取得できます。

たとえば、ブログ投稿コンテンツ・タイプの Author フィールドは、作成者コンテンツ・タイ プへの参照です。ブログ投稿のコンテンツ・レイアウトでは、contentClient.getItems()が、 作成者参照アイテムの詳細を現在のブログ投稿アイテムに関連付けます。

# メディア URL の取得

contentClient.getRenditionURL()を使用して、イメージなどのデジタル・アセットのデフ ォルトのレンディションを取得できます。

thumbnail などの他のレンディションが必要な場合、contentClient.getItems()を使用して デジタル・アセットを取得し、item.data.renditions.default および item.data.renditions.thumbnail を参照できます。

# トリガーの呼出し

scsData.contentTriggerFunction(payload)を使用して、コンテンツ・レイアウトからトリ ガーを呼び出すことができます。

作成者名がクリックされたときにトリガーを呼び出す作成者コンテンツ・レイアウトの例を 次に示します:

```
{{#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}}
```

ペイロードは、現在選択されている作成者の検索問合せで、これは、ページ上の他のコンテ ンツ・リスト・アイテムがリスニングできます。



### 検索問合せを使用した検索ページへの移動

一般的なユースケースでは、コンテンツ・レイアウト内のリンクをクリックしたとき に動的検索問合せを使用して検索ページに移動します。

たとえば、コンテンツ・レイアウト内の「More articles from this author」リンクをク リックしたときに、検索ペイロードを渡して、「Authors」という名前の検索ページに 移動したいとします。これを実現するには次のコードを使用します。サイト・ページ 内で実行する場合、コンテンツ・レイアウト内でグローバル・オブジェクト scs およ び scSRenderAPI が使用可能であることに注意してください。

```
$('.more-from-author').click($.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++) {</pre>
        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));
```

同じページ内で同じコンテンツ・レイアウトを複数回使用することが想定される場合、\$('.more-from-author').click(…)のように、クラス・セレクタではなく CSS セレクタ内の一意の ID を使用することをお薦めします。



例:

# マクロの展開とリッチ・テキストのレンダリング

コンテンツ・アイテム内のリッチ・テキストにデジタル・アセットを埋め込むことができます。

このリッチ・テキストをコンテンツ・アイテム内に適切にレンダリングするために、リッチ・ テキスト・フィールドには contentClient.expandMacros() API が使用されます。これによ り、リッチ・テキスト内のデジタル・アセットへのすべての参照が解決されます。

data["starter-blog-post_content"] =
 contentClient.expandMacros(data["starter-blog-post_content"]);

Mustache をレンダリングに使用する場合、リッチ・テキストには HTML があるため、{{{ }}} を使用してリッチ・テキスト値をレンダリングする必要があります。変数の周りに{{{ }}} 使用する場合、Mustache では HTML がエスケープされません。

### 「詳細」ページへのリンク

「詳細」ページへのリンクは、scsData.detailPageLink を介して使用できます。

たとえば、「詳細」ページに移動してブログのタイトルをクリックしたときにブログの詳細を 表示したい場合、次のように「詳細」ページへのリンクを使用できます:

# コンテンツ・リスト問合せのマクロの展開

コンテンツ・リスト問合せ内で、ページの実行時に計算されるプロパティの値を定義し、最 近更新されたコンテンツを表示できます。

サイト内のコンポーネントのほとんどのプロパティは静的です。ユーザーは、コンポーネントのプロパティの1つに固定文字列または値を選択または入力しますが、これは、ページがいつどこで実行されるかとは関係なく変更されません。ただし、ページの実行時に計算されるプロパティの値は定義できます。これは、コンテンツ問合せで最近更新されたコンテンツを表示する際に役に立ちます。ユーザーは引き続き、「過去3日間」などの日付を入力できます。

Mustache JS 拡張機能を複数のプロパティに挿入できます。これらの文字列で参照される値は、ページの実行時に実行されるモデルから導出されます。すぐに使用可能なモデルにより、


コンテンツ REST API コール用としてフォーマットされた日付が処理されます。この モデルは、ユーザー要件を満たすために追加値を使用して拡張できます。

プロパティに入力できる文字列の例は、次のとおりです:

Content List component:

Additional Query String property: updatedDate gt "{{#content.date}}today - 3 days{{/ content.date}}"

日付に対するこの Mustache エントリが実行時に評価されることにより、実行されるタ イミングに応じて返される値が変更されるようにします(つまり、updatedDate gt "222018100206000000"に拡張されます)。この方法により、ユーザーが事前定義済の 値を入力する必要なく、複雑な日付文字列を構築できるようになります。

#### サポートされているコンポーネント・プロパティ

次のプロパティが Mustache JS テンプレート構文をサポートしています:

- コンテンツ・リスト
  - 追加問合せ文字列
  - 例:updatedDate gt "{{#content.date}}today 3 days{{/ content.date}}"
- タイトル/段落/テキスト
  - CKEditor を介して入力されたリッチ・テキスト
  - 例: "Content REST API format for date: {{#content.date}}now{{/ content.date}}"

### 🧨 ノート:

Mustache テンプレートのカスタム・モデルがない場合、タイトル/段落/テキ スト内の拡張機能はそれほど役に立ちません。ただし、追加問合せ文字列に 入力内容を確認することは非常に有効です。なぜなら、編集と表示を切り替 えるときに評価され、即時表示されるからです。

#### サポートされているコンポーネント構文

content.date オブジェクトは、すぐに使用可能なオブジェクトとしてサポートされて います。これには、today と now という 2 つのメイン・パラメータが採用されます。

today 値は、現在のブラウザ時間を採用し、本日の午前零時に変換し、この値を UTC 時間に変換します。

{{#content.date}}today{{/content.date}}は、本日の午前零時のブラウザ時間に拡張され、UTC 値に変換され、コンテンツ REST API 日付フォーマットに設定されます。例:

2220181008065959999



• これは、次によって補強できます:

```
today +/- [day | week | month | year]
```

- today 値はまた、加算または減算を行うと動作が変化します。減算すると、朝の時間が使用されます。加算すると、午前零時の時間が使用されます。例:
  - {{#content.date}}today 1 day{{/content.date}}の場合、昨日の1日の始まり まで拡張されます。
  - {{#content.date}}today + 2 days{{/content.date}}の場合、明後日の午前零時まで拡張されます。

now 値は、現在のブラウザ時間を採用し、調整なしで UTC 時間に変換します。

- {{#content.date}}now{{/content.date}}は、UTC 値に変換された現在のブラウザ時間 に拡張され、コンテンツ REST API 日付フォーマットに設定されます。
- now はまた、hour を使用して補強することもできます。このため、次のようになります:

now +/- [hour | day | week | month | year]

- 例:
  - {{#content.date}}now + 2 hours{{/content.date}} 現在から2時間後で、UTC 値に変換され、コンテンツ REST API 日付フォーマットに設定されます
  - {{#content.date}}now 1 day{{/content.date}} 昨日のこのブラウザ時間で、
     UTC 値に変換され、コンテンツ REST API 日付フォーマットに設定されます

#### サポートされているコンポーネント構文の使用

追加問合せ文字列でマクロ拡張機能を使用するときに、過去3週間のすべてのデータを返したい場合は、次のように入力します:

updatedDate gt "{{#code.date}}today - 3 weeks{{/code.date}}"

日付のみが返されるため、コンテンツ REST API コールを操作するために、静的値を入力す る場合のように問合せ文字列を構築するときに引用符を追加します。

#### **MustacheJS**

構文については、mustache.github.io/mustache.5.html にある Mustache JS テンプレート・ページを参照してください。

文字列の拡張時に実行される Mustache インスタンスに 1 つの変更が行われています。 Mustache には、{{ }}を使用する text 拡張機能と{{{ }}}を使用する html 拡張機能の両方 が用意されています。これらの 2 つの違いは、テキスト格納機能では文字列に対して HTML エンコードが行われる点です。つまり、値が a < b に拡張された場合、結果が a &lt; b にな ります。これは、URLの文字列を構築する際に求めていない動作です。ユーザーには HTML 拡張機能を使用するよう伝えることはできますが、{{{ }}}を使用する必要がある理由を説明 するなど、余計な作業であり、さらに多くの問題が生じます。

これを回避するために、Mustache は、{{ }}を使用するときに値をエスケープしないよう設定されています。つまり、{{ }}と{{{ }}}は両方とも同じように動作します。これにより、



任意の結果をエンコードすることが必要な場合もこの作業をユーザーのために残して おくことができます。

#### OOTB Mustache モデル

Mustache には、拡張のためにテンプレートに適用するモデルが必要です。前の例で は、{{#content.date}}は、すぐに使用可能なものとしてすでに定義されています。一 方、{{person}}などの新しいプロパティは開発者が追加できます。モデル内にない値 をユーザーが Mustache テンプレートに入力すると、その結果は空の文字列になりま す。このため、Hello {{person}}の場合、開発者が person をモデルに追加しない限 り、単に Hello に拡張されます。

使用されているモデル・オブジェクトは、SCSMacros と呼ばれるグローバル・オブジェ クトです。開発者は、このオブジェクトに別の任意のエントリを自由に追加できます。 このオブジェクトは、テンプレートが評価されるときに Mustache に渡されます。

すぐに使用可能なモデル・オブジェクトは現在、content.date オブジェクトのみをサ ポートしています:

```
{
   content: {
    date: <lambda implementation>
   }
}
```

#### カスタム Mustache モデル

サポートされているオブジェクトは、開発者が要件に基づいて拡張できます。このため、lastTwoDaysオブジェクトを導入し、拡張を単なる{{lastTwoDays}}に簡素化できます。

前の例で Hello {{person}}のようなものをサポートするためにモデルを拡張するに は、person オブジェクトを SCSMacros に追加する必要があります。これは、ページ が実行される前に行う必要があります。これは、ページ・レイアウト内で先頭にスク リプト・タグを追加することによって実現できます。例:

```
<script type="text/javascript">
window.SCSMacros = window.SCSMacros || {}; // define/get the SCSMacros
object
window.SCSMacros.person = "World";
</script>
この変更を行った後、Hello {{person}}テンプレートは Hello World に拡張されま
```

```
す。
```

オブジェクトに値を渡したい場合(Hello {{#person}}personId{{/person}}など)、 mustache lambda を実装し、実装内の値をラップおよび拡張する必要があります。

例:

```
<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>
```

この変更を行った後、Hello {{#person}}111{{/person}}テンプレートは Hello Small に拡張され、Hello {{#person}}222{{/person}}テンプレートは Hello Big に拡張されます。

✓ ノート:

Mustache 拡張機能は同期して実行されます。非同期値を取得する必要がある場合、 Mustache 拡張機能を実行しようとする前にこれらの値をモデル内で解決する必要 があり、これは現在サポートされていません。ただし、特別注文の実装は可能です。

### 強力なコンテンツ・レイアウトの開発

コンテンツ・レイアウトは、コンテンツ REST コールから取得する 3 タイプのレスポンス・ データに対して堅牢である必要があります:

- コンテンツ・アイテム: expand=all パラメータを使用する場合、参照が拡張され、テキスト・フィールドが大きくなります。
- コンテンツ・アイテム: expand=all を使用しない場合、参照は拡張されませんが、テキスト・フィールドは大きくなります。
- コンテンツ問合せの場合、参照は拡張されず、テキスト・フィールドは大きくなりません。

パフォーマンスを向上させるために、19.2.3 でリリースされた Oracle Content Management の時点で、アセット・データを取得するコンテンツ REST コールに expand=all パラメータは 含まれていません。expand パラメータは、Oracle Content Management に、参照されるすべ てのアイテムおよびレスポンス内の現在のアイテムをドリルダウンおよび取得するように指 示します。expand パラメータを使用して取得されたデータに依存するカスタム・コンテン ツ・レイアウトは、取得されたデータに参照フィールド値が含まれない事例を処理するため に更新する必要があります。

### コンテンツ・アイテムのレンダリング

コンテンツ・レイアウトを使用して、コンテンツ・アイテムをレンダリングします。これは コンテンツ・アイテム・データを受け取り、HTML にレンダリングし、ページに挿入します。

デフォルトでは、コンテンツ・レイアウトは、Mustache テンプレートを活用してコンテン ツ・アイテムをレンダリングしますが、これらは任意の JavaScript テクノロジに実装できま す。Mustache テンプレートをレンダリングする場合、データが特定のフォーマットになるこ



とが想定されます。コンテンツ・レイアウトの render.js ファイルでは、テンプレートに渡すモデルがそのフォーマットと一致していることを確認する必要があります。

コンテンツ・レイアウトは、複数のユースケースでレンダリングされます:

- Oracle Content Management のアセット管理 UI で使用される場合、データは、ユ ーザーが変更を保存する前にプレビューできるように「編集済」状態である場合が あります。
- Oracle Content Management サイトでコンテンツ・リストまたはコンテンツ・アイ テム内で使用される場合、データは、実行されているサイトに関する追加情報を使 用して補強されます。
- コンテンツ SDK の contentClient.renderLayout()コールを介して使用される場合、コンテンツ SDK のユーザーは、必要なデータは何であれコンテンツ・レイアウトに直接渡します。

パフォーマンスについては、必要なデータをすべて返す単一の問合せと複数の問合せ の間にトレードオフがあります。これにより、高速な初回問合せを使用して輪郭がで きるだけ速くレンダリングされ、後続の問合せを使用して領域はその後に埋められま す。どのモデルを選択するかは、データとユースケースによって異なります。

また、コンテンツ・レイアウトに渡されるデータは、データが取得された方法が原因 でフォーマットが異なる場合がありますたとえば、コンテンツ REST を拡張パラメー タと一緒に使用すると、別のコンテンツ・アイテムへのフィールド参照(個別またはグ ループとしてのいずれか)が戻されます。参照されたコンテンツ・アイテムが含まれな い場合は、追加の REST コールを行う必要があります。

すべてのケースを処理するには、コンテンツ・レイアウト開発者は、受信したデータ のフォーマットに関して柔軟であるよう努める必要があります。また、必要に応じて、 開発者は、追加データをフェッチし、レンダリング・テンプレートによって想定され るフォーマットにデータを強制的に変更できます。

### コンテンツ・レイアウトのデータ構造の標準化

コンテンツ・レイアウト開発者は、コンテンツ・レイアウトが受信するデータの構造 を標準化する必要があります。

すべてのデータが存在する場合は、コンテンツ・レイアウトでコンポーネントを単に レンダリングできます。データの一部が存在しない場合、コンテンツ・レイアウトで 問合せを追加する必要がある場合があります。いずれの場合も、コンテンツ・レイア ウトでは特定のデータ・フォーマットが想定されることはなく、レンダリングされる フォーマットにデータが強制的に変換されます。

データがすべて期待どおりであることを確認する必要があります。データが存在しない場合は、追加の問合せを行う必要があります。データから次のフィールドが欠落している可能性があります:

- 参照フィールドの fields エントリ
- 大きいテキスト・フィールド

コンテンツ・レイアウトは特定のコンテンツ・タイプ用に設計されているため、コン テンツ・レイアウトの開発者は必要なフィールドのリストを把握します。これらの各 フィールドについて、コンテンツ・レイアウトがレンダリングできるようにデータを フェッチする必要があります。欠落データをフェッチしてから完全なデータでレンダ リングする、または、すぐにレンダリングしてから欠落データをフェッチして空白を 埋める、という2つのオプションがあります。



#### オプション 1: 欠落データのフェッチおよび完全データでのレンダリング

約束を作成して必要なデータを取得し、すべての約束が戻ったときにレンダリングを続行し ます。

たとえば、対応するフィールドを持つ次のコンテンツ・タイプがあります:

- starter-blog-author
  - フィールド
    - * starter-blog-author_name テキスト・フィールド
    - * starter-blog-author_bio-テキスト・フィールド
- starter-blog-post
  - フィールド
    - * starter-blog-post_title テキスト・フィールド
    - * starter-blog-post_content 大きいテキスト・フィールド
    - * starter-blog-post_author starter-blog-author アイテムへの参照

```
コンテンツ・レイアウトには次のテンプレートがあり、想定されるフィールド値をレンダリ
ングします:
```

```
</div>
```

```
\{ \{ / \texttt{fields} \} \}
```

コンテンツ・レイアウトは、次の問合せからデータを使用してコールできます:

- expand を使用したアイテム問合せ 指定されたすべてのデータ
  - /content/published/api/v1.1/items/{id}?expand=fields.starter-blogpost_author&channelToken=8dd714be0096ffaf0f7eb08f4ce5630f
  - これは、テンプレート内のすべての値を正常に移入するために必要なデータのフォーマットです。他のいずれかの問合せを使用した場合は、データをフェッチしてこのフォーマットに変換する追加作業が必要です。

```
- "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": "..."
    }
}
```

 アイテム問合せ(expand なし) - 参照アイテム・フィールドが"starter-blogpost_author.fields "にありません:

```
- /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"
    }
}
```

SCIM 問合せ - 大きいテキスト・フィールド"starter-blog-post_content"がありません。参照アイテム・フィールド"starter-blog-post_author.fields"がありません:

```
- /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"
    }
}
```

これらのいずれかの問合せで一貫してレンダリングできるようにするには、コンテン ツ・レイアウトからの render.js を使用して、参照されるすべてのフィールドが展開 され、大きいテキスト・フィールドが存在することを確認する必要があります。

そうでない場合は、これらのバックアップを問い合せてデータを修正し、完全なデー タを使用してレンダリングする必要があります。

```
サンプルの render()関数:
```

```
render: function (parentObj) {
    var self = this,
        template,
        contentClient = self.contentClient,
        content = self.contentItemData;
```

```
var getRefItems = function (contentClient, ids) {
        // Calling getItems() 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
         referedFields = ['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
     referedFields.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++) {</pre>
        if(!fieldsData[largeTextFields[i]]) {
           // need to fetch this content item directly to get all the large
text fields
            fetchIDs.push(content.id);
            break;
     }
    // now we have the IDs of all the content items we need to fetch, get
them all before continuing
    getRefItems(contentClient, fetchIDs).then(function (referenceData) {
        var items = referenceData && referenceData.items || [];
        // add the data back in
        items.forEach(function (referencedItem){
            // check if it's the current item
            if(referencedItem.id === content.id) {
               // copy across the large text fields
               largeTextFields.forEach(function (fieldName) {
                   fieldsData[fieldName] = referencedItem.fields[fieldName];
                });
            } else{
                // check for any referenced fields
                for (var i = 0; i < referedFields.length; i++) {</pre>
                    if(referencedItem.id === fieldsData[referedFields[i]].id)
```

```
{
                        // copy across the fields values
                        fieldsData[referedFields[i]].fields =
referencedItem.fields;
                       break;
            }
        });
        // now data is fixed up, we can continue as before
        try{
           // Mustache
           template = Mustache.render(templateHtml, content);
             if(template) {
                $(parentObj).append(template);
             }
        } catch (e) {
            console.error(e.stack);
    });
}
```

オプション 2: 即座にレンダリングした後、欠落しているデータをフェッチして空白に 入力

パフォーマンスを向上するには、存在しない可能性があるアイテムを分離し、それら を2番目のパスでレンダリングします。これには2つの Mustache テンプレートが必 要です。最初のテンプレートが初回のレンダリングを実行し、「穴」をそのままにしま すが、この「穴」は、データが完成したときに2回目のレンダリングで埋められます。

この場合、「穴」に個別のテンプレートを使用するか、モデルに実際の値ではなくテン プレート・マクロを戻させることにより、複数のパスをサポートするよう Mustache テ ンプレートを設定する必要があります。どちらの場合も、データが取得されるまでこ れらの「穴」を非表示にしてから移入し、適切な UI アニメーションを使用してページ が「動き回り」すぎないようにする必要があります。

### サンプル・ブログ・テンプレートの作成

BlogTemplate サンプルは、コンテンツ・レイアウト機能を示します。

**Developer Cloud Service** でテンプレートを作成し、テンプレート内のコンテンツ・レ イアウトを調査し、検索機能をテストできます:

**1.** タイプ BlogTemplate のテンプレートを作成します:

cec create-template MyBlogTemplate -f BlogTemplate

BlogTemplate をコピーすることによって新しいテンプレート MyBlogTemplate が 作成されます。



- 2. http://localhost:8085/を開き、「テンプレート」、「MyBlogTemplate」の順にクリックしま す。
- 3. ブログ投稿のリストが表示されます。そのうち1つをクリックします。これにより、レン ダリング対象として3つの異なるコンテンツ・レイアウトが使用されている「詳細」が表 示されます。
- 4. この作成者の別の記事をクリックします。検索ページに移動します。
- 5. 作成者名をクリックします。トリガーが呼び出され、右側に記事が表示されます。
- 6. 「ホーム」ページに再度移動し、検索をテストします。

### テンプレートへのコンテンツ・レイアウト・マッピングの追 加

コンテンツ・レイアウトを作成したら、コンテンツ・レイアウト・マッピングを追加することにより、Developer Cloud Service プロジェクト内のローカル・テンプレートにコンテンツ・ レイアウトを追加できます。

cec add-contentlayout-mapping コマンドを使用して、テンプレートにコンテンツ・レイア ウト・マッピングを追加します。コンテンツ・レイアウトのベースとなるコメント・タイプ (-c)、およびマッピングの対象となるテンプレート(-t)が必要です。例:

cec add-contentlayout-mapping Blog-Post-Detail-Layout -c Blog-Post -t BlogTemplate

デフォルトのコンテンツ・レイアウト・マッピングは、デスクトップのデフォルト・スタイ ルです。-s <layoutstyle>オプションを指定し、名前別に異なるレイアウト・スタイル(概要 や詳細など)を指定できます:

cec add-contentlayout-mapping Blog-Post-Detail-Layout -c Blog-Post -t BlogTemplate -s Details

また、-mオプションを使用してモバイル用のマッピングを設定することもできます:

cec add-contentlayout-mapping Blog-Post-Detail-Layout -c Blog-Post -t BlogTemplate -m

### ローカル・テスト・ハーネスを使用したコンテンツ・レイア ウトのテスト

コンテンツ・タイプおよびコンテンツ・レイアウトのマッピングをテンプレートに追加した ら、ローカル・テスト・ハーネス内のコンテンツ・レイアウトをテストできます。

ローカル・テスト・ハーネスを使用したテストを参照してください。

### ローカル・テスト・ハーネスを使用したテスト

ローカル・テスト・ハーネスでのカスタム・コンポーネント、テンプレートおよびコンテン ツ・レイアウトを、Oracle Content Management にインポートする前に実行します。 ローカル・テスト・ハーネスを開始するには:

- 1. ターミナル・ウィンドウに cd cec を入力します。
- 2. cec develop &または cec develop --server <server-name> &を入力します
- http://localhost:8085 でブラウザを開き、ローカル・テスト・ハーネスで実行されているコンポーネント、テンプレートおよびコンテンツ・レイアウトを表示します。
- 4. 次のディレクトリ内のコンポーネント、テンプレート、テーマなどを検索できます:
  - cec/src/main/components
  - cec/src/main/templates
  - cec/src/main/themes

## Oracle Content Management へのコンテンツ・レイアウトを含むテンプレートのインポート

コンテンツ・レイアウトを開発およびテストした後、コンテンツ・レイアウトが含ま れるテンプレートを Developer Cloud Service プロジェクトからエクスポートし、この テンプレートを Oracle Content Management にインポートできます。

テンプレートのエクスポートを参照してください。



Oracle Content Management サイトで使用するコンポーネントを開発できます。

- コンポーネントについて
- コンポーネントの開発について
- コンポーネントの作成
- Developer Cloud Service によるカスタム・コンポーネントの開発
- 多言語サイト用の翻訳可能コンポーネントの開発
- 「設定」パネルを使用した H1 コンポーネントの構築
- HTML テンプレートを使用した簡易コンポーネントの作成
- ローカル・コンポーネントとリモート・コンポーネントの比較
- コンポーネント設定のレンダリング
- ローカル・コンポーネントの実装
- コンポーネントのスタイル・クラス
- 組込みコンポーネントのスタイリング方法
- コンポーネント・プロパティの設定
- インライン・フレームでレンダリングされたコンポーネント
- インライン・フレームでレンダリングされたコンポーネントのインスタンス ID および構造について
- リモート・コンポーネントのセキュリティ
- リモート・コンポーネントの登録
- コンポーネントの削除
- サイト SDK

### コンポーネントについて

コンポーネントは、サイト内のページに追加できる特定のタイプのコンテンツです。Oracle Content Management 内のコンポーネントには、段落、タイトル、イメージ、区切りなどのア イテムが含まれます。

Oracle Content Management では、次のタイプのコンポーネントがサポートされています:

- ローカル・コンポーネント: ファイルは Oracle Content Management に格納されます
- リモート・コンポーネント:ファイルはリモート・サーバーに格納されます

ローカル・コンポーネントは、ページ内で直接レンダリングしたり、ページ内のインライン・ フレーム内でレンダリングしたりするよう設定できます。リモート・コンポーネントは常に、 インライン・フレーム内でレンダリングします。



Oracle Content Management には、デフォルトのコンポーネント・セットとともに各 テンプレートが用意されています(また、テンプレートにはテーマとサイトが含まれま す)。新しいコンポーネントを作成し、コンポーネント内でこれらのデフォルトのコン ポーネントを使用することもできます。サイトにコンポーネントを追加したら、ペー ジ・コンテンツ、フォントとフォント・サイズ、イメージのフレームと配置、および 他のスタイルを指定することにより、要件に合せてコンポーネントのプロパティ設定 を編集できます。編集できる設定の内容は、コンポーネント・タイプによって異なり ます。

これらのコンポーネントは、Oracle Content Management に含まれています。

名前	タイプ	ID
タイトル	scs-title	scs-title
段落	scs-paragragh	scs-paragragh
テキスト	scs-title	scs-text
イメージ	scs-image	scs-image
ギャラリ	scs-gallery	scs-gallery
ギャラリ・グリッド	scs-gallerygrid	scs-gallerygrid
ドキュメント	scs-document	scs-document
ボタン	scs-button	scs-button
マップ	scs-map	scs-map
区切り	scs-divider	scs-divider
スペーサ	scs-spacer	scs-spacer
YouTube	scs-youtube	scs-youtube
ソーシャル・バー	scs-socialbar	scs-socialbar
記事(カスタム・コンポーネン ト)	scs-component	scs-comp-article
ヘッドライン(カスタム・コン ポーネント <b>)</b>	scs-component	scs-comp-headline
フォルダ・リスト	scs-app	フォルダ・リスト
ファイル・リスト	scs-app	ファイル・リスト
ドキュメント・マネージャ	scs-app	ドキュメント・マネージャ
Facebook のいいね	scs-app	Facebook のいいね
Twitter のフォロー	scs-app	Twitter のフォロー
<b>Twitter</b> の共有	scs-app	Twitter の共有
Facebook のおすすめ	scs-app	Facebook のおすすめ
イメージおよびテキスト <b>(</b> カス タム・コンポーネント <b>)</b>	scs-component	scs-comp-image-text

*Oracle Content Management でのサイトの構築*のページ・コンテンツの配置を参照してください。

### コンポーネントの開発について

独自のカスタムコンポーネントを開発することにより、選択したページ・テクノロジを使用 してサイト・ページ内に埋め込むことのできる複合要素を作成できます。これにより、Oracle Content Management で提供されるコンポーネントのリストを効果的に拡張できます。

登録されたすべてのコンポーネントのエントリは、コンポーネント・カタログに格納されま す。これは、登録されたすべてのコンポーネントのエントリを含む、Oracle Content Management 内のフォルダです。

テンプレート、テーマ、コンポーネント、サイトまたはサイト・ページについて、authsite、 content、pages、scstemplate_*、_comps、_components、_compsdelivery、_idcservice、 _sitescloud、_sitesclouddelivery、_themes、_themesdelivery といった名前を使用しないでく ださい。documents および sites は、サイト・ページには使用できますが、テンプレート、テ ーマ、コンポーネントまたはサイトには使用しないでください。

#### コンポーネント・タイプ

コンポーネント・カタログでは、次のタイプのコンポーネントがサポートされています:

- ローカル・コンポーネント
- インライン・フレームでレンダリングされるローカル・コンポーネント
- リモート・コンポーネント

コンポーネントのタイプは、コンポーネント・フォルダの拡張属性 xScsAppType として格納 されます。有効な値は次のとおりです。

タイプ	説明
ローカル・コンポーネント	すべての依存関係がローカルであることが想定されていま す。
インライン・フレームでレンダリングさ れるローカル・コンポーネント	コンポーネントは、Oracle Content Management と同じド メインから処理されます。
リモート・コンポーネント	コンポーネントは、リモートの場所から提供されます。

#### コンポーネント・ファイル構造

各タイプのコンポーネントは、コンポーネント・カタログで作成された場合、それらの実装 方法に基づいて異なるファイル・セットを含んでいます。

### **ローカル・コンポーネント**:

/Components/component-name
 appinfo.json
 _folder_icon.jpg
 assets
 settings.html
 render.js



#### インライン・フレームを使用するローカル・コンポーネント:

/Components/component-name
 appinfo.json
 _folder_icon.jpg
 assets
 settings.html
 render.js
 js
 sites.min.js
 knockout.min.js
 jquery.min.js

### リモート・コンポーネント:

/Components/component-name
 appinfo.json
 _folder_icon.jpg
 keys.json

### フォルダのメタデータ

コンポーネント登録データは、フォルダ・メタデータに格納されます。次のプロパティは、コンポーネントとそのタイプを一意に識別するために使用されます。

プロパティ	説明
app name	コンポーネント名を指定するフォルダ名。
app description	フォルダの説明。
app guid	すべてのコンポーネントが、GUID に関連付けられており、拡張属性 xScsItemGUID として格納されます。 GUID は、コンポーネント作成時にサーバーによって生成されます。
app type	コンポーネント・タイプを指定するプロパティ。拡張属 性 xScsApType として格納されます。

### 💉 ノート:

拡張属性 xScsAppIconUrl として格納されているプロパティ iconUrl は推奨 されていません。

#### appinfo.json ファイル

各タイプのコンポーネントの appinfo.json 登録ファイルには、フォルダ・メタデー タで利用できないデータのみが含まれています。フォルダ・メタデータで定義されて いるコンポーネント・プロパティは、appinfo.json ファイル上で重複していませ ん。



**ローカル・コンポーネント**:

{

}

```
"settingsData":{
    "settingsHeight":80,
    "settingsRenderOption"; "dialog",
    "settingsWidth":300,
    "componentLayouts":[],
    "triggers":[],
    "actions":[]
},
"initialData":{
    "customSettingsData":[],
}
```

```
インライン・フレームでレンダリングされるローカル・コンポーネント:
```

```
{
       "endpoints": {
            "settings": {
                   "height": "300",
                   "width": "400"
            }
       }
       "initialData": {
              "customSettingsData": {}
       }
}
リモート・コンポーネント
{
       "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": {}
       }
}
ローカル・コンポーネントには、登録された url 値がありません。それらでは、コンテンツ
を編集できる特定のファイルが使用されますが、その場所や名前は変更できません。
```

- ローカル・コンポーネントでは、ファイル assets/render.js および assets/ settings.html が使用されます。
- インライン・フレームでレンダリングされるローカル・コンポーネントでは、ファ イル assets/render.html および assets/settings.html が使用されます。
- リモート・コンポーネントでは、url 値が指定されているものが使用されます。

### コンポーネントの作成

**Oracle Content Management** で使用するためにカスタマイズされたコンポーネントを 作成できます。

Oracle Content Management には、Web サイトの構築に使用する多数の事前定義済コ ンポーネントが用意されていますが、開発者は、同じ機能にアクセスできるカスタム・ コンポーネントを作成できます。これらのコンポーネント・タイプには、ローカル、 インライン・フレームでレンダリングされるローカル、およびインライン・フレーム でレンダリングされるリモートがあります。

### ローカル・コンポーネント

ローカル・コンポーネントを作成する場合、完全に機能するサンプル・コンポーネントが提供され、これを独自のコンポーネントの作成の基準として使用できます。「デフォルト」を選択してコンポーネントの Knockout バージョンを作成するか、「テンプレート」を選択して Mustache ベースのバージョンを作成します。Oracle Content Management でのサイトの構築のローカル・コンポーネントまたはレイアウトの作成を参照してください。

#### インライン・フレームを使用したローカル・コンポーネント

インライン・フレームを使用してローカル・コンポーネントを作成する場合、完全に 機能するサンプル・コンポーネントが提供され、これを独自のコンポーネントの作成 の基準として使用できます。ローカル・コンポーネントの作成と同じ手順を使用しま す。また、ローカルに保存されるコンポーネントのインライン・フレーム・バージョ ンを作成するために、サンドボックス・オプションも選択します。インライン・フレ ームでレンダリングされたコンポーネントを参照してください。

#### リモート・コンポーネント

インライン・フレームを使用するリモート・コンポーネントを作成する場合、**サンド ボックス・**オプションを選択します。

作成したファイルをリモート・サーバーにコピーし、リモート・コンポーネントを登録します。コンポーネントを、公開したサイトに含める前にテストします。

インライン・フレームでレンダリングされたコンポーネントおよびリモート・コンポ ーネントの登録を参照してください。

### Developer Cloud Service によるカスタム・コンポーネ ントの開発

**Oracle Developer Cloud Service** は、**Oracle Content Management** のテンプレート、テ ーマおよびカスタム・コンポーネントを開発する上で役に立ちます。



Developer Cloud Service と Oracle Content Management の統合により、テンプレートととも に、テンプレートおよびコンポーネントを開発するためのツールが提供されます。また、初 めにサンプルのユニット・テストも提供されます。統合には、テンプレートおよびコンポー ネントを開発する上で役に立つ Git リポジトリとツールのほか、テンプレート、テーマおよび カスタム・コンポーネントを迅速かつ反復的に開発するためにローカル・テスト・ハーネス が含まれます。

**Developer Cloud Service** は、次のタスクを実行する上で役に立ちます:

- テンプレート、テーマおよびコンポーネントのローカル開発およびテスト用として Oracle Content Management インスタンスを使用するようローカル開発環境を設定します
- サンプルまたはスタータからテンプレートおよびコンポーネントを作成し、これらをテスト・ハーネスで実行し、これらを調査し、テンプレートを開発し、Developer Cloud Service 環境でテンプレート、テーマおよびコンポーネントを開発します
- ソース管理および将来の開発のために、Oracle Content Management から作成されたテン プレートまたはコンポーネントを Developer Cloud Service 環境にインポートします
- Web サイトで使用するために、インポートされる Developer Cloud Service 環境からテン プレートまたはコンポーネントを Oracle Content Management にエクスポートします
- 既存のコンポーネントをコピーします
- ユニット・テストを作成します
- コンポーネントを最適化します
- コンポーネントを Oracle Content Management にデプロイします

次のトピックでは、Oracle Content Management のための開発を目的として Developer Cloud Service の統合を使用する方法について説明します:

- Oracle Content Management のカスタム・コンポーネントの開発
- コンポーネントの最適化(縮小)によるパフォーマンスの向上
- 連続統合ジョブの実行

### Oracle Content Management のカスタム・コンポーネントの開発

**Developer Cloud Service** およびローカル・マシンを使用して、**Oracle Content Management** のカスタム・コンポーネントを開発します。

次のトピックでは、**Developer Cloud Service** を使用して **Oracle Content Management** コンポ ーネントを開発およびテストするステップについて説明します:

- 1. ローカル・マシンでの OCE ツールキットの設定。
- 2. Oracle Content Management 用の Developer Cloud Service コンソールへのサインイン.
- 3. Developer Cloud Service でのプロジェクトの作成。
- 4. 新規 GIT リポジトリ内のプロジェクト・コードへの OCE ツールキットの追加.
- 5. カスタム・コンポーネントの開発
- 6. ローカル・テスト・ハーネスを使用したテスト
- 7. ユニット・テストの作成および実行



### カスタム・コンポーネントの開発

**cec** コマンドライン・ユーティリティを使用して、新規コンポーネントを作成し、コン ポーネントをローカルで開発して、コンポーネントを **Oracle Content Management** に エクスポートします。

#### コンポーネントの作成

カスタム・コンポーネントをローカルで作成および開発するには、次のコマンドを使 用します:

cec create-component <component-name> -f <source>

source には、次のいずれかの値を選択します:

- 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
- 例:

cec create-component MyLocalComponent1 -f local

コンポーネントは git リポジトリの cec-components/src/main/components の下に作 成されます。

src/main/components ディレクトリには、Sample-To-Do コンポーネントがシードされ ています。作成したコンポーネントはすべてこのディレクトリに 格納されます。



#### コンポーネントのコピー

cec copy-component コマンドを使用して、Developer Cloud Service 内のコンポーネントをコ ピーできます:

cec copy-component <source> [<destination>]

このコマンドは、<<mark>source></mark>という名前の既存のコンポーネントを<destination>にコピーしま す。

#### Oracle Content Management へのコンポーネントのエクスポートまたはデプロイ

コンポーネントがローカル・サーバーで開発およびテストされたら、次のコマンドを使用し てコンポーネントをエクスポートできます。これで、コンポーネントの zip ファイルが作成さ れます。このコンポーネント zip を Oracle Content Management に手動でインポートできま す。

cec export-component <component name>

または、次のコマンドを使用して、コンポーネントを Developer Cloud Service から Oracle Content Management に直接デプロイできます。

cec deploy <component name>

cec deployAll コマンドは、src/main/components 内のすべてのコンポーネントをデプロイ します。

#### Developer Cloud Service へのコンポーネントのインポート

**Oracle Content Management** サーバーから作成されたコンポーネント **ZIP** ファイルがある場合、それをさらに開発するために **Developer Cloud Service** にインポートできます。次のコマンドを使用します。

cec import-component <location of the component zip file>

#### 重要:

コンポーネントのソース・コードは、src/main/components にあります。Oracle Content Management ローカル・サーバーの機能に必要であるため、src/main/ components の外部のファイルは変更しないでください。



### ユニット・テストの作成および実行

ユニット・テストを記述し、それらをカスタム Oracle Content Management コンポー ネント用に実行するサンプル・ユニット・テストから開始します。

#### サンプル・ユニット・テストでの開始

**Oracle Content Management** 用の **Developer Cloud Service** サンプルには、ユニット・ テストの記述を支援するために次のファイルが含まれています:

- src/test/unit: Sample-To-Do コンポーネントのユニット・テストが含まれています。これは、RequireJS モジュールを含むコンポーネント JavaScript コードのユニット・テストを記述するためのサンプルとして役立ちます。JavaScript ユニット・テストのための Mocha および Chai フレームワークが使用されています。
- index.html: ユニット・テストを実行します。Mocha、Chai、メイン・テスト・フ アイル test-main.js をロードします。
- test-main.js: ユニット・テスト・モジュール Sample-To-Do-Test をロードし、 Mocha ユニット・テストを実行します。
- Sample-To-Do-Test.js:
  - 実際のテストを定義します。RequireJS を使用して、コンポーネント・ソース・コード components/Sample-To-Do/assets/render.js をロードします。
  - add()、delete()、title()およびplaceholder()メソッドをテストします。

#### コンポーネントに対するユニット・テストの記述

カスタム・コンポーネントのユニット・テストそれぞれを記述するには、次のステップを実行します:

- 1. Sample-To-Do-Test.js に似たユニット・テストを作成します。
- 2. 作成したテストを test-main.js にロードします。

#### ユニット・テストのローカルでの実行

ユニット・テストをローカルで実行するには、次のステップに従います:

- 1. Git リポジトリをローカルにクローンします。
- 2. ローカル・マシンのターミナル・ウィンドウで、npm install と入力します。
- 3. npm start と入力します。
- 4. npm test を入力するか、ブラウザで http://localhost:8085/unit/を開きます。



### コンポーネントの最適化(縮小)によるパフォーマンスの向上

**Developer Cloud Service** を使用して JavaScript コードを縮小することにより、コンポーネントのパフォーマンスを改善し、ダウンロード・サイズを削減します。

#### コンポーネントの最適化

Sample-Text-With-Image コンポーネントには、コンポーネント・ソース・コードを最適化す る例(オプション)が提供されています。RequireJS オプティマイザを使用して、ソース・コー ドを縮小して組み合わせることができます。コンポーネントを縮小すると、JavaScript コー ド内の不要なスペースが削除され、パフォーマンスが向上し、ダウンロード・サイズが縮小 されます。最適化には次の操作が含まれます:

- JavaScript を縮小します。
- **CSS**を圧縮します。
- JavaScript、HTML、および CSS を 1 つのファイル render. js に結合します。

この最適化により、ダウンロード・サイズが削減され、パフォーマンスが向上します。

#### コンポーネントの最適化の有効化

Sample-Text-With-Image コンポーネントでは、最適化によって、render.js (template.html、data-defaults.js および design.css)の依存関係としてマークされている すべてのファイルが、単一のファイルとして render.js に縮小されて結合されます。すでに 最適化をサポートしている他のコンポーネントは次のとおりです:

- Sample-Folder-List
- Sample-File-List
- Sample-Documents-Manager
- Sample-Process-Start-Form
- Sample-Process-Task-List
- Sample-Process-Task-Details

ローカル・コンポーネントの最適化を有効にするには、次のステップを実行します(必須):

- **1.** Sample-Text-With-Image コンポーネントからコンポーネントに gulpfile.js をコピー します。
- render.jsの依存関係のために gulpfile.jsのモジュール名を修正します。 gulpfile.jsのコメントを参照してください。

🖉 ノート:

assets の下に css フォルダがある既存のコンポーネントがある場合は、css フォル ダの名前を styles に変更して、最適化中にエラーが発生しないようにします。

### 連続統合ジョブの実行

連続統合ジョブを実行し、サイトを最新の状態に保つことができます。

Developer Cloud Service の「CEC カスタム・コンポーネント開発のスタート・ガイド」Wiki を参照してください。連続統合の構築ジョブを構成できます。

### 多言語サイト用の翻訳可能コンポーネントの開発

カスタム・コンポーネントの開発者は、カスタム・コンポーネント内のどの文字列を、 多言語(MLS)サイトのページの翻訳で使用するかを指定できます。

これを実行するには、customSettingsData オブジェクトにデータを保存する際に、最 上位レベルに nls プロパティを導入する必要があります。

例:

```
SitesSDK.setProperty('customSettingsData', {
    'nls': {
        linkText: 'More...'
     }
});
```

翻訳ジョブが作成されると、Oracle Content Management は、customSettingsDataオ ブジェクトのトップレベルのプロパティを確認し、ページ上の各カスタム・コンポー ネント・インスタンスに対して nls オブジェクト全体をエクスポートします。トラン スレータがこれらの値を変換すると、これらをサイトにインポートし戻すことができ るようになります。

サイト翻訳がインポートされたら、翻訳されたロケール用として nls オブジェクトの 正しいバージョンが customSettingsData オブジェクトで返されます。

たとえば、サイトをフランス語に翻訳してから、フランス語ロケールでページをレン ダリングした場合、カスタム・コンポーネントに渡された customSettingsData オブジ ェクトの nls オブジェクトの値は次のように更新されます:

```
{
    'nls': {
        'linkText': 'Plus...'
    }
},
```

customSettingsData の nls オブジェクトのフォーマットは、名前と値のペアに限定す る必要があります。これにより、翻訳が容易になり、翻訳済ロケールでページがレン ダリングされるときに、サイトのベース値に翻訳済の値を正しく適用できるようにな ります。

# 「設定」パネルを使用した H1 コンポーネントの構築

簡易 HTML テンプレートおよび CSS を持つ最小限の Oracle Content Management コ ンポーネントを作成できます。この H1 コンポーネントには、design.json 内のテーマ 用として簡易設定パネルおよびエントリが用意されているため、他の Oracle Content Management ユーザーがエディタでコンポーネントを使用するときに**3**つの組込みスタイル から選択できます。

新しいコンポーネントを作成する場合、すぐに使用可能な一連のシード済ファイルが用意されています。これらのシード済ファイルは、製品内のコンポーネントのほとんどの機能を網 羅しています。シード済コードを変更して独自のコンポーネントを作成できますが、これに は、最終結果を達成するために必要なのはシード済コードの小さいサブセットのみです。

設定パネルを使用した H1 コンポーネントの構築は5つの手順で実行できます:

- 1. 新規ローカル・コンポーネントの作成
- 2. 基本的な H1 コンポーネントの構築
- 3. コンポーネントへの CSS の追加
- 4. 設定パネルの追加によるヘッダー・テキストの変更
- 5. 他のユーザーが H1 コンポーネント・スタイルを選択するためのテーマの更新

### 新規ローカル・コンポーネントの作成

Oracle Content Management を使用して、ページに即時ドロップできるローカル・コンポーネントを作成します。これは、新しいコンポーネントを作成するための開始ポイントです。

新しいローカル・コンポーネントを作成するには:

- サイド・ナビゲーションで「開発者」をクリックし、「すべてのコンポーネントの表示」 をクリックします。
- 2. 「作成」 > 「ローカル・コンポーネントの作成」を選択します。
- 3. 名前(H1_Component など)を入力し、必要に応じて説明を入力します。
- 4. 「作成」をクリックして、新しいコンポーネントを作成します。

コンポーネントを正常に作成できたら、これはコンポーネント・ページのコンポーネントの リストのほか、作成するサイトの「追加」>「カスタム」コンポーネント・パレットに表示さ れます。次のステップを使用して、コンポーネントの作成を確認します(チェックポイント 1):

- シード・テンプレートを使用して新しいサイトを作成します。たとえば、 StarterTemplate テンプレートを使用して ComponentTest という名前のサイトを作成し ます。
- 2. 「編集」オプションを選択し、サイトの更新を作成し、これをエディタで開きます。
- 3. 作成したサイト内のページを編集します。
- 左側のバーの「追加」ボタン(+)をクリックし、カスタム・コンポーネントのリストの「カ スタム」を選択します。
- カスタム・コンポーネント・パレットから「H1_Component」を選択し、ページにドロ ップします。

これで、作成したローカル・コンポーネントのデフォルトのレンダリングが表示されま す。

- **6.** コンポーネントのコンテキスト・メニューを選択します。
- 7. ドロップダウン・メニューから「設定」を選択します。

設定を変更し、シード・コンポーネントのレンダリングがどのように変化するかを確認で きます。



シード・ファイルを変更し、新しいカスタム・コンポーネントを作成できます。

### 基本的な H1 コンポーネントの構築

シード済ファイル内のコンテンツの大部分を削除して、H1 コンポーネントを作成できます。これにより、viewModelの作成時にシードする見出しテキストが表示されます。 後で、コンポーネントの設定およびスタイルを指定できます。

ローカル・コンポーネントの構造を確認するには:

- Oracle Content Management デスクトップ同期アプリケーションを使用して、コン ポーネントを検索し、それをファイル・システムと同期させます。
  - デスクトップ同期アプリケーションの最新バージョンで、「同期化の開始」または同期化するフォルダの選択オプションを選択します。
  - デスクトップ同期アプリケーションがない場合は、Oracle Content Management の「コンポーネント」タブでコンポーネントを選択し、ドリルダ ウンしてファイルを表示できます。
- 2. コンポーネントの下のファイルをリストする場合は、次のようなファイルが表示されます:
  - assets フォルダ内のコンポーネント・ファイル:
    - render.js
    - settings.html
  - appinfo.json: コンポーネント説明を含む JSON ファイル。
     コンポーネントの開発についてを参照してください。
  - folder_icon.jpg: コンポーネント・カタログに表示されるアイコン。

H1 コンポーネントを構築するには:

1. appinfo.json ファイルを開き、その内容を次の行と置き換えます:

```
{
   "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. 任意のテキスト・エディタで assets フォルダ内の render.js ファイルを開きます。
- 3. render.jsのコンテンツを次の行に変更します:

```
/* 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();
   11
   // Handle property changes from the Settings panel
   11
   this.updateCustomSettingsData = $.proxy(function(customData) {
     this.headingText(customData && customData.headingText);
   }, this);
   this.updateSettings = function(settings) {
     if (settings.property === 'customSettingsData') {
       this.updateCustomSettingsData(settings.value);
     ł
   };
   // Register your updateSettings listener to recieve SETTINGS_UPDATED
events
SitesSDK.subscribe(SitesSDK.MESSAGE_TYPES.SETTINGS_UPDATED, $.proxy(this.u
pdateSettings, this));
   11
   // Get the initial settings data for the component and apply it
   11
   SitesSDK.getProperty('customSettingsData',
this.updateCustomSettingsData);
 };
 // -----
 // Create a knockout based component implemention
 // -----
 var SampleComponentImpl = function(args) {
   // Initialze 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>';
     };
     11
     // SDK Callbacks
     \ensuremath{{//}} setup the callbacks expected by the SDK API
     11
     SampleComponentImpl.prototype.setupCallbacks = function() {
       11
       // callback - render: add the component into the page
       11
       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);
       11
       // callback - dispose: cleanup after component when it is
   removed from the page
       11
       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. assets フォルダで、コンポーネントの単純な HTML テンプレートになるよう
   render.html という新しいファイルを作成します。
```

5. render.html ファイル内の次のコンテンツを使用します:

```
<hl data-bind="text: headingText()"></hl>
```

コンポーネントの assets フォルダに、3 つのファイルが含まれるようになります。

- render.html
- render.js
- settings.html

新しい H1 コンポーネントをページに追加します(チェックポイント 2)。

### コンポーネントへの CSS の追加

コンポーネントのデフォルト・スタイルを提供する CSS を追加できます。

CSS を追加するには:

1. 次のコンテンツとともに、design.css ファイルをコンポーネントの assets フォルダに追加します:

```
.hl-component-default-style .scs-component-content {
  font-family: "Helvetica Neue", "Helvetica", "Arial", sans-serif;
  font-size: 24px;
  color:red;
  font-weight: normal; }
```

コンポーネントをスタイルするために使用するスタイル・クラスの接頭辞を宣言するために appinfo.json に追加します。h1-componentの styleClassName を追加する場合、コンポーネントがページにドロップされると、デフォルト・スタイルが h1-component-default-style になります。appinfo.jsonの新しいコンテンツは、次のようになります:

```
{
   "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. 2 行目(define(['knockout', 'jquery', 'text!./render.html'], function(ko, \$, template) {)を次の行に置き換えて、CSS ファイルをロードす るために render.js を更新します。

define(['knockout', 'jquery', 'text!./render.html', 'css!./
design.css'], function(ko, \$, template, css) {

```
N-F:
```

appinfo.json ファイルを変更したため、ブラウザをリフレッシュし、コ ンポーネントを再度ページに追加して、変更内容を反映させる必要があ ります。

コンポーネントで appinfo.json ファイルからそのデフォルト・データが取得されるようになることとを確認します。

### 設定パネルの追加によるヘッダー・テキストの変更

settings.html ファイルを更新して、H1 コンポーネントのテキストを設定するために 使用できる設定パネルを提供します。

設定パネルを追加して見出しテキストを変更するには:

1. settings.html ファイルを更新して、次の内容が含まれるようにします:

```
<!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"</pre>
rel="stylesheet">
      <!-- include supporting files -->
      <script type="text/javascript" src="/_sitescloud/renderer/app/</pre>
apps/js/knockout.min.js"></script>
      <script type="text/javascript" src="/_sitescloud/renderer/app/</pre>
apps/js/jquery.min.js"></script>
      <!-- include the Sites SDK -->
       <script type="text/javascript" src="/_sitescloud/</pre>
renderer/app/sdk/js/sites.min.js"></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script>
</head>
<body data-bind="visible: true" style="display:none; margin:0px;</pre>
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. サイト・ビルダーでコンポーネントを選択し、「設定」をクリックします。

設定パネルは小さく、コンポーネント設定パネルの最上部に埋め込まれるよう設定されて います。「見出しテキスト」というタイトルのセクションを探します。

見出しテキストを変更できることを確認するには(チェックポイント 4):

設定パネル内の見出しテキストを変更すると、コンポーネントが更新されて新しいテキスト が表示されます。



### 他のユーザーが H1 コンポーネント・スタイルを選択する ためのテーマの更新

コンポーネントのスタイルをテーマに登録することにより、コンポーネントに指定し たスタイルを他のユーザーが設定パネルまたは**「スタイル」**タブから切り替えられる ようにすることができます。

他のユーザーがコンポーネント・スタイルを選択できるようテーマを更新するには:

 コンポーネントの design.css ファイルに別のスタイルを追加します。 appinfo.json に定義されているとおり、各スタイルにコンポーネントの登録済 styleClassName 接頭辞を付けます。このコンポーネントの場合、接頭辞は h1component です。

h1-component-gothic-style および h1-component-courier-style の2つの別の スタイルが追加されています。

design.css の新しいコンテンツは次のようになります:

```
.hl-component-default-style .scs-component-content {
  font-family: "Helvetica Neue", "Helvetica", "Arial", sans-serif;
  font-size: 24px;
  color:red;
  font-weight: normal; }
.hl-component-gothic-style .scs-component-content {
  font-family: "Century Gothic", "CenturyGothic", "AppleGothic", sans-
serif;
  font-size: 32px;
  font-weight: bold; }
.hl-component-courier-style .scs-component-content {
  font-family: "Courier";
  font-size: 32px;
  font-weight: bold; }
```

 スタイルをテーマ design.json ファイルに登録します。このファイルは、サイト が使用しているテーマ内にあります。デザイン・ファイル内のテーマ・ファイルに ドリルダウンしてから、デフォルトのフォルダにドリルダウンし、コンポーネント のセクションを design.json に追加します。

後ろに続く**太字**テキストが、追加すべき内容の例です:



```
"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": [
            "icons": [
```

追加された design.json スニペットの名前("Plain"、"Courier"、"Gothic")は、次のように スタイル・タブ上のコンポーネントの設定パネルに表示されます。選択した場合、これらに より、対応するスタイル("h1-component-default-style"、"h1-component-courier-style"、"h1component-gothic-style")がそれぞれコンポーネントに適用されます。

### HTML テンプレートを使用した簡易コンポーネント の作成

Mustache JS テンプレートとともに HTML および CSS を使用する Oracle Content Management コンポーネントを作成し、設定パネルで入力するデータに基づいてコンテンツ をレンダリングできます。

設定パネルを提供し、HTML テンプレートでレンダリングするデータを編集できるようにす ることができます。

この手順の例では、Mustache JS テンプレートを使用します。

HTML テンプレートを使用して簡易コンポーネントを2つの手順で作成できます:

- 1. HTML コンポーネントの作成
- 2. HTML コンポーネントへのユーザー指定データの追加

### HTML コンポーネントの作成

Oracle Content Management コンポーネント・カタログで新しいローカル・コンポーネント を作成し、アセット・フォルダでファイルを追加および変更することによって HTML コンポ ーネントを構築できます。

HTML コンポーネントを作成および構築するには:

1. コンポーネントを作成し、HTML Component という名前を付けます。

新規ローカル・コンポーネントの作成を参照してください。

2. mustache.min.js ファイルを HTML コンポーネント内の assets フォルダに追加します。



3. assets フォルダ内に、コンポーネントの本体の HTML を含む render.html という 名前の新しいファイルを作成します。この例では、次のコンテンツで render.html ファイルを作成します:

```
<h1 class="title">One</h1>
      Lorem ipsum dolor sit amet, consectetur
  adipiscing elit, sed do eiusmod tempor incididunt ut labore et
  dolore magna aliqua.
     <h1 class="title">Two</h1>
       Lorem ipsum dolor sit amet, consectetur
  adipiscing elit, sed do eiusmod tempor incididunt ut labore et
  dolore magna aliqua.
      <h1 class="title">Three</h1>
      Lorem ipsum dolor sit amet, consectetur
  adipiscing elit, sed do eiusmod tempor incididunt ut labore et
  dolore magna aliqua./p>
      <h1 class="title">Four</h1>
      Lorem ipsum dolor sit amet, consectetur
  adipiscing elit, sed do eiusmod tempor incididunt ut labore et
  dolore magna aliqua.
     4. assets フォルダ内に、コンポーネントの CSS である design.css という名前の新
   しいファイルを作成します。この例では、design.css ファイルのコンテンツとし
  て次の行を追加します:
   .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. assets フォルダ内の render.js ファイルを開き、コンテンツを次のように変更します。
   前の手順で使用した HTML および CSS に関係なく、次の render. js ファイルでは、
   HTML および CSS がページにレンダリングされます:
   /* globals define */
   define(['jquery', './mustache.min', 'text!./render.html', 'css!./
   design.css'], function($, Mustache, template, css) {
     'use strict';
     // -----
     // Create a Mustache-based component implemention
     // ------
    var SampleComponentImpl = function(args) {
      this.SitesSDK = args.SitesSDK;
      // Initialze 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));
    };
    11
     // SDK Callbacks
     // setup the callbacks expected by the SDK API
     11
```

```
SampleComponentImpl.prototype.setupCallbacks = function() {
    11
   // callback - render: add the component into the page
   11
   this.render = $.proxy(function(container) {
      this.container = $(container);
this.SitesSDK.getProperty('customSettingsData', $.proxy(this.update,
 this));
    }, this);
   //
   // callback - SETTINGS_UPDATED: retrive new custom data and re-
render the component
   11
this.SitesSDK.subscribe(this.SitesSDK.MESSAGE_TYPES.SETTINGS_UPDATED
, $.proxy(this.updateSettings, this));
    11
    // callback - dispose: cleanup after component when it is
removed from the page
    11
   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;
});
```

HTML コンポーネントが構築されたことを確認するには(チェックポイント 1):

- 1. コンポーネント内の assets フォルダに次の 5 つのファイルがあることを確認し ます。
  - design.css
  - mustache.min.js
  - render.html
  - render.js
  - settings.html
- 新しい HTML コンポーネントをテスト・サイト内のページに追加します。エディ タの編集モードおよびプレビュー・モードでは、次のようにページ上にコンポーネ ントが表示されます。

編集モード





#### プレビュー・モード



### HTML コンポーネントへのユーザー指定データの追加

HTML コンポーネントで定義したすべてのデータ要素のためのフィールドを提供する、「カス タム設定」パネルを追加できます。ユーザーは、その後、「カスタム設定」パネル内のページ で、コンポーネント・インスタンスの値を入力できます。

HTML コンポーネントにユーザー入力データを追加するには:

 render.html ファイル内の HTML テンプレートを更新して、必要なデータ要素が含まれ るようにします。次の例では、Mustache JS テンプレート作成言語が使用されます。その ため、次のように{{および}}構文を使用してデータ要素を追加する必要があります:

```
<h1 class="title">{{title1}}</h1>
  {{text1}}
 <h1 class="title">{{title2}}</h1>
  {{text2}}
 <h1 class="title">{{title3}}</h1>
  {{text3}}
 <h1 class="title">{{title4}}</h1>
  {{text4}}
```


2. HTML コンポーネント内の appinfo.json ファイルのコンテンツを変更すること で、次のデータ・フィールドのデフォルト値を提供します:

```
{
     "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. HTML テンプレートの render.html ファイル内のデータ値を探す、「カスタム設
   定」パネルを追加し、ユーザーがエディタで変更できるように設定を作成します。
   settings.html ファイルのコンテンツを次のような行に変更します:
   <!DOCTYPE html>
```

```
<html lang="en">
<html lang="en">
<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"</pre>
rel="stylesheet">
       <!-- include supporting files -->
     <script type="text/javascript" src="/_sitescloud/renderer/app/apps/js/</pre>
knockout.min.js"></script>
     <script type="text/javascript" src="/_sitescloud/renderer/app/apps/js/</pre>
jquery.min.js"></script>
    <!-- include the Sites SDK -->
    <script type="text/javascript" src="/_sitescloud/renderer/app/sdk/js/</pre>
sites.min.js"></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></
</head>
<body data-bind="visible: true" style="display:none; margin:0px;</pre>
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);
                                ł
                       }, self);
           };
           // apply the bindings
           ko.applyBindings(new SettingsViewModel());
  </script>
</body>
     ノート:
     appinfo.json ファイルを変更したため、ブラウザをリフレッシュし、コ
      ンポーネントを再度ページに追加して、変更内容を反映させる必要があ
```

```
ORACLE
```

ります。

コンポーネントで appinfo.json ファイルからそのデフォルト・データが取得されるようにな ることと、「カスタム設定」パネル内の値を変更できることを確認するには(チェックポイント 2):

- appinfo.jsonファイルを変更した後、ブラウザをリフレッシュし、コンポーネントを再度ページに追加して、変更内容を反映させることを覚えておいてください。
- 2. HTML コンポーネント内の「カスタム設定」パネルを開き、データを入力します。

HTML_Component	HTML_Component Settings ×			
123	General	Custom Settings ×		
Lorem ipsum dolor s amet, consectetur adipiscing elit, sed d elusmod tempor incidio ut labore et dolore maj aliqua.	Alignment E Width	title1 123 text1 Lorem Ipsum dolor sit amet, consectetur ac title2 Two	br sit tur d do ididunt nagna	Four Lorem ipsum dolor sit amat, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
Sample	Spacing TOP 5 V	text2 Lorem ipsum dolor sit amet, consectetur ac title3 Three text3 Lorem ipsum dolor sit amet, consectetur ac title4	ıt	
		Four text4 Lorem ipsum dolor sit amet, consectetur at		

3. Oracle Content Management コンポーネントで HTML および CSS を Mustache JS テン プレートと組み合せて使用して、「カスタム設定」パネルに入力したデータに基づいてコ ンテンツがレンダリングされることを確認します。

# ローカル・コンポーネントとリモート・コンポーネ ントの比較

ローカル・コンポーネントとリモート・コンポーネントは異なる方法で実装されます。

次の表は、コンポーネントの実装方法の違いを示しています。これは、インライン・フレー ムでレンダリングされたリモート・コンポーネントとローカル・コンポーネントのどちらを 使用するかを決定する際に役に立つ場合があります。

ローカル・コンポーネントの実装	リモート・コンポーネントの実装
複雑なコンテンツ中心のユーザー・インタフェー	アプリケーション・ロジックを Oracle Content
スを <b>Oracle Content Management</b> に統合します	Management に統合します
ページ内のコンテンツをレンダリングする	ページ内のコンテンツをレンダリングする
JavaScript を実行します	<iframe> HTML タグを使用します</iframe>



ローカル・コンポーネントの実装	リモート・コンポーネントの実装
JavaScript テクノロジ・スタックを使用します	JavaScript のみではなく任意のテクノロジを使用 できます
<b>Oracle Content Management</b> の JavaScript スタックを活用し、Oracle Content Management コンポーネントを再使用できます	Oracle Content Management の JavaScript スタ ックとは統合されません
<b>Oracle Content Management</b> サーバーに格納さ れている/assets ドキュメントからアセットの 依存関係を追加します	すべてのアセット・アクセスに対して HTTPS URL および対応する証明書を必要とします
テンプレートを使用してホストされ、Oracle Content Management によってテーマとは関係 なく公開されます	URL エンドポイント用としてホストされたミドル 層サーバー
コードはページ内で実行されるため、コンポーネ ント・コードが破損している場合、ページが破損 する可能性があります	プロジェクトはインライン・フレームのコンテン ツとは関係なくレンダリングされるため、コンポ ーネントに障害が発生してもページは破損しませ ん
コンポーネントのレンダリング速度が遅い場合、 ページのレンダリング速度が遅くなる可能性があ ります	プロジェクトはインライン・フレームのコンテン ツとは関係なくロードされるため、ページは可能 な限り高速でロードされ、任意のインライン・フ レームのコンテンツがロードされます

### 🖋 ノート:

ローカル・コンポーネントとリモート・コンポーネントのレンダリング方法 は異なりますが、サイト・ビルダーでの「設定」パネルの実装方法は同じで す。両タイプのコンポーネントの「設定」パネルはインライン・フレームを 使用してレンダリングされ、境界を越える通信とドメインを越える通信の両 方を可能にするために同じ JavaScript SDK を使用します。

# コンポーネント設定のレンダリング

コンポーネント設定をレンダリングするには、コンポーネント設定 URL およびコンポ ーネント設定レンダリング・オプションを使用できます。

#### コンポーネント設定 URL

コンポーネント設定 URL は、インライン・フレームを使用してレンダリングされ、ペ ージにドロップされる実際のコンポーネントの特定の設定を可能にするパラメータを 使用してコールされます。設定 URL のフォーマットは次のとおりです:

{Component Settings URL}?instance=<appinstance>&width=<width>&currCompId=<id of the app associated with the settings panel>&locale=<locale>

名前	タイプ	説明
コンポーネント設定 URL	URL	コンポーネントのコンポーネント設 定 URL



名前	タイプ	説明
width	数値	設定インライン・フレームの幅(ピク セル単位)
currCompId	文字列	「設定」パネルによって編集されるコ ンポーネントの現在のコンポーネン ト ID
locale	文字列	ホスト・サイトの現在のロケール(サ イト・ビルダー)。フォーマットは <language>_<dialect>です。例: En_us。</dialect></language>

#### ローカル・コンポーネント設定のレンダリング・オプション

インライン・フレームのサイズと複雑さに基づいて、ローカル・コンポーネントの「設定」 パネルでインライン・フレームをレンダリングするために3つのオプションが用意されてい ます。各オプションは、settingsRenderOptionページに指定されます。

/ ノート:

これらのオプションは、ローカル・コンポーネントに対してのみ使用できます。リ モート・コンポーネントの「設定」パネルは常にダイアログ内でレンダリングされ ます。

- inline
  - このオプションを使用するのは、ユーザーが入力する小さいプロパティがいくつか存 在する場合のみです。
  - このインライン・フレームは、「設定」パネルに移動するために一般的なボタンのかわりに「一般」タブに挿入されます。
  - 最も統合されたソリューションを提供し、ユーザーに対して最も少ないクリック数を 要求しますが、領域は制限されています。
- panel
  - このオプションを使用するのは、プロパティのリストが長い場合ですが、これらのプロパティは依然として標準「設定」パネルの 300 ピクセル内に適切に表示できます。
  - このインライン・フレームはビュー内にスライドし、ユーザーを「一般」タブに戻す ために「戻る」ボタンが表示されます。
  - ページ上のインライン・フレームを操作できる統合ソリューションを提供します。
- dialog
  - 複雑なユーザー・インタフェースを必要とする、より一般的な設定レイアウトを処理 するためのデフォルト・モード。
  - ページ上のモーダル・ダイアログにインライン・フレームが表示されます。

### ローカル・コンポーネントの実装

コンポーネント・インスタンスは、コンポーネント・ファクトリ・オブジェクトによ って作成されます。

コンポーネント登録ファイル内のコンポーネント・ファクトリ・オブジェクト・エン トリは、initData.componentFactoryです。これは、RequireJSでコンポーネント・ ファクトリのロードに使用される、JavaScriptファイルへの参照です。これは、 componentFactory インタフェースを実装する JavaScript オブジェクトを返す必要が あります

コンポーネント・ファクトリ・オブジェクトは、コンポーネントの各インスタンスを 作成する、customComponentFactory.createComponent(*args*)という **API** を実装する必 要があります。

args には、次のような設定が含まれます:

- SitesSDK: Oracle Content Management のサイト SDK。
- id: ページに追加されたコンポーネントの一意の ID (GUID)。
- viewMode: ページがレンダリングされている現在のモード。ページが編集中の場合、それは「編集」です。ページがプレビュー中の場合、それは「ナビゲート」です。ランタイム時(サイトの公開時)には、この値は定義されていません。各モードで公開する必要がある機能に基づいて、様々な実装を提供できます。たとえば、リンクは、ページが編集モードで実行されているときはアクティブにできません。

### 🎤 ノート:

コンポーネントに JQuery または Knockout を使用する必要はありませんが、 ネストされたコンポーネントなどの Oracle Content Management 機能を活 用するには、Oracle Content Management から提供される Knockout のバー ジョンを使用する必要があります。このバージョンの Knockout では、コン ポーネント登録およびハンドラが拡張されています。それらは、それ以外で は利用できません。

コンポーネント自体については、コンポーネントがページ・ライフサイクルとやり取 りできるようインスタンス化されるときに、SDK が渡されます。ページ・ライフサイ クル関数は、コンポーネントによって実装され、コンポーネントをページ上にレンダ リングするために Oracle Content Management によってコールされる必要がありま す。

コンポーネントを実装するために、必須およびオプションの API が提供されています。

#### 必須の API

*customComponent*.render(container): コンポーネントに、提供された DOM コンテナ 要素にそれ自体を挿入するよう指示します。

• container: カスタム・コンポーネント HTML の DOM コンテナ要素。



#### オプションの API

customComponent.dispose(): コンポーネントがページから削除されているときにコールさ れます。必要なくなったリソースをコンポーネントで削除する機会を提供します。

# コンポーネントのスタイル・クラス

ユーザーがあなたのコンポーネントに適用できるスタイルの定義済リストを作成できます。

コンポーネントの事前定義済スタイル・リストを作成する作業では、テーマのデザイン・ファイルを介して Oracle Content Management によって提供されるコンポーネントのスタイル・クラスを定義する場合と同じモデルを採用します。カスタム・スタイル・クラスの名前は、コンポーネントの appinfo.json ファイルに記載します。

追加スタイルは、design.css および design.json ファイルに定義します。json ファイ ルは、ユーザー・インタフェースに表示される名前から実際に基礎となる css クラス名への マッピングを提供し、css ファイルは、各スタイル・クラスの詳細を提供します。

design.jsonファイルには、コンポーネントの次の構造があります:

```
{
    "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": []
}
```

}

"styles":[]の各エントリには、特定のコンポーネントのスタイルのリストが含まれま す。"name"は、組込みローカライズ文字列への参照である場合、または使用対象の指 定値である場合があります。たとえば、タイトル・コンポーネントには、次のデフォ ルトのスタイルが用意されています:

```
{
   "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"
              }
]
}
名前値は、次のように、ユーザー・インタフェースに表示する実際の単語にマップさ
れます:
   "COMP_STYLE_FLAT": "Flat",
   "COMP STYLE HIGHLIGHT": "Highlight",
   "COMP STYLE DIVIDER": "Divider",
css ファイルは、クラス値の定義を提供します:
   .scs-title-default-style {
     color: #333333;
     display: block;
     font-family: "Helvetica Neue", "Helvetica", "Arial", sans-serif;
     font-size: 24px;
     font-weight: normal;
                           }
```



例として、テーマの design.json ファイルには、components.json ファイルに定義する initialData.compomentId 値に基づいてコンポーネントのエントリを追加できます:

```
"componentId": "news-article"
```

design.json ファイル内の対応するエントリは、次のようになります:

design.css ファイル内の対応するエントリは、次のようになります:

```
.news-article-default-style .scs-image {...}
.news-article-style-1 .scs-image {...}
```

# 組込みコンポーネントのスタイリング方法

組込みスタイルをオーバーライドおよび拡張することによって、Oracle Content Management 内の組込みコンポーネントをスタイル設定するために独自のルック・アンド・フィールを作 成します。

組込みコンポーネントは、次の2箇所からそれらの視覚的なスタイルを取得します:

- comp.css: 各コンポーネントの*基本的な*外観を指定する組込み CSS ファイル
- design.css: サイトで使用しているテーマの一部である CSS ファイル

design.css ファイルでは、組込み comp.css スタイルをオーバーライドおよび拡張して、独 自のルック・アンド・フィールを作成できます。テーマでは、design.css ファイルは designs/ default ディレクトリにあります。

次のトピックでは、すべての組込みコンポーネントに共通する comp.css ファイル内のクラス を説明し、テーマの定義の概要を示します:

- コンポーネント・スタイルの基本
- コンポーネント固有のスタイル
- コンポーネント・プロパティの設定

### コンポーネント・スタイルの基本

組込みサイト・コンポーネントはすべて、同様の CSS クラス構造を共有します。

各コンポーネントには、最も外側の<div>要素に適用される次の3つのCSSクラスがあります:

scs-component scs-type design-style

*type* は、コンポーネント・タイプ(image、gallery または divider など)です。*design-style* は、テーマのファイルに定義されているとおりに、コンポーネントに選択された スタイル・クラスです。

#### テーマ・スタイルの基本

テーマの design.json ファイルには、コンポーネントの各タイプ(button または image など)に適用できるすべてのスタイル(frame、shadow、highlighted など)がリストされ ます。各スタイルには、display name と class name の両方があります。表示名は、 「設定」パネルの「スタイル」タブに表示されます。クラス名は、テーマの design.css ファイル内の CSS セレクタを参照します。たとえば、button コンポーネントのエン トリは次のようになります:

```
"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"
        }
},
```

組込みコンポーネントの名前は翻訳されるため、リソース・バンドルからスタイル名 を取得するためのキーが表示されます。button コンポーネントをページに追加してか ら、「設定」>「スタイル」パネルで「シンプル」スタイルを選択すると、design.json ファイルにより、表示名「シンプル」(キー COMP_STYLE_SIMPLE)がクラス名 scs-buttonstyle-3 に関連付けられます。button は、次のクラスを使用してレンダリングされま す:

scs-component scs-button scs-button-style-3

特定のコンポーネントに対してスタイルが選択されていない場合、デフォルトのスタ イル scs-type-default-style が使用されます。前の例では、ボタンは次のクラスを使 用してレンダリングされます:

scs-component scs-button scs-button-default-style



#### scs-component-content スタイル

すべての組込みコンポーネントについて、前述の scs-component <div>内に、CSS クラス scs-component-content を持つ content <div>があります。つまり、次のようになります:

scs-component scs-type design-style
scs-component-content

design.css ファイル内では、多くの場合、scs-component-content クラスを使用して、コン ポーネントの周りの「ボックス」をスタイルします(たとえば、枠線またはシャドウを適用し ます)。

組込み comp.css ファイル内では、他の CSS プロパティ 間で position:relative および display:inline-block を使用して共通の scs-component-content クラスが定義されている ことに注意してください。

scs-component-content はコンポーネントの周りの「ボックス」をスタイルする上で役に立ちますが、コンポーネントを完全にスタイルするにはコンポーネント固有のクラスが必要です。コンポーネント固有のスタイルを参照してください。

### コンポーネント固有のスタイル

イメージ、ボタン、ドキュメント、段落、タイトル、マップおよび他のコンポーネントには 固有のスタイルを適用できます。

#### イメージ・コンポーネント

イメージ・コンポーネントには、scs-component-content クラスの下に次の CSS クラス構造 があります:

scs-image-container
scs-image-link
scs-image-image
scs-image-caption

scs-image-image クラスは、<img>タグ自体に適用されます。scs-image-caption クラスは、 キャプションが存在する場合にキャプションをスタイリングするために使用します。

scs-image-link クラスは、イメージにリンクがアタッチされている場合のみ、存在します。 通常、このクラスも scs-image-container クラスも、カスタム・スタイルを必要としません。

デフォルトでは、イメージのキャプションは、イメージの下部に広がった半透明のオーバー レイとしてレンダリングされます。





.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;
}

イメージの上部にキャプションを配置し、色を変更するには、design.json ファイル 内のイメージ・コンポーネントに別のスタイルを追加してから、design.css ファイル でこれに対して **CSS** を定義します。





```
.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;
}
```

#### ボタン・コンポーネント

ボタン・コンポーネントには、次のクラス構造があります:

scs-button-button scs-button-text

scs-button-button クラスは、ボタンに見えるようにスタイルが設定されたクリック可能な <div>です。scs-button-text クラスは、ボタン内のテキストをスタイリングするために使用 します。

たとえば、ボタン・コンポーネントのルック・アンド・フィールの変更をテストするには、 design.json ファイル内のこのコンポーネントに別のスタイルを追加してから、design.css ファイルでこれに対して **CSS** を定義します。



# **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;
}
```



```
.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;
}
```



#### ドキュメント

ドキュメント・コンポーネントには、次のクラス構造があります:

scs-document-container
scs-document-cap
scs-document-title
scs-document-desc

scs-document-container クラスは、ドキュメント・ビューアをラップし、通常はスタイルが 設定されません。

#### ギャラリ

ギャラリ・コンポーネントには、基礎となる JSSOR スライダ・コンポーネントをラップする 単一のクラスがあります:

scs-gallery-container

JSSOR スライダは、同様にスタイリングできる複数のクラスを使用します:

```
jssorb14 (navigator)
jssora021 (left arrow)
jssora02r (right arrow)
jssort07 (thumbnails)
```

#### ギャラリ・グリッド

ギャラリ・グリッド・コンポーネントに使用されるクラスは、「設定」パネルで選択したレイ アウトおよび切取りによって異なります:

```
scs-gallerygrid-container scs-gallerygrid-layout
scs-gallerygrid-cell
scs-image (multiple)
```

ギャラリ・グリッドに対して選択したレイアウトおよび切取り設定に応じて、layout \の値は、stretch、crop、fit または flowing になります。

scs-gallerygrid-cell クラスは、列レイアウトに対してのみ存在します。

ソーシャル・バー

ソーシャル・バー・コンポーネントには、次のクラス構造があります:

```
scs-socialbar-container
scs-socialbar-icon
```

scs-socialbar-icon クラスは、ソーシャル・バー内の各<img>タグに適用されます。



#### 段落

段落コンポーネントには、実際の段落テキストをラップする単一のクラスのみがあります:

scs-paragraph-text

たとえば、段落コンポーネントでコントリビュートするテキストに、金属に刻印され たテキストのような効果を適用するには、design.json ファイルに別のスタイル・ク ラスを追加し、design.css ファイルでこれに CSS を定義します。



```
.scs-paragraph-style-7 {
font-size: 24px;
font-family: Arial, Helvetica, sans-serif;
font-weight: 700;
padding: .3em;
color: #000000;
background: #6666666;
text-shadow: 0px 1px 1px #ffffff;
}
```

または、凝ったデザインにしたい場合は、次の例のようにすることができます。



.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;
}
```

#### タイトル

タイトル・コンポーネントにも、実際のテキストをラップする単一のクラスのみがあります:

scs-title-text

#### マップ

マップ・コンポーネントには、マップ・レンディションをラップする単一のクラスがありま す:

scs-map-content

このクラスは通常、スタイルが設定されません。

#### カスタム・ローカル・コンポーネント

カスタム・ローカル・コンポーネントには、実際のコンポーネントをラップする単一のクラ スのみがあります:

scs-custom-component-wrapper

カスタム・ローカル・コンポーネントのカスタム・ビューをレンダリングするために使用す る必要がある CSS スタイルを完全に制御できます。ローカル・コンポーネントはインライン でレンダリングされます。つまり、テーマ内または design.css ファイル内に定義された CSS スタイルを直接適用できます。

#### カスタム・リモート・コンポーネント

カスタム・リモート・コンポーネントには、iframe をラップする単一のクラスのみがあります:

scs-app-iframe-wrapper

カスタム・リモート・コンポーネントに定義されている CSS スタイルを適用する以外にも、 サイト SDK を活用してホスト・サイトから design.css ファイルをフェッチできます。

```
// 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);
}
}
});
```

これにより、ホスト・スタイルのスタイルをコンポーネントに継承させることができ ます。

#### 区切り

区切りコンポーネントにはコンポーネント固有のクラスがありませんが、<hr>タグ自体のスタイルを設定できます。

たとえば、点線区切りを作成できます:

```
.design-style .scs-divider hr {
border-top: 1px dotted #333333;
}
```

#### ビデオ、YouTube、スペーサ

ビデオ・コンポーネント、YouTube コンポーネントまたはスペーサ・コンポーネント には、コンポーネント固有のクラスはありません。

### コンポーネント・プロパティの設定

ノート:

サイトで使用するコンポーネント・プロパティを構成できます。

コンポーネントは、Web ページの個別部分であり、テキスト、タイトル、イメージ、 ボタン、区切り、マップ、ギャラリ、ビデオなどが含まれます。テーマのデザインを 作成する場合、コンポーネントの各タイプのデフォルト設定も指定する必要がありま す。各コンポーネントには、コンポーネントの外観および動作を定義するサイズ、位 置合せ、間隔、色、枠線などの設定があります。この設定は、コンポーネントによっ て異なります。コンポーネントが新しいテーマのサイトで使用可能になったときにコ ンポーネント・プロパティをユーザーが変更できるようにするかどうかを選択するこ ともできます。

例として、次の手順は、段落コンポーネントの設定を構成する方法を示しています:

- 1. 開発サイトが編集モードで開いているときに、段落コンポーネントが含まれるページを選択するか、段落コンポーネントを追加します。
- 段落コンポーネントをクリックし、コンポーネントの隅にあるをクリックし、
   「設定」を選択します。

「設定」パネルが表示され、コンポーネントに関する選択を行うことができます。

設定オプションは、コンポーネントの各タイプによって異なります。ロ ーカル・カスタム・コンポーネントまたはリモート・カスタム・コンポ ーネントを操作している場合、**「カスタム設定」**リンクが表示されます。



Paragraph Settings		×		
Ge	General		tyle	
Alignment				
Left	Center	Right		Fill
Width				
	_			× ^
Spacing				
TOP		BOTTOM		
5			~	^
LEFT				
5				^

3. たとえば、位置合せ、幅、間隔、フォント、スタイル、色、サイズなどに関する設定を変 更できます。

コンポーネントの外観は新しい設定に合せて変更されます。

4. 終了したら、 をクリックします。設定がページに適用されます。

# インライン・フレームでレンダリングされたコンポ ーネント

インライン・フレーム内にレンダリングされるコンポーネントは、外部サーバー(**リモート・ コンポーネント**と呼ばれる)のコンポーネントを登録および追加することによって Oracle Content Management ページで指定することも、ローカル・コンポーネントの作成時にインラ イン・フレーム・オプションを選択することによって指定することもできます。このタイプ のコンポーネントは、ソーシャル・コンポーネントやカート精算コンポーネントの追加など、 サイトのために機能を拡張できます。

リモート・コンポーネントの場合は、エンドポイント URL を指定する必要があります。イン ライン・フレームでレンダリングされるローカル・コンポーネントの場合、URL は、コンポ ーネント・カタログ内のコンポーネントの名前から導出されます。

- エンドポイント URL: ウィジェットのコンテンツは、この URL からフェッチされ、イン ライン・フレーム内に埋め込まれます。
- 設定 URL: この URL は、コンポーネントがページ上にドロップされた後にそれを構成するために、インライン・フレームでレンダリングされます。

コンポーネントでは、静的または動的なデータ視覚化をレンダリングし、サイトの機能を拡 張するフォームまたはその他の対話型ユーザー・インタフェースを表示できます。コンポー ネント設定のレンダリングを参照してください。



#### Oracle Content Management によって提供されるコンポーネント

インライン・フレーム内にレンダリングされる一連のコンポーネントは、Oracle Content Management で提供されます。

名前	タイプ	ID
フォルダ・リスト	scs-app	フォルダ・リスト
ファイル・リスト	scs-app	ファイル・リスト
ドキュメント・マネージャ	scs-app	ドキュメント・マネージャ
Facebook のいいね	scs-app	Facebook のいいね
Twitter のフォロー	scs-app	Twitter のフォロー
Twitter の共有	scs-app	<b>Twitter</b> の共有
Facebook のおすすめ	scs-app	Facebook のおすすめ

#### コンポーネント登録

インライン・フレームでレンダリングされるリモート・コンポーネントをサイトで使 用できるようにするには、HTTPS を使用してエンドポイント URL を登録する必要が あります。この情報は、コンポーネント・カタログに格納されます。エンドポイント では、URL をインライン・フレームで表示できるようにする必要があります。ヘッダ ーで X-Frame-Options="sameorigin"を設定しないでください。ローカル・コンポーネ ントの場合、ファイルは Oracle Content Management サーバーに格納されているため、 このエンドポイント基準が自動的に満たされます。

インライン・フレームでレンダリングされるコンポーネントが登録されると、そのコ ンポーネントを表す新しい GUID が生成されます。このようなコンポーネントが Oracle Content Management インスタンスに複数回登録されている場合、複数の GUID が取得されますが、これは、これらがそのコンポーネントの各登録を表すからです。 コンポーネントが登録されると、GUID に対して Oracle Content Management に格納 されているすべての説明が表示されます。コンポーネントは依然としてそのリモー ト・エンドポイントから実行されます。リモート・コンポーネントの登録を参照して ください。

インライン・フレームでレンダリングされたコンポーネントが登録されると、インス タンス ID も生成されます。このインスタンス ID は、特定の Oracle Content Management テナントに登録されているコンポーネントを表します。同じ Oracle Content Management テナント内に複数回登録されているか、異なる Oracle Content Management テナントに登録されている同じコンポーネントは、異なるインスタンス ID を持ちます。

#### リモート・コンポーネント設定の永続性

インライン・フレームでレンダリングされるコンポーネントのための設定 URL は、「設 定」パネル・ダイアログでレンダリングされ、インスタンス ID とコンポーネント ID が 両方とも提供されます。これにより、コンポーネントが、インスタンス ID とコンポー ネント ID で索引付けされ、その固有サーバー内の設定自体を永続化するよう選択でき るようになります。また、Oracle Content Management のサイト SDK を使用すると、 サイト・ページのページ・モデルにコンポーネント ID に対して最大 1.5KB の JSON デ ータを保存できます。



サイト SDK を使用して設定を永続化すると、インライン・フレームでレンダリングされるコンポーネントにとって 2 つの利点があります:

- コンポーネントは、ページ・バージョン、ページ更新およびサイト公開モデルに簡単に参加できます。
- コンポーネントは、バックエンド・システムで実行するのとは対照的にブラウザで実行される HTML エンドポイントで構成できます。

Oracle Content Management SDK を参照してください。

# インライン・フレームでレンダリングされたコンポ ーネントのインスタンス ID および構造について

コンポーネント・インスタンス **ID** は、サイト内のインライン・フレームでレンダリングされ たコンポーネントの一意の識別子です。

インライン・フレームでレンダリングされたコンポーネントをユーザーがコンポーネント・ カタログからサイト・ページにドラッグ・アンド・ドロップすると、Oracle Sites Cloud Service に対してプロビジョニング・コールが行われ、新しい一意のコンポーネント・インスタンス ID が生成されます。この ID は一意であることが保証されており、そのサービスでプロビジョ ニングされるそのようなコンポーネント・インスタンスはすべて、同じインスタンス ID を取 得します。

コンポーネント・インスタンス ID には、インライン・フレームでレンダリングされたコンポ ーネントの設定および使用を保護するために使用できる追加情報が含まれるため、コンポー ネントは、設定の更新が信頼できる場所からのものであることを確信できます。

開発者は、インスタンス ID パラメータを使用して、サイトを特定し、コール元を認証できま す。コール元は、コンポーネントの秘密鍵を使用して生成されたデジタル署名を検証するこ とによって認証されます。秘密鍵は、コンポーネントの登録プロセス中に生成されます。

コンポーネント・インスタンスは、「.」によって区切られたデータと構造の2つの部分で構成 されています。

#### コンポーネント・インスタンス: データ

インライン・フレームでレンダリングされたコンポーネントのインスタンスのデータ部分は、 Base64 JSON でエンコードされた文字列です。JSON 文字列の構造を次に示します:

```
{
   "instanceid": "BBDC7614F693B75110D811E6C0B77C935FAEC5112E5E",
   "permissions": "",
   "entitlements": "",
   "signdate": "1435426735293",
   "sitedomain": "service1-tenant4.localhost"
}
```

フィールド名	説明
instanceid	<b>Oracle Content Management</b> テナント用としてインライン・フレームでレ ンダリングされたコンポーネントの一意の識別子。
signdate	署名の生成日。



フィールド名	説明
sitedomain	<b>Oracle Content Management</b> インスタンスのドメイン名。
permissions	サイト・メンバーの権限セット。編集モードでは、値 SITE_OWNER を持ちま す。そうでない場合、値を持ちません。
entitlements	サイト所有者によって購入されたプレミアム機能のリスト。

#### コンポーネント・インスタンス:署名

コンポーネント・インスタンスのデータ部分は、APP_SECRET_KEY によって署名される 前にシリアル化されます。コンポーネントの登録時には、この秘密鍵が生成され、開 発者に示される必要があります。署名は、次に示すように、秘密鍵を使用してコンポ ーネント・インスタンスのデータ部分(シリアル化された JSON 構造)のハッシュを生成 することによって計算されます:

\$signature = HMAC (serialized JSON structure, APP_SECRET_KEY)

署名の生成に使用されるハッシュ・アルゴリズムは SHA256 です。このトークンは、 次に示すように、シリアル化された JSON 構造と生成された署名コンポーネントを連 結したものです:

\$instance = {base64encoded serialized JSON structure}.
{base64encoded \$signature}

例:

//base64 encoded serialized object //signature
eyJpbnN0YW5jZWlkIjoiQTRGOTE3REY5OTZEN0Q3ODBCMjUzODZFOTFEMDA3ODJGMjVBRjY2
Rjc3OTIiLCJzaWduZGF0ZSI6IjE0NDU2MzcwNTk5MTciLCJzaXRlZG9tYWluIjoic2Vydmlj
ZTEtdGVuYW50MS51cy5vcmFjbGUuY29tIiwicGVybWlzc2lvbnMi0iJTSVRFX09XTkVSIiwi
ZW50aXRsZWllbnRzIjoiIn0=.5p3of7t110wuysF3zpm+YgICSHH8C/BHczdbVZx2VH8=

### リモート・コンポーネントのセキュリティ

Oracle Content Management では、サード・パーティの開発者が、カスタム・コンポ ーネントを Oracle Content Management プラットフォームに統合できますが、リモー ト・サーバーに格納することもできます。

各リモート・コンポーネントは、設定およびレンダリングのエンドポイントが Oracle Content Management に登録されている必要があります。エンドポイントに加えて、開 発者は、登録済コンポーネント固有の秘密キーを提供する必要もあります。

Oracle Content Management は、登録済コンポーネントのエンドポイントを呼び出し て、サイト・ページ内のコンテンツを実現します。これらのエンドポイントはパブリ ック・インターネットに公開されるため、開発者は、登録済リモート・コンポーネン トのエンドポイントが Oracle Content Management から呼び出されていることを確認 する必要があります。呼出し元の信頼性を確認するために、署名済トークンが URL の 登録済エンドポイントに送信されます。呼出し元は、登録プロセス中に提供されたリ モート・コンポーネントの秘密キーとともに署名済トークンに埋め込まれたデジタル 署名を確認することで、認証されます。



トークンのフォーマットは次のとおりです:

{base64 encoded serialized JSON data}.{base64 encoded signature}

登録済アプリケーション・エンドポイントに渡されたサンプル・トークンは次のとおりです:

eyJpbnN0YW5jZWlkIjoiQTRGOTE3REY5OTZEN0Q3ODBCMjUzODZFOTFEMDA3ODJGMjVBRjY2Rjc30 TIiLCJzaWduZGF0ZSI6IjE0NDU2MzcwNTk5MTciLCJzaXRlZG9tYWluIjoic2VydmljZTEtdGVuYW 50MS51cy5vcmFjbGUuY29tIiwicGVybWlzc2lvbnMiOiJTSVRFX09XTkVSIiwiZW50aXRsZWllbnR zIjoiIn0=.5p3of7t110wuysF3zpm+YgICSHH8C/BHczdbVZx2VH8=

トークンは、「.」 デリミタで区切られた 2 つの別個の部分であるデータと署名で構成されています。

一般的なガイドラインとして、開発者は、コンポーネントの登録済エンドポイントへのアクセス権を付与する前に、必ず編集モードかプレビュー・モードでトークンを認証する必要があります。さらに、設定エンドポイントで呼出し元を認証する際に、開発者は常にトークンのpermissionsフィールドでSITE_OWNER値を調べるよう注意する必要があります。トークンのpermissionsフィールドには、編集モードの場合のみSITE_OWNER値が表示されます。 編集セッション中に生成されたトークンは、ページ・モデルには永続化されず、permissionsフィールドでNULL値を持つランタイム・トークンで切り替えられます。

#### データ

インスタンスのデータ部分は、Base64 JSON でエンコードされた文字列です。JSON 文字列 の構造を次に示します:

```
{
   "instanceid": "BBDC7614F693B75110D811E6C0B77C935FAEC5112E5E",
   "permissions": "",
   "entitlements": "",
   "signdate": "1435426735293",
   "sitedomain": "service1-tenant4.localhost"
}
```

```
フィールド名説明instanceidOracle Content Management テナントに対するコンポーネントの一意の識<br/>別子。signdate署名の生成日。sitedomainOracle Content Management インスタンスのドメイン名。permissionsサイト・メンバーの権限セット。編集モードでは、値 SITE_OWNER を持ちま<br/>す。そうでない場合、値を持ちません。entitlementsサイト所有者によって購入されたプレミアム機能のリスト。
```

#### 署名

リモート・コンポーネント・インスタンスのデータ部分は、APP_SECRET_KEY によって署名さ れる前にシリアル化されます。コンポーネントの登録時には、この秘密鍵が生成され、開発 者に示される必要があります。署名は、次に示すように、秘密鍵を使用してコンポーネント・



インスタンスのデータ部分(シリアル化された JSON 構造)のハッシュを生成すること によって計算されます:

\$signature = HMAC (serialized JSON structure, APP_SECRET_KEY)

署名の生成に使用されるハッシュ・アルゴリズムは SHA256 です。このトークンは、 次に示すように、シリアル化された JSON 構造と生成された署名コンポーネントを連 結したものです:

\$instance = {base64encoded serialized JSON structure}.
{base64encoded \$signature}

### リモート・コンポーネントの登録

サイトでリモート・コンポーネントを使用するには、それを Oracle Content Management に登録する必要があります。

サードパーティのリモート・コンポーネントおよび自分で開発したリモート・コンポ ーネントを登録できます。

**Oracle Content Management** インスタンスで使用するためにリモート・コンポーネントを登録するには:

- 1. 「開発者」をクリックし、「すべてのコンポーネントの表示」をクリックします。
- 2. 「作成」をクリックして、「リモート・コンポーネントの登録」を選択します。
- 3. 「リモート・コンポーネントの登録」ウィンドウで、次のような情報を入力または 選択します。
  - 名前: ユーザーが参照するコンポーネントの名前。
  - 説明: ユーザーが参照するコンポーネントの説明。
  - コンポーネント URL: ページにコンポーネント・コンテンツをレンダリングするために、iframeで使用されるエンドポイント。HTTPSである必要があります。
  - 設定 URL: ページに追加されるリモート・コンポーネントの設定をレンダリン グするために、iframe で使用されるエンドポイント。HTTPS である必要があ ります。
  - 設定幅: コンポーネント設定パネルのデフォルトの幅を設定します(ピクセル 単位)。
  - 設定高さ: コンポーネント設定パネルのデフォルトの高さを設定します(ピク セル単位)。
  - キー: リモート・コンポーネントに関連付けられ、コンポーネントのプロビジョニング時に署名済ハッシュ・トークンの作成に使用される 192 ビットのAES キー。これは暗号化して、コンポーネント設定のセキュアな読取りおよび書込みを確保するために使用されます。
- 4. 「登録」をクリックします。



リモート・コンポーネントが作成されると、名前がコンポーネントのリストに表示されま す。コンポーネントのリストでコンポーネント名をクリックすると、コンポーネントを登 録するのに使用したファイルを検索できます。

コンポーネントの登録情報は同じ Oracle Content Management インスタンスに作成されたサ イトで使用されるカタログに格納されますが、コンポーネントはリモート・サービスのまま となります。

コンポーネント所有者として、コンポーネント・アイコンは、サイト・ビルダー内の「カス タム・コンポーネント」パネルに、コンポーネントに割り当てた名前で追加されます。コン ポーネントは他のユーザーと共有でき、サイト・ビルダー内の「カスタム・コンポーネント」 パネルにコンポーネントが表示されます。

# コンポーネントの削除

適切な権限を持っている場合、コンポーネント・マネージャからコンポーネントを削除でき、 その結果コンポーネントは使用できなくなります。コンポーネントを削除すると、コンポー ネント・フォルダとその関連フォルダおよびファイルはごみ箱に移動されます。

コンポーネント・マネージャからコンポーネントを削除できるのは、コンポーネントを作成 した(コンポーネントの所有者である)場合か、他のユーザーがあなたとコンポーネントを共有 していて、あなたにマネージャ・ロールを付与している場合です。

/ ノート:

コンポーネントは、サイトまたは更新(ごみ箱内にあるサイトまたは更新を含む)で 使用中である場合、削除できません。

コンポーネントを削除するには:

- 1. ホーム・ページで、「開発者」をクリックします。
- 「すべてのコンポーネントの表示」をクリックします。
   現在登録されているコンポーネントはすべて表示されます。
- コンポーネント名を選択して、右クリック・メニューの「削除」を選択するか、アクション・バーのをクリックします。 アクションを確認するよう求められます。
- 削除アクションを確定するには、「はい」をクリックします。削除アクションを中止する には、「いいえ」をクリックします。 削除を確定すると、コンポーネントとそのすべての関連フォルダおよびファイルはごみ箱 に移動されます。

削除したコンポーネント・フォルダは、次までごみ箱にあります:

- フォルダを復元します。
- フォルダを完全に削除します。
- ごみ箱の割当て制限に達しました。
- ごみ箱は、サービス管理者が設定した間隔に基づいて自動的に空になります。デフォルト 値は 90 日です。



### サイト SDK

Oracle Content Management 用として開発されたコンポーネントは、サイト内でコン ポーネントとしてレンダリングされ、サイトの特定のページ上の任意の場所でドラッ グ・アンド・ドロップできます。

サイト SDK は、コンポーネントとページの間のすべての通信を処理します。

- Sites.Settings.getProperty(propertyName, callbackFunction):カスタム・コンポーネント・インスタンスに対してリクエストされたプロパティを取得するためのコールバックを提供します。
- Sites.Settings.setProperty(propertyName, propertyValue):カスタム・コンポ ーネント・インスタンスに対してリクエストされたプロパティを格納します。

すべてのカスタム・コンポーネント・プロパティに対して Sites.Settings.getProperty または Sites.Settings.setProperty を使用してから、 メッセージ・イベントをリスニングするための SitesSDK.publish および SitesSDK.subscribe を使用します。

サイト SDK リファレンスを参照してください。



# 24 コントローラ・ファイルのカスタマイズ

Oracle Content Management サイトでは、コントローラ・ファイルを使用してサイト内の各 ページが表示されます。ブラウザが Web ページのリクエストを送信すると、サーバーがコン トローラ・ファイルのコピーを使用して応答します。リクエストされたすべてのページにつ いて、コントローラ・ファイルの同じコピーがサーバーから配信されます。

- コントローラ・ファイルについて
- デフォルトの controller.html ファイルの変更
- SCS オブジェクトについて
- カスタマイズできないコントローラ・ファイル・セクション
- トークンを使用したカスタム・コントローラ・ファイルのポータビリティの許可
- カスタム・コントローラ・ファイルのサンプル

### コントローラ・ファイルについて

コントローラ・ファイルは、ページの残りのレンダリング・シーケンスを動的に開始する小 さい HTML ページです。これは、サイトのすべてのページの動作に影響を与えるためにカス タマイズを適用できる最初の場所です。

コントローラ・ファイルの第一のタスクは、コントローラ JavaScript の実行環境を提供しホ ストすることです。コントローラ JavaScript は、続いてページをロードし表示します。コン トローラ・ファイル内のカスタマイズは、コントローラ JavaScript の操作をオーバーライド し影響を与える機能を提供します。

### デフォルト・コントローラ・ファイル

**Oracle Content Management** サイトの初回作成時、デフォルト・コントローラ・ファイルで ある controller.html がサイトに関連付けられます。

```
<!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. -->
<img id="scsWaitImage" style="display: none; margin-top: 5%; margin-
left: auto; margin-right: auto;" src="data:image/png;base64,..." />
</body></html>
```

# デフォルトの controller.html ファイルの変更

サイトの controller.html ファイルをダウンロードし、各サイト・ページの動作に影響す るように変更します。サイトの controller.html ファイルをダウンロードするには:

- 1. 変更するサイトをサイト・ビルダーで開き、「**編集」**に設定します。
- 2. 既存の更新を選択するか、新たに作成します。
- 3. サイド・ナビゲーション・メニューの「設定」、「サイト」の順にクリックします。



 「コントローラ・ファイル」セクションで、「デフォルト・コントローラ・ファイル のダウンロード」をクリックします。



- 5. ローカル・ドライブに保存し、編集を使用して controller.html ファイルをカスタマイズします。
- 6. 編集が完了したらサイトの設定ページに戻り、「コントローラ・ファイル」 セクションで、 「アップロードするファイルの選択」 をクリックします。
- 7. 変更したコントローラ・ファイルに移動してアップロードします。

/ ノート:

コントローラ・ファイルに対するカスタマイズは、オンライン・サイトでのみ使用 されます。サイトの編集時またはサイト・プレビューでは使用されません。サイト のカスタム controller.html ファイルをアップロードした後に変更を反映するには、 サイト更新をコミットし、サイトを公開する必要があります。

## SCS オブジェクトについて

コントローラ JavaScript は、Web ページをレンダリングするために、グローバル SCS オブ ジェクトで定義されている変数およびプロパティを使用します。グローバル JavaScript オブ ジェクトは、controller.js ファイルの包含前にコントローラ・ファイルで定義する必要があり ます。カスタマイズに使用できる SCS オブジェクトには 2 つの主要プロパティがあります:

- SCS.sitePrefix
- SCS.preInitRendering
- SCS.getDeviceInfo

### SCS.sitePrefix

SCS.sitePrefix 変数は、オンライン・サイトのパス接頭辞を定義します。通常、これは site/<*siteName>*/にデフォルト設定されます。

コントローラ JavaScript は、この値を使用して、どの Web ページを表示するかを決定しま す。たとえば、ブラウザが/site/SampleSite/products/index.html のページをリクエストし ている場合、sitePrefix /site/SampleSite/により、JavaScript はサイトの products/ index.html ページを表示するように計算できます。sitePrefix は、文字"/"で始まり、終わる 文字列値である必要があることに注意してください。

プロキシまたは Akamai などの他のインフラストラクチャおよび URL マッピング・ルールが 使用されている場合、この変数を変更することで、サイト接頭辞をカスタマイズできます。 たとえば、sitePrefix を/intranet/ExampleSite/に設定すると、Web サイトをデフォルトの/ site/SampleSite/パス接頭辞ではなく、そのパス名によりブラウザで配信できます。

### N ノート:

ブラウザ内の URL がサイト接頭辞と一致しない場合、内部的に、デフォルト・コントローラ JavaScript は"/"を sitePrefix として使用します。これにより、カスタマイズすることなく、バニティ・ドメインを使用してサイトを提供できます。



### SCS.preInitRendering

定義されている場合、計算ロジックが実行される前に、SCS.preInitRendering 関数が コントローラ JavaScript コードによってコールされます。これは、コントローラ JavaScript の基本操作をオーバーライドするカスタム・コントローラ・ファイルでの 定義で役立つ関数です。

定義されている場合、SCS.preInitRendering が関数である必要があります。これは、 引数なしでコールされ、戻り値は必要なく、処理されません。

### SCS.getDeviceInfo

SCS.getDeviceInfo 関数により、コントローラ JavaScript 内のデバイス検出ロジック のカスタマイズが可能になります。この検出は、ページのモバイル・バージョンまた はレスポンシブ・バージョンのいずれを配信するかどうかの決定に使用されます。

SCS.getDeviceInfo 関数は、引数を取得せず、2 つのプロパティを含む JavaScript を 返します:

- isMobile 現在のデバイスがスマートフォンなどのモバイル・クライアントである ことを示すブール・プロパティ。
- islOS 現在のデバイスが islOS ベースのオペレーティング・システムで実行されていることを示すブール・プロパティ。

カスタム・コントローラが SCS.getDeviceInfo をオーバーライドしない場合、組込み のデフォルト実装が使用されます。

# カスタマイズできないコントローラ・ファイル・ セクション

コントローラ・ファイルの特定のセクションは、CDN を使用して URL キャッシュ・ セグメントを定義するために Oracle Content Management サーバーによって処理され ます。これらは、次のサンプルのインラインで注記されています。

```
<!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. -->
```

```
<img id="scsWaitImage" style="display: none; margin-top: 5%; margin-left:
auto; margin-right: auto;" src="data:image/png;base64,..." />
<!-- The id of the wait image tag should not be changed -->
```

</body></html>

# トークンを使用したカスタム・コントローラ・ファ イルのポータビリティの許可

サイトが作成されると、サイトの名前がコントローラ・ファイルで使用されます。これは、 この名前のサイトにコントローラ・ファイルを関連付け、サイト間でコントローラ・ファイ ルを使用できないようにします。

コントローラ・ファイル内で使用されている名前と異なる名前のサイトでのコントローラ・ ファイルの使用は機能しません。また、これはカスタム・コントローラ・ファイルを含むサ イト・テンプレートから作成されたサイトにも適用されます。動的に評価される次のトーク ンにより、コントローラ・ファイルを複数のサイトで使用できます。

トークン	説明
[!\$SCS_SITE_PREFIX]	これは、現在のサイトのサイト接頭辞に評価されます。値の例は次のと おりです:
	/site/MySite/
	または
	/site/authsite/MySecureSite/



トークン	説明
[!\$SCS_SITE_PATH]	これは、現在の製品 CDN の場所に評価されます。値には末尾の'/'文字 は含まれません。値の例は次のとおりです:
	/site/MySite/_cache_0000
	または
	/site/authsite/MySecureSite
[!\$SCS_PRODUCT_PATH]	これは、キャッシュ・キーなど(該当する場合)現在の製品 URL に評価 されます。値には末尾の'/'文字は含まれません。値の例は次のとおり です:
	https://www.example.com/cdn/cec/v21.1.2.23
	または
	https://www.example.com/cdn/cec/v21.1.3.18

# カスタム・コントローラ・ファイルのサンプル

次のカスタム・コントロール・ファイルのサンプルが提供されています:

- サイト接頭辞の変更
- 待機グラフィックのカスタマイズ
- ファビコンのカスタマイズ
- JavaScript 以外のクローラ用の<noscript>および<meta>タグのカスタマイズ
- JavaScript ファイルのプリフェッチ
- 追加のマークアップによるサイトの所有権の確認
- デバイス検出の補強
- トークンを使用した controller.htm ポータビリティの拡張

### サイト接頭辞の変更

次のサンプル・コントローラ・ファイルは、複数の接頭辞でのサイトの配信を可能に する preInitRendering 関数を定義しています。



🖉 ノート: 🖉

このサンプルを使用するには、CDN または他のプロキシを、定義済接頭辞に応答す るように構成する必要があります。また、関数で定義されている追加の接頭辞のい ずれもブラウザの URL に一致しない場合、デフォルトの動作が使用されます。

```
<!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/'
     1;
     // 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++) {</pre>
           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>
<img id="scsWaitImage" style="display: none; margin-top: 5%; margin-left:</pre>
auto; margin-right: auto;" src="data:image/png;base64,..." />
</body></html>
```

### 待機グラフィックのカスタマイズ

次のサンプル・コントローラ・ファイルは、ネットワーク遅延によりレンダリングが遅延し ている場合に表示される待機グラフィックをカスタマイズします。 🖋 ノート:

このサンプルでは、レンダリング中に個別のリクエストを受けるかわりにイ メージのデータ URL が使用されます。

```
<!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>
<img id="scsWaitImage" style="display: none; margin-top: 5%; margin-</pre>
left: auto; margin-right: auto;" src="
//MAP+Z//+ZzP+Zmf+ZZv+ZM/+ZAP9m//9mzP9mmf9mZv9mM/9mAP8z//
8zzP8zmf8zZv8zM/8zAP8A//8AzP8Amf8AZv8AM/8AAMz//8z/zMz/mcz/Zsz/M8z/AMzM/
8zMzMzMmczMZszMM8zMAMyZ/8yZzMyZmcyZZsyZM8yZAMxm/8xmzMxmmcxmZsxmM8xmAMwz/
8wzzMwzmcwzZswzM8wzAMwA/8wAzMwAmcwAZswAM8wAAJn//5n/zJn/mZn/Zpn/M5n/AJnM/
5nMzJnMmZnMZpnMM5nMAJmZ/5mZzJmZmZmZZZpmZM5mZAJlm/5lmzJlmmZlmZplmM5lmAJkz/
5kzzJkzmZkzZpkzM5kzAJkA/5kAzJkAMZkAZpkAM5kAAGb//2b/zGb/mWb/Zmb/M2b/AGbM/
2bMzGbMmWbMZmbMM2bMAGaZ/2aZzGaZmWaZZmaZM2aZAGZm/2ZmzGZmmWZmZmZmAGYz/
2YzzGYzmWYzZmYzM2YzAGYA/2YAzGYAmWYAZmYAM2YAADP//zP/zDP/mTP/ZjP/MzP/ADPM/
zPMzDPMmTPMZjPMMzPMADOZ/zOZzDOZmTOZZjOZMzOZADNm/zNmzDNmmTNmZjNmMzNmADMz/
zMzzDMzmTMzZjMzMzADMA/zMAzDMAmTMAZjMAMzMAAAD//wD/zAD/mQD/ZqD/MwD/AADM/
wDMzADMmQDMZgDMMwDMAACZ/wCZzACZmQCZZgCZMwCZAABm/wBmzABmmQBmZgBmMwBmAAAz/
wAzzAAzmQAzZqAzMwAzAAAA/
wAAZAAAMQAAZqAAM+4AAN0AALsAAKoAAIqAAHcAAFUAAEQAACIAABEAAADuAADdAAC7AACqA
ACIAAB3AABVAABEAAAiAAAAAAAA7qAA3QAAuwAAqqAAiAAAdwAAVQAARAAAIqAAEe7u7t3d3
bu7u6qqqoiIiHd3d1VVVURERCIiIhEREQAAACH/C05FVFNDQVBFMi4wAwEAAAAh/
h9HaWZCdWlsZGVyIDAuMy4yIGJ5IFl2ZXMqUGlndWV0ACH5BAQAAP8ALAAAAABKACAAAAj9A
FcJHEiwoMGDCBMqXMiwocOHECNKnEixosWLqwBh3HgRkEeOICN6HOkvpEmGI1f9K3my5UGN/
v79c0mTYMyVGWvqjJnxo86aIz2y/
NkyJU6iRVfdfKjRYNOCTwlGHThVYNWcNmWWvOrUJ1WvVsH2nJpSqtiyA3miRRqUEMu2b4PGJ
RmW7lihdfGatcv241Gic1UG/
ut3sGGohRXClElVKWOrjmdCXjr5cU7Kly03xnzw5tHInz0PFZ1WZmjTo033wyqQtMKYQlt/
lh3bcW3YBXFn3QpWN9KOcn+fBCzc5GLJxStGdZ1841rfzUUGj46ROPWOka9fZK6dIvTu4ATD
MwwIACH5BAQKAP8ALAAAAAAAAAAAAAizAFcJHEiwoMGDCBMqXMiwYcN9q/
ZJ1BjRIMSJGC1CJJhxX6CMHCUG0keS5MeJAyWSvMKSJUmMAvetbGnFpT6Uq0a2vNJip76TEP
W1ZBEAQAAWNvUBCtqyKACj0/
09FMrzqVWaUgEBonrFKVQsLKXeDDS0aIAWWARdGXmzn86dLAUJ+rmPFbh/
MrleSTv35j5+dlXqLYkzoki2P4FuNIxRMcWCHWFaNFxxscPLmDNr3syZYEAAIfkEBAOA/
wAsAAAABkAIAAACMQAVwkcSLCgwYMIEypcyLChw4cQI0p8uK+ixX0QKwbayNFiQ40b9YkMp
C9QxYUq9V1ZyXLjSYQqWcpc6RLjQY0zc5q0WXAfoJIsWQQAEIBFS32AeAr0CSifzKEAiMrMB
yqpwapN8wm60iKqV5b5qFYlyNTp1itQiWIBO7YnoECC414RSrQFlq0lk/
YzyKpVK7kz45YMxOrfP4P8+qpSKVOwy8SHDcbUuRPlPo4iBldmeJHjTqWcL6560bG06dOoU6
```

tenTogACH5BAQKAP8ALAQAAAAdACAAAAjpAFcJHEiwoMGDCBMqXMiwocOHECNKjLhv1b6LFy

1KrIixY8aKDjleDESSZMeHGEnqW6mvJMaGKQNdmUnzSsuXC0fKrFnzJsiEKXny9AlyHyCDRlfSZBE AQAAWNFcC4gio6k+jgPLVbArAac18VquK9SdQbL58gmZ2XUvzrNhVYvn9I4tVa9orXL22tWoRED9/ //4JTCqo8BWmTqGmbTmln0DAgckKZNWqlWGhaUmy6if48dyB+/

hRFqSPZ+GWgUJzzrkvUGmhrjMy7FiSpUvZIXXe5ojSoseXPyHy5j2xuPHjyJMrX85cYEAAIfkEBAo A/wAsDAAAAB0AIAAACOgAVwkcSLCgwYMIEypcyLChw4cQI0qMuG/

VvosXLUqsiLFjxooOOV4MRJJkx4cYS+pbWRJjw5SBrsiceUVfIJcLR8akSdNmRoUpefL0+XMfIJAD jdqcySIAgAAsZto8KhCQVaRGAeWj6RTAU5r5rlod66+q1Xz5BMn0ynYm2rFZAfH7VzbrVrVXun51e 9WiXH///

gk0GkiQ4StNn0ZVO3VfP4GAA5cVyKpVq8NCld5k1U8wZLpJ+VUWpI+nYZs3+XUGOrK0UJNIc+oMtB LlzZ8vYbb8SNGjy9ghLf6eSLy48ePIkytfPjAgACH5BAQKAP8ALBQAAAAdACAAAAjoAFcJHEiwoMG DCBMqXMiwocOHECNKjLhv1b6LFy1KrIixY8aKDjleDESSZMeHGEvqW1kSY8OUga7InH1FXyCXC0fG pEnTZkaFKXny9PlzHyCQA43anMkiAIAALGbaPCoQkFWkRgHlo+kUwFOa+a5aHeuvqtV8+QTJ9Mp2J tqxWQHx+1c261a1V7p+dXvVolx///

4JNBpIkOErTZ9GVTt1Xz+BgAOXFciqVavDQtXeZNVPMGS6SflVFqSPp2GbN/

l1BjqytFCTSHPqDLQS9c2fL2G2/EjRo8vYIS3+nki8uPHjyJMrXz4wIAAh+QQECgD/

ACwcAAAAHQAgAAAI5gBXCRxIsKDBgwgTKlzIsKHDhIAeMgREUSJEihEtIsRoMdAqjwYzNgxEsiTJg v7+jSypr6VJgv4YlrxCs+aVkhJJ2rxipeZJhzp37vw5UOTAQID0BarJIgCAACx86gMEkiNBQPny2X QK4KnNrBExUoy5CmM+K4Jodl1b00o+jhj5/YuJNOuVtFe4eq0Jtii/lCo/

UhVE+ErTvWlJRuwnEPDcgaxatSosNHEgVv8Cr0pJViC/

yIL07SSslORnzQlnCsWZk6VL1hpNwtZ49ONBo7SvVsytezfvor5/

C8QtPKVwlMeTJwwIACH5BAQKAP8ALCQAAAAdACAAAAj9AFcJHEiwoMGDCBMqXMiwocNVgPY9VLgPk EWJEw9a30gvI0GJgTbu+9fP476TJ0MGYtWv5EOU+wLJDMSvVT9+FQM5RDkz0JVAgviFtKiT4qqYMq 8oXRrIipWNLhOm9Lm06hWnEf91RDjVqtd8gVr5+3cUkEGVVK+wCAAgAIulgmyykihyYEVA+aqyBdC WKb9+dzcC2roxnxVBSvkqZirTIkTHWsvivYL4yt6+SmXGNAtx1Viyq1QKGq2WbQCliDWfFPg581GW /

PhVtooYLMaBY7cKjN1KUFrUggznAxToNlekvzPnG25RN8WePnuKdC31KHLoIff1+wfaKEyVqyO3c3 e+EOTGovxyexQ82GPBwKuouy/rmft8gq3Jz1d/v//8gAAh+QQEDwD/ACwsAAAAHgAgAAAI/

QBXCRS4r+A+fQgRHiw4sKFDhwYTSlTI8KHFiAmvaLwi0aBFiAcR5rMHwB6LfBwT7lu18iPBkPpIAi iZr2ZKfRVdFkR4ZaZPjTZV5ny4E6HMkliAolTJ8mNRjSTtacTSAujNoQ2f2sRCNSqLKzWZtszKUiR KqD6lbqTYFCTPsz19AlCaEuvLtyhbHFULVOjFkBuvsIha9UrhugP3AWq5M7Dhw3pLfqW4EpDllY0F b2zB0fJMqUwti+7Hb59NFoU7t5A71+0q0ftasTqo+UpS1XsFCVJIEBC/

VsBpg7X9mPPgkld04xzbD7hsmIK4RjbJ2bbu3U0bz+YZeO9h5XZkMVpdLXfj0uUXM571jBRsPrZjX 9KG2+K4VCyC3rvO+vSK18JcKbefW/

r4llZgAqLnFGCs0UWZSwQllNtwfumE0Ug06XZVfDpNdJ2AdlkIkz7X4aQghPwt1FGIKLJk0IsMcfh QQAA7"/>

```
</body></html>
```

### ファビコンのカスタマイズ

次のサンプル・コントローラ・ファイルは、サイトのカスタム・ファビコンを定義します。 これは、ブラウザが、レンダリングされたページのドキュメント・オブジェクト・モデルか らファビコンを動的にロードしない場合に便利です。

```
<!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"></
</pre>
```


#### </head>

```
<body id="scsControllerBody"><noscript>This site requires JavaScript to
be enabled.</noscript>
<img id="scsWaitImage" style="display: none; margin-top: 5%; margin-
left: auto; margin-right: auto;" src="data:image/png;base64,..." />
</body></html>
```

# JavaScript 以外のクローラ用の<noscript>および<meta>タグ のカスタマイズ

次のサンプル・コントローラ・ファイルは、JavaScript を処理しないクローラの <noscript>メッセージおよび<meta>タグをカスタマイズします。

```
✓ ノート:
コントローラ・ファイルが Web サイトの各ページに提供されているので、同
じメタ・タグがすべてのページに表示されます。
```

```
<!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>
<img id="scsWaitImage" style="display: none; margin-top: 5%; margin-</pre>
left: auto; margin-right: auto;" src="data:image/png;base64,..." />
</body></html>
```

### JavaScript ファイルのプリフェッチ

一部のブラウザでは、現在または後続のナビゲーションで必要とされるリソースの宣言が許可されます。コントローラ・ファイル内のマークアップは、プリフェッチまたは事前ロードする必要があるリソースを宣言できます。



### 🖍 ノート:

すべての OCE Web ページでは、require.js および renderer.js が使用されるため、こ れらは事前ロードおよびプリフェッチ技術用の優れた候補です。また、href が <link>タグで最初にリストされている場合、その値は、CDN から配信するために OCE サーバーによって修正されます。

```
<!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"</pre>
rel="preload" as="script">
<link href="/SampleSite/_sitesclouddelivery/renderer/renderer.js"</pre>
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>
<img id="scsWaitImage" style="display: none; margin-top: 5%; margin-left:</pre>
auto; margin-right: auto;" src="data:image/png;base64,..." />
</body></html>
```

# 追加のマークアップによるサイトの所有権の確認

サードパーティ・クローラおよび検索エンジンでは、サイトの所有権を確認するために追加 のマークアップが必要となる場合があります。これは、必要に応じて、コントローラ・ファ イルにインジェクトできます。

```
<!DOCTYPE html>
<html>
<html>
<html>
<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>
<img id="scsWaitImage" style="display: none; margin-top: 5%; margin-left:</pre>
```



auto; margin-right: auto;" src="data:image/png;base64,..." />
</body></html>

## デバイス検出の補強

新規モバイル・デバイスまたはカスタマイズされたデバイス user-agent 識別子の認識 は、コントローラ内のデバイス検出コードのカスタマイズによって実現できます。

```
/ノート:
```

- getDeviceInfo 関数は、モバイル・デバイスが使用されているかどうか を特定するために、コントローラによってコールされます。このサンプ ルは、デフォルトの getDeviceInfo コールをオーバーライドします。
- isMobile プロパティは、ページのレンダリング時に使用されるレイアウトを決定します。
- islOS プロパティは、マップ・コンポーネントでどのプロバイダを使用するかを決定します。
- 組込みのデフォルト isMobile テストは次のとおりです

```
/Mobi|iPhone|iPod|BlackBerry|IEMobile|Opera Mini/
i.test(userAgent) && !/iPad/i.test(userAgent)
```

• 組込みのデフォルト isIOS テストは次のとおりです

/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</pre>
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>
<img id="scsWaitImage" style="display: none; margin-top: 5%; margin-left:
auto; margin-right: auto;" src="data:image/png;base64,..." />
</body></html>
```

## トークンを使用した controller.htm ポータビリティの拡張

デフォルトで、サイト名は controller.html ファイルでハードコードされているため、サイトの 名前変更、または複数のサイトでの同一のカスタム・コントローラの再利用は困難です。ト ークンを使用すると、この問題を緩和できます。次のサンプルは、ポータビリティ・トーク ンの使用方法を示しています。

### ✓ ノート:

サンプルの<link>タグは、デモのみを目的にしています。ベースライン controller.html には、これらは含まれません。

```
<!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/</pre>
controller.js"></script>
<link rel="shortcut icon" href="[!--$SCS_SITE_PATH--]/favicon.ico">
<link rel="preload" href="[!--$SCS_PRODUCT_PATH--]/_sitesclouddelivery/</pre>
renderer/require.js" as="script">
<link rel="preload" href="[!--$SCS_PRODUCT_PATH--]/_sitesclouddelivery/</pre>
renderer/renderer.js" as="script">
</head>
<body id="scsControllerBody"><noscript>Please enable JavaScript to view this
site properly.</noscript>
<img id="scsWaitImage" style="display: none; margin-top: 5%; margin-left:</pre>
auto; margin-right: auto;" src="data:image/png;base64,..." />
</body></html>
```



第∨部

# 他のツールを使用したサイトの開発

**Oracle Content Management** は、**Oracle Content Management** の内部、または他のサード・ パーティ・ツールを使用して **Oracle Content Management** の外部で開発されたアプリケーションおよびエクスペリエンスと対話するためのいくつかの方法を提供します。

#### OCE ツールキットの使用

Oracle Content Management OCE ツールキットおよび SDK は、Oracle Content Management リポジトリで管理されているコンテンツを消費するカスタム・アプリケーショ ンの開発を支援します。これらのアプリケーションは、Oracle Content Management で、ま たはサード・パーティ・ツールを使用して開発できます。

OCE ツールキットは、次のタスクを実行する上で役に立ちます:

- コンポーネント、テンプレート、テーマおよびコンテンツ・レイアウトのローカル開発お よびテスト用として Oracle Content Management インスタンスを使用するようローカル 開発環境を設定します
- サンプルからコンポーネント、サイト・テンプレートおよびコンテンツ・レイアウトを作成し、テスト・ハーネスでこれらを実行して確認し、Developer Cloud Service 環境でコンポーネント、テンプレート、テーマおよびコンテンツ・レイアウトを開発します
- ソース管理および将来の開発のために、Oracle Content Management で作成されたコンポ ーネントおよびサイト・テンプレートを Developer Cloud Service プロジェクトおよび環 境にインポートします
- Oracle Content Management で使用するために、Developer Cloud Service 環境からコン ポーネント、テンプレートおよびコンテンツ・レイアウトをエクスポートします
- 既存のコンポーネント、テンプレートまたはコンテンツ・レイアウトをコピーします
- ユニット・テストを作成します
- コンポーネントを最適化します
- コンポーネントおよびテンプレートを Oracle Content Management にデプロイします

追加の情報は、https://github.com/oracle/content-and-experience-toolkit#readme にあります。

#### エクスペリエンス・オーケストレーション

Oracle Content Management 以外のツールを使用してエクスペリエンスを作成する場合は、 Oracle Content Management リポジトリをこれらのエクスペリエンスに接続することで、コ ンテンツ作成者が作業中にサイトの変更をプレビューし、コンテンツの変更や公開時に自動 ビルドをトリガーできます。このエクスペリエンス・オーケストレーションは、コンテンツ・ プロバイダとサイト開発者の間のワークフロー・プロセスを自動化して、エクスペリエンス の管理と公開を簡素化します。



# 25 OCE ツールキットでの開発

OCE ツールキットは、Oracle Content Management のサイト・テンプレート、テーマ、カス タム・コンポーネントおよびコンテンツ・レイアウトを開発する上で役に立ちます。

OCE ツールキットを使用して開発環境で作業すると、Oracle Content Management のアセット・リポジトリ、ファイルおよびフォルダを使用できます。OCE ツールキットには、テーマおよびコンテンツ・レイアウトを含め、カスタム・コンポーネントおよびサイト・テンプレートを作成および開発するためのツールがあります。これには、迅速かつ反復的に開発するためのローカル・テスト・ハーネスおよび開始するためのサンプル・ユニット・テストが含まれています。

次のトピックでは、OCE ツールキットを設定し、これを使用してローカル・マシンまたは Developer Cloud Service プロジェクトとして開発する方法について説明します:

- ローカル・マシンでの OCE ツールキットの設定
- jQuery 3.5.x へのアップグレード
- Developer Cloud Service による Oracle Content Management の開発
- OCE ツールキットによるテストから本番への変更の伝播
- テンプレートからのサイトの作成およびコンテンツの同じ GUID の維持
- タクソノミのインポートおよびエクスポート
- レコメンデーションのインポートおよびエクスポート
- OCE ツールキットを使用したカスタム・フィールド・エディタの開発
- あるサーバーから別のサーバーへのサイトの転送または更新
- OCE ツールキットによるサイト・ページの索引付け
- OCE ツールキットによる多言語サイトの索引付け
- コンポーネント開発を容易にするための簡潔なコンポーネントの作成
- サイトのコンパイルによるサイト・ページのランタイム・パフォーマンスの改良
- Oracle Content Management サーバーでの新規のサイトまたはアセットの翻訳ジョブの 作成
- 言語サービス・プロバイダでのサイトの翻訳

# ローカル・マシンでの OCE ツールキットの設定

GitHub からローカル・コンピュータに OCE ツールキットをダウンロードできます。

OCE ツールキットを設定するには、次の場所にある手順に従います

https://github.com/oracle/content-and-experience-toolkit/blob/master/
README.md



設定を完了して、OCE ツールキットの使用準備をします:

- 1. npm を介した依存関係のインストール
- 2. cec コマンドライン・ユーティリティの使用
- 3. ローカル・テスト・ハーネスを使用したテスト

### npm を介した依存関係のインストール

**npm (**ノード・パッケージ・マネージャ**)**を使用して、プロジェクトのサイトの依存関係 をインストールします。

プロキシを使用してインターネットにアクセスしている場合、npm のプロキシを npm config コマンドで設定します。https://docs.npmjs.com/misc/config を参照してください。bower にプロキシを設定するには、https://bower.io/docs/config/を参照してください。

# N ノート:

ローカル・コンピュータに Node.js 8.0.0 以降(https://nodejs.org/)がインスト ールされていることを確認してください。

### cec コマンドライン・ユーティリティの使用

cec クロスプラットフォーム・コマンドライン・ユーティリティは、テンプレートおよ びコンポーネントを作成および管理するコマンドを提供します。

cec コマンドライン・ユーティリティを使用する前に、Oracle Content Management サ ーバーへの接続が必要なコマンドのためにローカル開発環境でソースを作成します。

統合ヘルプでは、コマンドを実行するための情報が例とともに提供されます。cec 統合 ヘルプを表示するには、コマンドラインで cec コマンドを入力します。

- ターミナル・ウィンドウで、cec ディレクトリに移動します。
- cec と入力して、cec コマンドをリストします。

cec share-folder <name>
users and groups on OCM server.

cec ______ Usage: cec <command> [options] Run cec <command> -h' to get the detailed help for the command. Commands: Documents cec create-folder <name> Creates a folder or folder hierarchy on OCM server. [alias: cfd

[alias: cfd] [alias: sfd]



cec unshare-folder <name> access to a shared folder on OCM server. cec list-folder <path> hierarchy on OCM server. cec download-folder <path> OCM server. cec upload-folder <path> server. cec delete-folder <path> server. cec download-file <file> from OCM server. cec upload-file <file> OCM server. cec delete-file <file> server. Components cec create-component <name> <name>. cec copy-component <source> [<destination>] component named <source> to <destination>. cec import-component <zip> from <zip>. cec export-component <name> <name> as a zip file. cec download-component <names> components <names> from the OCM server. cec upload-component <names> <names> to the OCM server. cec control-component <action> <action> on components on OCM server. cec share-component <name> users and groups on OCM server.

cec unshare-component <name> access to a component on OCM server.

#### Templates

```
cec create-template <name>
<name>.
  cec create-template-from-site <name>
<name> from site <site> on the OCM server.
  cec download-template <name>
<name> from the OCM server.
  cec compile-template <source>
within the template.
 cec copy-template <source> [<destination>]
template named <source> to <destination>.
  cec import-template <zip>
<zip>.
  cec export-template <name>
<name> as a zip file.
  cec upload-template <name>
<name> to the OCM server.
  cec delete-template <name>
```

Deletes user or group [alias: usfd] Displays folder [alias: lfd] Downloads folder from [alias: dlfd] Uploads folder to OCM [alias: ulfd] Deletes folder on OCM [alias: 1 Downloads file <file> [alias: dlf] Uploads file <file> to [alias: ulf] Deletes file on OCM [alias: 1

Creates the component [alias: cc] Copies an existing [alias: cpc] Imports a component [alias: ic] Exports the component [alias: ec] Downloads the [alias: dlcp] Uploads the components [alias: ulcp] Performs action [alias: ctcp] Shares component with [alias: sc] Deletes user or group [alias: usc]

Creates the template [alias: ct] Creates the template [alias: ctfs] Downloads the template [alias: dlt] Compiles the site [alias: cmpt] Copies an existing [alias: cpt] Imports a template from [alias: it] Exports the template [alias: et] Uploads the template [alias: ult] Deletes the template



<name> on the OCM server. [alias: ] cec share-template <name> Shares template with users and groups on OCM server. [alias: stm] cec unshare-template <name> Deletes user or group access to a template on OCM server. [alias: ustm] cec update-template <action> Performs action on [alias: ut] a local template. cec describe-template <name> Describes the template <name> package. [alias: dst] cec create-template-report <name> Generates an asset usage report for the template <name> package. [alias: cttr] Themes Adds a component cec add-component-to-theme <component> to a theme. [alias: actt] cec remove-component-from-theme <component> Removes a component from a theme. [alias: rcft] cec control-theme <action> Performs action <action> on theme on OCM server. [alias: ctt] Shares theme with cec share-theme <name> users and groups on OCM server. [alias: sth] cec unshare-theme <name> Deletes user or group access to a theme on OCM server. [alias: usth] Sites cec create-site <name> Creates Enterprise Site <name>. [alias: cs] cec copy-site <name> Copies Enterprise Site <name>. [alias: cps] cec update-site <name> Update Enterprise Site <name>. [alias: us] cec transfer-site <name> Transfers a site from one OCM server to another. [alias: ts] cec transfer-site-content <name> Creates scripts to transfer site content from one OCM server to another. [alias: tsc] cec validate-site <name> Validates site <name>. [alias: vsl cec control-site <action> Performs action <action> on site on OCM server. [alias: cts] cec share-site <name> Shares site with users and groups on OCM server. [alias: ssl cec unshare-site <name> Deletes user or group access to a site on OCM server. [alias: uss] cec get-site-security <name> Gets site security [alias: gss] on OCM server. cec set-site-security <name> Sets site security on OCM server. [alias: sss] cec index-site <site> Index the page content of site <site> on OCM server. [alias: is] cec create-site-map <site> Creates a site map for site <site> on OCM server. [alias: csm] cec create-rss-feed <site> Creates RSS feed for site <site> on OCM server. [alias: crf]

cec create-asset-report <site> 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. Assets

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.

#### Content

```
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>
users and groups on OCM server.
  cec unshare-channel <name>
access to a channel on OCM server.
  cec create-localization-policy <name>
policy on OCM server.
  cec list-server-content-types
from server.
  cec share-type <name>
and groups on OCM server.
  cec unshare-type <name>
```

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]

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]

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 [alias: sch] Deletes user or group [alias: usch] Creates a localization [alias: clp] Lists all content types [alias: lsct] Shares type with users [alias: st] 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 [alias: recommendation from the OCM server. dlrl 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] Creates content cec add-contentlayout-mapping <contentlayout> 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] Removes a field cec remove-field-editor <name> 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 l cec compile-content Compiles the content items generating HTML renditions. [alias: cmpc] cec upload-compiled-content <path> Uploads the compiled content to OCM server. [alias: ulcc] Taxonomies Downloads a cec download-taxonomy <name> 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. [alias: ltj] cec create-translation-job <name> Creates a translation job <name> for a site on OCM server. [alias: ctj] cec download-translation-job <name> Downloads translation job <name> from OCM server. [alias: dtj] cec submit-translation-job <name> Submits translation job <name> to translation connection <connection>. [alias: stj] cec refresh-translation-job <name> Refreshes

translation job <name> from translation connection. cec ingest-translation-job <name> Get <name> from translation connection and ingest. [a cec upload-translation-job <name> Upl <name> to OCM server. [alias cec create-translation-connector <name> Cre connector <name>. [a cec start-translation-connector <name> Sta connector <name>. [ cec register-translation-connector <name> Reg connector. [alias

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.

```
.. [alias: rtj]
Gets translated job
[alias: itj]
Uploads translation job
[alias: utj]
Creates translation
[alias: ctc]
Starts translation
[alias: stc]
Registers a translation
[alias: rtc]
```

Creates an OCM group on [alias: cg] Deletes an OCM group on [alias: ] Adds users and groups [alias: amtg] Removes users and [alias: rmfg]

Create an encryption [alias: cek] Registers a OCM [alias: rs] Set OAuth token for a [alias: sot] Lists local or server [alias: 11 Makes an HTTP GET [alias: exeg] Creates source [alias: i] Starts a test [alias: d] Starts a sync [alias: scs] Starts a webhook [alias: whs]

```
Options:
--version, -v Show version number [boolean]
--help, -h Show Help
```

```
-----
cec create-folder
-----
Usage: cec create-folder <name>
```

cec webhook-server

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:

server.



```
--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:Compl/assets
_____
cec download-folder
_____
Usage: cec download-folder <path>
Downloads folder and all its content from OCM server. Specify the server
with -s <server> or use the one specified in cec.properties file. Optionally
specify -f <folder> to save the folder on the local system.
Options:
  --help, -h
               Show Help
  --folder, -f <folder> Local folder to save the folder on OCM server
  --server, -s <server> The registered OCM server
Examples:
  cec download-folder Releases/1
                                                          Downloads folder
Releases/1 from OCM server and save to local folder src/documents/
  cec download-folder /
                                                          Downloads all
documents from OCM server and save to local folder src/documents/
  cec download-folder Releases/1 -s UAT
                                                          Downloads folder
Releases/1 from the registered server UAT and save to local folder src/
documents/
```



cec download-folder Releases/1 -f ~/Downloads Downloads folder Releases/1 from OCM server and save to local folder ~/Download/ cec download-folder Releases/1 -f . Downloads folder Releases/1 from OCM server and save to the current local folder cec download-folder site:blog1 -f ~/Downloads/blog1Files Downloads all files of site blog1 and save to local folder ~/Download/blog1Files cec download-folder theme:blog1Theme Downloads all files of theme blog1Theme and save to local folder src/documents/ blog1Theme/ cec download-folder component:Comp1/assets Downloads all files in folder assets of component 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 bloq1Theme 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> The registered OCM server
  --server, -s
  --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:Compl/assets/render.js
                                                               Downloads
the js file from folder assets of component Compl and save to local folder
src/documents/Comp1/assets/
_____
cec upload-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/
           Uploads the file to folder settings/misc of site blog1
misc
  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/comp1.js -f component:Comp1/
assets
                   Uploads the js file to folder assets of component
Comp1
_____
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 Compl
 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 Comp1 Copies Sample-To-Do to Comp1.
------
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> Folder to upload the component zip file
  --folder, -f
```



```
--publish, -p Publish the component
                <server> The registered OCM server
  --server, -s
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
Compl on the server specified in cec.properties file
 cec control-component publish -c Comp1 -s UAT
                                                     Publish component
Compl 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 Compl with user user1 and user2 and assign Manager role to them cec share-component Compl -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 Compl 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 Compl -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 Templ -s Sitel -r UAT Create template Temp1 based on site Site1 on the registered server UAT cec create-template EnterpriseTempl -s StandardSitel -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> The registered OCM server --server, -r 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 -1 <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:
                                Show Help
  --help, -h
                                The registered OCM server
  --server, -s
  --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
specifed in the page list
  --debug, -d
                                Start the compiler with "--inspect-
brk" option to debug compilation
                                Do not generate compiled detail pages
  --noDetailPages, -e
  --noDefaultDetailPageLink, -o Do not generate compiled detail page
for items/content lists that use the default detail page
                                The target device type when using
  --targetDevice, -a
```

```
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, -1
                               Include default locale when creating pages
  --verbose, -v
                               Run in verbose mode to display all warning
messages during compilation.
  --iqnoreErrors, -i
                               Ignore compilation errors when calculating
the exit code for the process.
Examples:
  cec compile-template Temp1
                                                           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 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 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 Temp1 Temp2 Copies Temp1 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 Templ Exports the template Templ. _____ cec upload-template _____ Usage: cec upload-template <name> Uploads the template <name> to the Content Management server. Specify the server with -s <server> or use the one specified in cec.properties file. Optionally specify -f <folder> to set the folder to upload the template zip file. Optionally specify -p to publish theme and components after import. Options: --help, -h Show Help --folder, -f <folder> Folder to upload the template zip file --server, -s <server> The registered OCM server --optimize, -o Optimize the template --excludecontenttemplate, -x Exclude content template --excludecomponents, -e Exclude components --publish, -p Publish theme and components Examples: cec upload-template StarterTemplate Uploads the template StarterTemplate. cec upload-template StarterTemplate -s UAT Uploads the template StarterTemplate to the registered server UAT. cec upload-template StarterTemplate -f Import/Templates Uploads file StarterTemplate.zip to folder Import/Templates and imports the template StarterTemplate. cec upload-template StarterTemplate -p Publish the theme and all components in StarterTemplate.zip after import cec upload-template StarterTemplate -o Optimizes and uploads the template StarterTemplate. cec upload-template StarterTemplate -x Exclude the "Content Template" from the template upload. "Content Template" upload can be managed independently. cec upload-template StarterTemplate -e Exclude all components from the template upload. Components can be uploaded independently. _____ cec delete-template

_____

Usage: cec delete-template <name>



```
Deletes the template <name> on the Content Management server. Specify the
server with -s <server> or use the one specified in cec.properties file.
Optionally specify -p to permanently delete the template.
Options:
  --help, -h
                  Show Help
  --server, -s
                  <server> The registered OCM server
  --permanent, -p flag to indicate to permanently delete the template
Examples:
 cec delete-template BlogTemplate
 cec delete-template BlogTemplate -p
 cec delete-template BlogTemplate -s UAT
_____
cec share-template
------
Usage: cec share-template <name>
Shares template with users and groups on OCM server and assign a role.
Specify the server with -s <server> or use the one specified in
cec.properties file. The valid roles are
 manager
 contributor
 downloader
  viewer
Options:
  --help, -h Show Help
  --users, -u The comma separated list of user names
  --groups, -g The comma separated list of group names
  --role, -r The role [manager | contributor | downloader | viewer] to
assign to the users or groups [required]
  --server, -s <server> The registered OCM server
Examples:
 cec share-template Template1 -u user1, user2 -r manager
Share template Template1 with user user1 and user2 and assign Manager role
to them
  cec share-template Template1 -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 Template1 -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: Show Help --help, -h --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 Template1 -u user1, user2 cec unshare-template Template1 -u user1, user2 -g group1, group2 cec unshare-template Template1 -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 Template1 cec update-template rename-asset-id -t Template1 -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 -
                                       The report will be saved to
0
StarterTemplateAssetUsage.json at the current local location
  cec create-template-report StarterTemplate -o ~/
Documents
                                   The report will be saved to ~/
Documents/StarterTemplateAssetUsage.json
  cec create-template-report StarterTemplate -o ~/Documents/
StarterTemplateReport.json The report will be saved to ~/Documents/
StarterTemplateReport.json
  cec create-template-report StarterTemplate -
i
                                       Include validating page links
_____
cec add-component-to-theme
_____
Usage: cec add-component-to-theme <component>
Adds a component to a theme. Optionally specify -c <category> to set the
component category.
Options:
              Show Help
  --help, -h
               <theme> Theme [required]
  --theme, -t
  --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 Themel with user user1 and user2 and assign Manager role to
them
  cec share-theme Themel -u user1, user2 -g group1, group2 -r manager
Share theme Themel 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> Default language, required for
  --defaultLanguage, -d
enterprise site
  --description, -p
                          <description> Site description
  --sitePrefix, -x
                          <sitePrefix> Site Prefix
                          Keep the existing id for assets
  --update, -u
  --server, -s
                          <server> The registered OCM server
Examples:
  cec create-site Sitel -t StandardTemplate
Creates a standard site
 cec create-site Sitel -t Templatel -r Repository1 -l L10NPolicy1 -d en-US
Creates an enterprise site with localization policy L10NPolicy1
  cec create-site Sitel -t Templatel -r Repository1 -d en-US
Creates an enterprise site and uses the localization policy in Template1
 cec create-site Sitel -t Templatel -r Repository1 -d en-US -s UAT
Creates an enterprise site on server UAT
 cec create-site Sitel -t Templatel -u -r Repositoryl -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 site [required]
  --target, -t
  --repository, -r Repository, required for enterprise site
  --description, -d Site description
  --sitePrefix, -x Site Prefix
                   The registered OCM server
  --server, -s
Examples:
  cec copy-site Sitel -t SitelCopy
                                                          Copies a
standard site
 cec copy-site Site1 -t Site1Copy -r Repository1
                                                          Copies an
enterprise site
 cec copy-site Sitel -t SitelCopy -r Repositoryl -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 Sitel using the
content from template Template1
  cec update-site Sitel -t Templatel -x Updates site Sitel 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:
```

--help, -h

Show Help

```
The registered OCM server the site is from
  --server, -s
[required]
  --destination, -d
                            The registered OCM server to create or update
the site [required]
  --repository, -r
                            Repository, required for creating enterprise site
  --localizationPolicy, -l Localization policy, required for creating
enterprise site
  --sitePrefix, -f
                            Site prefix
  --publishedassets, -p
                            The flag to indicate published assets only
  --repositorymappings, -m The repositories for assets from other
repositories
  --excludecontent, -x
                           Exclude content
  --excludecomponents, -e Exclude components
  --excludetheme, -c
                           Exclude theme
  --excludetype, -t
                           Exclude content types
  --includestaticfiles, -i Include site static files
  --suppressgovernance, -g Suppress site governance controls
Examples:
  cec transfer-site Sitel -s DEV -d UAT -r Repository1 -l
L10NPolicy1
Creates site Sitel on server UAT based on site Sitel on server DEV
  cec transfer-site Sitel -s DEV -d UAT -r Repository1 -l L10NPolicy1 -
                                                           Creates site
р
Sitel on server UAT based on site Sitel on server DEV with published assets
  cec transfer-site Site1 -s DEV -d UAT -r Repository1 -l L10NPolicy1 -
                                                           Creates site
х
Sitel on server UAT based on site Sitel on server DEV without content
  cec transfer-site Site1 -s DEV -d UAT -r Repository1 -l L10NPolicy1 -
е
                                                           Creates site
Sitel on server UAT based on site Sitel on server DEV without transferring
components to server UAT
  cec transfer-site Sitel -s DEV -d UAT -r Repository1 -l L10NPolicy1 -e -
С
                                                        Creates site Site1
on server UAT based on site Sitel on server DEV without transferring
components and theme to server UAT
  cec transfer-site Sitel -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 Sitel on server DEV and transfter the assets
from repository Shared Images and Shared Video
  cec transfer-site Sitel -s DEV -d UAT -r Repository1 -l L10NPolicy1 -
i
                                                           Creates site
Sitel on server UAT based on site Sitel on server DEV with static files from
DEV
  cec transfer-site Site1 -s DEV -d
UAT
                 Updates site Sitel on server UAT based on site Sitel 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:
```

help, -h	Show Help
server, -s	The registered OCM server the site is
from [required]	
destination, -d	The registered OCM server to transfer the
content [required]	
repository, -r	The site repository [required]
publishedassets, -p	The flag to indicate published assets only
addtositecollection, -l	Add assets to the site collection
repositorymappings, -m	The repositories for assets from other
repositories	
number, -n	The number of items in each batch,
defaults to 500	
execute, -e	Execute the scripts

```
Examples:
```

cec transfer-site-content Sitel -s DEV -d UAT -r Repository1 Generate script Site1_downloadcontent and Site1_uploadcontent cec transfer-site-content Site1 -s DEV -d UAT -r Repository1 е Generate script Site1_downloadcontent and Site1_uploadcontent and execute them cec transfer-site-content Sitel -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 -Only the р published assets will be transferred cec transfer-site-content Sitel -s DEV -d UAT -r Repository1 -1 The assets from the site repository will be added to site default collection on destination server cec transfer-site-content Sitel -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 Sitel 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> Site [required]
  --site, -s
  --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> The registered OCM server
  --server, -r
Examples:
  cec control-site publish -s Site1
                                                Publish site Sitel on the
server specified in cec.properties file
 cec control-site publish -s Sitel -u
                                                Publish the site and all
assets added to the site's pages
  cec control-site publish -s Sitel -c
                                                Compile and publish site
Site1
  cec control-site publish -s Sitel -t
                                                Only publish the static
files of site Site1
 cec control-site publish -s Site1 -f
                                                Do a full publish of Site1
                                                Publish site Sitel on the
  cec control-site publish -s Site1 -r UAT
registered server UAT
  cec control-site unpublish -s Sitel -r UAT
                                                Unpublish site Site1 on the
registered server UAT
  cec control-site bring-online -s Sitel -r UAT Bring site Sitel online on
the registered server UAT
  cec control-site take-offline -s Sitel -r UAT Take site Sitel 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 Sitel -u userl, user2 -r manager Share site Sitel 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 Sitel with user user1 and user2 and group group1 and group2 and assign Manager role to them cec share-site Sitel -u user1, user2 -r manager -s UAT Share site Sitel with user userl 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 Sitel -u user1, user2 cec unshare-site Sitel -u user1, user2 -g group1, group2 cec unshare-site Sitel -u userl, 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 Sitel
 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 Sitel -s no
make the site publicly available to anyone
 cec set-site-security Sitel -s no -r UAT
make the site publicly available to anyone on server UAT
 cec set-site-security Sitel -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 userl
```


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</contenttype>
publish, -p	publish page index items
server, -s	<server> The registered OCM server</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:

--help, -h Show Help

--url, -u <url> Site URL [required]

--changefreq, -c How frequently the page is likely to

change.
```



--file, -f Name of the generated site map file --languages, -l <languages> The comma separated list of languages used to create the site map --publish, -p Upload the site map to OCM server after creation --toppagepriority, -t Priority for the top level pages, a decimal number between 0 and 1 --server, -s <server> The registered OCM server --newlink, -n Generate new 19.3.3 detail page link --noDefaultDetailPageLink, -o Do not generate detail page link for items/ content lists that use the default detail page Examples: cec create-site-map Site1 -u http://www.example.com/site1 cec create-site-map Sitel -u http://www.example.com/sitel -s UAT cec create-site-map Site1 -u http://www.example.com/site1 -t 0.9 cec create-site-map Sitel -u http://www.example.com/sitel -f sitemap.xml cec create-site-map Site1 -u http://www.example.com/site1 -p cec create-site-map Sitel -u http://www.example.com/sitel -c weekly -p cec create-site-map Sitel -u http://www.example.com/sitel -l de-DE,it-IT _____ cec create-rss-feed _____ Usage: cec create-rss-feed <site>

Creates RSS feed for site <site> on OCM server. Specify the server with -s <server> or use the one specified in cec.properties file. Optionally specify -x <template> to specify the RSS template. Optionally specify -p to upload the RSS feed to OCM server after creation.

```
Options:
```

help, -h	Show Help
url, -u	<url> Site URL [required]</url>
query, -q	Query for content items [required]
limit, -l	The limit of the items returned from the query
[required]	
orderby, -o	The order by for the query [required]
language, -i	The language for the query
template, -x	The RSS xml template
javascript, -j	Javascript file that contains functions to process
Mustache data	
title, -t	The RSS feed title
description, -d	The RSS feed description
ttl	How long the data will last in number of minutes
file, -f	Name of the generated RSS feed file
publish, -p	Upload the RSS feed to OCM server after creation
server, -s	<server> The registered OCM server</server>
newlink, -n	Generate new 19.3.3 detail page link

## Examples:

```
cec create-rss-feed Site1 -u http://www.example.com/site1 -q 'type eq
"BlogType"' -l 10 -o name:asc -t "Blog RSS"
    cec create-rss-feed Site1 -u http://www.example.com/site1 -q 'type eq
"BlogType"' -l 10 -o name:asc -t "Blog RSS" -x ~/Files/RSSTemplate.xml
```

```
cec create-rss-feed Sitel -u http://www.example.com/sitel -q 'type eq
"BlogType"' -1 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 Sitel
 cec create-asset-report Sitel -s UAT
 cec create-asset-report Sitel -o
                                                             The
report will be saved to SitelAssetUsage.json at the current local
location
 cec create-asset-report Sitel -o ~/Documents
                                                             The
report will be saved to ~/Documents/SitelAssetUsage.json
 cec create-asset-report Sitel -o ~/Documents/SitelReport.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
            The site on OCM server [required]
  --site, -s
  --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
                    Site name
  --name, -n
  --description, -p Site description
  --sitePrefix, -x Site Prefix
Examples:
 cec migrate-site Sitel -s ICServer -d ECServer -r
Repo1
                           Migrates site Sitel from ICServer to
ECServer
  cec migrate-site Sitel -s ICServer -d ECServer -r Repol -n
newSite
                  Migrates site Sitel from ICServer to ECServer and
rename to newSite
 cec migrate-site Sitel -d ECServer -t ~/Documents/SitelTemplate.zip -
r Repol Migrates site Sitel to ECServer with template
SitelTemplate.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 to fetch the assets
  --query, -q
  --assets, -a
                        The comma separated list of asset GUIDS
                        The file with an array of asset GUIDS
  --assetsfile, -f
                        The name for this download, default to the
  --name, -n
channel or repository name
  --server, -s
                        The registered OCM server
Examples:
  cec download-content
Site1Channel
Download all assets in channel SitelChannel and save to local folder
src/content/Site1Channel
  cec download-content SitelChannel -n
SitelAssets
                                                      Download all
assets in channel SitelChannel and save to local folder src/content/
SitelAssets
  cec download-content SitelChannel -
                                                        Download
р
```

published assets in channel SitelChannel cec download-content SitelChannel -s UAT Download all assets in channel SitelChannel on server UAT cec download-content SitelChannel -q 'fields.category eq "RECIPE"' Download assets from the channel Site1Channel, matching the query, plus any dependencies cec download-content SitelChannel -r Repol -c Collection1 Download assets from the repository Repol, collection Collection1 and channel Site1Channel cec download-content Site1Channel -r Repo1 -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 -1 <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> The channel to add the content --channel, -c --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 Upload content types and taxonomies only --types, -p Examples: cec upload-content SitelChannel -r Repol Upload content to repository Repol, creating new items, and add to channel SitelChannel cec upload-content SitelChannel -r Repol -u Upload

content to repository Repol, updating existing content to create new versions, and add to channel SitelChannel

cec upload-content SitelChannel -r Repol -l SitelCollection Upload content to repository Repol and add to collection SitelCollection and

```
channel SitelChannel
  cec upload-content SitelChannel -r Repol -p
Upload content types from content SiteChannel to the server
  cec upload-content SitelChannel -r Repol -s UAT
Upload content to repository Repol on server UAT and add to channel
SitelChannel
  cec upload-content Template1 -t -r Repo1 -c channel1
Upload content from template Template1 to repository Repol 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 Repol
and add to channel channel1
cec control-content
_____
Usage: cec control-content <action>
Performs action <action> on channel items on OCM server. Specify the
channel with -c <channel>. Specify the server with -s <server> or use
the one specified in cec.properties file. The valid actions are
  publish
  unpublish
  add
  remove
Options:
  --help, -h
                    Show Help
  --channel, -c
                   Channel
  --repository, -r Repository, required when <action> is add
  --collection, -l Collection
  --query, -q
                   Query to fetch the assets
  --assets, -a
                   The comma separated list of asset GUIDS
  --server, -s
                   The registered OCM server
  --date, -d
                   Date to publish items
  --name, -n
                   Name of the scheduled publishing job to create
Examples:
  cec control-content publish -c
Channel1
                     Publish all items in channel Channell 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 Channell on the registered server
UAT
  cec control-content unpublish -c Channel1 -s
UAT
       Unpublish all items in channel Channell on the registered server
UAT
```

cec control-content add -c Channell -r Repol -s UAT Add all items in repository Repol to channel Channell on the registered server UAT cec control-content add -c Channell -r Repol -q 'type eq "BlogType"' -s Add all items in UAT repository Repol, matching the query to channel Channell 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 Channell -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 Channell on the registered server UAT cec control-content add -l Collection1 -r Repo1 -s Add UAT all items in repository Repol to collection Collection1 on the registered server UAT cec control-content remove -1 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" -Create a publishing job n Name 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: --help, -h Show Help --server, -s The registered OCM server the content is from [required]

--destination, -d The registered OCM server to transfer the content
[required]
--channel, -c The channel
--publishedassets, -p The flag to indicate published assets only
--number, -n 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 Channell in repository Repository1 cec transfer-content Repository1 -s DEV -d UAT -c Channel1 -p Transfer the items published to channel Channell 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 name --channel, -c --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: List all assets cec list-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 Channell List all assets from channel Channel1 cec list-assets -r Repol -l Collection1 List all assets from collection Collection1 and repository Repo1 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 ^{\odot} 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]
                   The slug for the asset when create a single asset
  --slug, -l
  --attributes, -a The JSON file of asset attributes
                   The registered OCM server
  --server, -s
Examples:
 cec create-digital-asset -f ~/Documents/logo.jpg -t Image -r
Repo1
                                                      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 Repo1 -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
Repo1
                                                        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
  --sluq, -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 -1
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 -1 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
--assets, -a
--target, -t
                   Query to fetch the assets
                   The comma separated list of asset GUIDS
                   The target repository [required]
  --server, -s
                   The registered OCM server
Examples:
 cec copy-assets Repol -t
Repo2
                                                   Copy all assets in
repository Repol to Repo2
  cec copy-assets Repol -t Repo2 -s
UAT
                                          Copy all assets in
repository Repol to Repo2 on server UAT
 cec copy-assets Repol -a GUID1,GUID2 -t
Repo2
                                    Copy asset GUID1 and GUID2 and all
their dependencies in Repol to Repo2
 cec copy-assets Repol -q 'fields.category eq "RECIPE"' -t
Repo2
                  Copy assets from repository Repol, matching the
query, plus any dependencies to Repo2
 cec copy-assets Repol -c Channell -t
Repo2
                                       Copy assets from the repository
Repol and channel Channell to Repo2
 cec copy-assets Repol -1 Collection1 -t
Repo2
                                    Copy assets from the repository
Repol and collection Collection1 to Repo2
 cec copy-assets Repol -c Channell -q 'fields.category eq "RECIPE"' -t
Repo2 Copy assets from repository Repo1, channel Channell, 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>

channel1, channel2 -1 en-US -s UAT

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
                         The description for the repository
  --description, -d
                         The repository type [asset | business]. Defaults to
  --type, -p
asset
                         The comma separated list of content types for the
  --contenttypes, -t
repository
                         The comma separated list of publishing channels to
  --channels, -c
use in this repository
  --defaultlanguage, -1 The default language
  --server, -s
                         The registered OCM server
Examples:
  cec create-repository Repol
  cec create-repository BusinessRepo -p business
  cec create-repository Repo1 -d "Blog Repository" -t BlogType, AuthorType -c
```



```
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
                 The comma separated list of user names
  --users, -u
                 The comma separated list of group names
 --groups, -g
                 The role [manager | contributor | viewer] to assign to the
  --role, -r
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 userl, 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:
```

help, -h	Show Help
description, -d	The description for the channel
type, -t	The channel type [public   secure]
publishpolicy, -p	The publish policy [anythingPublished
onlyApproved]	
localizationpolicy, -l	The localization policy for the channel
server, -s	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 channell with localization policy en-fr and everything can be published
```

cec create-channel channell -t secure -p onlyApproved Create secure channel channell and only approved items can be published

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: --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] --server, -s <server> The registered OCM server



Examples: cec share-channel Channell -u user1, user2 -r manager Share channel Channell with user user1 and user2 and assign Manager role to them cec share-channel Channell -u user1, user2 -g group1, group2 -r manager Share channel Channell with user user1 and user2 and group group1 and group2 and assign Manager role to them cec share-channel Channell -u user1, user2 -r manager -s UAT Share channel Channell 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 Channell -u user1, user2 cec unshare-channel Channell -u user1, user2 -g group1, group2 cec unshare-channel Channell -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 -1 <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 --required languages, -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:
                                      Download content type BlogType and
 cec download-type BlogType
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
                       the content form [required]
  --objectname, -o
 --contenttype, -c the content type [required]
--template, -t The tormlate type
                        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 -
                            Associate content form form1 with content
s
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
                       Change not to use form1 when create or edit
UAT
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 Repo1 -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
                          The registered OCM server
  --server, -r
                          <style> Content layout style: detail | overview
  --style, -s
  --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
                    Add custom settings when used in Sites
BlogTemplate -a
  cec create-contentlayout Blog-Post-Overview-Layout -c Blog-Post -
                       Use content type Blog-Post from the server
r
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 mapping
  --mobile, -m
                    <server> The registered OCM server
  --server, -r
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]
  --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 typ
--contenttype, -c The content type [required]
                        The template the content type is from [required]
                         The field the field editor is for [required]
  --field, -f
  --contenttemplate, -n Flag to indicate the template is a content template
Examples:
  cec remove-field-editor editor1 -t BlogTemplate -c BlogPost -f
```

```
Remove editor1 as the appearance for field summary
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 Repo1 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:
  --help, -h
                       Show Help
  --server, -s
                       The registered OCM server
                       The comma separated list of asset GUIDS
  --assets, -a
                       Compile all the published assets of this content
  --contenttype, -t
type.
  --repositoryId, -i
                       Id of the repository for content type queries.
  --renditionJobId, -r Server invoked rendition job id for a publishing job
  --debug, -d
                       Start the compiler with "--inspect-brk" option to
debug compilation
  --verbose, -v
                       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: cpublishingJobId> 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:
  --help, -h
               Show Help
  --server, -s 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
                    The name of the new taxonomy
  --name, -n
  --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 -
С
                                           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
                    The registered OCM server
  --server, -s
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 jobl -s Sitel -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 SitelJob
 cec download-translation-job SitelJob -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 SitelJob1 -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 SitelJob1
  cec refresh-translation-job SitelJob1 -s UAT Refresh translation job
SitelJobl 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 SitelJobl Ingest local translation job
  cec ingest-translation-job SitelJob1 -s DEV Ingest translation job
SitelJob1 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 SitelJob
                                                             File will
be uploaded to the Home folder.
  cec upload-translation-job SitelJob -s UAT
                                                             File will
be uploaded to the Home folder on registered server UAT
 cec upload-translation-job SitelJob -f Import/TranslationJobs File will
be uploaded to folder Import/TranslationJobs.
  cec upload-translation-job SitelJob -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, optionlly specify <idcsurl>, <clientid>, <clientsecret> and <scope> for headless commands.

```
Options:
```

```
--help, -h
                     Show Help
  --endpoint, -e
                     <endpoint> Server endpoint [required]
                    <user> User name [required]
  --user, -u
  --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 in millisecond when try to login to the
  --timeout, -m
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
```

m

The server is a tenant on Oracle Public cloud cec register-server server1 -e http://server1.com -u user1 -p samplePass1 -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
                      The registered OCM server for sync source
  --server, -s
[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
                      The username used to authenticate the web hook
  --username, -u
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
                    Set port. Defaults to 8086.
  --port, -p
  --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
                   The webhook server type [seo] [required]
  --type, -t
  --contenttype, -c The content type [required]
  --detailpage, -d The full url of the site detail page for this type
[required]
                    The registered OCM server [required]
  --server, -s
  --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

## ローカル・テスト・ハーネスを使用したテスト

ローカル・テスト・ハーネスでのカスタム・コンポーネント、テンプレートおよびコンテン ツ・レイアウトを、Oracle Content Management にインポートする前に実行します。

ローカル・テスト・ハーネスを開始するには:

- 1. ターミナル・ウィンドウに cd cec を入力します。
- 2. cec develop &または cec develop --server <server-name> &を入力します
- 3. http://localhost:8085 でブラウザを開き、ローカル・テスト・ハーネスで実行されているコンポーネント、テンプレートおよびコンテンツ・レイアウトを表示します。
- 4. 次のディレクトリ内のコンポーネント、テンプレート、テーマなどを検索できます:
  - cec/src/main/components


- cec/src/main/templates
- cec/src/main/themes

# jQuery 3.5.x へのアップグレード

開発者は、OCE ツールキットを使用すると、不適切に閉じた html タグを特定できます。

jQuery 3.5.x にアップグレードして、HTML パーサーでセキュリティ 修正を取得します。

https://blog.jquery.com/2020/04/10/jquery-3-5-0-released

jQuery 3.5.x で特定の HTML 文字列を解析すると、3.4.x で同じ文字列を解析した場合 と異なる結果になります。これらの文字列には、"<div />"など、HTML 標準に違反し たセルフクローズしたタグが含まれています。特に多数のタグの一部である場合に、 このようなタグを 3.5.x で解析すると、3.4.x と異なる結果になる場合があります。

cec create-asset-report <site>を実行すると、ページの JSON ファイルおよびコン ポーネントの HTML と JS ファイルで不適切に閉じた html タグを検出できます。

# Developer Cloud Service による Oracle Content Management の開発

OCE ツールキット、Developer Cloud Service 統合は、Oracle Content Management の サイト・テンプレート、テーマ、カスタム・コンポーネントおよびコンテンツ・レイ アウトを開発する上で役に立ちます。

OCE ツールキットで、Oracle Content Management のアセット・リポジトリ、ファイ ルおよびフォルダを使用できます。OCE ツールキットには、テーマおよびコンテン ツ・レイアウトを含め、カスタム・コンポーネントおよびサイト・テンプレートを作 成および開発するためのツールがあります。これには、Git リポジトリ、迅速かつ反復 的に開発するためのローカル・テスト・ハーネスおよび開始するためのサンプル・ユ ニット・テストが含まれています。

OCE ツールキットは、次のタスクを実行する上で役に立ちます:

- コンポーネント、テンプレート、テーマおよびコンテンツ・レイアウトのローカル 開発およびテスト用として Oracle Content Management インスタンスを使用する ようローカル開発環境を設定します
- サンプルからコンポーネント、サイト・テンプレートおよびコンテンツ・レイアウトを作成し、テスト・ハーネスでこれらを実行して確認し、Developer Cloud Service環境でコンポーネント、テンプレート、テーマおよびコンテンツ・レイアウトを開発します
- ソース管理および将来の開発のために、Oracle Content Management で作成された コンポーネントおよびサイト・テンプレートを Developer Cloud Service 環境にイ ンポートします



- Oracle Content Management で使用するために、Developer Cloud Service 環境からコン ポーネント、テンプレートおよびコンテンツ・レイアウトをエクスポートします
- 既存のコンポーネント、テンプレートまたはコンテンツ・レイアウトをコピーします
- ユニット・テストを作成します
- コンポーネントを最適化します
- コンポーネントおよびテンプレートを Oracle Content Management にデプロイします

次のトピックでは、Developer Cloud Service を設定してカスタム・コンポーネント、サイト・ テンプレートとテーマおよびコンテンツ・レイアウトを開発する方法について説明します。

- 1. Developer Cloud Service の使用について
- 2. Oracle Content Management 用の Developer Cloud Service コンソールへのサインイン
- 3. Developer Cloud Service でのプロジェクトの作成
- 4. 新規 GIT リポジトリ内のプロジェクト・コードへの OCE ツールキットの追加
- 5. ローカル・テスト・ハーネスでのカスタム・コンポーネント、テンプレートおよびコンテ ンツ・レイアウトのテスト
- 6. 変更のマージ

次のトピックでは、Oracle Content Management OCE ツールキットの使用について説明します:

- cec コマンドライン・ユーティリティの使用
- Developer Cloud Service によるカスタム・コンポーネントの開発
- Developer Cloud Service によるテンプレートの開発
- コンテンツ・レイアウトの開発

#### Developer Cloud Service の使用について

Oracle Developer Cloud Service は、クラウドベースのソフトウェア開発 Platform as a Service (PaaS)であり、アプリケーション開発インフラストラクチャ用ホスティング環境です。オープン・ソースの標準をベースとする統合で、Oracle Cloud 内でのアプリケーションの開発やコラボレート、デプロイが可能です。

Developer Cloud Service は Oracle Cloud 上にホストされたソフトウェアとサービスの集合 体で、Git、Maven、問題および Wiki との統合でより効率的にアプリケーション開発のライフ サイクルの管理に役立ちます。Oracle Developer Cloud Service を使用すると、Oracle Cloud 上の Git リポジトリに対してアプリケーションのソース・コードをコミットし、割り当てられ た問題と欠陥をオンラインで追跡し、Wiki ページを使用して情報を共有し、ソース・コード を相互評価し、プロジェクトの構築をモニターできます。テストが成功すると、プロジェク トを Oracle Content Management にデプロイできます。

### Oracle Content Management 用の Developer Cloud Service コンソー ルへのサインイン

**Developer Cloud Service** コンソールで **Oracle Content Management** のカスタム・コンポーネ ントの開発を開始します。



**Oracle Cloud** サービスの管理者として、**My Service Administration** を使用してクラウ ド・サービスを作成および管理できます。**Oracle Content Management** のサービス・ インスタンス管理者および **Standard Developer Service** のサービス管理者の場合、設 定して使用を開始できます:

- 1. アカウントに提供された情報を使用して、Oracle Cloud にサインインします。
- 2. My Service Administration にサインインして、Oracle Content Management インス タンスと Standard Developer Service を作成および管理します。



- 3. 要望どおりに Oracle Developer Cloud Service 電子メールを確認します。
- サービスのサブスクリプション詳細を使用して Oracle Content Management イン スタンスを設定し、インスタンスの Oracle Content Management URL に移動しま す。
- 5. Standard Developer Service の URL にアクセスします。
- 6. Oracle Developer Cloud Service アカウントにサインインします。

Developer Cloud Service URL にアクセスし、コンソールにサインインします。

#### Developer Cloud Service でのプロジェクトの作成

Content Experience Cloud プロジェクト・テンプレートを使用して Developer Cloud Service でプロジェクトを作成できます。または、空の Git リポジトリを使用してプロ ジェクトを作成し、OCE ツールキットを Oracle Content Management インスタンスか らインポートできます。

- Oracle Content Management テンプレートを使用した Developer Cloud Service プロジェクトの作成
- Oracle Content Management からの OCE ツールキットのダウンロードを使用した Developer Cloud Service でのプロジェクトの作成
- 新規 GIT リポジトリ内のプロジェクト・コードへの OCE ツールキットの追加



# Oracle Content Management テンプレートを使用した Developer Cloud Service プロジェクトの作成

**Developer Cloud Service** でカスタム・コンポーネント、テンプレート、テーマおよびコンテ ンツ・レイアウトを開発するためのプロジェクトを作成します。

プロジェクトを作成するには:

- **1.** Developer Cloud Service コンソールにサインインした後、「新規プロジェクト」をクリックします。
- 2. テンプレートのリストで、「コンテンツ管理」を選択して、「次」をクリックします。
- 3. 「プロジェクト・プロパティ」の下にあるプロパティの「Wiki マークアップ」フィールド で「Confluence」を選択します。

Oracle Content Management からの OCE ツールキットのダウンロードを 使用した Developer Cloud Service でのプロジェクトの作成

> **Developer Cloud Service** でカスタム・コンポーネント、テンプレート、テーマおよびコンテ ンツ・レイアウトを開発するためのプロジェクトを作成します。

プロジェクトを作成するには:

- **1.** Developer Cloud Service コンソールにサインインした後、「新規プロジェクト」をクリックします。
- 2. プロジェクトの名前を指定し、必要なプロジェクト詳細を入力または選択して、「次」を クリックします。
- 3. テンプレートのリストで、「初期リポジトリ」,を選択して、「次」をクリックします。
- 4. 「プロジェクト・プロパティ」の下にあるプロパティで、初期リポジトリに「空のリポジト リ」を選択します。「終了」をクリックします。

### 新規 GIT リポジトリ内のプロジェクト・コードへの OCE ツー ルキットの追加

プロジェクトの新しい空の Git リポジトリに OCE ツールキットを追加できます。

- 1. 新しいプロジェクトの REPOSITORIES の下で、プロジェクトの Git リポジトリの HTTP URL をコピーします。
- 2. ターミナル・ウィンドウを開き、git clone <your-project >.git コマンドを入力しま す。
  - a. 求められたら、Developer Cloud Service のパスワードを入力します。
  - **b.** 「git is not a command」というエラーが表示されたら、https://git-scm.com/ downloads から Git をインストールし、git clone コマンドを再入力します。
- 3. git clone git@github.com:oracle/content-and-experience-toolkit.git

または、https://github.com/oracle/content-and-experience-toolkit/archive/ master.zip からダウンロードすることもできます。



- 4. cp -R content-and-experience-toolkit/sites/cec-components <yourproject>
- cd <your-project>
- 6. git add cec-components
- 7. git commit -a -m "<your comments>"
- 8. git push

### ローカル・テスト・ハーネスでのカスタム・コンポーネン ト、テンプレートおよびコンテンツ・レイアウトのテスト

ローカル・テスト・ハーネスでのカスタム・コンポーネント、テンプレートおよびコ ンテンツ・レイアウトを、Oracle Content Management にインポートする前に実行し ます。

ローカル・テスト・ハーネスを開始するには:

1. ターミナル・ウィンドウに cd cec-components を入力します。

npm start &と入力します

http://localhost:8085 でブラウザを開き、ローカル・テスト・ハーネスで実行されているコンポーネント、テンプレートおよびコンテンツ・レイアウトを表示します。

コンポーネントをローカル・サーバーでテストするときに、コンテンツの使用をロー カル・テンプレートからにするか、Oracle Content Management サーバーからにする かを選択できます。

#### 変更のマージ

コンポーネント、テンプレートまたはコンテンツ・レイアウトを作成した後、または ローカル・マシン上のソース・コードを編集した後、新規および変更済のコンポーネ ントとテンプレートをプロジェクトの GIT リポジトリにマージする必要があります。

変更を GIT リポジトリにマージするには、ターミナル・ウィンドウで次のコマンドを 順に入力します。

cd cec-components git pull git add . git status git commit -a -m "Your comments" git pull git push

# OCE ツールキットによるテストから本番への変 更の伝播

サイト・テンプレートを開発した後で、OCE ツールキットのコマンドライン・インタ フェース(CLI)を使用して、Oracle Content Management サーバー上で開発からテスト と本番までテンプレートを伝播できます。



変更を伝播する場合、ツールキット・コマンドを使用して、開発、テストおよび本番サーバ ー上でサイトを作成したりライフサイクルを管理します。開発環境でサイトに変更を行い、 これらの変更をテストおよび本番環境に伝播します。この一連のコマンドライン・ユーティ リティをスクリプト環境に組み入れてデプロイメントを管理することもできます。CLIユー ティリティを使用すると、アセットやコンポーネントなどの新規アイテムや既存のコンテン ツの更新をロールアウトできます。

次のステップでは、OCE ツールキットの CLI を使用して開発からテストおよび本番まで変更 を伝播する方法を示します。

 同じリポジトリおよびローカリゼーション・ポリシーで開発、テストおよび本番サーバー を設定します。

開発サーバーからテスト・サーバーと本番サーバーに変更を伝播するには、3 つのサーバ ーのそれぞれに同じ名前およびローカリゼーション・ポリシーでリポジトリを設定する必 要があります。アセット・リポジトリのデフォルト・ローカリゼーション・ポリシーは en-US ですが、3 つのサーバーのすべてのポリシーが同じであれば、異なるものを使用で きます。

アセット・リポジトリの設定を参照してください。

2. 開発、テストおよび本番サーバーを Oracle Content Management に登録します。

変更をサイトに伝播する前に、各サーバーを登録する必要があります。OCE ツールキットで提供されている cec register-server コマンドでサーバーを登録できます。

cec register-server <name>

次のコマンド・オプションを指定します。

- サーバー URL の場合、-e <endpoint>
- サーバーに接続する場合、-u <user>および-p <password>
- サーバー・タイプを設定する場合、オプションの-t <type>。デフォルト値は、pod_ec です。

**Oracle Public Cloud** で **Oracle Content Management** テナントに接続する場合、 pod_ec のみを使用します。

たとえば、次のコマンドは、Oracle Public cloud のテナントであるサーバーを登録します。

cec register-server DEV -e https://DEV.example.com -u user1 -p <password>

次のコマンドは、Oracle Content Management のスタンドアロン開発インスタンスを登録 します:

cec register-server DEV -e https://DEV.git.oraclecorp.example.com -u
user1 -p <password>

**Oracle Content Management** サーバーを登録した後で、コンテンツを cec list ツールキット・コマンドでリストできます。



<pre>cec-compontents&gt; cec list -s DEV - Logged in to remote server: &lt;. Channels: Name StarterSite</pre>	host:port> Token <site-id></site-id>	
Components: Name Published FooterBar StarterComponent StarterFooter StarterNavMenu	Type Component group Local component Component group Local component	
Localization policies: Name Optional Languages en-US	Required languages en-US	
Repositories: Name r		
Sites: Type Published Online Name Enterprise	Theme StarterSiteTheme	
Templates: Name StarterTemplate	Theme StarterTheme	Type Standard
サイト・テンプレートを開発サーバー を作成します。	にアップロードし、テンプレート	<b>-</b> からサイト
cec create-template コマンドでサイ を開発サーバーにアップロードできま ンプレートからサイトを作成できます テンプレートをアップロードします。	ト・テンプレートを作成して、ラ す。その後、cec create-site I 。次のコマンドは、テンプレート	テンプレート コマンドでテ ヽを作成して
- cec create-template blog -f Blo	ogTemplate	

次のコマンドは、開発サーバーのコンテンツをリストします。

- cec upload-template blog -s DEV

次のコマンドは、アップロードしたテンプレートから blog という名前のサイトを 作成します。

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

3.

- get repository



```
- get localization policy
    - creating enterprise site . . .
      name
                          bloq
      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. サイトを公開して、開発サーバーでこれをオンラインにします。
   サイトを作成した後で、cec control-site コマンドを使用すると、サイトを公開してオ
   ンラインにできます。
   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 Sitel on
   the server
    cec control-site publish -s Site1 -r UAT
                                                  Publish site Site1 on
   the registered server UAT
    cec control-site unpublish -s Sitel -r UAT
                                                  Inpublish site Site1 on
   the registered server UAT
    cec control-site bring-online -s Sitel -r UAT Bring site Sitel online
   on the registered server UAT
     cec control-site take-offline -s Sitel -r UAT Take site Sitel 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
- サイトを DEV から UAT に移動するには、サイトをパッケージ化する必要があります。サイト間でサイトを移動するためのパッケージング・モデルはテンプレートです。開発サーバーで作成したサイトから新しいテンプレートを作成し、テンプレートをダウンロードします。

次の例の cec create-template-from-site コマンドは、blog から blog2 という名前のサイト・テンプレートを作成します。

cec create-template-from-site blog2 -s blog

cec download-template コマンドで開発サイトから作成したテンプレートをダウ ンロードします。

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 tmplate content file

*** template is ready to test: https://localhost:8085/templates/



blog2 cec upload-template blog2 --server UAT

6. テンプレートをアップロードして、コンテンツ・タイプおよびコンテンツ・レイアウト・ マップを作成します。

cec upload-template blog2 --server UAT

テンプレートをアップロードしますが、コンテンツ・アイテム(コンテンツ・テンプレート)をテンプレートから除外します。

cec upload-template blog2 --server UAT -x

これを実行して、元のサイトと同じ GUID を持つコンテンツを含むサイトを作成します。 コンテンツを含むテンプレートからサイトを作成する場合、新しいサイト内のコンテンツ はすべて新しい GUID を持つことになります。新しいコンテンツを作成するのではなく、 コンテンツの更新を可能にするため、テンプレートからコンテンツを除外する必要があり ます。

8. テンプレートからサイトを作成します。

cec create-site blog -t blog2 -r r -l "en-US" -d "end-US" --server UAT

コンテンツ・テンプレートをサイトのチャネルおよびコレクションにアップロードします。ステップ7でテンプレートからこれを除外したため、これを実行する必要があります。

cec upload-content blog2 -t -r r -c blog -l "blog site" --server UAT

10. サイトを公開して、テスト・サーバーでこれをオンラインにします。

```
cec control-site コマンドを使用し、サイトを公開してオンラインにします。
```

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. DEV** サーバーのサイト blog に変更を加えた場合、**UAT** サーバー上にすでに作成している サイトに変更を伝播できます。
- 12. テンプレートに変更が反映されるように、サイトから別のテンプレートを作成します。

cec create-template-from-site blog3 -s blog --server DEV

13. テンプレートをダウンロードします。

cec download-template blog3 -s DEV

**14.** テンプレートをアップロードしてこのテンプレートからサイトを作成し、変更をテ スト環境に伝播します。

cec upload-template blog3 -s UAT

このコマンドは、変更されたコンポーネントおよびテーマを作成または更新します が、コンテンツを除外します。

15. update-site コマンドを使用し、コンテンツを取得してページを更新します。

cec update-site blog -t blog3 - UAT

#### 例:

```
cec-components> cec update-site blog -t blog3 --server UAT
   Updating site: blog3
    - Logged in to remote server: https://<host>:<port>
                       : updating file# 6 of 6 files
    - pages
    - content
                        : updating file# 3 of 3
                                                    files
                        : updating file# 5
                                             of 5
                                                     files
    - System Files
    - controller
                        : no files in update, removing files on
   server
    - favicons
                        : no files in update, removing files on
   server
                        : no files in update, removing files on
    - misc
   server
                        : no files in update, removing files on
    - seo
   server
   - system
                         : no files in update, removing files on
   server
    - created content file /Users/<user-name>/devenv/git/webclient/
   developer/sites-toolkit/cec-compnents/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. サイトをチェックして、変更が伝播されていることを確認します。
```

**17.** UAT サーバーから PROD サーバーに移動するには、DEV から UAT にサイトを移動する際に実行したステップと同じステップを実行します。



18. 本番サーバーでサイトを作成し、これをオンラインにして変更を確認します。

cec list コマンドを使用して、本番サイトのコンテンツを表示したり、開発環境で行った変更が含まれていることを確認できます。また、サイトをチェックして、変更が本番に伝播されていることを確認できます。

#### パスワードの暗号化

サーバーを OCE ツールキットで登録する場合、パスワードを暗号化してサーバーをローカル で使用できるようにする必要があります。

- **1.** Oracle Content Management サーバーをプレーン・テキストのパスワードを含む cec register-server コマンドで登録します。
- 2. cec create-encryption-key コマンドでパスワードを暗号化します。

cec create-encryption-key <file> [alias: cek]
Create an encryption key to encrypt/decrypt password for servers.

3. サーバーを暗号化キーで再度登録します。これにより、サーバーがローカルの開発および テストに使用できるようになります。

暗号化されたパスワードはサーバーの接続ファイルに格納されます。パスワードは登録済サ ーバーに接続すると復号化されます。

#### サーバーの登録

OCE ツールキットでサーバーを登録できます。

cec register-server コマンドを暗号化キーとともに使用して、ローカル開発およびテスト 用に Oracle Content Management サーバーを登録します。

サーバーの登録時にパスワードを暗号化すると、サーバーを OCE ツールキットで使用できる ようになります。パスワードの暗号化を参照してください。

#### サイトの使用状況および権限レポートの作成

レポートを作成して、サイトのテストから本番への伝播用にターゲット・サーバー・メンバ ーの権限を検証および修正できます。

**Cec** create-asset-report を使用して次のように確認します。

- すべてのサイト・アーティファクトのメンバーシップおよびチャネル割当てを確認します。
  - テーマ
  - テンプレート
  - コンポーネント
  - コンテンツ・タイプ
- 2. 対応可能な問題にフラグを設定します。



例:

cec create-asset-report blog1 -s <registered-server> -o

cec create-asset-report trbcent -s <registered-server> -o

レポートが JSON ファイルに生成され、これにより、使用状況および権限に関する問題を確認できます。次のコマンドが権限の修正に使用可能です:

- cec share-type: Oracle Content Management サーバーのユーザーとタイプを共有します。
- cec unshare-type: Oracle Content Management サーバーの指定されたユーザー のタイプへのアクセス権を削除します。
- cec share-repository: Oracle Content Management サーバーのリポジトリ(およびリポジトリで使用されるタイプ)を共有します。
- cec unshare-repository: Oracle Content Management サーバーのリポジトリへのユーザー・アクセス権を削除します。
- 例:

cec share-repository Repol -u <user-namel>,<user-name2> -r manager t -s <registered-server>

# ドキュメントおよびフォルダのダウンロードおよびアッ プロード

Oracle Content Management サーバーに対してドキュメントおよびフォルダをダウン ロードおよびアップロードできます。

次のコマンドがドキュメントとフォルダのダウンロードおよびアップロードに使用可 能です:

```
cec download-folder <path>
                                     Downloads folder from CEC
                    [alias: dlfd]
server.
 cec upload-folder <path>
                                     Uploads folder to CEC
server.
                        [alias: ulfd]
 cec download-file <file>
                                     Downloads file <file> from CEC
               [alias: dlf]
server.
 cec upload-file <file>
                                     Uploads file <file> to CEC
                   [alias: ulf]
server.
 cec-share-folder <name>
 cec-unshare-folder <name>
```

cec-share-folder <name>の場合、Oracle Content Management サーバー上でユーザ ーとフォルダを共有でき、ユーザーにロールを割り当てることができます。サーバー を-s <server>で指定するか、cec.properties ファイルで指定されたサーバーを使用し ます。有効なロールは次のとおりです。

- manager
- contributor



- downloader
- viewer

ダウンロードの場合、フォルダ階層を指定できます。

# テンプレートからのサイトの作成およびコンテンツ の同じ GUID の維持

開発者は OCE ツールキットのコマンドを使用して、コンテンツの同じ GUID を維持したま ま、テンプレートから Oracle Content Management サイトを作成できます。

次の OCE ツールキットのコマンドを使用します。

update create-site-from-template --reuse-content

このコマンドは、Oracle Content Management サーバーにサイトを作成し、サイトの作成時 にコンテンツ ID を保持します。コンテンツ ID の保持は、本番へのテストの複数の実行でタ ーゲット・サーバーに重複したコンテンツ・アイテムを生成しないようにするために必要で す。

# 標準サイトからのエンタープライズ・テンプレート の作成

開発者は、OCE ツールキットの CLI コマンドを使用して、標準サイトからエンタープライズ・テンプレートを作成できます。

デフォルトでは、create-template コマンドは、サイトが標準サイトの場合に標準テンプレートを、サイトがエンタープライズ・サイトの場合にエンタープライズ・テンプレートを作成します。同様に標準サイトからエンタープライズ・テンプレートを作成できます。

新しい-enterprise オプションを指定して cec create-template を実行します。

cec create-template EnterpriseTemp1 -s StandardSite1 -e

# タクソノミのインポートおよびエクスポート

OCE ツールキットのコマンドを使用して、ローカル・マシンから Oracle Content Management サーバーにタクソノミをインポートしたり、サーバーからローカル・マシンに タクソノミをエクスポートします。

cec download-taxonomy <name>コマンドは、Oracle Content Management からタクソノミを エクスポートします。Oracle Content Management サーバーからタクソノミをダウンロード します。

このコマンドで次のオプションを使用できます。

- --status, -t [promoted | published] [required]: タクソノミ・ステータスを指定します。
- --id, -i: 同じ名前の別のタクソノミがある場合、タクソノミ ID を指定します。



- --server, -s: 登録済 Oracle Content Management サーバーを指定するか、 cec.properties ファイルで指定したものを使用します。
- --help, -h: コマンドのヘルプを表示します。

download-taxonomy コマンドの例は次のとおりです。

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 <taxonomy>コマンドは、Oracle Content Management にタクソ ノミをインポートします。Oracle Content Management サーバーにタクソノミをアッ プロードします。

このコマンドで次のオプションを使用できます。

- --createnew, -c: 新規タクソノミを作成します。
- --name, -n: 新規タクソノミの名前を指定します。
- --abbreviation, -a: 新規タクソノミの略語を指定します。
- --description, -d: 新規タクソノミの説明を指定します。
- --file, -f: タクソノミをファイルで指定するかどうかを指定します。
- --server, -s: 登録済 Oracle Content Management サーバーを指定するか、 cec.properties ファイルで指定したものを使用します。
- --T2P: 同じ ID を持つアセットまたはタクソノミがターゲットの Oracle Content Management インスタンスに存在するかどうかを確認します。true の場合、新規バ ージョンとして、タクソノミはドラフト(既存のドラフトをオーバーライド)として 追加されます。それ以外の場合、新規アセットまたはタクソノミが同じ ID で作成 されます。
- --New: 常に新しいアセットまたはタクソノミをターゲットの Oracle Content Management インスタンスに作成します。
- --help, -h: コマンドのヘルプを表示します。

upload-taxonomy コマンドの例は次のとおりです。

```
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

OCE ツールキットのテストから本番の CLI ユーティリティを 使用すると、コンテンツ・モデ ルとその依存関係とともに、ソース Oracle Content Management サーバーからターゲット Oracle Content Management サーバーへのアセットのインポートまたはエクスポートを自動 化できます。

マネージャまたはコンテンツ管理者は、テストから本番環境で OCE ツールキットのコマンド を使用してタクソノミをインポートまたはエクスポートできます。マネージャ権限で、タク ソノミのドラフトを追加できます。コンテンツ管理者権限で、新規タクソノミを作成できま す。

プロモート、リポジトリへの割当ておよびタクソノミの公開などのタクソノミ・ライフサイ クル操作は、テストから本番に使用できます。アセットをカテゴリ化情報およびタクソノミ とともにインポート・ファイルにインポートしたり、アセットをカテゴリ化情報およびタク ソノミとともにエクスポート・ファイルからエクスポートできます。

# レコメンデーションのインポートおよびエクスポー ト

OCE ツールキットのコマンドを使用して、Oracle Content Management サーバーからローカ ル・マシンにレコメンデーションをインポートしたり、ローカル・マシンからサーバーにレ コメンデーションをエクスポートします。

cec download-recommendation <name>コマンドは、Oracle Content Management からレコメ ンデーションをエクスポートします。Oracle Content Management サーバーからレコメンデ ーションをダウンロードします。

このコマンドで次のオプションを使用できます。

- --status, -t [promoted | published] [required]: レコメンデーション・ステータス を指定します。
- --id, -i:同じ名前の別のレコメンデーションがある場合、レコメンデーション ID を指定します。
- --server, -s: 登録済 Oracle Content Management サーバーを指定するか、 cec.properties ファイルで指定したものを使用します。
- --help, -h: コマンドのヘルプを表示します。

download-recommendation コマンドの例は次のとおりです。

cec download-recommendation Recommendation1 -t promoted

cec download-recommendation Recommendation1 -i 6A6DE836572C468B90F2A1C17B7CE5E4 -t promoted

cec download-recommendation recommendation -t published -s UAT



cec upload-recommendation <name>コマンドは、Oracle Content Management にレコ メンデーションをインポートします。Oracle Content Management サーバーにレコメ ンデーションをアップロードします。

このコマンドで次のオプションを使用できます。

- --createnew, -c: 新規レコメンデーションを作成します。
- ▶ --name, -n: 新規レコメンデーションの名前を指定します。
- --abbreviation, -a: 新規レコメンデーションの略語を指定します。
- --description, -d: 新規レコメンデーションの説明を指定します。
- --file, -f: レコメンデーションをファイルで指定するかどうかを指定します。
- --server, -s: 登録済 Oracle Content Management サーバーを指定するか、 cec.properties ファイルで指定したものを使用します。
- --help, -h: コマンドのヘルプを表示します。

upload-recommendation コマンドの例は次のとおりです。

#### 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 uploadrecommendation Create a new recommendation or a draft of an existing recommendation in <file-name>.json -f and upload the JSON file

### コレクション・コンテンツの追加または削除

開発者は、OCE ツールキットのコマンドを使用して、コンテンツをコレクションに追加したり、コレクションからコンテンツを削除できます。

次の cec-control-content コマンドでは、**Repo1** リポジトリ内のすべてのアイテムを **UAT** 登録済サーバーの **Collection1** コレクションに追加します。

cec control-content add -l Collection1 -r Repo1 -s UAT



次の cec-control-content コマンドでは、**UAT** 登録済サーバーの **Collection1** コレクション からすべてのアイテムを 削除します。

cec control-content remove -l Collection -s UAT

# OCE ツールキットを使用したカスタム・フィールド・ エディタの開発

OCE ツールキットでは、フィールドの外観タイプのコンポーネントを開発するためのサポートを提供しています。開発者はカスタム・フィールド・エディタを作成および管理できます。

フィールドの外観タイプのコンポーネントの場合、次のタスクを実行できます。

- コンポーネントを開いたり、コピーまたは削除します
- コンポーネントを公開または非公開にします
- コンポーネントをエクスポートまたはインポートします
- コンポーネントのメンバーを追加または削除します
- プロパティを表示します
- コンポーネント・ロゴを選択します

フィールドの外観タイプでコンポーネントのリストをフィルタできます。

次の OCE ツールキットのコマンドをカスタム・フィールド・エディタの開発に使用できま す。

```
cec add-field-editor <name>
                                                   Adds a field editor to a
field in a content type.
                                               [alias: afe]
                                                     Removes a field editor
  cec remove-field-editor <name>
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

--contenttype, -c The content type [required]
                       The template the content type is from [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

**ORACLE**[°]

```
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

フィールドの外観コンポーネントの次のサンプルは、OCE ツールキットに含まれています。

- TextFieldEditor
- SliderFieldEditor
- MapFieldEditor

次の図は、サンプルのフィールドの外観コンポーネントの開発に使用できる OCE ツー ルキットのコマンドを示しています。



```
2
 3
    # create
    cec cc editor1 -f TextFieldEditor
 4
 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
```

ローカル・サーバー上ですぐに使用可能なフィールドの外観コンポーネントを作成し、これ らをテストしてから、Oracle Content Management インスタンスにアップロードできます。 次の図は、localhost:8085 でこれらのコンポーネントを示しています。



ローカル・サーバーでスライダなどの各コンポーネントをテストできます。コンポーネント のプロパティを選択して、これを保存します。



ORACLE* Content and Experience Cloud						
slider1						
Developer >	Local Content	Ser	erver Content	Theme Design		Theme Resources
Components > slider1	None	\$	None \$	None	\$	None 🗘
Editor Properties		×	Content Field E	ditor		
<ul> <li>Handles multiple value</li> </ul>	s			0	118	
Select data types supporte	- ed by this editor					
■ Text						
Large Text						
Date			Content Field E	ditor View		
				0	118	
■ Boolean				0		
Embedded Content						
	Save to appinfo.json					

マップ・コンポーネントの場合、マップの周りをクリックして、エディタ値として場 所を指定できます。

Map1				
Developer > Local Content	Server Content	Theme Design	Theme Resource	S
Components > Map1  None	None \$	None	\$ None	\$
Editor Properties  Handles multiple values Select data types supported by this editor	× Content Field	Editor	Westside Jewish Community School	
🖬 Text	Constry Screol	The second second	St.	
■ Large Text		Wasin c	MTA- h St-B & C Lines	
Date	83r0 St			
Number		Rodeph Sholom	D W Ro	
Decimal	Ser.		entra de la sala sa	
Embedded Content	500 ft			
Save to appinfo.json	Content Field	Editor View Control of the state of the sta		

コンポーネントの HTML ファイルを編集して、背景色などの設定を変更できます。



•••		eur.ntm — components	
C,	EXPLORER	◇ edit.html ●	
Q	<ul> <li>OPEN EDITORS 1 UNSAVED</li> <li>OPEN edit.html editor1/assets</li> <li>COMPONENTS</li> </ul>	editor1 > assets > ↔ edit.html > 1 月IDOCTYPE html5 2 This is a plain #4c89c5 types on</th <th>:</th>	:
% ष्र	<ul> <li>✓ editor1</li> <li>✓ assets</li> <li>♦ edit.html</li> <li>♦ view.html</li> </ul>	3 <html> 4 <head> 5 <style> 6 body { 7 margin: 0; 7 margin: 0;</th><th>>></th></tr><tr><th>ß</th><th><ul> <li>_folder_icon.png</li> <li>_folder.json</li> <li>appinfo.json</li> <li>Mant</li> </ul></th><th>s     page119; 0;       9     font: normal 1       10     }       11     input {       12     width: 199%;</th><th></th></tr><tr><th>A</th><th>> Sample-To-Do > slider</th><th><pre>13 height: 32px; Shorthand property for setting most background properties at the same place in the 14 font-size: 14p style sheet. 15 color: □#333; 16 background: □#4c69C5; 17 border: 1px solid □#dfe4e7; 18 border-radus: 2px; 19 padding: 4px 5px; 10 box-sizing: border-box; 21 } 22 </style> 24  25  26  27 <!-- load the field editor SDK ---> 28 <script src="/documents/static/gemini/api/field-editor-sdk-1.0.js"></script> 29 <script> 20 </script> 20  21  22  23  24  25 &lt; 26 <script src="/documents/static/gemini/api/field-editor-sdk-1.0.js"></script> 29 <script> 20 </script> 20  21  22  23  24  25 &lt; 26  27  29 <script> 20 </script> 20  20  20  20  21  22  23  24  25  26  27  28  29  20  21  22  23  24  25  26  27  28  29  20  20  20  20  20  20  20  20  20  21  22  23  24  25  26  27  28  29  20 <th></th></head></html>	
500	> OUTLINE > MAVEN PROJECTS	31 /* globals editorSDK */	
⊗ 0	∆ 0	Ln 1, Col 1 Tab Size: 4 UTF-8 LF HTML ③	

OCE ツールキットのコマンドを使用すると、フィールド・エディタをコンテンツ・タイプの フィールドとローカルに関連付けることができます。

cec add-field-editor editor1 -t SimpleContent -n -c SimpleType -f title

cec add-field-editor slider -t SimpleContent -n -c SimpleType -f value

カスタム・フィールド・エディタの構成およびテストが完了すると、OCE ツールキットのコ マンドを使用して、これらを Oracle Content Management インスタンスにアップロードでき ます:

# upload editors
cec ulcp editor1,slider -p -s Latest

# upload content

cec cr Repo5 -s

cec upload-content SimpleContent -r Rpo5 -s

カスタム・フィールド・エディタをアップロードすると、コンポーネントが Oracle Content Management インスタンスにインポートされます。



1	Home	$\equiv$ OR/	ACLE' Content and Experience Cloud	Search All		Q	ج ؟	0
Ę	Assets	Compon	ents 🗛 -				Cr	eate 🔻
ī	Sites	Developer > Co	omponents					
	Recommendations				All 👻	50 per Page 👻	Name -	· III
G	Developer		editor1 Created just now by you CONTENT FIELD EDITOR					0
L	<u>II.</u> Analytics	1	slider Created just now by you CONTENT FIELD EDITOR					٥
c	OLLABORATION	•						
1	Documents							
•	Conversations							
	DMINISTRATION							
3	🛠 System							
-	E Integrations							
•	Content							

# あるサーバーから別のサーバーへのサイトの転送 または更新

開発者は OCE ツールキットのコマンドを使用して、サイトを作成したり、その内容を サーバー A からサーバー B に更新できます。

デフォルトで、すべてのアセットが転送され、オプションで、公開済アセットのみを 転送するには-pを指定します。ソース・サーバーを-s <server>で、宛先サーバーを d <destination>で指定します。サイトに他のリポジトリのアセットが含まれる場合、 オプションで、リポジトリ・マッピングを指定しますが、そうでない場合、これらの アセットは転送されません。

次のコマンドを使用して、サイトを更新したり、サイトをテストから本番に転送でき ます。

追加のオプションについては、cec コマンドライン・ユーティリティの使用を参照して ください。

### コンテンツ・アイテムのないサイトの転送

開発者は、OCE ツールキットを使用して Oracle Content Management サーバー間でコ ンテンツ・アイテムのないサイトを転送できます。

多数のコンテンツ・アセットを含むサイトを転送するには、サイトとそのコンテンツ を分ける必要があります。--excludecontent オプション(ショートカット-x)を transfer-site コマンドに追加できます。

このオプションを設定すると、サイトが転送されます。



例:

cec transfer-site Sitel -s DEV -d UAT -r Repositoryl -l LocalizationPolicyl - x

# グループでのサイトのコンテンツ・アイテムのダウ ンロードまたはアップロード

開発者が Oracle Content Management サーバー間でサイトを転送する場合、サイトのコンテンツ・アイテムをグループでダウンロードまたはアップロードできます。

多数のコンテンツ・アセットを含むサイトを転送するには、サイトとそのコンテンツを分け る必要があります。コンテンツ(cec transfer-site --excludecontent)なしでサイトを転送 した後で、cec transfer-site-content を使用してサイトのコンテンツを転送します。

例:

cec transfer-site-content <name>

このコマンドはスクリプトを作成して、Oracle Content Management サーバー間でエンター プライズ・サイト・コンテンツを転送します。このコマンドは多数のコンテンツ・アイテム の転送に使用し、アイテムはバッチで転送されます。デフォルトでは、このコマンドはスク リプトを実行せず、すべてのアセットが転送されます。-pを指定すると、公開済アセットの みを転送できます。

ソース・サーバーを-s <server>で、宛先サーバーを -d <destination>で指定します。

#### オプション

```
--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]
```

オプション--execute が設定されていない場合、cec transfer-site-content の完了後、生成されたスクリプト<site name>_downloadcontent を実行してコンテンツをソース・サーバーからダウンロードし、<site name>_uploadcontent を実行してダウンロード済コンテンツを宛先サーバーにアップロードします。

#### 例

```
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
```



# OCE ツールキットによるサイト・ページの索引付 け

OCE ツールキットを使用すると、サイト・ページ上のテキストのコンテンツ・アイテ ムを作成し、サイトのページ検索を有効にできます。

次の各項では、OCE ツールキットでページを索引付けする方法について説明します。

- 1. サイト・ページ・テキストのコンテンツ・タイプの作成
- 2. OCE ツールキットによるページ索引のコンテンツ・アイテムの作成
- 3. Oracle Content Management のサイトへのコンテンツ検索の追加

### サイト・ページ・テキストのコンテンツ・タイプの作成

コンテンツ・タイプに名前、必要なフィールド値、およびタイプのデフォルト・コン テンツ・レイアウトを指定します。

• タイプ名

有効なコンテンツ・タイプ名を指定します。

フィールド
 次のフィールドが必須です。

フィールド名	フィールド・タイプ	データ・フィールド値の数	説明
site	テキスト	単一	サイト名
pageid	テキスト	単一	ページ ID
pagename	テキスト	単一	ページ名
pageurl	テキスト	単一	ページ URL
pagedescription	テキスト	単一	ページ説明
keywords	テキスト	複数 <b>(</b> 最大値なし <b>)</b>	ページのすべてのテキスト およびページのコンテン ツ・アイテムのすべてのテ キスト・フィールドの値で あり、OCE ツールキット index-site コマンドで取 得されます

content.fields.pageFullURL = SCSRenderAPI.getSitePrefix() +
content.fields.pageurl;

• タイプのコンテンツ・レイアウトを作成します。

コンテンツ・レイアウトは、サイト名およびページに移動するための URL を表示します。 たとえば、layout.html では次のようになります。

```
{{#fields}}
<div class="indextype"></div>
<div>
<a href="{{pageFullURL}}"title="
{{pagename}}">{{pagename}}</a>
</div>
{{fields}}
```

render.js では、ページの完全な URL を生成します。

```
content.fields.pageFullURL =
SCSRenderAPI.getSitePrefix() +
content.fields.pageurl;
```

コンテンツ・レイアウトをタイプのデフォルト・コンテンツ・レイアウトとして設定します。

```
content.fields.pageFullURL =
SCSRenderAPI.getSitePrefix() + content.fields.pageurl;
```

### OCE ツールキットによるページ索引のコンテンツ・アイテム の作成

OCE ツールキットのコマンドを使用すると、ページ索引のコンテンツ・アイテムを作成できます。

前提条件:

- OCE ツールキットはインストール済で、ローカル・マシンで設定されています。
- Oracle Content Management のサイトは公開されています。
- サイト・ページのコンテンツ・アイテムはサイト・チャネルに公開されています。

コマンドライン・インタフェースで、次の OCE ツールキットのコマンドを入力します:

cec index-site site name -c content type name -p

コマンドで、site name はサイトの名前で、content type name はページ・テキストに作成 したコンテンツ・アイテムです。オプション-p は作成後にページ索引のコンテンツ・アイテ ムを公開することを示します。

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 Sitel -c PageIndex

cec index-site Sitel -c PageIndex -p
```

使用方法を確認するには、cec index-site -hと入力します

### Oracle Content Management のサイトへのコンテンツ検索の 追加

コンテンツ検索を検索ページおよび検索フィールドとともに Oracle Content Management サイトに追加できます。

コンテンツ検索をサイトに追加するには:

- 1. サイトへの検索ページの追加
- 2. テーマへの検索フィールドの追加

#### サイトへの検索ページの追加

検索ページをサイトに追加したり、コンテンツ・リスト・コンポーネントを検索ペー ジに追加できます。

検索ページを追加します。

- 1. ページをサイトに追加し、これを検索ページとして設定します。
- 2. コンテンツ・リスト・コンポーネントを検索ページに追加します。
- 3. 「コンテンツ・タイプ」を以前に作成したページ索引コンテンツ・タイプに設定し ます。

#### テーマへの検索フィールドの追加

サイトの各ページに検索フィールドが表示されるようにするには、検索フィールドを テーマのレイアウト HTML ページに追加します。

例:

```
<div align="center">
<input id="searchonpage" type="text" size="30" placeholder="Search on
page. . ."/>
</div>
```



1. 入力フィールドを追加します:

```
<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. JavaScript を HTML ボディの最後に追加します。

# OCE ツールキットによる多言語サイトの索引付け

OCE ツールキットを使用すると、翻訳、およびページとコンテンツ・アイテムの検索に多言 語(MLS)サイトを索引付けできます。

多言語サイトの索引を作成し、サイトの公開前にこれをテストできます。OCE ツールキット の cec index-site コマンドを使用して、多言語サイトを索引付けします。ceccomponents ディレクトリに移動し、オプションを指定せずにこのコマンドを発行してコマン ドのヘルプ情報を表示します。

```
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 pate, 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

ページの索引アイテムはページごとおよび言語ごとに存在します。言語ごとに作成されたページの索引コンテンツ・アイテムは、デフォルト言語ページの索引アイテムの 翻訳として作成されます。実行サイトで問合せを実行すると、検索およびコンテンツ・ リストはサイト URL から言語を取得します。これにより、検索が自動的にフィルタさ れます。

多言語サイトを公開する前に、これを索引付けして翻訳する必要があります。ここで、 翻訳ジョブが必要になります。Oracle Content Management サーバーでの新規のサイ トまたはアセットの翻訳ジョブの作成を参照してください。

デフォルト言語(英語)は必須です。サポートされている各言語(必須およびオプション) に対して、索引の作成を実行し、索引アイテムの翻訳を作成します。この索引付けを 2回実行すると、索引が更新されます。

OCE ツールキット・コマンドを使用して多言語サイトを索引付け、翻訳および公開するには:

- サイトのコンテンツ・タイプを作成し、リポジトリで使用できるようにします。サイト・ページ・テキストのコンテンツ・タイプの作成を参照してください。
- 2. 検証ポリシーを選択します。
  - a. 左側のナビゲーション・メニューの「**アセット」**をクリックします。
  - **b. 「アセット」**メニューで「**ローカリゼーション・ポリシー」**を選択します。
  - c. ローカリゼーション・ポリシーを選択します。
  - d. 必要に応じてローカリゼーション・ポリシーを変更して、サイトの索引付けお よび翻訳に使用する言語を含めます。たとえば、ポリシーに英語しかない場 合、フランス語およびスペイン語を追加できます。

すべての翻訳は英語から行われます。

- 3. 翻訳ジョブをダウンロードします。サイトで使用されているアセットのみ翻訳で きます。
- 4. サイトを翻訳します。サイト翻訳は、手動で、または統合コネクタを通じて実行で きます。
- 5. 翻訳が完了したら、翻訳ジョブをアップロードします。
- **6.** cec index-site コマンドを使用して、サイトを索引付けします。-c <contenttype> を指定して、ページ索引コンテンツ・タイプを設定します。

-pオプションを指定して、サイトを公開することもできます。その後、索引付け および翻訳を検証して、変更をライブ・サイトに公開します。



```
たとえば、次の cec index-site コマンドは、英語、フランス語およびスペイン語を使用
するサイトにサイト索引を作成します。サイトでサポートされている言語は、デフォルト
言語を含め、割り当てられた L10n ポリシーに基づいています。
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 fx-FR,es-ES
 - query site stucture 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 stucture 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. サイトを公開して翻訳を含めます。

# コンポーネント開発を容易にするための簡潔なコ ンポーネントの作成

OCE ツールキットを使用すると、簡素化したコンポーネントを作成して容易に開発で きるようになります。

簡素化されたコンポーネントの SimpleHTML は OCE ツールキットで使用可能で、カス タム・コンポーネントの開発を容易に開始できるようになります。

cec create-component -f SimpleHTML

JET コンポーネントのサンプルを使用して開始することもできます。

cec create-component MyComp -f JET-CCA-Demo-Card

# サイト・コンパイル・サービスのセットアップ

コンパイル・サーバーの Docker イメージを作成して、Oracle Content Management で サイト・コンパイル・サービスをセットアップできます。このサービスにより、公開 前に使用するようにコンパイル・サーバーを定義するオプションが与えられます。

コンパイル・サーバーには、「管理」>「システム」>「サイトとアセット」ページの 「コンパイル・エンドポイント URL」フィールドに URL 形式のエンドポイントがあり ます。その後、サーバーに登録する完全修飾 URL を入力し、「テスト」をクリックし てエンドポイントを検証します。コンパイル・エンドポイント URL の設定を参照して ください。

サイト・コンパイル・サービスは、公開されたサイトにアクセスすると HTML ページ が返されるように、サイトのページをコンパイルします。Oracle Content Management



サーバーがサイトを公開すると、サービスが構成されていれば、サイト・コンパイル・サー ビスをコールしてページをコンパイルします。

サイトのエンドポイントを指定すると、「サイトのプロパティ」ダイアログの「静的サイト配 信」タブで自動コンパイルを有効にできます。サイトを公開すると、コンパイル・サーバー でコンパイルされます。

サイト・コンパイル・サービスは、ツールキット・コマンドの拡張機能です。cec compilationserver を実行し、Docker イメージを使用すると、標準モードでコンパイル・サーバーを作成 してサービスの構成を調整できます。

Docker イメージを作成して Docker ファイルを公開するには:

- GitHub から Docker イメージ情報をダウンロードします。 GitHub から情報をダウンロードすると、コンパイル・サーバーの下に追加の3つのファ イルが表示されます。Docker イメージのいずれかはコンパイル・サーバーで、Docker フ ァイルおよび readme ファイルが含まれています。
- run.sh コマンドは、Docker イメージが作成されると実行されます。
   このコマンドは、OCE ツールキットをダウンロードしてインストールし、ソース・ディレクトリを作成します。
- その後、環境をカスタマイズして、run.sh コマンドを実行します。使用するポート、タイムアウトおよび変更する必要があるものすべてを指定してから、更新および.sh コマンドを使用します。
- Docker イメージの情報をダウンロードしてコンパイル・サーバーのディレクトリを変更 した後、Docker ファイルを作成できます。 すでに OCE ツールキットをダウンロードしている場合、すべてを再度ダウンロードしな いように no-cache オプションを削除できます。
- 5. 1 つまたは複数の Docker イメージをリストします。
- 6. Docker イメージを使用できるようになったら、これを登録します。
- 正常に機能することを検証します。
   REST API の例外ファイルに移動します。これにより、サポートされているバージョンを 確認できます。このテストにより、実行されていることを検証できます。
- 8. サーバーを登録する必要があります。サーバー名の最初の部分が必要ですが、内部にある その他すべてのものが処理されます。
- 9. システム・ディレクトリおよびサイト・アセットに戻り、ここでコンパイル・エンドポイ ントを登録して「**保存」**をクリックします。
- 10. サイトのプロパティに移動して、公開するものおよび公開するタイミングを指定します。

# サイトのコンパイルによるサイト・ページのランタ イム・パフォーマンスの改良

Oracle Content Management でサイトをコンパイルすると、サイト・ページのランタイム・ パフォーマンスおよび動作を改良できます。コンパイルでは、元のファイルと完全に同じよ うに動作するサイト内のページごとに静的 HTML ファイルを作成することによってこれを実 現します。



#### サイトのコンパイルの概要

サイトの「ページ」を構成するメタデータ・ファイルはコンパイル時に結合され、通 常、実行時に必要なサーバーのリクエストを回避します。ページごとのビューのレン ダリングが1回のみのコンパイル時間になり、効率が向上します。

実行時のリクエスト数を減らすと、パフォーマンスの改善が実現されます。これにより、すべてのコンテンツ問合せの結果を直接ページにコンパイルできるため、これらのコンテンツ問合せを回避できます。

ページが即座にレンダリングされるため、動作の改善が実現されます。したがって、 「スタイルが設定されていないコンテンツのフラッシュ」の問題、コンポーネントが予 測できない順序でページに表示される問題、またはページのレンダリング時にフッタ ーがすぐに表示されてから他のアイテムとしてページの下部に移動する問題を回避で きます。

Oracle Content Management サイト内のページをコンパイルするには、テンプレート としてサイトをエクスポートし、OCE ツールキットを使用してテンプレートをコンパ イルし、元のサイトに対して作成された静的ページをアップロードする必要がありま す。次のステップで、テンプレートのコンパイル・プロセス、およびこのモデルを使 用してサイトをコンパイルする方法を説明します。

#### プリレンダーとのインタラクション

プリレンダーは検索エンジンに適した結果を返す機能に重点を置いています。プリレ ンダーによって作成されたページは、元のページと同じようには実行されず、 JavaScript は実行されません。ブラウザでプリレンダーされた HTML を返し、SEO テ キスト検索をサポートします。

サイト・コンパイルで作成された静的ページはブラウザで実行されるように想定され、 元の動的サイト・ページと同様に動作します。これには、コンパイル前のコンポーネ ントの動的な実行、およびユーザーが結果に完全な最新データを必要とするランタイ ム問合せの実行が含まれていて、コンパイル時のデータの修正は含まれていません。

プリレンダーのソリューションはテナント管理者によって有効化されると、静的ページをプリレンダー・サーバーで配信することもできます。これにより、インデクサお よびクローラが静的ページの一部を処理できるようになり、このページをコンテンツ・ リストのように動的にレンダリングできます。

コンテンツ問合せはコンテンツ・アイテムを返し、これにはリッチ・テキスト・フィ ールドの値が含まれているため、コンテンツ・レイアウトがコンテンツ・リストに使 用される場合に値を別々にフェッチする必要がなくなります。

#### コントローラ・サイト・ページのレンダリング

cec compile-template コマンドを使用すると、Oracle Content Management サイト・ ページ・コントローラを使用しないで、サイト・ページをブラウザで直接レンダリン グできます。

コンパイラを使用して静的 HTML ページを作成せずに、Oracle Content Management サイトはコントローラ・モデルを使用してページをレンダリングします。これによっ て、ブラウザで実際に表示される前に、サイト、サイト内のページ、およびページを レンダリングする際に使用されるテンプレートに関する情報を取得するために多数の リクエストが発生します。



🥒 ノート:

controller.js ファイルの取得など、さらに多くのリクエストがあります。

#### コンパイル済サイト・ページのレンダリング

cec compile-template コマンドを使用すると、静的 HTML ファイルにあるすべてのこれらの ステップをコンパイルでき、これらのサーバー・リクエストを削減または除去することによ り、ランタイム・サイトのパフォーマンスが改善されます。さらに、実行時に変更する必要 のないリソースのリクエストを削減し、これらを静的ページにコンパイルできます。

デフォルト・コンパイル・プロセスは、既存のランタイム・コントローラの作業を実行し、 サイト内の page.json ファイルごとに物理的な HTML ページを作成します。これらのページ はサイトにデプロイでき、サイトとともに公開され、対応するページ URL がヒットするとき に使用されます。

#### テンプレートのコンパイル

サイトをコンパイルするには、サイト内のテーマおよびコンポーネントへのアクセス権が必要です。サイト・テンプレートはサイトのパッケージング・モデルです。テンプレートには、 サイトの移行および実行に必要なすべてのリソースが含まれています。

サイト・テンプレートを使用すると、OCE ツールキット環境にサイトをエクスポートでき、 ここでテンプレート内のサイトをコンパイルできます。次の項で、OCE ツールキットでシー ド済の BlogTemplate 内のサイトのコンパイルを説明します。

#### 設定

cec compile-template コマンドは OCE ツールキットで使用でき、ツールキットがインスト ールされたときにインストールされます。OCE ツールキットをインストールする標準の手 順に従います。

OCE ツールキットをインストールすると、cec コマンドを実行してテンプレートを作成します。

1. OCE ツールキットの開発ディレクトリをインストールします。

mkdir cec-src
cd ./cec-src
cec install

# Start up the CEC Toolkit Development server for testing cec develop &

 テンプレートを作成します。 シード済 BlogTemplate はカスタム・コンパイラで更新されています。このソースから テンプレートを作成します。

 $\ensuremath{\texttt{\#}}$  create a new template (must be in the cec-src folder if not already there)



cd ./cec-src
cec create-template BlogTemplate -f BlogTemplate

シード済 BlogTemplate はカスタム・コンパイラで前述のテンプレートを拡張します。

• コンテンツ・レイアウト・コンパイラ

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

ページ・レイアウト・コンパイラ

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/about-compile.js

#### テンプレートのコンパイル

テンプレートをコンパイルすると、テンプレート内のサイトの structure.json ファイルにある各ページの静的 HTML ページが作成されます。

コンパイル・ステップではページ・メタデータ(たとえば、100.json)をページ・レイア ウト(たとえば、about.html)を結合するため、HTMLページが即座に表示され、実行時 にコントローラの必要がありません。カスタム・コンパイラを使用すると、すでにナ ビゲーションおよびコンポーネントなどの動的アイテムがページに事前にコンパイル されているため、ランタイム JavaScript 実行を減らすことができます。

cec compile-template コマンドは、サイトの各ページに対して次のアクションを実行 します。

- サイトおよびページのメタデータを読み取ります
- ページ・レイアウトを読み取り、ページ・レイアウト・コンパイラを適用します
- ページの各スロットに対して次を行います:
  - page.json ファイルで定義されたグリッドでスロットを拡張します
  - スロット内の各コンポーネントに対して次を行います:
    - * コンポーネント・コンパイラを適用します
    - * 生成したコンポーネントのマークアップをスロット内の対応する場所に挿 入します
- コンパイル済マークアップでマクロを拡張し、実行時に Oracle Content
   Management レンダラで使用される SCS JavaScript オブジェクトを挿入します
- コンパイル済ページ・マークアップを src/templates/<template>/static フォル ダに保存します



サイトをコンパイルするには、次を実行します:

```
> 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/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- forcontent-and-experience-cloud createPage: Processing detail pageId 105. Preview URL: http://localhost:8085/templates/BlogTemplate/post-detail/1481786063052-dynamic-dom- manipulation-incontent-layout createPage: Processing detail pageId 105. Preview URL: http://localhost:8085/templates/BlogTemplate/post-detail/1481786063045-getting-media-url-in- thecontent-lavout createPage: Processing detail pageId 105. Preview URL: http://localhost:8085/templates/BlogTemplate/post-detail/1481786063053-getting-reference- items-incontent-layout createPage: Processing detail pageId 105. Preview URL: http://localhost:8085/templates/BlogTemplate/post-detail/1481786063048-navigating-to-a- search-pagewith-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- contentlayout createPage: Processing detail pageId 105. Preview URL: http://localhost:8085/templates/BlogTemplate/post-detail/1481786063044-rendering-the-content-layoutusing-mustache-template

All detail page creation calls complete.

#### コンパイル・エラー

コンパイル時に、3つのタイプのメッセージ(情報、警告、エラー)が表示されます。


- 1. 情報メッセージは認識する必要がある問題に関するものですが、これらのメッセージはコンパイルの一環としてみなされます。
  - プレースホルダ・コンテンツ・アイテムであり、実行時にレンダリングされます。
  - ページ・レイアウト・コンパイラがありません。ページ・レイアウトに動的要素がない可能性があります。そのため、動的要素がない場合は問題とみなされません。
  - "アクセス時にレンダリング"としてマークされたアイテム。すなわち、ページ をコンパイルしたときでも、サイト開発者はこのコンポーネントを実行時に動 的にレンダリングする必要があります。
- 警告メッセージは実行しているサイトのパフォーマンスに影響を及ぼす可能性が ある問題に関するもので、可能であれば修正する必要があります。このカテゴリの アイテムは次のとおりです。
  - コンテンツ・アイテムまたはカスタム・コンポーネント・コンパイラがありません。これらがなくても、コンポーネントは以前と同様に動的にページにレンダリングされます。
  - コンテンツ・レイアウト・マップがありません。システムのデフォルト・コン テンツ・レイアウトを使用したコンテンツ・アイテムのレンダリングが必要に なる可能性は低くなります。
- 3. エラー・メッセージは、コンパイルの失敗を示す問題に関するものです。ページは 可能であればコンパイルを続行しますが、全体のコンパイルはエラーで終了しま す。
  - これは、カスタム・コンパイラの JavaScript エラーのために起きる可能性が高く、エラーは解決する必要があります。

🖋 ノート:

繰返しメッセージを減らすために、同じメッセージは、複数ページで発生す るかどうかに関係なく、コンパイルごとに1回のみ表示されます。

前述の例は、次のコンパイル終了ステータスになります。

Compilation completed with 0 errors and 3 warnings. to display warnings, run with --verbose (-v) option.

"-v"オプションでコンパイル・ステップを再実行する場合、次のような出力が表示され ます。

> 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-e901-48f6-8044-36803c836aal" of type "scs-
contentlist" marked as
"render on access", will not be compiled.
Info: Component: "ba9f3711-4367-444e-ae38-71289fc10e73" of type "scs-
contentlist" marked as
"render on access", will not be compiled.
createPage: Processing pageId 403. Preview URL:
http://localhost:8085/templates/BlogTemplate/authors.html
All page creation calls complete.
Creating detail pages:
createPage: Processing detail pageId 105. Preview URL:
http://localhost:8085/templates/BlogTemplate/post-detail/1481786063051-developing-content-layout-for-
content-and-experience-cloud
createPage: Processing detail pageId 105. Preview URL:
http://localhost:8085/templates/BlogTemplate/post-detail/1481786063052-dynamic-dom- manipulation-in-
content-layout
createPage: Processing detail pageId 105. Preview URL:
http://localhost:8085/templates/BlogTemplate/post-detail/1481786063045-getting-media-url-in- the-
content-layout
createPage: Processing detail pageId 105. Preview URL:
http://localhost:8085/templates/BlogTemplate/post-detail/1481786063053-getting-reference- items-in-
content-lavout
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/1481786063046raising-triggers-from- content-layout createPage: Processing detail pageId 105. Preview URL: http://localhost:8085/templates/BlogTemplate/post-detail/1481786063044-rendering-the-contentlayout-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/cecsrc/src/templates/BlogTemplate/static

### 詳細ページのコンパイル

詳細ページのコンパイルでは、コンパイル時に見つかったすべてのコンテンツ・アイ テムが照合されます。その後、スラグ値を使用して新しい詳細ページの URL を定義し て、見つかったコンテンツ・アイテムごとに詳細ページを再コンパイルします。

前述のコンパイルでは、次の2つのセクションで出力があります。

- 1. ページのコンパイル
- 2. 詳細ページのコンパイル

前述の例では、Starter-Blog-Author ページにコンテンツ・レイアウト・マップ・エン トリがないことに関して警告が表示されます。ただし、Starter-Blog-Authorの詳細ペ ージは不要です。詳細ページは、Starter-Blog-Post コンテンツ・アイテム専用です。 これらのエラーを削除するには、設定で参照される明示的な詳細ページがない場合、 次のオプションを指定して、詳細ページのコンパイルからコンテンツ・アイテムを除 外します。

--noDefaultDetailPageLink, -o Do not generate compiled detail page for items/content lists

that use the default detail page.

したがって、コンパイルを再実行し、デフォルトの詳細ページを使用する詳細ページ の作成を除外すると、次のように出力されます。

> 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/search.html



All page creation calls complete.

Creating detail pages: createPage: Processing detail pageId 105. Preview URL: http://localhost:8085/templates/BlogTemplate/post-detail/1481786063051developing-contentlayout-for-content-and-experience-cloud createPage: Processing detail pageId 105. Preview URL: http://localhost:8085/templates/BlogTemplate/post-detail/1481786063052-dynamic-dom -manipulation-incontent-lavout createPage: Processing detail pageId 105. Preview URL: http://localhost:8085/templates/BlogTemplate/post-detail/1481786063045-getting-media-url-in- thecontent-layout createPage: Processing detail pageId 105. Preview URL: http://localhost:8085/templates/BlogTemplate/post-detail/1481786063053-getting-reference- items-incontent-layout createPage: Processing detail pageId 105. Preview URL: http://localhost:8085/templates/BlogTemplate/post-detail/1481786063048-navigating-to-a-search-pagewith-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/cecsrc/src/templates/BlogTemplate/static

### 特定のページのコンパイル

テンプレートのすべてのページをコンパイルする必要はありません。コンパイルするページ を選択できます。これは、サイト全体を継続的に再コンパイルするのではなく、デバッグま たは更新する特定のページで作業する場合に便利です。

特定のページをコンパイルするには、--pages (-p)オプションを使用し、その後にコンパイ ルするページのリストを指定します。

💉 ノート:

ノート: コンテンツ・アイテムがリスト参照詳細ページ内のページにある場合、詳細 ページが明示的に含まれない場合でも、これらはコンパイルされます。

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/cecsrc/src/templates/BlogTemplate/static

## サイトのコンパイル

前述のステップでは、ローカル・テンプレートを作成してコンパイルする方法の概要 が説明されています。通常、実際の Oracle Content Management サイトがコンパイル されます。

Oracle Content Management サイトをコンパイルするには、サイトをテンプレートに エクスポートしてからテンプレートをコンパイルする必要があります。テンプレー ト・パッケージには、サイト、およびテンプレートともにエクスポートされるテーマ とコンポーネントにあるサイトのコンパイルに必要なカスタム・コードが含まれてい ます。

### サイト・コンパイルの前提条件

次のステップでは、Oracle Content Management サーバーで BlogSite という名前のサ イトを作成していることを想定しています。前述の BlogTemplate をアップロードし て、テンプレートからこのサイトを作成できます。

例:

> cec upload-template BlogTemplate --sever UAT

```
> cec create-site BlogSite --template BlogTemplate --repository
<yourRepository> --
localizationPolicy <yourLocalizationPolicy> --defaultLanguage en-US --
server UAT
```

サイトを選択または作成すると、次の項のステップを実行してサイトをコンパイルで きます。

### サイトのコンパイル

**OCE** ツールキットの cec create-template-from-site コマンドはサイトのテンプレートを作成できるため、サイトをコンパイルできます。

```
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]
```

サイトをコンパイルするには:



1. サイトからテンプレートを作成します。

```
> cec create-template-from-site BlogTemplate --site BlogSite --
includeunpublishedassets -
server UAT
```

- 2. テンプレートをダウンロードします。
  - > cec download-template BlogTemplate --server UAT
- 3. テンプレートをコンパイルします。

```
> cec compile-template BlogTemplate --noDefaultDetailPageLink --verbose --
server UAT
```

--channelToken elbb88cdcle025c8dd278f6b676877a3

### / ノート:

サイトの公開チャネルのサーバーでサイトのチャネル・トークン(-channelToken (-c)オプション)を取得する必要があります。これは、テンプレ ートのサイト内にあるすべての問合せに使用されます。

コンパイルしたサイト・ページをアップロードします。
 コンパイルした静的ファイルをサイトの静的フォルダにコピーします。

```
> cec upload-static-site-files src/templates/BlogTemplate/static --site
BlogSite --server UAT
```

#### コンパイル前の動作に戻す

コンパイル前の動作に戻すには、サイトにアップロードした静的ファイルを削除する必要が あります。

> cec delete-static-site-files BlogSite --server UAT

サイトが現在公開済である場合、再度サイトを公開します。これによってサイトから静的な フォルダが削除されますが、サイトは削除されません。

## カスタム・コンパイラ

コンパイル・プロセスは、カスタム・コンパイラでさらに拡張できます。カスタム・コンパ イラをコールして、ページ・レイアウト、セクション・レイアウト、カスタム・コンポーネ ントまたはコンテンツ・レイアウトをページにコンパイルでき、実行時にコンポーネントを 動的に追加する必要がなくなります。

カスタム・コンパイラがコンポーネントに対して存在しないか、コンポーネント・コンパイ ラがマークアップを返さない場合、コンパイルされていないようにコンポーネントが実行時 にレンダリングされます。

カスタム・コンパイラは、次のコンポーネントに対してサポートされています。

- ページ・レイアウト
- セクション・レイアウト
- カスタム・コンポーネント
- コンテンツ・レイアウト

次のサンプルは、ページ・レイアウトおよびコンテンツ・レイアウト・コンパイラを 示しています。セクション・レイアウトおよびカスタム・コンポーネントのコンパイ ラは、コンテンツ・レイアウトのコンパイラと同じモデルに従います。

カスタム・コンパイラが存在しない場合、コンポーネントは render.js ファイルでレ ンダリングします。

カスタム・コンパイラが存在してコールされると、生成された HTML がページに挿入 されます。カスタム・コンパイラが、コンポーネントで実行時に JavaScript ハイドレ ーションが必要であることを示す場合もあります。このような場合、render.js ファイ ル内のコンポーネントの hydrate()関数がコールされます。コンポーネントでハイド レーションが不要な場合、コンポーネントの render.js ファイルはロードされません。

/ ノート:

コンパイルは実行時(公開済サイト)専用機能です。サイトが編集、ナビゲー ションまたはプレビュー・モードで表示される場合、ページは通常どおりに 動作し、すべてのコンポーネントは常に動的にページに追加されます。

### 制約

cec compile-template 操作は NodeJS アプリケーションであり、ブラウザの外部で実行されます。ページはブラウザでレンダリングされないため、DOM またはウィンドウ・オブジェクトがなく、JQuery、VueJS または KnockoutJS などのクライアント側の JavaScript ライブラリは機能しません。

JSDOM などのライブラリを使用して DOM オブジェクトを作成し、これらのクライア ント側ライブラリを実行できますが、これを実行する利点はほとんどありません。単 純な HTML 解析の場合、cheerio などの NodeJS HTML パーサーを使用できます。

マークアップから返された HTML は、有効な HTML である必要があります。これはパ ーサーから渡され、解析された HTML のみがページに追加されます。これは、コンパ イルされた HTML にスロットを分割する可能性のある一致しないタグが含まれていな いことを確認するためです。

## カスタム・コンパイラのデバッグ

カスタム・コンパイラの開発時に、コードのデバッグが必要になります。

cec compile-template コマンドには--debug (-d)オプションがあり、これはデバッガ をプロセスにアタッチできるように設定された--inspect-brk フラグでコンパイラを 起動します。標準のノード・デバッグに従って、コードを確認できます。

cec compile-template BlogTemplate --noDefaultDetailPageLink --debug



Debugger listening on ws://127.0.0.1:9229/8a8eba83-42d2-476b-adc1b29ab4e92642
For help see https://nodejs.org/en/docs/inspector

さらに、--pages (-p)オプションを使用すると、特定のページにコンパイルされるページを 制限できます。

### ページ・レイアウト・コンパイラ

ページ・レイアウト・コンパイラは NodeJS (CommonJS) JavaScript モジュールであり、対応するページ・レイアウトをコンパイルします。

特定のページ・レイアウトのページ・レイアウト・コンパイラは、-compile.js 拡張機能との名前の関連付けによって定義されます。

- src
  - themes
    - * <yourTheme>
      - layouts
        - * <yourPageLayout>.html
        - * <yourPageLayout>-compile.js

-compile.js がページ・レイアウトに対して存在しない場合、カスタム・コンパイルが適用 されません。

ページ・レイアウト・コンパイラは compile()インタフェースを実装する必要があり、これ は、about-compile.js など約束を返します。

```
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</pre>
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();
```

## コンポーネント・コンパイラ

カスタム・コンポーネント・コンパイラはページ・コンパイラと同じモデルにすべて 従い、セクション・レイアウト、カスタム・コンポーネントおよびコンテンツ・レイ アウト用に作成できます。

コンパイル時に、cec compile-template コマンドは、コンポーネントの render.js ファイルと同じ場所で compile.js ファイルを検索します。

- src
  - components
    - * <yourComponent>
      - * assets
        - * render.js
          - compile.js

このファイルが存在しない場合、コンポーネントはコンパイルされず、実行時にレン ダリングされます。

ファイルが存在する場合、約束を返す compile()インタフェースを実装する必要があり ます。たとえば、次の Starter-Blog-Author-Summary はカスタム・コンテンツ・レイ アウト・コンパイラです。

```
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',</pre>
    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;
```

### コンテンツ・レイアウト・コンパイラ

指定されたコンテンツ・タイプのコンテンツ・アイテムが公開されたときに、コンテ ンツ・レイアウト・コンパイラによりコンテンツ・レイアウトの HTML が出力されま す。

コンパイル時に、cec compile-content コマンドは、コンポーネントの render.jsフ ァイルと同じ場所で compile.js ファイルを検索します。

- src
  - components
    - * <yourComponent>
      - * assets
        - * render.js
        - compile.js

このファイルが存在しない場合、レイアウトはコンパイルされず、実行時にレンダリ ングされます。

ファイルが存在する場合、約束を返す compile()インタフェースを実装する必要があり ます。次にコンテンツ・レイアウト・コンパイラ、続いて結果の layout.html 出力の例 を示します:

```
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',</pre>
    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,
```



```
module.exports = ContentLayout;
```

結果の layout.html は Web アプリケーション内、またはコンテンツ・レイアウトの静的 HTML 出力が必要などの場所にも挿入できます。layout.html は、compile.js ファイルと同じディレク トリに出力されます。

### **SCSCompileAPI**

SCSRenderAPIと同様に、各コンパイル関数に渡される SCSCompileAPI があります。

これには次のプロパティおよび関数が含まれています。

- プロパティ:
  - navigationRoot: サイトのルートであるノードの ID。
  - **navigationCurr**: 現在のページ・ノードの ID。
  - structureMap: ID によってアクセスされる、サイト階層のすべてのノード。
  - siteInfo: すべてのサイト・プロパティ。
- 関数:
  - getContentClient: コンテンツ API のコールで使用するための contentClient インス タンスを取得します。
  - getCustomSiteProperty: サイトのコンパイル時にカスタム・プロパティを取得します。

siteinfo.json ファイルにサイト・レベルのカスタム・データおよびプロパティを格納 して、データとプロパティに基づいてサイトをカスタマイズできます。たとえば、ペ ージの背景のテクスチャ、フッターのリンクまたは Web サイト名を変更できます。

### コンポーネント・ハイドレーション

ハイドレーションでは、HTML がブラウザでレンダリングされるときに、JavaScript 動 作をページのコンパイル済 HTML に追加する処理を参照します。

たとえば、マスター/詳細としてレンダリングするページに2つのコンポーネントがあ る場合、マスターのアイテムをクリックすると、詳細コンポーネントを更新する必要 があります。これは、ページで実行される JavaScript によってすべて処理されます。 これを動作させるには、on click イベント・ハンドラをマスター・コンポーネントの 要素に追加し、リスナーを詳細コンポーネント・コンテナに追加することで、2つのコ ンポーネントがページにレンダリングされた後で、これらのコンポーネントの HTML を*ハイドレーション*して、on click イベントがイベントを渡したペイロードに基づい て発生したときに再度レンダリングする必要があります。

コンポーネント・コンパイラが HTML をページに挿入します。コンポーネントでイベ ント・ハンドラのようなものを追加するために実行時に実行される追加の JavaScript が必要な場合、インラインの JavaScript またはハイドレーション関数など、オプショ ンがいくつかあります。選択するソリューションは要件によって異なります。

### インライン JavaScript

返されたコンパイル済マークアップに<script>タグを直接挿入できます。スクリプト では、ページ実行を実行します。

#### 例:

```
<script src="/_sitesclouddelivery/renderer/libs/scs-core/jssor-</pre>
slider/js/jssor.slider.min.js" type="text/javascript"></script>
<div id="slider container c46b122d-978a-429d-aa25-9b5698428f6f"</pre>
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
```

### ハイドレーション関数

JavaScript をインライン化するのではなく、ハイドレーション関数を render.js ファイルに 含め、コンパイル済マークアップを返すときに実行時にコンポーネントで*ハイドレーション* が必要になることに注意します。このハイドレーションでは<script>タグの繰返しを回避 し、既存の JavaScript コードを活用してイベントを管理できます。

render.js ファイルをロードしても、render()関数はハイドレーション時にコール*されませ* ん。hydrate()関数のみがコールされます。

💉 ノート:

コンパイル済コンポーネントでハイドレーションが不要な場合、コンポーネントの render.js ファイルはロードされません。

たとえば、カスタム・コンテンツ・レイアウト・コンパイラでは- { hydrate: true }が返さ れます。

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 });

さらに、必要に応じて、カスタム・コンパイラでは*ハイドレーション*・プロパティを追加で き、これは実行時に検索されます。例:

```
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}}">
最後に、コンポーネントでハイドレーションが必要な場合、実行時にコンポーネント
の render.js ファイルがロードされ、hydrate() 関数がコールされ、コンパイル済マー
クアップを含むコンテナ<div>が渡されます。
たとえば、render.jsでは、hydrate()関数は次のようになります。
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 $parent0bj = $(parent0bj),
           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);
   }
```



## 公開

コンパイル済の静的ページを生成してサイトの静的フォルダにアップロードした後、アクティブにするには、そのページのサイトを公開または再公開する必要があります。同様に、コンパイル前のサイトの配信動作に戻すには、サイトから静的ファイルを削除した後に公開または再公開する必要があります。

公開時に、アップロードした静的ページは配信できるようになります。これらのファイルは 公開プロセス時にコピーされるため、公開操作のパフォーマンスがファイル数に応じて低下 する可能性があります。

公開操作では現在のセットの静的ファイルを使用し、これらを配信できるようにします。こ れらのファイルは動的サイトで行われた変更内容と同期されている場合があり、動的サイト をミラーリングする場合もあります。また、そうでない場合もあります。適切なタイミング での静的ファイルのコレクションの更新は、サイト開発者に任されています。

### 静的サイト配信の優先度

関連付けられた静的ファイルがサイトにある場合、これらのファイルが配信され、サーバー に送信される URL と照合されます。受信 URL が静的ファイルと一致しない場合、サイトの controller.html ファイルがリクエストに対して返されます。これは、サイト配信の既存の 動的モデルに従います。

Oracle Content Management サイトは、関連付けられた JSON ファイルによって 301 および 302 リダイレクトも定義できます。リダイレクトが構成されている場合、リダイレクトが静 的ファイルよりも優先されます。URL がリダイレクト・ルールと静的ファイルの両方と一致 する場合、リダイレクトがサーバーから配信されます。

サイト配信の URL 評価は次のフローのとおりです。

- URL は構成済リダイレクトと一致しますか。
   一致する場合、リダイレクト・レスポンスを発行します。
- 2. URL は静的ファイルに対応していますか。 モバイル静的ユーザー・エージェント・リストがサイトに対して構成されていて、リクエ ストがリストに一致するブラウザから送信される場合、モバイル静的ファイルを配信しま す。
- 3. それ以外の場合、動的サイトの controller.html ファイルを配信します。

### N ノート:

モバイル静的ファイルがサイトに関連付けられていて、ユーザーが配信に CDN を 使用している場合、CDN (通常は Akamai)が標準デスクトップ・リクエストと別に モバイル・ブラウザ・リクエストをキャッシュするように構成されている必要があ ります。

CDN が別々のモバイル/標準キャッシュで構成されていない場合、モバイル・ブラ ウザが標準レスポンスを受信し、デスクトップ・ブラウザがモバイル・ブラウザ用 のレスポンスを取得する可能性があります。



### キャッシュ・ヘッダー

Web サーバーからのレスポンスで HTTP ヘッダーは、ブラウザがページをキャッシュ する方法を決定するのに役立ちます。静的ページも、ブラウザ・キャッシングを簡単 に実行できるようにキャッシュ・ヘッダーとともに配信されます。

セキュアなサイトの場合、次のヘッダーがレスポンスで送信されます。

- Cache-Control: no-store
- Pragma: no-cache

標準のセキュアでないサイトの場合、次のヘッダーが送信されます。

- Cache-Control: max-age=300
- Edge-Control: !no-store,max-age=2592000,downstream-ttl=1800
   Edge-Control ヘッダーは CDN キャッシュ・ヘッダーを簡単に実行するのに役立 ちます

これらの2つの領域のいずれかでヘッダーをカスタマイズした場合、レスポンスには、 ここでリストした標準のものではなく、カスタム・ヘッダーが含まれます。

テナント・レベルまたはサイト・レベルでこれらのレスポンスを制御できます。

### 詳細ページ

**Oracle Content Management** サイトの詳細ページを使用すると、1ページに多数のコン テンツ・アイテムの情報を表示できます。

たとえば、同じ詳細ページを使用して多数の URL を処理できます。これらの URL の それぞれには同じページ構造が表示されますが、それぞれスラグ値が item1.html、 item2.html および item3.html であるコンテンツ・アイテムに関するコンテンツが表 示されます。このような状況では、cec テンプレート・コンパイラで 4 つのファイルが 作成される可能性があります。

- /detail/item1.html
- /detail/item2.html
- /detail/item3.html
- /detail.html

最後のファイルを使用すると、サイトを再コンパイルおよび再公開することなく、新 しく公開された資料を Web サイトに表示できます。この例では、スラグ値 item4.htmlを持つコンテンツ・アイテムが、サイトのオンライン後に公開されます。 静的な/detail.html ページでは、その新規アイテムをサイトで動的に表示できます。 URL /detail/item4.html では detail.html ページを配信しますが、item4.html コン テンツ・アイテムに関するコンテンツを表示します。

cec コンパイラでは detail.html ページを生成してコンテンツ・アイテムを表示しま す。このため、コンパイル済 detail.html ページ内の相対 URL に追加の親セグメント (../)が含まれます。したがって、直接参照すると、detail.html ページ自体が適切に 表示されません。このような理由から、detail.html ページ自体の参照またはページ・ ナビゲーションへの追加を行わないでください。



## コンテンツ・アイテムのチャネルへの追加

OCE ツールキットの control-content コマンドを使用すると、コンテンツ・アイテムを Oracle Content Management サーバーのチャネルに追加できます。

control-content <action>コマンドには、コンテンツ・アイテムを Oracle Content Management チャネルに追加するための add というアクションがあります:

cec control-content add -c Channell -r Repol -s UAT

このコマンドは、リポジトリ Repo1 内のすべてのアイテムを 登録済サーバーの UAT 上のチャネル Channel1 に追加します。

サーバーを-s <server>で指定することも、cec.properties ファイルで指定されたサーバー を使用することもできます。

content-usage コマンドの有効なアクションは次のとおりです。

- publish
- unpublish
- add
- remove

content-usage コマンドのオプションは次のとおりです。

- --channel, -c Channel [必須]
- --repository, -r Repository [<action>が add の場合必須]
- --server, -s 登録された Oracle Content Management サーバー
- --help, -h ヘルプを表示します[boolean]

control-content コマンドの例は次のとおりです。

#### cec control-content publish -c Channel1

Publish all items in channel Channell on the server specified in the cec.properties file

#### cec control-content publish -c Channel1 -s UAT

Publish all items in channel Channell on the registered server UAT

cec control-content unpublish -c Channel1 -s UAT

Unpublish all items in channel Channell on the registered server UAT

cec control-content add -c Channell -r Repol -s UAT

Add all items in repository Repol to channel Channell on the registered server UAT.

cec control-content remove -c Channel1 -s UAT



Remove all items in channel Channell on the registered server UAT

## モバイル・デバイスのコンパイル・サイト

OCE ツールキットを使用すると、サイト Web ページのモバイル・レイアウトをコン パイルできます。モバイル・レイアウトは、同じコンテンツのデスクトップ・ページ・ レイアウトと異なるものにできます。また、モバイル・レイアウトとデスクトップ・ レイアウトを同じものにすることもできます。

サイト・エディタで、モバイル・デバイスにデスクトップ・レイアウトと同じページ・ レイアウトを選択することも、異なるページ・レイアウトを指定することもできます。 OCE ツールキットで、モバイル・デバイスの静的レイアウトを別々にコンパイルでき ます。

モバイル・デバイスにサイト・ページを別々に表示できます。モバイル・デバイスで レンダリングされるページは、デスクトップ・レイアウトのページとバナーが異なる 場合があります。

OCE ツールキットで、cec compile-template のヘルプ・ページに、サイト・テンプ レートのコンパイル時に特定のデバイスをターゲットとする targetDevice オプショ ンが表示されます。

C:\git\webclient\developer\test\sites-compiler\cec-install>cec compile-templatehelp 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 - < <snerver> to make content queries against this server (requires channelToken). Optionally specify - &lt; <snermaltoken any="" content="" generating="" urls.<br="" when="">Optionally specify - &lt; <snermaltoken any="" content="" generating="" urls.<br="" when="">Optionally specify - &lt; <snermaltoken any="" content="" generating="" urls.<br="" when="">Optionally specify - &lt; <snermatrix compile.<br="" of="" pages="" to="">Optionally specify - &lt; <snermatrix description="" flag.<br="" of="" the="" withinspect-brk="">Optionally specify - &lt; <snermatrix creating="" default="" locale="" of="" pages.<br="" the="" when="">Optionally specify - &lt; <snermatrix creating="" default="" locale="" of="" pages.<br="" the="" when="">Optionally specify - &lt; <snermatrix creating="" default="" locale="" of="" pages.<br="" the="" when="">Optionally specify - &lt; <snermatrix creating="" default="" locale="" of="" pages.<br="" the="" when="">Optionally specify - &lt; <snermatrix creating="" default="" locale="" of="" pages.<br="" the="" when="">Optionally specify - &lt; <snermatrix creating="" default="" locale="" of="" pages.<br="" the="" when="">Optionally specify - &lt; <snermatrix creating="" default="" locale="" of="" pages.<="" td="" the="" when=""></snermatrix></snermatrix></snermatrix></snermatrix></snermatrix></snermatrix></snermatrix></snermatrix></snermatrix></snermaltoken></snermaltoken></snermaltoken></snerver>			
Options:       The registered CEC server        channelToken, -c       The channel access token to use        type, -t       The type of content to retrieve        pages, p       The list of pages to compile        necturse, -r       Compile all child pages of thos        noDefaultDetailPageLink, -o       Do not generate compiled detail        targetDevice, -a       The Linget device type when usis        includeLocale, -l       Include default locale when cree        verbose, -v       Run in verbose mode to display of	for content URLs from the serve [published   draft] e specified in the page list set-brk" option to debug compilation pages page for items/content lists that use the default detail page gradarbue Jayouts [desktop   mobile] ting pages ill warning messages during compilation.		
Examples: Con cec compile-template Temp1 - Con cec compile-template Temp1 - c channelToken Con cec compile-template Temp1 - c channelToken - s UAT -t draft Con cec compile-template Temp1 - p 184,112,183 -r Con cec compile-template Temp1 - d Way	mpiles the site in template Temp1 using content stored in the template. mpiles the site in template Temp1 using the given channelToken for any content URLs. mpiles the site in template Temp1 retrieving draft content from the specified server. mpiles the specified pages in the site in template Temp1 including all child pages. its for the debugger to be attached. Once attached, compiles the site in template Temp1.		
C:\git\webclient\developer\test\sites-compiler\cec-install>_			

サイトのコンパイル時に、desktop 用にコンパイルするか、mobile 用にコンパイルす るかを指定できます。デスクトップ・ファイルは、static/_files の下にあります。 モバイル・ファイルは、static/_mobilefiles の下にあります。



C:\git\webclient\developer\test\sites-compiler\cec-install>cec compile-template Corporate-Site-TemplatetargetDevice 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 10. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/developing-templates/html createPage: Processing pageId 10. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/developing-templates/html createPage: Processing pageId 10. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/developing-templates/html createPage: Processing pageId 10. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/developing-templates/navges.html createPage: Processing pageId 10. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/developing-templates/navgeton.html createPage: Processing pageId 10. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/developing-templates/navgetont.html createPage: Processing pageId 10. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/developing-templates/corporate.html createPage: Processing pageId 10. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/developing-templates/corporate.html createPage: Processing pageId 200. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/developing-templates/corporate.html createPage: Processing pageId 200. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/developing-templates/corporate-Site- Implate/developing-templates/sign-Im.html All page creation calls complete.
Compilation completed with 0 errors and 5 warnings. to display warnings, run withverbose (-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-TemplatetargetDevice 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 10. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/developing-templates.html createPage: Processing pageId 10. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/developing-templates/hemes.html createPage: Processing pageId 10. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/developing-templates/navges.html createPage: Processing pageId 10. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/developing-templates/navges.html createPage: Processing pageId 10. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/developing-templates/navges.html createPage: Processing pageId 10. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/developing-templates/navges.onth.html createPage: Processing pageId 10. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/developing-templates/comporate.html createPage: Processing pageId 10. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/developing-templates/comporate.html createPage: Processing pageId 10. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/developing-templates/comporate.html createPage: Processing pageId 200. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/developing-templates/comporate.html createPage: Processing pageId 200. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/developing-templates/comporate.html createPage: Processing pageId 200. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/privacy-policy.html
Compilation completed with 0 errors and 6 warnings. to display warnings, run withverbose (-v) option.
<pre>*** compiled template is ready to test *** compiled tites from under: C:\git\webclient\developer\test\sites-compiler\cec-install\src\templates\Corporate-Site-Template\static *** 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 ***</pre>

モバイル・デバイスのテンプレートをコンパイルすると、OCE ツールキットのコマンド upload-static-site-files ではモバイル・ファイルがサポートされます。

## サイトのライフサイクルおよびコンパイル済ページ

サイトからテンプレートを作成する場合、コンパイル済ページはテンプレートに組み込まれ ません。これは、サイト開発者が動的ページを配信するように想定した場合に静的ページが 配信される問題を回避するためです。

その後、テンプレートからサイトを作成する場合、サイト・ページをコンパイルして、これ らを新規サイトにアップロードする必要があります。

# Oracle Content Management サーバーでの新規のサイト またはアセットの翻訳ジョブの作成

**OCE** ツールキットを使用して、**Oracle Content Management** のサイトまたはアセットの翻訳 ジョブを作成します。

多言語サイトを索引付けする前に、翻訳ジョブが必要です。翻訳ジョブを作成するには:

- 1. 「サイト」ページの上部メニューで「翻訳」をクリックします。
- 2. 「翻訳ジョブの作成」ダイアログでジョブの名前を入力し、デフォルト・ソース言語、タ ーゲット言語および翻訳ジョブ・コンテンツを選択します。

翻訳パッケージにすべてのサイト・コンテンツとターゲット指定されたアセットを含める ことも、サイト・コンテンツのみを含めることも、サイトの公開チャネルにターゲット指 定されたアセットのみを含めることもできます。

「翻訳しない」テキスト設定で構成されているすべてのコンテンツ・アイテムを翻訳から 除外します。たとえば、通常、製品名は翻訳しません。

3. 「作成」をクリックして翻訳ジョブを作成します。

4. OCE ツールキットのコマンドを使用して使用可能なジョブをリストします。

cec componen Asset transl	ts> cec list-translation-jobs ation jobs:			
Name Source Langu Languages	age Target Languages	Sta	tus Pending	
Site transla Name Source Langu Languages	tion jbs: age Target Languages	Sta	tus Pending	
demol US	fr-FR,es-ES	REA fr	DY -FR,es-ES	en-
searchdemol US	fr-FR,es-ES	TRA	NSLATED	en-

5. 翻訳ジョブをダウンロードします。

cec components> cec download-translation-job demo1
 - translation job downloaded to /Users/<user-name>/Dev/webclient/
developers/sites-toolkit/cec-components/demo.zip
 - update the translation job status to INPROGRESS.
cec components> cec translate dmo1.zip -1 all -t demo1-xlate.zip
 - target languages: fr-FR,ex-ES
 - translation finished: /Users/<user-name>/Dev/webclient/
developers/sites-toolkit/cec-components/demo1-xlate.zip

6. 翻訳バンドルを開き、翻訳する言語のリソース・フォルダを作成します。

```
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
  inflating
```

7. 翻訳ジョブをインポートします。

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

# 言語サービス・プロバイダでのサイトの翻訳

OCE ツールキットのコマンドライン・インタフェースおよび言語サービス・プロバイダ(LSP)を使用して、複数言語のサイトの翻訳を管理できます。

サイトのローカリゼーション・ポリシーでは、そのサイト用に英語(米国) (en-US)など デフォルト言語、およびドイツ語やフランス語など1つ以上の代替言語を指定します。 サイトのテキスト文字列は指定した代替言語に翻訳できます。翻訳前にサイトの言語 を変更しても、テキスト文字列はデフォルト言語で表示されます。

OCE ツールキットでは、次の翻訳オプションをコマンドライン・インタフェースで提供します。

Translation	
cec list-translation-jobs	Lists translation
jobs.	[alias:
ltj]	
cec create-translation-job <name></name>	Creates a
translation job <name> for a site on CEC server.</name>	
[alias: ctj]	
cec download-translation-job <name></name>	Downloads
translation job <name> from CEC server.</name>	
[alias: dtj]	
cec submit-translation-job <name></name>	Submits translation
job <name> to translation connection <connection>.</connection></name>	[alias: stj]
cec ingest-translation-job <name></name>	Gets translated job
<name> from translation connection and ingest.</name>	[alias: itj]
cec upload-translation-job <name></name>	Uploads translation
job <name> to CEC server.</name>	[alias: utj]
cec create-translation-connector <name></name>	Creates translation
connector <name>.</name>	[alias: ctc]
cec start-translation-connector <name></name>	Starts translation
connector <name>.</name>	[alias: stc]
cec register-translation-connector <name></name>	Registers a
translation connector.	
[alias: rtc]	

cec list-translation-jobs コマンドを使用すると、サーバー上にすでに存在する翻 訳ジョブをリストできます。例:

```
cec ltj -s
Server: <server-name>
Asset translation jobs:
Name
Target Languages
testHash
fr-FR,de-DE
Site translation jobs:
Name
Target Languages
```

Status Source Language Pending Languages INPROGRESS en-US fr-FR,de-DE

Status Source Language Pending Languages



demoTest DE,fr-FR TRANSLATED en-US de-

パラメータを指定しないか、-h を指定して cec コマンドを入力すると、コマンドのヘルプが 表示されます。cec コマンドライン・ユーティリティの使用を参照してください。

次の各項では、LSP を使用したサイトの翻訳について説明します。

- 1. OCE ツールキットによる翻訳ジョブの作成
- 2. 翻訳ジョブのリスト
- 3. 翻訳コネクタの作成
- 4. 多言語サイトのサイト・マップの生成
- 5. 言語サービス・プロバイダへの翻訳ジョブの送信
- 6. サーバーへの翻訳ジョブのアップロード

### OCE ツールキットによる翻訳ジョブの作成

OCE ツールキットのコマンドを使用すると、ローカル・システムでサイト翻訳ジョブを作成できます。

サイトの新規翻訳ジョブを作成するには、cec create-translation-job コマンドを使用しま す。このコマンドでは、サイトのすべてのアセットを検索し、このサイトから翻訳する必要 のあるすべてに対して zip ファイルを作成します。

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

翻訳オプションの詳細は、「Oracle Content Management サーバーでの新規のサイトまたはア セットの翻訳ジョブの作成」を参照してください。

### 翻訳ジョブのリスト

サーバー上の翻訳ジョブをリストすると、ジョブが作成されて作業する準備ができているこ とを確認できます。

cec list-translation-jobs -s			
Server: <server-name></server-name>			
Asset translation jobs:			
Name	Status	Source Language	Target
Languages	Pending Languages		
testHash	INPROGRESS	en-US	fr-
FR, de-DE	fr-FR,de-DE		



Site translation jobs:		
Name	Status	Source Language
Target Languages	Pending Languages	
demoTest	TRANSLATED	en-US
de-DE,fr-FR		
FridayDemo	READY	en-US
de-DE,fr-FR	de-DE,fr-FR	

FridayDemo ジョブは READY 状態です。

### 翻訳コネクタの作成

言語サービス・プロバイダ(LSP)はサイトの翻訳に役立ちます。LSP への翻訳コネクタ を使用すると、翻訳ジョブを送信および取得できます。

翻訳ジョブを送信する前に、翻訳コネクタを作成する必要があります。LSP なしでサ イトを翻訳するには、モック翻訳コネクタを作成して実行できます。cec createtranslation-connector コマンドを使用して翻訳コネクタを作成し、cec starttranslation-connector コマンドを使用してこれを起動します。

```
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
```

OCE ツールキットを使用し、必要な API でこれを実行して翻訳コネクタをテストします。

1. OCE ツールキットでコネクタを登録します。

>cec register-translation-connector

2. ツールキットを開いて、翻訳の接続ページに移動します。

>http://localhost:8085/public/translationconnections.html

 翻訳コネクタの検証ページでステップを実行します。これらのステップはコネク タ環境で/data フォルダにある translationBundle.zip ファイルを使用して、コ ネクタを検証します。

翻訳コネクタの SDK を使用すると、Oracle Content Management の翻訳コネクタを開 発できます。この SDK は、翻訳コネクタ API のサンプル NodeJS 実装です。サンプル では Oracle Content Management 翻訳ジョブの zip ファイルを受け入れ、ファイル内 のすべてのリソースを翻訳し、すべての翻訳を含む新しい zip ファイルを返します。



SDK では、ユーザーが LSP へのアクセス権を持ち、実際の文字列の翻訳を実行できる必要が あります。モック LSP サーバーは SDK に含まれており、対象となるロケールを文字列の先 頭に付加することで LSP からのレスポンスを模倣します。

翻訳コネクタ SDK は3つの主なモジュールから構成されます。

- コネクタ: 必要な Oracle Content Management 翻訳コネクタ API を実装している翻訳コネクタ。
- ジョブ・マネージャ: 言語サービス・プロバイダで翻訳されるときに、コネクタ・ジョブの状態を維持するファイル・システムベースのサンプル・ジョブ・マネージャ。
- プロバイダ:翻訳のドキュメントを送信し、翻訳したドキュメントを取得するために LSP で必要な特定のセットの API の実装。

モック翻訳プロバイダ JS をコピーして、その内部にすべてのメソッドを実装できます。

### 多言語サイトのサイト・マップの生成

OCE ツールキットを使用して多言語サイトのサイト・マップを生成し、マップをサイトに公開します。

cec create-site map <site>コマンドを使用して、Oracle Content Management サーバー上 に多言語サイトのサイト・マップを作成できます。例:

cec create-site-map Site1 -u http://www.example.com/site1

このコマンドはサイト構造を横断し、サイト・ページ階層と一致するサイト・マップ階層を 作成し、Oracle Content Management サーバー上に指定したサイト URL でサイト・マップを 作成します。

コマンドのオプションは次のとおりです。

```
--url, -u <url> Site [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 [boolean]
```

<changefreq>オプションの有効な値は次のとおりです。

- always
- hourly
- daily
- weekly
- monthly
- yearly
- never
- auto



cec create-site-map コマンドの例は次のとおりです。

cec create-site-map Site1 -u http://www.example.com/site1

cec create-site-map Sitel -u http://www.example.com/sitel -f sitemap.xml

cec create-site-map Site1 -u http://www.example.com/site1 -p

cec create-site-map Sitel -u http://www.example.com/sitel -c weekly -p

サイト・マップを公開するには、サイト更新を作成し、サイト・マップを更新し、更 新をコミットします。

### 言語サービス・プロバイダへの翻訳ジョブの送信

OCE ツールキットでは言語サービス・プロバイダに送信可能な zip ファイルを指定して、翻訳ジョブでの作業を開始します。

翻訳コネクタによって翻訳ジョブをLSP に送信できます。コネクタがファイルを解凍 して個々のファイルをすべて LSP に送信する必要があるため、送信に時間がかかりま す。その後、LSP は翻訳ジョブのプロジェクトを作成できます。ファイルがプロジェ クトにインポートされたら、翻訳用ファイルの選択を開始できます。その後、LSP は 翻訳ステータスのモニターを開始します。

ステータスを確認するには、オプションを指定しないで cec list-translationjobs コマンドを使用して翻訳ジョブをローカルでリストします。ジョブのステータ スが READY TO INGEST の場合、LSP から zip ファイルをダウンロードすると翻訳ジョ ブを取得できます。翻訳コネクタが zip ファイルを LSP に送信し、LSP がファイルの リストを翻訳し、コネクタがダウンロードして取得可能な zip ファイルで LSP からフ ァイルを戻します。

cec list-tra	inslation-jobs		
Local transl	ation jobs:		
Name		Status	Source
Language Tar	get Languages		
FridayDemo		READY TO INGEST	en-
US	de-DE,fr-FR		
demoTest		READY TO INGEST	en-
US	de-DE,fr-FR		

zip ファイルを取得すると、翻訳ジョブがコネクタから OCE ツールキットに取り込ま れます。

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>/ceccomponents/src/main/translationJobs/FridayDemo



zip ファイルの取得後、翻訳ジョブをローカルにリストすると、翻訳ジョブのステータスは TRANSLATED になります。

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		

翻訳されたジョブを Oracle Content Management サーバーにアップロードできます。通常、 ジョブが初期のクイック翻訳を処理し、これをレビュー用に送信します。サイトの翻訳には、 LSP から返された翻訳ジョブの取得、翻訳への接続および翻訳ジョブの再送信があるため、 完了に数週間かかる場合があります。

## サーバーへの翻訳ジョブのアップロード

翻訳ジョブを取得した後で、これを Oracle Content Management サーバーにアップロードしてから自分のサイトで翻訳を確認できます。

```
cec upload-translation-job コマンドを使用して、翻訳 zip ファイルをサーバーにアッ
プロードします。
```

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

翻訳ジョブをアップロードすると、サーバー上のジョブのステータスは INPROGRESS になります。

cec list-translation-jobs -s			
Server: <server-name></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			



翻訳を確認するには、翻訳されるサイトでアセットのテキスト文字列を確認します。





次の付録を使用できます。

- チュートリアル: Knockout でのコンポーネントの開発
- サイト SDK リファレンス
- トラブルシューティング



# 26

# チュートリアル: Knockout でのコンポーネン トの開発

このチュートリアルでは、JavaScript オブジェクト・セットの作業の概要を示します。これ は、標準の Knockout ViewModel およびテンプレート機能を活用して、Oracle Content Management コンポーネント・カタログに格納されるコンポーネントを作成します。

- Knockout によるコンポーネント開発の概要および前提条件
- ステップ 1: コンポーネントの作成
- ステップ 2: ローカル・コンポーネントのレンダリング構造のレビュー
- ステップ 3: ローカル・コンポーネント設定の構造のレビュー
- ステップ 4: コンポーネントでの新規プロパティの表示
- ステップ 5: トリガーの登録
- ステップ 6: トリガーの呼出し
- ステップ 7: アクションの登録
- ステップ 8: アクションの実行
- ステップ 9: コンポーネントのインスタンスごとの個別タイトルの作成
- ステップ 10: インライン編集によるネストしたコンポーネントの使用
- ステップ 11: 別のレイアウトのサポート
- ステップ 12: カスタム・スタイルの定義
- ステップ 13: インライン・フレームでのコンポーネントのレンダリング
- ステップ14:コンポーネントがインライン・フレームでレンダリングされる場合のカスタム・スタイルの使用
- ステップ 15: ページの「元に戻す」および「繰返し」動作との統合
- ステップ 16: アセットの管理
- チュートリアル・レビュー

# Knockout によるコンポーネント開発の概要および前 提条件

このチュートリアルでは、JavaScript オブジェクトを使用してサンプル・コンポーネントを 作成するためのステップおよび検証手順を示します。これは標準の Knockout JS ViewModel テンプレート機能を活用します。



これらのステップ(コンポーネントを作成したときにシードされるファイルで提供済) で参照されるコードを取得して.html テンプレートおよび JavaScript viewModel のみ を自分のコードで更新できる必要があります。

/ ノート:

Oracle Content Management ではコンポーネントの作成に使用する JavaScript テクノロジは指定していませんが、通常、どの JavaScript フレー ムワークを選択しても、ファクトリ JavaScript 関数はコンポーネントの実装 ごとに同じです。

#### 前提条件

このチュートリアルでは、コンポーネントの実装のみに重点を置いています。コンポ ーネントの一般的な情報の詳細は、「コンポーネントの開発」を参照してください。

このチュートリアルのステップを完了するには、次の要件を満たす必要があります。

- サイトおよびコンポーネントを作成する権限を持つ Oracle Content Management インスタンスにアクセスできる必要があります。
- Oracle Content Management インスタンス・サーバーは、Oracle Content Management デスクトップまたはカスタム・コンポーネントを使用するローカル・ コンピュータと同期されています。Developer Cloud Service によるカスタム・コ ンポーネントの開発を参照してください。

さらに、次の JavaScript の概念およびフレームワークに精通している必要があります。

- JavaScript ブラウザ・デバッグ
- JavaScript クロージャ
- JavaScript 非同期モジュール定義 (AMD)の開発
- RequireJS および KnockoutJS フレームワーク

「ステップ1:コンポーネントの作成」に進みます。

# ステップ 1: コンポーネントの作成

このステップでは、Oracle Content Management でカスタム・コンポーネントを作成 する方法について説明します。

カスタム・コンポーネントを作成すると、これを登録して Oracle Content Management で使用可能にする必要があります。Oracle Content Management にコンポーネントを 認識させるには、サイト・ビルダーの「コンポーネント」ページを使用してコンポー ネントを登録します。

登録するコンポーネントには次の2つのタイプがあります。

- ローカル・コンポーネント:
  - これは、ファイルが Oracle Content Management インスタンス・サーバーに 格納されているコンポーネントです。
  - 主な長所は、ファイルがユーザーのサイトに配置されているため、ドメイン間 またはプロトコル間の問題を気にしなくてもよい点です。



- 短所は、Oracle Content Management サーバーでミドル層ロジックを実行できない点です。そのため、CORS をサポートするリモート・サーバーに対して REST API を使用する必要があります。
- このタイプのコンポーネントをページに直接組み込むか、インライン・フレームを使用することを選択してページにコンポーネントをレンダリングします。
- ・ リモート・コンポーネント:
  - コンポーネントのファイルはリモート・サーバーに格納され、URLのみをコンポーネントのレンダリングおよび設定パネルに登録します。
  - リモート・コンポーネントには、コンポーネントのコンテンツの作成時に実行する必要があるサーバー側ロジックがある場合にメリットがあります。
  - 短所は、ドメイン間の問題およびセキュリティの問題がこれらの URL へのアクセス
     時に解決されるようにする必要があるという点です。
  - リモート・コンポーネントは常にインライン・フレームを使用して、ページにレンダ リングされます。

#### ローカル・コンポーネントを作成および登録するには:

**1.** Oracle Content Management ホーム・ページで、「開発者」をクリックします。

「開発者」ページが表示されます。

- 2. 「すべてのコンポーネントの表示」をクリックします。
- 3. メニューから、「ローカル・コンポーネントの作成」を選択します。
- 4. A_Local_Component など、コンポーネントの名前を入力します。
- 5. 説明を入力します(オプション)。
- 6. 「作成」をクリックします。

これを実行すると、コンポーネントのリストに A_Local_Component という名前のコンポ ーネントが表示されます。

#### ステップ1の結果の確認

これでコンポーネントが正常に作成されました。作成したサイトの「コンポーネント」パレットでコンポーネントが表示されます。次のステップを使用して、コンポーネントの作成を 確認します:

- 1. localComponentTest という名前のサイトを作成します。
- 2. サイトを選択し、「開く」をクリックします。
- 3. 「編集」をクリックします。
- 4. サイトの更新を作成し、名前およびオプションで説明を指定します。
- 5. サイトでページを選択します。
- 6. サイド・パレットで をクリックして、「カスタム」を選択してカスタム・コンポーネン トのリストを表示します。
- 7. 「カスタム」コンポーネント・リストから A_Local_Component を選択し、ページにこれを ドラッグ・アンド・ドロップします。 これで 作成したローカル・コンポーネントのデフォルトのレンダリングが表示されま

これで、作成したローカル・コンポーネントのデフォルトのレンダリングが表示されます。



- **8.** ページにドロップしたコンポーネントのバナーで **三**を選択します。
- 9. 「設定」を選択します。
- 10. 位置合せを変更し、コンポーネントのスタイルを設定します。
- 11.「設定」パネルを閉じます。

次のステップでは、カスタム・コンポーネントを構築する方法および自分専用にこれ を変更する方法について説明します。「ステップ 2: ローカル・コンポーネントの構造の レビュー」に進みます。

# ステップ 2: ローカル・コンポーネントのレンダリ ング構造のレビュー

このステップでは、ローカル・コンポーネント用に作成されたデフォルト・ファイルの構造をレビューします。

単純な Hello World 例の場合、4 つの JavaScript オブジェクトおよびコード行の数は 多すぎるように見えますが、これはさらに複雑なコンポーネントを構築し、Oracle Cloud Sites Service ページのライフサイクルを持つインタラクションを処理するため の基本を提供するためのものです。

ローカル・コンポーネントの構造を確認するには:

- Oracle Content Management ホーム・ページで、「開発者」をクリックします。
   「開発者」ページが表示されます。
- 2. 「すべてのコンポーネントの表示」をクリックします。
- 3. メニューから、「ローカル・コンポーネントの作成」を選択します。
- 4. A_Local_Component など、コンポーネントの名前を入力します。
- 5. 説明を入力します(オプション)。
- 6. 「作成」をクリックします。

これを実行すると、コンポーネントのリストに A_Local_Component という名前の コンポーネントが表示されます。

Oracle Content Management デスクトップ同期クライアントを使用して、コンポーネントを検索し、それをファイル・システムと同期させます。

デスクトップ・クライアントがない場合は、すべてのコンポーネントを表示し、 Oracle Content Management インタフェースの「コンポーネント」ページでコンポ ーネントを選択し、ドリルダウンしてファイルを表示できます。

2. コンポーネントの下にあるファイルをリストする場合は、次のようなファイルが表示されます。

assets render.js settings.html appinfo.json _folder_icon.jpg

3. /assets ディレクトリにある render.js ファイルを開きます。



render.js ファイルの主要なポイントは次のとおりです。

- JavaScript AMD モジュールとして構造化されるため、ページで必須にできます。
- これには、Oracle Content Management ページの一部としてすでにロードされている KnockoutJS および JQuery への参照も含まれています。

render.js ファイルの構造を考えます。

render.jsファイルのコンテンツには、2つの JavaScript オブジェクトがあり、これは必要 な Oracle Content Management コンポーネントの API (sampleComponentFactory および SampleComponentImpl)を実装しています。これらのオブジェクトは、コンポーネントに基づ いた KnockoutJS を作成するための実装例です。これらのオブジェクトの実装は、使用してい るテクノロジに基づいて変化します。

- sampleComponentFactory
  - このオブジェクトは、render.js AMD モジュールによって返されます。
  - これは非常に単純なファクトリ・オブジェクトであり、1つの createComponent()インタフェースを実装しています。
  - さらに複雑な実装では、viewMode パラメータに基づいてコンポーネントの別の実装 を返すために渡される args 値を使用できます。これにより、サイト・ビルダーと比 べると、かなり軽量に実行時のコンポーネントを実装できるようになります。
- SampleComponentImpl
  - このオブジェクト内の主な関数は render 関数で、ページのコンポーネントをレンダ リングする際に使用されます。

Knockout コンポーネントをページにレンダリングするには、render 関数は動的にテ ンプレートをページに追加してから、viewModel バインドをテンプレートに適用しま す。

残りの実装では、viewModel パラメータとテンプレートの初期化、およびページとコンポーネント間のメッセージ送信を処理します。

render.jsファイル内の最後の2つのオブジェクト(sampleComponentTemplate および SampleComponentViewModel)は、コンポーネントに対してカスタムの実装を提供します。次の 実装は、要件によって異なります。

- sampleComponentTemplate
  - このオブジェクトでは KnockoutJS テンプレートの作成を提供しています。これは、 コンポーネントですべてのデータが初期化されてから表示を試みるまで待機します。
- SampleComponentViewModel
  - viewModel はコンポーネントのかわりに Oracle Content Management によって格納 された情報を取得し、そのデータに基づいて適切にコンポーネントをレイアウトする 方法を選択します
  - 一般的な Knockout オブザーバブルはテンプレートで使用され、コンポーネントのかわりに格納されたメタデータへのアクセスを処理します。

```
self.imageWidth = ko.observable('200px');
self.alignImage = ko.observable();
self.layout = ko.observable();
self.showTopLayout = ko.observable();
self.showStoryLayout = ko.observable();
```

- トリガーとアクションの統合:

**トリガー**: ページの他のコンポーネントのアクションにバインド可能なコンポ ーネントから Oracle Content Management トリガーを呼び出す関数。

```
self.imageClicked = function (data, event) {
   self.raiseTrigger("imageClicked"); // matches appinfo.json
};
```

**アクション**: コンポーネントが指定されたペイロードでアクションを実行する ように指示されたときにコールバックを処理する関数。

```
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);
            }
        });
    }
};
```

**コールバック**は必要に応じて登録済アクションを実行します。

SitesSDK.subscribe(SitesSDK.MESSAGE_TYPES.EXECUTE_ACTION, \$.proxy(self.executeActionsListener, self));

- コンポーネント・ライフサイクルへのサブスクリプション:
  - * コンポーネントの初期化: すべてのデータがフェッチされるまでコンポー ネントはレンダリングされません。これは Knockout オブザーバブルによ って処理されます。

self.componentLayoutInitialized = ko.observable(false); self.customSettingsDataInitialized = ko.observable(false);

必要なプロパティの初期値を取得します。これはコールバックによって処 理され、データを取得します。

SitesSDK.getProperty('componentLayout', self.updateComponentLayout); SitesSDK.getProperty('customSettingsData', self.updateCustomSettingsData);


* メタデータの更新: ユーザーが「設定」パネルを起動してデータを更新するとき など、コンポーネントのかわりに格納されたコンポーネント・メタデータが変更 されるたびに、コールバックします。

SitesSDK.subscribe(SitesSDK.MESSAGE_TYPES.SETTINGS_UPDATED, \$.proxy(self.updateSettings, self));

### N-F:

Oracle Content Management サーバーでは、常に.html ファイルに mime タイプを 設定するため、.html ファイルをアップロードしたり、これをロードするのに必要 な"text!"プラグインを使用することはできません。したがって、テンプレートの場 合、別の拡張機能を使用して"text!"プラグインによりこれをロードするか、シード 済データに表示されているように JavaScript に直接インラインでロードする必要 があります。

#### ステップ2の結果の確認

カスタム・コンポーネントのレンダリング構造がどのように作成されているかについて概要 を示します。動作を確認するには:

 render.js ファイルの sampleComponentTemplate オブジェクトを更新して、次の行を変 更します。次のコードを変更します:

'<!-- ko if: initialized -->'+

かわりに次のコードを使用します:

```
'<!-- ko if: initialized -->'+
'<div data-bind="text:\'image width is: \' + imageWidth()"></div>' +
```

- コンポーネントを Oracle Content Management インスタンス・サーバーに同期またはア ップロードします。
- 3. サイト内のページを編集し、A_Local_Component カスタム・コンポーネントをページにド ロップします。
  - この時点で、コンポーネントに image width is: 260px が表示されます。
- 4. 「設定」パネルを開き、「カスタム設定」ボタンをクリックします。
- 5. 「イメージの幅」フィールドを 300px に変更します。
- 6. この時点で、コンポーネントは次の2つのようになります。
  - a. デフォルト・イメージはサイズが 260px から 300px に拡張されます。
  - **b.** 追加したテキストが image width is 300px に更新されます。

「ステップ3:ローカル・コンポーネント設定の構造のレビュー」に進みます。



# ステップ 3: ローカル・コンポーネント設定の構造 のレビュー

このステップでは、ローカル・コンポーネント用に指定された設定の構造をレビュー します。

/assets ディレクトリの render.js ファイルと同様に、同じディレクトリに事前作 成済の settings.html ファイルがあります。settings.html ファイルは、コンポ ーネントのカスタム設定データをレンダリングします。デフォルトの実装では、カス タム設定データにプロパティ imageWidth が1つあります。

ローカル・コンポーネントの構造を確認するには:

Oracle Content Management デスクトップ同期クライアントを使用して、コンポーネントを検索し、それをファイル・システムと同期させます。

デスクトップ同期クライアントがない場合は、Oracle Content Management Web インタフェースの「**コンポーネント」**タブでコンポーネントを選択し、ドリルダウ ンしてファイルを表示できます。

2. コンポーネントの下にあるファイルをリストする場合は、次のようなファイルが表示されます。

assets render.js settings.html appinfo.json _folder_icon.jpg

/assets ディレクトリにある settings.html ファイルを開き、コンテンツをレビュ ーします。render.js ファイルと異なり、settings.html ファイルはサイト・ビル ダーの「設定」パネルのインライン・フレームを使用します。そのため、インライン・ フレームで正しくレンダリングするために、サポートしているファイルへのアクセス 権も必要になります。サイト・ビルダーは、サイトを管理するために必要です。これ により、JavaScript コードのエラーをサイト・ビルダーから分離できるようになりま す。そのため、settings.html ファイルはインライン・フレームを使用します。

次に、settings.html ファイルの主な領域を示します。

「設定」パネルをレンダリングする Knockout テンプレート。



 「設定」パネルがレンダリングされた後にインライン・フレームの高さを調整するための カスタム・バインド・ハンドラ。

ko.bindingHandlers.scsCompComponentImpl

• Knockout テンプレートに適用する Knockout ViewModel。

SettingsViewModel

次に、SettingsViewModelの主な要素を示します。

- コンポーネント・ライフサイクルへのサブスクリプション:。
- コンポーネントの初期化:
  - すべてのデータがフェッチされるまでコンポーネントがレンダリングされないよう にします。これは Knockout オブザーバブルによって処理されます。

self.initialized = ko.observable(false);

- 準備が完了するまでデータの更新を試行しないようにします。

self.saveData = false;

必要なプロパティの初期値を取得します。これはコールバックによって処理され、データを取得します。

```
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;
});
```

プロパティの変更をカスタム設定データに保存します。

```
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);
```

取得する必要がある別のプロパティを追加するには、いくつかのステップが必要です。

- 1. ユーザー・インタフェースを更新して新しい値を表示します。
- 2. コンポーネントに対して格納された現在値に値を初期化します。
- 3. 値の変更内容をコンポーネントに保存します。



別のプロパティをカスタム・コンポーネントに追加するには、これらの変更内容を settings.html ファイルに加えます。

1. 別のオブザーバブルを追加して新規プロパティを処理します。次のコードを変更 します:

self.width = ko.observable();

かわりに次のコードを使用します:

```
self.width = ko.observable();
self.imageBannerText = ko.observable();
```

2. 「設定」パネルが最初に表示されたときに、新規プロパティの現在値を取得します。 次のコードを変更します:

self.width(data.width);

かわりに次のコードを使用します:

self.width(data.width); self.imageBannerText(data.imageBannerText);

3. この新規プロパティの変更内容を保存します。次のコードを変更します:

'width': isNaN(self.width()) ? self.width() : self.width() +
'px'

かわりに次のコードを使用します:

'width': isNaN(self.width()) ? self.width() : self.width() + 'px', 'imageBannerText': self.imageBannerText()

 ユーザー・インタフェースを追加して新しいフィールドを表示します。次のコード を変更します:

<lpre><label id="widthLabel" for="width" class="settings-heading" databind="text: 'Image Width'"></label> <input id="width" data-bind="value: width" placeholder="example: 200px or 33%" class="settings-text-box">

かわりに次のコードを使用します:

<label id="widthLabel" for="width" class="settings-heading" databind="text: 'Image Width'"></label> <input id="width" data-bind="value: width" placeholder="example: 200px or 33%" class="settings-text-box">

<lpre><label id="imageBannerTextLabel" for="imageBannerText"
class="settings-heading" data-bind="text: 'Image Banner'"></label>
<input id="imageBannerText" data-bind="value: imageBannerText"</pre>



placeholder="Text to display above an image" class="settings-text-box">

5. settings.html ファイルを同期またはアップロードします。

これを実行する場合、フィールドが表示されます。ただし、「設定」パネルは自動的に変更さ れません。パネルのサイズを増やしたため、components.json 登録エントリを新しいサイ ズに更新する必要もあります。

1. コンポーネントの assets/ディレクトリと同じレベルである appinfo.json ファイル をダウンロードして、「設定」パネルのサイズを更新します。次のコードを変更します:

"settingsHeight": 90,

かわりに次のコードを使用します:

"settingsHeight": 160,

2. appinfo.json ファイルを同期またはアップロードします。

#### ステップ3の結果の確認

「設定」パネルに追加した新規プロパティを表示および入力できる必要があります。

- サイト・ビルダーでコンポーネントへの変更内容を反映できるように、サイトでページを リフレッシュします。
- 2. ページを編集モードで取り込みます。
- 3. コンポーネントをページにドラッグ・アンド・ドロップします。
- 4. 「設定」パネルをコンポーネントに対して開きます。
- 5. 「カスタム設定」ボタンをクリックします。

settings.html ファイルにあるそれぞれのプロパティに対して表示される2つのフィールドを確認します。

「ステップ4:コンポーネントでの新規プロパティの表示」に進みます。

# ステップ 4: コンポーネントでの新規プロパティの表示

この項の終わりで、「設定」パネルで新規プロパティの値を入力し、カスタム・コンポーネントの変更内容を表示して新しい値を反映できるようになります。プロパティへの更新はページにも自動的に保存されます。

render.jsファイルで、コンポーネントで2つのJavaScriptオブジェクトを更新する必要があります。

- SampleComponentViewModel
- sampleComponentTemplate



render.js を編集して SampleComponentViewModel コンポーネントを更新し、新規プロパティを含めます。次のプロパティを変更します:

self.showStoryLayout = ko.observable();

かわりに次を使用します:

self.showStoryLayout = ko.observable(); self.imageBannerText = ko.observable();

SampleComponentViewModelを取得して値の変更を取得します。次のプロパティを変更します:

self.imageWidth(customData && customData.width);

かわりに次を使用します:

self.imageWidth(customData && customData.width); self.imageBannerText(customData && customData.imageBannerText);

sampleComponentTemplateを変更して、新規プロパティを表示します。次のプロパティを変更します:

'<div data-bind="text: \'image width is: \' + imageWidth()"></div>' +

かわりに次を使用します:

'<div data-bind="text: imageBannerText"></div>' +

コンポーネントを Oracle Content Management サーバーに同期またはアップロードします。

新規プロパティを表示するためにコンポーネントはすでに変更されています。ページ のインライン・フレームに組み込まれている「設定」パネルと異なり、コンポーネン トはページに直接挿入されるため、サイズが大きくなると、それに対して使用可能な 領域は自動的に大きくなります。

#### ステップ4の結果の確認

表示された新規プロパティを確認するには:

- 1. サイト・ビルダーでコンポーネントへの変更内容を反映できるように、サイトでペ ージをリフレッシュします。
- 2. ページを編集モードで取り込みます。
- 3. コンポーネントをページにドラッグ・アンド・ドロップします。
- 4. 「設定」パネルをコンポーネントに対して開きます。
- 5. 「カスタム設定」ボタンをクリックします。
- 6. Image Banner を Workspace に変更します。



ページにおける Worksapce へのコンポーネントの更新がイメージの上に表示されます。

「ステップ5:トリガーの登録」に進みます。

# ステップ 5: トリガーの登録

このステップで、Oracle Content Management トリガーを登録する方法をレビューします。 コンポーネントの「設定」パネルにある「リンク」タブの下の「トリガー・アクション」オ プションを使用して、これを選択できます。

トリガーは、Oracle Content Management のコンポーネント間の通信の一部です。任意のコ ンポーネントが任意の数のトリガーを呼び出すことができます。コンポーネントは、トリガ ーにペイロードを提供できます。このペイロードは、トリガーが呼び出されたときに実行さ れる任意のアクションに渡されます。各トリガーに対して実行されるアクションを選択でき ます。最後に、連携して動作するよう構築されたコンポーネントは、ユーザーによるコンポ ーネント間のインタラクションの定義が不要で、他のコンポーネントに対してアクションを 実行するためにトリガーを自動的に呼び出すことができます。

追加したコンポーネントの場合、トリガーは、コンポーネントの登録データの一部として登録されます。トリガーを追加するには、コンポーネントによってサポートされている各トリガーを使用して"triggers"プロパティ配列を更新します。また、トリガーがサポートしているペイロードを指定し、アクションによってサポートされるプロパティにユーザーがペイロード内の値をマップできるようユーザー・インタフェースを作成できるようにする必要があります。

appinfo.json ファイルを開き、"triggers":[],エントリをレビューします。

```
"triggers": [{
    "triggerName": "imageClicked",
    "triggerDescription": "Image clicked",
    "triggerPayload": [{
        "name": "payloadData",
        "displayName": "Trigger Payload Data"
    }]
}],
```

このエントリに、次が表示されます。

- triggerName の値は一意の値"imageClicked"で、通常、カスタム・コンポーネント ID で ネームスペースが指定されます。
- triggerDescription の値は"Image clicked"で、これはトリガーを表示するためにユーザ ー・インタフェース・ダイアログで使用されます。
- トリガーの場合、triggerPayloadの単一の値は"payloadData"です。ユーザーはこのペイ ロードのエントリを選択して、これらをアクションのフィールドにマップします。

#### ステップ5の結果の確認

コンポーネントの「設定」パネルにある**「リンク」**タブに移動するときに、トリガーを確認 して選択できます。

- 1. サイト・ビルダーでコンポーネントへの変更内容を反映できるように、サイトでページを リフレッシュします。
- 2. ページを編集モードで取り込みます。



- 3. コンポーネントをページにドラッグ・アンド・ドロップします。
- 4. 「設定」パネルをコンポーネントに対して開きます。
- 5. 「設定」パネルの最上部で「リンク」タブを選択します。
- 6. 「リンク・タイプ」として、「トリガー・アクション」をクリックします。
- 7. 登録したクリックされるイメージ・トリガーをクリックします。
- ダイアログで、「ページ・アクション」セクション内から「アラートの表示」アクションをドラッグします。(ページ・アクションは Oracle Content Management で 提供されている組込みアクションです。)
- 「メッセージ」フィールドで、トリガー・ペイロード・データ値を選択します。これは、トリガーを登録したときに表示されたペイロード内のエントリ名です。これは希望する名前に変更できます。

これで、トリガーを登録し、そのトリガーを組込みアクションにマップして値を渡せ るようになりました。次のステップで、トリガーを呼び出してアクションを実行する 方法をレビューします。

「ステップ 6: トリガーの呼出し」に進みます。

# ステップ 6: トリガーの呼出し

このステップでは、登録済と表示されたトリガーを呼び出す方法を示します。

トリガーは、コンポーネントが任意の時点で呼び出すことができます。通常、これは、 ボタンのクリックや表内の行の選択などのユーザー操作によって呼び出されます。た だし、REST コールが原因でデータが変更された場合など、コンポーネントが任意の 基準に基づいてトリガーを呼び出すことができます。

このサンプルの場合、イメージをクリックすると、whoAreYou プロパティの現在値を渡 すトリガーが呼び出されます。

render.js ファイルをレビューして、SampleComponentViewModel オブジェクトを確認します。

トリガーを呼び出すには:

 サイト SDK をコールしてトリガーを呼び出す SampleComponentViewModel オブジ ェクトの関数をレビューします。

```
self.raiseTrigger = function (triggerName) {
  SitesSDK.publish(SitesSDK.MESSAGE_TYPES.TRIGGER_ACTIONS, {
    'triggerName': triggerName,
    'triggerPayload': {
        'payloadData': 'some data here'
    }
  });
};
```



 トリガーを呼び出す関数をコールする際に、ユーザー・インタフェースに必要なものがあります。render.jsファイルをレビューして、次のエントリを含めるように sampleComponentTemplateオブジェクトを更新します。

'<div data-bind="attr: {style: imageStyle, \'data-layout\':
alignImage()}, click: imageClicked">' +

SampleComponentViewModel オブジェクトで、イメージがクリックされたときにコールされる JavaScript 関数が表示されます。この関数は、サイト SDK を呼び出し、トリガー "imageClicked"(ステップ2のclick バインドから渡される値です)に定義されているすべて のアクションをトリガーするよう命じます。これはまた、単一のフィールド payloadData を 持つ triggerPayload を渡し、静的な値'some data here'も渡します。値 imageClicked およ び whoAreYou は、トリガーが登録されている(前述のステップ)appinfo.json ファイルの値 と一致します。

サンプル・コードでは、トリガーは click バインドのデータバインドによって呼び出され、 トリガー名 imageClicked を渡します。ユーザーが選択したレイアウトに基づいて、現在、 <scs-image>コンポーネントの3つのレンダリングがあります。レイアウトごとにトリガー が呼び出されるようにするには、render.js ファイルを編集して次の変更を行います。

別のレイアウトからトリガーを呼び出します。次のコードの2つのエントリを見つけます。

```
'<div data-bind="attr: {style: imageStyle, \'data-layout\':
alignImage()}">' +
```

```
コードを次のように変更します。
```

```
'<div data-bind="attr: {style: imageStyle, \'data-layout\':
alignImage()}, click: imageClicked">' +
```

• トリガーに渡すペイロードを指定します。次のコードを変更します:

```
self.raiseTrigger = function (triggerName) {
   SitesSDK.publish(SitesSDK.MESSAGE_TYPES.TRIGGER_ACTIONS, {
        'triggerName': triggerName,
        'triggerPayload': {
            'payloadData': 'some data here'
        }
    });
   };

かわりに次のコードを使用します:
   self.raiseTrigger = function (triggerName) {
```

```
SitesSDK.publish(SitesSDK.MESSAGE_TYPES.TRIGGER_ACTIONS, {
    'triggerName': triggerName,
    'triggerPayload': {
        'payloadData': self.imageBannerText() // pass banner text as
payload
     }
```



}); };

 render.js ファイルを Oracle Content Management インスタンス・サーバーに同 期またはアップロードします。

これで必要なコードをレビューしました。ボタンがクリックされるとカスタム・コン ポーネントがこれを呼び出すように、トリガーを結び付けることができます。

#### ステップ6の結果の確認

トリガーに対して実行するアクションを登録して、トリガーが呼び出されたときにア クションが実行されるようにする必要があります。

- サイト・ビルダーでコンポーネントへの変更内容を反映できるように、サイトでペ ージをリフレッシュします。
- 2. ページを編集モードで取り込みます。
- **3.** コンポーネントをページにドラッグ・アンド・ドロップします。
- 4. 「設定」パネルをコンポーネントに対して開きます。
- 5. 「設定」パネルの最上部で「リンク」タブを選択します。
- 6. 「リンク・タイプ」として、「トリガー・アクション」を選択します。
- 7. 登録済と表示された imageClicked トリガーをクリックします。
- ダイアログで、「ページ・アクション」セクションから「アラートの表示」アクションをドラッグします。
- 9. 「メッセージ」フィールドで、payloadData 値を選択します。これは、トリガーを 登録したときに入力したペイロードです。
- 10.「設定」パネルを閉じて、サイト・ビルダーをプレビュー・モードに切り替えます。
- 11. コンポーネントでイメージをクリックします。

imageBannerText 値を指定していないため、no message defined を示すアラート が表示されます。

- 12. ページを編集モードにして、コンポーネントの「設定」パネルを再度開きます。
- 13.「カスタム設定」をクリックして Workplace.と入力します。
- 14.「設定」パネルを閉じて、ページをプレビュー・モードに切り替えます。
- 15. コンポーネントでイメージをクリックします。
  - これで、更新されたペイロード Workplace が表示されます。これは、click バインドに加えた変更から呼び出されます。
- トリガーが呼び出された場合、任意の数のアクションを実行できます。

### / ノート:

アクションが実行されるタイミングに関して事前定義済の順序はありません。各アクションはリストされている順序でコールされますが、現在のアクションが完了するまで待機しないと次のアクションがコールされないわけで はありません。アクションによって非同期コールが行われる場合、次のアクションが実行される前に完了しない可能性があります。



```
「ステップ7:アクションの登録」に進みます。
```

# ステップ 7: アクションの登録

トリガーが呼び出されると、コンポーネントで Oracle Content Management アクションがコールされます。

コンポーネントは、任意の数のアクションを登録できるほか、アクションがサポートしてい るペイロードを定義することもできます。ユーザーは、アクションを選択するときに、アク ションに渡すペイロードを移入できます。

トリガーを登録する場合と同じように、コンポーネントがサポートしているアクションを appinfo.json 登録データに登録できます。コンポーネント内のサンプル・アクションの登 録をレビューするには、appinfo.json ファイルを開いて"actions"コードを見つけます。

```
"actions": [{
   "actionName": "setImageWidth",
   "actionDescription": "Update the image width",
   "actionPayload": [{
        "name": "imageWidth",
        "description": "Image Width in pixels",
        "type": {
            "ojComponent": {
                "component": "ojInputText"
            }
        },
        rvalue": ""
    }]
}]
```

登録されたこのアクションは、コンポーネントの「設定」パネル上の「**リンク」**タブでトリ ガーをクリックするときに呼び出される「アクション」ダイアログに表示されます。

#### ステップ7の結果の確認

- サイト・ビルダーでコンポーネントへの変更内容を反映できるように、サイトでページを リフレッシュします。
- 2. ページを編集モードで取り込みます。
- 3. コンポーネントをページにドラッグ・アンド・ドロップします。
- 4. ボタン・コンポーネントをページにドロップします。
- 5. 「設定」パネルをボタン・コンポーネントに対して開きます。
- 6. 「一般」タブで、ボタンのラベルを「ここをクリック」に変更します
- 7. 「設定」パネルで「リンク」タブを選択します。
- 8. 「リンク・タイプ」として、「トリガー・アクション」を選択します。
- 9. ボタン・コンポーネントに対して「ボタンをクリック」トリガーをクリックします。
- **10.** ダイアログの左側のパレットで、A_Local_Component コンポーネントを展開します。
- **11. 画像の幅の更新**アクションを A_Local_Component コンポーネントからページにドラッ グ・アンド・ドロップします。



```
12. 300px を「画像の幅(ピクセル)」フィールドに入力します。
```

アクションの登録方法およびユーザー・インタフェース上でアクションの表示方法が 表示されました。次のステップでは、コールされたときのコンポーネント内でのアク ションの処理方法について学習します。

「ステップ8:アクションの実行」に進みます。

# ステップ 8: アクションの実行

このトピックの最後で、コンポーネント内のアクションを実行するコンポーネントを ページにドロップできるようになります。ここでは、前述のステップで作成したアク ション登録を活用します。

コンポーネントがアクションを実行するには、EXECUTE_ACTION メッセージをリスニン グする必要があります。このメッセージには、アクションに渡されるペイロードが含 まれ、ここから想定される値が抽出されます。

EXECUTE_ACTION メッセージをリスニングするには、render.js ファイルを編集し、 次のエントリを使用して SampleComponentViewModel オブジェクトを更新します:

SitesSDK.subscribe('EXECUTE_ACTION', \$.proxy(self.executeActionsListener
, self));

EXECUTE_ACTION メッセージを受信すると、関連するコールバック関数が実行されます。

```
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);
            }
        });
    }
}
```

これにより、アクションを実行するための JavaScript 関数が作成され、EXECUTE_ACTION メッセージが呼び出されるたびにサイト SDK を使用して関数がコールされます。

このコンポーネントは、EXECUTE_ACTION メッセージが呼び出されるたびにコールされ、処理対象として設計されているアクションのみを処理するかどうかはコンポーネント次第です。これを行うには、アクションの名前を確認し、コンポーネントが処理可能なものであることを確認する必要があります。

アクションのペイロードは値の配列です。通常、関心のあるペイロード値を配列から 探し出す必要があります。



### 🖍 ノート:

アクション・リスナーはコールバックであるため、JavaScript クロージャを使用するか、関数を適切にバインドし、関数の実行時に viewModel にアクセスできることを確認する必要があります。

#### ステップ8の結果の確認

- サイト・ビルダーでコンポーネントへの変更内容を反映できるように、サイトでページを リフレッシュします。
- 2. ページを編集モードで取り込みます。
- 3. コンポーネントをページにドラッグ・アンド・ドロップします。
- 4. ボタン・コンポーネントをページにドラッグ・アンド・ドロップします。
- 5. 「設定」パネルをボタン・コンポーネントに対して開きます。
- 6. 「一般」タブで、ボタンのラベルを「ここをクリック」に変更します
- 7. 「設定」パネルの最上部で「リンク」タブを選択します。
- 8. 「リンク・タイプ」として、「トリガー・アクション」を選択します。
- 9. ボタン・コンポーネントに対して「ボタンをクリック」トリガーをクリックします。
- **10.** ダイアログの左側で、A_Local_Component コンポーネントを展開します。
- **11. 画像の幅の更新**アクションを A_Local_Component コンポーネントから右側にドラッグ・アンド・ドロップします。
- 12. 300px を「画像の幅(ピクセル)」フィールドに入力します。
- 13. ページをプレビュー・モードに切り替えます。
- 14.「ここをクリック」ボタンをクリックします。

ここで、画像のサイズが 300px に増加します。

### 💉 ノート:

トリガーおよびアクションは、コンポーネント間の通信をサポートするように設計 されています。これらは状態を作成または管理するように設計されていません。ペ ージをリフレッシュすると、トリガーが呼び出されていないか、アクションが実行 されていないかのように、ページが元の状態に戻ります。

「ステップ9:コンポーネントのインスタンスごとの個別タイトルの作成」に進みます。

# ステップ 9: コンポーネントのインスタンスごとの個 別タイトルの作成

このステップでは、コンポーネントの各インスタンスに個別のタイトルを作成する方法について説明します。



ページにコンポーネントをドロップすると、A_Local_Component というコンポーネントのバナーが表示されます。ユーザーがコンポーネントのいずれかのみをページにドロップする場合は問題ありませんが、ユーザーがコンポーネントの各インスタンスを区別できるように、個別のタイトルの作成が必要になる場合があります。

サイト SDK を使用すると、コンポーネントのタイトルを更新できます。このステップ では、"imageBannerText"プロパティに基づいて更新します。

タイトルを更新するには、render.jsファイルを編集し、このコードを SampleComponentViewModel オブジェクトに追加します。

```
self.updateDescription = ko.computed(function () {
   SitesSDK.setProperty('description', self.imageBannerText());
});
```

この Knockout の計算では、imageBannerText オブザーバブルが変更されるたびに、コンポーネントの説明が変更されます。

#### ステップ9の結果の確認

- サイト・ビルダーでコンポーネントへの変更内容を反映できるように、サイトでペ ージをリフレッシュします。
- 2. ページを編集モードで取り込みます。
- 3. ページにコンポーネントをドロップします。
- 4. 「設定」パネルをコンポーネントに対して開きます。
- 5. 「カスタム設定」ボタンをクリックします。
- 6. Image Banner を Workplace に変更します。
- 7. 「設定」パネルを閉じて、カーソルをコンポーネントの上に置いてバナーを表示し ます。

A_Local_Component Workplace が表示されます。

「ステップ10:インライン編集によるネストしたコンポーネントの使用」に進みます。

# ステップ 10: インライン編集によるネストしたコ ンポーネントの使用

Oracle Content Management コンポーネントは、KnockoutJS コンポーネント・アーキ テクチャを使用して実装されています。これは、KnockoutJS を使用してコンポーネン トを実装している場合、Oracle Content Management 組込みコンポーネントを直接テ ンプレートに含めることができるということです。

### 🎤 ノート:

Oracle Content Management 組込みコンポーネントは Oracle Content Management ページでのみ実行できるため、コンポーネントがインライン・ フレームでレンダリングされる場合、ネストされたコンポーネントは使用で きません。



ネストされたコンポーネントを活用するには:

- 1. KnockoutJS を使用してコンポーネントを実装します。
- 2. RequireJS を使用してコンポーネントを含め、Oracle Content Management によって作成 される、同じ Knockout "ko"インスタンス変数を使用します。

**Oracle Content Management** が Knockout をコンポーネントで拡張して、KnockoutJS の 独自のインスタンスを使用していると、これらのコンポーネントが使用できなくなるた め、これは必要です。

このステップで、Oracle Content Management のイメージ、段落およびタイトル・コンポー ネントをカスタム・コンポーネントでレンダリングする方法をレビューします。ユーザーは、 これをページ内で直接編集したり、ネストしたコンポーネントの「設定」パネルにアクセス したりできるようになります。

これらのコンポーネントをテンプレートに含める方法を確認するには、render.jsファイル を編集して、sampleComponentTemplate オブジェクトを表示します。レンダリングされるデ フォルト・セクションがここに表示されます。

```
'<!-- ko if: alignImage() !== \'right\' -->' +
'<div style="display:flex;">' +
'<div data-bind="attr: {style: imageStyle, \'data-layout\': alignImage()},</pre>
click: imageClicked">' +
'<scs-image params="{ scsComponent: { \'renderMode\': mode, \'parentId\':</pre>
id, \'id\': \'imageId\', \'data\': imageData } }"></scs-image>' +
'</div>' +
'<div data-bind="attr: {style: paragraphStyle}">' +
'<scs-title params="{ scsComponent: { \'renderMode\': mode, \'parentId\':</pre>
id, \'id\': \'titleId\', \'data\': titleData } }"></scs-title>' +
'<scs-paragraph params="{ scsComponent: { \'renderMode\': mode,</pre>
\'parentId\': id, \'id\': \'paragraphId\', \'data\': paragraphData } }"></</pre>
scs-paragraph>' +
'</div>' +
'</div>' +
'<!-- /ko -->' +
```

ネストした<scs-image>コンポーネントを表示すると、次のエントリが表示されます。

'<scs-image params="{ scsComponent: { \'renderMode\': mode, \'parentId\': id, \'id\': \'imageId\', \'data\': imageData }}"></scs-image>' +

params テンプレート・バインドに渡される scsComponent データには、次のものが含まれて います。

- renderMode: これは、サイト・ビルダーのモードを参照します。これを使用すると、機能 を有効および無効にできます。たとえば、<scs-title>コンポーネントによって使用され ている場合、edit モードで実行しているときに、リッチ・テキスト・エディタが追加さ れます。
- parentId: Oracle Content Management コンポーネントはこれがネストしたコンポーネ ントとしてレンダリングされることを認識しているため、これは必須です。ネストしたコ ンポーネントへのすべての変更は、カスタム・コンポーネントのデータに保存されます。
- id: ネストしたコンポーネントの一意の ID。これは、さらにカスタム・コンポーネントの ID によってネームスペースが 指定されます。



 data: ネストしたコンポーネントの初期データ。コンポーネントが変更されない 場合、この初期データでレンダリングされます。

参照された id および mode 値は SampleComponentViewModel オブジェクトのカスタ ム・コンポーネントに渡されるので、これらの値を取得するようにオブジェクトを変 更する必要はありません。

// Store the args
self.mode = args.viewMode;
self.id = args.id;

他のすべてのサポートされているネストしたコンポーネントのコンポーネントは <scs-paragraph>と同じパターンです。たとえば、<scs-image>, <scs-title>, <scsbutton>です。

#### ステップ 10 の結果の確認

- 1. サイト・ビルダーでコンポーネントへの変更内容を反映できるように、サイトでペ ージをリフレッシュします。
- 2. ページを編集モードで取り込みます。
- 3. コンポーネントをページにドラッグ・アンド・ドロップします。
- **4.** コンポーネント内の As a page author, you can edit. . .テキストをクリック し、リッチ・テキスト・エディタを使用して説明を更新します。
- 5. プレビュー・モードに切り替えて、更新内容を確認します。
- 6. 編集モードに戻します。
- 7. 「設定」パネルをコンポーネントに対して開きます。
- 8. ネストしたコンポーネントが見つかると表示される「**コンポーネント」**リンクをク リックします。
- 9. 「段落」をクリックします。これは見つかったネストしたコンポーネントです。

コンポーネント内の段落コンポーネントに対してプロパティを更新できます。

### 💉 ノート:

コンポーネントがインスタンス化されるまで、Oracle Content Management ではテンプレート内に存在する可能性のあるネストしたコンポーネントを認 識しません。Oracle Content Management に非表示のネストしたコンポーネ ントを認識させるには、 SitesSDK.setProperty('visibleNestedComponents', []); API を使用しま す。非表示のネストしたコンポーネントをデフォルトで表示させるには、コ ンポーネント登録で"nestedComponents": []配列を更新する必要がありま

「ステップ11:別のレイアウトのサポート」に進みます。

ORACLE

す。

### ステップ 11: 別のレイアウトのサポート

このステップで、レイアウトをレビューし、これを使用すると、ユーザーはコンポーネント の表示方法を変更できます。

カスタム・コンポーネントは、ユーザーが選択可能ないくつかのレイアウトをサポートでき ます。これらのレイアウトのそれぞれで、カスタム・コンポーネントの表示方法を変更しま す。レイアウトは登録データの別の拡張機能です。

サンプル・コードでサポートされている 3 つのレイアウトをレビューするには、 appinfo.json ファイルの"componentLayouts"エントリをレビューします

```
"componentLayouts": [
    {
        "name": "default",
        "displayName": "IMAGE_LEFT_LAYOUT"
    },
    {
        "name": "right",
        "displayName": "IMAGE_RIGHT_LAYOUT"
    },
    {
        "name": "top",
        "displayName": "IMAGE_TOP_LAYOUT"
    }
],
```

「設定」パネルをカスタム・コンポーネントに対して開くと、レイアウトを切り替えるオプションが表示されます。コンポーネントを有効にして選択内容の変更に対応するために、 render.js ファイルには、現在選択されている値を取得し、この値の変更内容をリスニングするコードがあります。

render.js ファイルを編集して、SampleComponentViewModel オブジェクトを確認します。

• テンプレートで参照される layout オブザーバブルがあります。

self.layout = ko.observable();

• この値が変更されるたびに処理する更新関数があります。

```
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);
```



• 初期化コードでは、レイアウトの元の値を取得し、更新関数をコールします。

SitesSDK.getProperty('componentLayout', self.updateComponentLayout);

```
プロパティの変更リスナーはこのプロパティへの変更を確認し、更新関数をコール
します。
```

```
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));

最終的に、sampleComponentTemplate テンプレート・オブジェクトには、この値の変更を反映するためのコードが含まれます。

'<!-- ko if: alignImage() === \'right\' -->' +

また、これらの変更により、「設定」パネルでレイアウトを選択して、コンポーネント を更新できます。

#### ステップ 11 の結果の確認

- 1. サイト・ビルダーでコンポーネントへの変更内容を反映できるように、サイトでペ ージをリフレッシュします。
- 2. ページを編集モードで取り込みます。
- 3. コンポーネントをページにドラッグ・アンド・ドロップします。
- 4. 「設定」パネルをコンポーネントに対して開きます。
- 5. 「レイアウト」プロパティから「**イメージを右」**を選択します。 この時点で、コンポーネントが更新され、"<scs-image>"コンポーネントが表示さ れます。

「ステップ12: カスタム・スタイルの定義」に進みます。

### ステップ 12: カスタム・スタイルの定義

作成したコンポーネントは、サイトに使用されるテーマの design.json および design.css ファイル内の他のコンポーネントと同様に処理されます。

カスタム・コンポーネントの独自のスタイルを追加するには、コンポーネントを登録 したときに使用された id 値を確認します。appinfo.json ファイルでは、これは "id": "hello-world"でした。



この値を使用して、テーマの design.json ファイルを編集して、この id に対してサポート する新しいスタイルを追加します。たとえば、テーマ内の/designs/default/ design.json ファイルを編集して、このコードを追加します。

```
"hello-world": {
   "styles": [{
     "name": "Plain",
     "class": "hello-world-default-style"
   },
   {
    "name": "Gothic",
    "class": "hello-world-gothic-style"
   }]
},
```

「設定」パネルをコンポーネントに対して開くと、「スタイル」タブにリストされる2つのオ プションとして「プレーン」(デフォルト)および Gothic が表示されます。ただし、これらの オプション間で切り替えても、design.cssファイルにリストされたスタイル・クラスを実 際に定義するまで、何も処理されません。

テーマの design.css ファイルを編集して、スタイルのカスケード・スタイル・シート(CSS) のクラスを追加します。たとえば、テーマ内の/designs/default/design.css ファイル を編集して、このコードを追加します。

```
.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; }
```

ファイルを Oracle Content Management インスタンス・サーバーに保存および同期します。

#### ステップ 12 の結果の確認

- サイト・ビルダーでコンポーネントへの変更内容を反映できるように、サイトでページを リフレッシュします。
- 2. ページを編集モードで取り込みます。
- 3. コンポーネントをページにドラッグ・アンド・ドロップします。
- 4. 「設定」パネルをコンポーネントに対して開きます。
- 5. 「スタイル」タブに移動します。
- design.jsonファイルに定義されている Gothic と「プレーン」スタイル間で切り替え ます。

選択ごとに適用された CSS クラス間で切り替えると、変更内容を反映するようにコンポーネント内のフォント・サイズが調整されます。

「ステップ13:インライン・フレームでのコンポーネントのレンダリング」に進みます。



# ステップ 13: インライン・フレームでのコンポー ネントのレンダリング

今まで、サンプルでは、ページにインラインでレンダリングされたローカル・コンポ ーネントを表示していました。インライン・フレームでコンポーネントをレンダリン グすることを選択することもできます。

たとえば、コンポーネントでページに対して権限が不要の更新が行われる場合、イン ライン・フレームでコンポーネントをレンダリングするように選択できます。これを 行うと、プロパティが変更されるたびに、ページを再作成する必要があります。さら に、リモート・コンポーネントは常にインライン・フレームでレンダリングされます。

この項のサンプルは、ローカル・コンポーネントの作成時に、「iframe にレンダリング するコンポーネントの作成」オプションを選択したときに作成されたファイルから取 得したものです。ただし、リモート・コンポーネントに同様に適用されるように、こ れらのファイル・セットを選択してリモート・サーバー上にホストできます。

#### インライン・フレームのコンポーネントと非インライン・フレームのコンポーネント の類似点

#### 「設定」パネル

「設定」パネルは常にインライン・フレームのページに配置されるため、コンポーネン トがインライン・フレームを使用するかどうかに関係なく、「設定」パネルのコードは 変更されません。両方のユースケースに同じ「設定」パネルのコードを作成します。

#### サイト SDK API

SDK API は両方のユースケースで同じです。同じコードを使用して、トリガーを呼び 出し、アクションをリスニングし、プロパティ値を取得および設定します。特定のプ ロパティは両方のケースに使用できず(たとえば、インライン・フレームを使用しない コンポーネントの"height"プロパティは設定できません)、API は同じままです。した がって、これらのタイプのコンポーネントの両方のコードをコピーでき、このチュー トリアルで説明したサンプル・コードは両方のケースで動作します。

#### インライン・フレームのコンポーネントと非インライン・フレームのコンポーネント の相違点

#### ファイル構造および依存関係

ローカル・コンポーネントの作成時に「iframe にレンダリングするコンポーネントの 作成」を選択すると、次のファイルが作成され、表示されます。

```
<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

これらのファイルが作成され、ページのインライン・フレームでコンポーネントを即座に実 行できるようになります。この構造と標準ローカル・コンポーネントの構造の主な相違点は 次のとおりです。

- JavaScript の依存関係:
  - これらのファイルの完全なコピーが取得され、コンポーネントが実行されます。これらのファイルは、サンプルのインライン・フレームのコンポーネントを実行するために必要です。要件に基づいてこのディレクトリのコンテンツを追加および削除できます。
  - コンポーネントが公開されるときに、コンポーネントの assets ディレクトリにある すべてのものがパブリック・サイトにプッシュされるため、js ディレクトリのすべ てのものをサイト・ビルダーでも実行時でも使用できます。
  - ノート: これらのファイルは、それぞれの使用のために作成されます。インライン・ フレームのコンポーネントごとに別々のバージョンのファイルを作成するのではな く、テーマまたは他のパブリックな場所でこれらのファイルの統合を検討する必要が あります。
- render.html:
  - AMD モジュールである標準コンポーネントの render.js ファイルと対照的に、これは完全な HTML コンポーネントです。

#### コンポーネントの"高さ"の管理

インライン・フレームの使用時に起きる問題の1つは、インライン・フレーム自体の高さの 管理です。この問題が発生した場合、必要かどうかに関係なく、ページのコンポーネントに スクロール・バーが表示されます。

インライン・フレームの高さを管理するために、コンポーネントはページにインライン・フレームにどのくらいの高さが必要かを指示する必要があります。リモート・コンポーネントでドメイン間の問題を処理する場合、コンポーネントがページ上にレンダリングされた後で、サイト SDK メッセージングを使用して必要な高さにインライン・フレームを設定するように要求する必要があります。これは、SitesSDK.setProperty('height', {value}) APIを使用して行われます。(「Oracle Content and Experience SDK」を参照してください。)

たとえば、setHeight 関数およびカスタム・バインド・ハンドラを作成して、コンポーネント がページでレンダリングされるときにこれをコールします。

高さを更新する関数:

```
// 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');
};
```

 コンポーネントがページでレンダリングされるかプロパティが変更されるたびに setHeight をコールする Knockout カスタム・バインド・ハンドラ:

```
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();
    };
};
```

バインド・ハンドラをコールするテンプレート更新:

<div data-bind="sampleAppSetAppHeight: true"></div>

#### トリガーおよびアクションの登録

インライン・フレームにないコンポーネントのトリガー/アクションの登録は appinfo.jsonファイルに配置されていますが、インライン・フレームのコンポーネ ントの場合、コンポーネント自体がこの情報を提供します。これは、2つの API を使 用して行われます:

```
SitesSDK.subscribe('GET_ACTIONS', self.getAppActions);
SitesSDK.subscribe('GET_TRIGGERS', self.getAppTriggers);
```

次に、これらの API の使用例を示します。

```
// 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;
};
```

#### テーマ・スタイルへのアクセス

コンポーネントはインライン・フレームでレンダリングされるため、テーマで使用可能なス タイルにアクセスできません。サイト SDK は、これらのスタイルをインライン・フレーム内 の要素に適用できるように、これらを取得するための API を提供しています。

このトピックは、「ステップ **14**: コンポーネントがインライン・フレームでレンダリングされ る場合のカスタム・スタイルの使用」で詳細に説明されています。

#### HTTPS プロトコルと HTTP プロトコルの混在

Oracle Content Management では HTTPS プロトコルを使用しているため、ページ内で参照さ れているすべてのリソースは HTTPS も使用する必要があります。リソースにはベース.html ファイルが含まれていて、参照されているすべてのファイルとともにインライン・フレーム でレンダリングされます。

このリソースの要件は、通常、リモート・コンポーネントに適用され、この制約を認識して いる必要があります。インライン・フレームを使用しているローカル・コンポーネントのリ ソースは Oracle Content Management サーバーによって提供されているため、これらのコン ポーネントはすでに一致するプロトコルを使用しています。

「ステップ14: コンポーネントがインライン・フレームでレンダリングされる場合のカスタム・スタイルの使用」に進みます。

# ステップ 14: コンポーネントがインライン・フレーム でレンダリングされる場合のカスタム・スタイルの 使用

インライン・フレームでレンダリングされたコンポーネントは、design.cssファイルに直 接アクセスできません。かわりに、コンポーネントの design.cssの URL を取得し、これ をページに追加する追加のステップがあります。ユーザーが選択したスタイルを反映するよ うに、コンポーネントを更新する必要があります。

design.css ファイルをコンポーネントに含めて使用するには、render.html ファイルに 変更が必要になります。

- 1. design.css ファイルの URL を特定して含めます
- 2. 選択したスタイル・クラスの値が変更されるたびに、これを取得します
- 3. 選択した styleClass を反映するように、テンプレートを更新します



第 26 章 ステップ 14: コンポーネントがインライン・フレームでレンダリングされる場合のカスタム・スタイルの 使用

- 4. コンポーネントで選択したスタイル・クラスへの変更を反映します
- 5. スタイルが変更されるとインライン・フレームのサイズが変更されることを確認し ます

次に、render.html ファイルを編集するための詳細な手順を示します。

1. design.css ファイルの URL を特定して含めます。

動的に design.css ファイルをページの<head>セクションに追加します。これを ロードした後、スタイルを適用することによりインライン・フレームの高さが変更 されるため、インライン・フレームの高さを設定します。

次のコードを viewModel オブジェクトに追加します。

```
// 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++) {</pre>
                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) {</pre>
                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 + '")</</pre>
style>');
    $style.appendTo('head');
    // kickoff the polling
    pollFunction();
```

```
オブザーバブルを作成して、styleClass プロパティの値が変更されたタイミングを追跡
します。
```

self.selectedStyleClass = ko.observable();

```
スタイル・クラスが設定されるまでレンダリングできないことに注意してください。次の
コードを変更します:
```

```
self.customSettingsDataInitialized = ko.observable(false);
self.initialized = ko.computed(function () {
    return self.customSettingsDataInitialized();
}, self);
```

```
かわりに次のコードを使用します:
```

```
self.customSettingsDataInitialized = ko.observable(false);
self.styleClassInitialized = ko.observable(false);
self.initialized = ko.computed(function () {
    return self.customSettingsDataInitialized() &&
self.styleClassInitialized();
}, self);
```

```
次を追加して、選択したスタイル・クラスの初期値を取得します。
```

```
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. styleClass を反映するように、テンプレートを更新します。次のコードを変更します:



かわりに次のコードを使用します:

selectedStyleClass">

 コンポーネントで選択したスタイル・クラスへの変更を反映します。次のコードを 変更します:

```
if (settings.property === 'customSettingsData') {
    self.updateCustomSettingsData(settings.value);
}
```

かわりに次のコードを使用します:

```
if (settings.property === 'customSettingsData') {
    self.updateCustomSettingsData(settings.value);
}
if (settings.property === 'styleClass') {
    self.updateStyleClass(settings.value);
}
```

5. スタイルが変更されるとインライン・フレームのサイズが変更されることを確認し ます。次のコードを変更します:

```
\ensuremath{{\prime\prime}}\xspace 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();
```

かわりに次のコードを使用します:

 $\ensuremath{{\prime\prime}}\xspace$  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. ファイルを Oracle Content Management インスタンス・サーバーに保存および同 期します。

#### ステップ 14 の結果の確認

- サイト・ビルダーでコンポーネントへの変更内容を反映できるように、サイトでペ ージをリフレッシュします。
- 2. ページを編集モードで取り込みます。
- 3. コンポーネントをページにドラッグ・アンド・ドロップします。
- 4. 「設定」パネルをコンポーネントに対して開きます。



- 5. 「スタイル」タブに移動します。
- 6. design.jsonファイルに定義されている Gothic と「プレーン」スタイル間で切り替え ます。

選択ごとに適用された CSS クラス間で切り替えると、変更内容を反映するようにコンポーネント内のフォント・サイズが調整されます。

「ステップ15:ページの「元に戻す」および「繰返し」動作との統合」に進みます。

# ステップ 15: ページの「元に戻す」および「繰返し」 動作との統合

Oracle Content Management ではカスタム・コンポーネントのかわりにプロパティを格納す るため、これらのプロパティへの変更は、自動的にページの「元に戻す」および「繰返し」 動作の一部になります。

ユーザーが「元に戻す」または「繰返し」をクリックしたときに起きること明確にするため に、これらの「元に戻すイベント」は、ユーザーがページに対して何かを実際に実行したと きにのみ発生する必要があります。たとえば、カスタム・コンポーネントの「設定」パネル を開いても、ユーザーが実際にプロパティに変更を加えるまで、ページ内のプロパティは更 新されません。「設定」パネルでプロパティを初期化しただけでは、更新イベントは起動しま せん。

注意しないでこの動作を実行すると、予期しない動作が発生する可能性があります。ページ は実行されたままになりますが、ユーザー・エクスペリエンスが低下します。たとえば、次 の動作が発生する可能性があります。

- 「設定」パネルを開いただけで、「保存」ボタンがアクティブになります。
- 効果が表示されるまで、ユーザーが「**元に戻す」**を複数回クリックする必要があります。
- コンポーネントで予期しない変更が書き込まれ、「繰返し」スタックが新しい値で更新され るため、「繰返し」スタックが削除されます。

「設定」パネルのチュートリアルで提供されたサンプル・コードは、初期化時ではなく、実際 に saveData をコールする準備ができた場合にのみ書き込むようにする方法の例を示してい ます。通常、あまり注意する必要はありませんが、ユーザー操作が含まれていない場合、 customSettingsData を更新しないようにコンポーネント内で同様に注意する必要がありま す。

「ステップ16:アセットの管理」に進みます。

### ステップ 16: アセットの管理

このステップでは、コンポーネントで使用されるアセットを管理する方法について説明しま す。

アセットには、アセットのライフサイクルを管理するために、Oracle Content Management が認識する必要があるコンポーネントおよびカスタム・コンポーネントが含まれています。

#### Oracle Content Management コンテンツ・フォルダ

Oracle Content Management で作成した各サイトには専用の content フォルダがあります。 これは非表示のフォルダで、通常は表示されません。サイトが公開されると、content フォル ダ内のすべてのファイルもファイル・システムに公開されます。



たとえば、イメージ・コンポーネントを使用してイメージを選択した場合、Oracle Content Management では選択したイメージのコピーを作成し、これをコンテンツ・フ ォルダに配置します。元のイメージを削除した場合にサイトが破損しないように、 URL は常にこのコピーしたバージョンのイメージを指します。これは、Oracle Content Management で提供されている他のコンポーネント(ギャラリ、ギャラリ・グ リッド、ドキュメント、ソーシャル・バー、ファイル・ダウンロード、およびスロッ トと componentGroups の背景イメージ)にも適用されます。

このアセットのライフサイクルに関与するカスタム・コンポーネントの場合、カスタム・コンポーネントは、サービスがかわりに管理するアセットを Oracle Content Management に認識させる必要があります。これにはアセットのコピーの作成が含まれているため、管理方法を認識するように、カスタム・コンポーネントは Oracle Content Management API を使用してアセットを選択する必要もあります。

#### URL の管理

アセットの URL は基準の数によって変化します。

- コンポーネントのランタイム URL はコンポーネントのサイト・ビルダーの URL と 異なります
- ページをコピーすると、Oracle Content Management ではコンテンツ・フォルダ内の参照されたすべてのアセットのコピーを作成するため、2 つのコンポーネントが コンテンツ・フォルダ内の同じアセットを指すことはありません
- componentGroup をページにドロップすると、componentGroup のコンポーネント で参照されているアセットの新しいコピーが作成されます

さらに、相対 URL がローカル・コンポーネントには問題ない可能性がありますが、完 全修飾 URL でインライン・フレーム・コンテンツをレンダリングできるように、リモ ート・コンポーネントでは、Oracle Content Management でかわりに管理するアセッ トの完全修飾 URL が必要になります。

静的なままの URL に依存できないため、アセットをレンダリングする場合、コード内 でアセットに対して ID への参照のみを維持し、アセットの URL を取得する必要があ ります。

#### アセットの管理

これらのサイト SDK API はアセットの管理に使用できます。

SitesSDK.getProperty('componentAssets', callback);

- これにより、現在のアセットの配列が取得されます
- 各アセットのエントリは次から構成されます。
  - id: アセットの一意の ID。
  - title: Oracle Content Management のタイトル・メタデータ。
  - **description**: Oracle Content Management の説明メタデータ。
  - fileName: 選択したファイルの元の名前。カスタム・コンポーネントの「設定」 パネルで表示する際に便利です。これにより、ユーザーは自分が選択したファ イルを認識します。これは、コンテンツ・フォルダにコピーしたファイルの名 前ではありません。



- source: アセットの URL を有効にしたマクロ。この値は時間が経つと変化するので、コンポーネントで参照しないでください。アセットの一部として保存する必要があります。
- **url**: getPropert()がコールされたコンテキストに基づいたアセットの完全修飾 URL。

SitesSDK.setProperty('componentAssets', [assets]);

- これをコールして、Oracle Content Management がかわりに管理するすべてのアセットを 保存します。
- これをコールしない場合、アセットは保存されません。
- この配列にないアセットは、サイトが公開されると削除されます。
- assets パラメータは、getProperty によって返されたものおよび filePicker で返された ものと同じ形式のアセットの配列です。

✓ ノート:

url 値は格納されません。アセットを要求したときに、この値は動的に作成されます。

SitesSDK.filePicker(options, callback);

- アセットのリストを選択するためのファイル・ピッカーを表示する API。
- 選択したアセットの配列を渡してアセットを正常に選択すると、コールバックがコールされます。
- この時点では何も保存されません。setProperty('componentAssets', [assets]);をコ ールして、保存する他のすべてのアセットと組み合せた選択内容のアイテムを保存するか どうかは、コンポーネント次第です。

#### アセットの選択例

この項では、アセットを選択し、その ID を格納し、格納されたアセットから実際の値を再度 フェッチする方法を示します。

- 1. settings.html ファイルを編集します。
- 2. Image selection を含めるようにテンプレート・オブジェクトを変更します。

```
<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="imageAsset" type="button" class="selectbutton" data-
bind="click: showFilePicker">Select Image</button</pre>
</div>
```

3. オブザーバブルを追加するように viewModel を変更して、選択したアセットの ID を格納 します。

```
self.imageID = ko.observable();
```



4. ファイル・ピッカーを表示して選択したアセットの名前を表示することで、アセットの選択内容を管理するように viewModel を変更します。

```
11
   // handle component assets
   11
   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++) {</pre>
           if (self.assets[i].id === imageID) {
               imageName = self.assets[i].fileName;
               break;
           }
       }
       return imageName
   }, self);
5. customSettingsData を取得する前に先にアセットを取得するように、viewModel
   を更新します。self.ImageID()オブザーバブルが変更されたときに、このコード
   により、self.imageName も起動されます。
   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. 最後に、imageIDを保存するように save 関数を更新し、参照したアセットのリストで
   componentAssets を更新します。
   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);
```

#### アセットの選択の結果の確認

- サイト・ビルダーでコンポーネントへの変更内容を反映できるように、サイトでページを リフレッシュします。
- 2. ページを編集モードで取り込みます。
- 3. コンポーネントをページにドラッグ・アンド・ドロップします。
- 4. 「設定」パネルを開きます。
- 5. 「イメージを選択」ボタンをクリックします。
- 参照(またはアップロード)してイメージを選択します。
   選択したイメージを示すイメージの名前が格納されていることに注意してください。
- 7. 「設定」パネルを閉じます。
- 8. 「設定」パネルを再度開きます。

イメージの名前が再度反映されていることに注意してください。

#### アセットのレンダリング例

この項では、アセットを取得してこれらをコンポーネントにレンダリングし、「設定」パネル で値が変更されるときは常にコンポーネントを動的に更新する方法を示します。

/ ノート:

これはページのインライン・フレームに存在するローカル・コンポーネントの例を 示していますが、同様のコードがページ内でインラインでレンダリングされている コンポーネントに対して機能します。



- 1. render.html ファイルを編集します。
- 2. アセットを含めるようにテンプレートを更新します。

3. viewModel では、2 つのオブザーバブルを作成して customSetttingsData から imageID を取得し、アセットの格納済リストから取得される imageURL を格納しま す。

```
self.imageID = ko.observable();
self.imageURL = ko.observable();
```

imageID が変更されるときは常に対応するイメージのアセット URL を取得するように、viewModel を更新します。

```
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;
            }
        });
});</pre>
```

5. customSettingsData から ID を取得するように、viewModel を更新します。

#### アセットのレンダリングの結果の確認

- サイト・ビルダーでコンポーネントへの変更内容を反映できるように、サイトでペ ージをリフレッシュします。
- 2. ページを編集モードで取り込みます。
- 3. コンポーネントをページにドラッグ・アンド・ドロップします。
- 4. 「設定」パネルを開きます。
- 5. 「イメージを選択」ボタンをクリックします。
- 参照(またはアップロード)してイメージを選択します。
   選択したイメージを示すイメージの名前が格納されていることに注意してください。
- 「設定」パネルを閉じます。
   この時点で、コンポーネントでレンダリングされる選択済イメージが表示されます。

```
「チュートリアル・レビュー」に進みます。
```



### チュートリアル・レビュー

このチュートリアルでは、Knockout コンポーネント・ファクトリを使用して、カスタマイズ されたコンポーネントを作成する方法の概要を説明します。

このチュートリアルの主な目的は、このパターンを使用して、SampleComponentViewModel お よび sampleComponentTemplate JavaScript オブジェクトを更新するだけで、カスタム・コン ポーネントを作成することです。sampleComponentFactory および SampleComponentImpl オ ブジェクトは、チュートリアルで確認したとおりに変更されていません。ページとのやり取 りを行う必要がない、これらの変更を実装し、次のタスクを実行できました。

- 変更内容を「設定」パネルからコンポーネントに伝達し、これらの変更内容を維持しま す。
- トリガーおよびアクションを実行し、ページの他のコンポーネントを操作します。
- レイアウトを作成し、ネストされたコンポーネントを活用します。
- コンポーネント固有のスタイルを定義します。

この例ではカスタム・コンポーネントを複数のファイルに分割していますが、これはチュートリアルをわかりやすくするためです。最適化のために、ファイルを適切にパッケージ化することを検討して複数のダウンロードを回避する必要があります。

最後に、このチュートリアルはコンポーネントに基づいた Knockout に適切ですが、AngularJS などの別の JavaScript テクノロジを使用してカスタム・コンポーネントを作成する場合、 SampleComponentImpl オブジェクトを再実装して、実際のコンポーネントのテクノロジ固有 の実装とともに、フレームワークとの対応する通信を作成する必要があります。この作業は、 このチュートリアルの範囲外です。



# 27 サイト SDK リファレンス

Oracle Content Management JavaScript SDK (サイト SDK)には、コンポーネントが Oracle Content Management と統合されたエクスペリエンスを持つことを実現する一連の機能があります。

サイト SDK は、Oracle Content Management サーバーからダウンロードできます:

http://{server}/_sitesclouddelivery/renderer/app/sdk/js/sites.min.js

サイトのアプリケーション・プログラミング・インタフェース(API)定義は、グローバル・オ ブジェクト SitesSDK のすべての定義済ネームスペースに対応しています:

- 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

サイト SDK の主要目的は、ユーザーがアプリケーションを構築および管理できるようにする ことです。次の複数の機能が用意されています:

- インライン・フレームおよびドメインの境界をまたぐやり取り
  - コンポーネントがホストされている場所に関係なく、コンポーネントがページとやり 取りできるようにします
  - 次のページ・モデル・ライフ・サイクルへの参加が含まれます:
    - * コンポーネントの作成
    - * 格納されたプロパティによるコンポーネントのレンダリング
    - * コンポーネントの更新およびそれらの変更内容の保存
    - * ページからのコンポーネントの削除
  - ページ内で使用するために、コンポーネントの構成に使用されるカスタム設定を格納 および取得します
  - コンポーネント間のやり取りを使用してトリガーとアクションに関与します



- ページへのジオメトリ・ヒント(カスタム・コンポーネントには当てはまらない)
  - コンポーネントによって指示されたときにページでインライン・フレームを動 的にサイズ変更できるようにします
  - コンポーネントがページの一部として表示されるようになり、コンポーネントのサイズ変更イベントに応じてページが移動されるようになります(スクロール・バーなし)
- Oracle Content Management によっていつ呼び出されたかに関するコンポーネントの認識
  - コンポーネントとページとの間のやり取りには、秘密キーによって署名された
     トークンが含まれています。
  - サイト SDK により、コンポーネントで、トークンをデコードすることや、呼 出し元の確認を容易にすることができます。

スクリプトの既知のパスを提供することにより、サイト SDK の関数を含めます。例:

```
<script type="text/javascript"
src="<sdk_install_dir_path_prefix>/sites.min.js">
</script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script>
```

サイト SDK には、次のグローバル・オブジェクトがあります:

- SitesSDK
- SitesSDK.Utils

#### SitesSDK

SitesSDK グローバル・オブジェクトは、レンダリングおよび設定のエンドポイントで 使用できます。

コマンド	タスク
SitesSDK.getProperty(propertyName, callback)	ホスト・サイトから、指定したプロパ ティの値を取得します。
SitesSDK.setProperty(propertyName, propertyValue)	指定したプロパティの値を <i>property</i> Value に設定します。
SitesSDK.getSiteProperty(propertyName, callback)	指定されたサイト・プロパティ値をホ スト・サイトから取得します。たとえ ば、ホスト・サイトによって使用され ている現在のテーマ・デザインをフェ ッチできます。
SitesSDK.publish(messageType, payload)	サーバーにメッセージを送信し、ペイ ロードとして messageType および JSON オブジェクトを受け入れます。
SitesSDK.subscribe(messageType, callback)	ホスト・サイトからディスパッチされ るメッセージのメッセージ・リスナー です。このコールは非同期です。こ れらは JavaScript コールバックであ るため、関数で JavaScript クロージ ャを使用するか、関数を適切にバイン ドし、適切なコンテキストにアクセス できることを確認する必要がありま す。



#### SitesSDK.Utils

SitesSDK.Utils ネームスペースには、Oracle Content Management コンポーネントのすべて のエンドポイントによってコールできるユーティリティ 関数があります。このネームスペー スは、リモート・コンポーネントの「設定」パネルで使用できますが、インライン・ローカ ル・コンポーネントでは使用できません。

### 🔺 注意:

これらの関数は、オラクル社によって公式にサポートされているわけではなく、サ ンプル実装を目的としています。自己責任において使用してください。

コマンド	タスク
SitesSDK.Utils.Logger	サイト SDK で使用されているロガー・ オブジェクトを返します。ログ・レベル は、次のいずれかに設定できます: debug log info error warn
SitesSDK.Utils.addSiteThemeDesign(cssUrl)	コンポーネントに現在のサイト・テー マ・デザインを追加します。
Utils.getStyle(elem,styleProp)	インライン・フレームの高さを計算する ユーティリティ ・メソッドです。

### SitesSDK.getProperty('componentAssets', callback)

この関数は、サイト内のコンポーネントのかわりに現在格納されているアセットのリストを 返します。

パラメータ

名前	タイプ	説明
componentAssets	文字列	リスト作成者を呼び出します。
callback	関数	コールバック関数

#### 使用方法

```
// 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
11
// returns a list of selected assets. If user cancels out of the
picker,
// the callback is never called
SitesSDK.filePicker({options}, callback);
11
// example 'componentAssets' JSON returned:
11
'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)

この関数は、ホスト・サイトから、指定したプロパティの値を取得します。

パラメータ

名前	タイプ	説明
propertyName	文字列	プロパティの名前。
callback	関数	コールバック関数です。

#### 関連プロパティ

プロパティ名	説明	設定	レンダリン グ
height	フレームの高さ	該当なし	はい
width	フレームの幅	該当なし	はい
customSettingsData	設定データ	はい	はい
styleClass	選択したスタイル・クラス	はい	はい
borderColor、borderRadius、 alignment、borderStyle、 margin などのその他のスタイ			

### 使用方法

ル属性



### レンダリング URL からフェッチされる追加の関連プロパティ

getProperty()の使用によるプロパティ値のフェッチに加えて、レンダリング URL からフェ ッチできるその他の関連プロパティを次に示します。

プロパティ名	説明
id	コンポーネント ID
instance	暗号化されたトークン
width	iframe の幅
height	<b>iframe</b> の高さ
viewMode	レンダリング・モードの値
locale	サイトまたはブラウザで設定されている言語
settingsId	現在の設定 ID

### 設定 URL からフェッチされる追加プロパティ

getProperty()の使用によるプロパティ値のフェッチに加えて、設定 URL からフェッチでき るその他のプロパティを次に示します。

プロパティ名	説明
currCompId	編集対象のコンポーネントの ID
instance	暗号化されたトークン
width	設定の iframe の幅
locale	サイトまたはブラウザで設定されている言語
OrigSettingsId	コンポーネントの元の設定 ID
settingsId	新しい設定 <b>ID</b>

### SitesSDK.getSiteProperty(propertyName, callback)

この関数は、指定されたサイト・プロパティ値をホスト・サイトから取得します。例では、ホスト・サイトによって使用されている現在のテーマ・デザインをフェッチしています。

名前	タイプ	説明
propertyName	文字列	サイト・プロパティの名前



名前	タイプ	説明
callback	関数	プロパティ 値を読み取るためのコール バック関数

### サイト・プロパティ

プロパティ名	説明	設定	レンダリン グ
theme	現在のテーマ・デザインの URL	該当なし	はい

### 使用方法

```
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])

この関数は、カスタム・コンポーネントのかわりに格納されるサイト・アセットのリ ストを更新します。

名前	タイプ	説明
componentAssets	文字列	コンポーネント・アセットのリストを 作成します。
assets	JSON オブジェク ト	選択したアセットのリストを返しま す。
引数		
プロパティ名	説明	
filetypes	サポート ト。	されているファイル・タイプのリス
multiSelect	単一ファ す。	イルまたは複数ファイルを選択しま



```
// 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
11
// args:
// fileTypes - list of supported file types
// multiSelect - single or multiple file select
11
// returns a list of selected assets. If user cancels out of the picker,
// the callback is never called
SitesSDK.filePicker({options}, callback);
11
// example 'componentAssets' JSON returned:
11
'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)

この関数は、指定したプロパティの値を propertyValue に設定します。

パラメータ

名前	タイプ	説明
propertyName	文字列	プロパティの名前。
propertyValue	JSON オブジェク ト	プロパティの値。

#### 関連プロパティ

この表は、setProperty()を使用してフェッチできる一連の関連プロパティを示しています。

プロパティ名	説明	設定	レンダリング
height	フレームの高さ	該当なし	はい



プロパティ名	説明	設定	レンダリング
width	フレームの幅	該当なし	はい
customSettingsData	設定データ	はい	はい
styleClass	選択したスタイル・クラス	はい	はい
description	ページ上のローカル・コンポーネン トのインスタンスのバナーで使用 されます		
layout	ローカル・コンポーネントのみに対 して現在選択されているレイアウ トを返します。appinfo.json ファイルの値に基づいて「設定」パ ネルから更新できます。		
renderStatus	レンダリングが終了したことをカ スタム・コンポーネントがレポート できるようにします。	true	
borderColor, borderRadius,			

alignment、borderStyle、 margin などのその他のスタイル 属性

### 使用方法

## SitesSDK.filePicker({options}, callback)

この関数は、選択したファイルのリストを返します。

名前	タイプ	説明
options	文字列	返す選択済ファイルを指定するオプ ションです。
callback	関数	コールバック関数です。ユーザーが ピッカー外で取り消した場合、コール バックは呼び出されません。



```
// 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
11
// args:
// fileTypes - list of supported file types
// multiSelect - single or multiple file select
11
// returns a list of selected assets. If user cancels out of the picker,
// the callback is never called
SitesSDK.filePicker({options}, callback);
11
// example 'componentAssets' JSON returned:
11
'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)

```
この関数は、選択したファイルのリストを返します。
```

SitesSDK.openDocumentPicker()関数は引数を1つ使用し、選択したドキュメントに解決する約束を返します。使用可能なオプションは、Oracle Content Management の統合および拡張にあるドキュメント・ビューの組込み UI (documentsView Component)の API V2 の documentsView コンポーネントのオプションと一致しています。

### 使用方法

```
SitesSDK.openDocumentPicker({
   selectable: "any",
   layout: "grid"
}).then(function (selection) {
   console.log(selection);
});
```

ドキュメントが取得されたら、Oracle Content Management に、 SitesSDK.getProperty('componentAssets', callback)および SitesSDK.setProperty('componentAssets',



[assets])プロパティを使用して、これらのドキュメントの ID をサイトに格納するように 指示する必要があります。そうしないと、サイトとともに公開されず、削除される可 能性があります。

### SitesSDK.openAssetPicker(options)

この関数は、選択したファイルのリストを返します。

SitesSDK.openAssetPicker()関数は引数を1つ使用し、選択したアセットに解決する 約束を返します。使用可能なオプションは、Oracle Content Management の統合およ び拡張にあるアセット・ビューの組込み UI (assetsView コンポーネント)の API V2 の assetsView コンポーネントのオプションと一致しています。

#### 使用方法

```
SitesSDK.openAssetPicker({
   select: "single"
}).then(function (selection) {
   console.log(selection);
});
```

アセットが取得されたら、Oracle Content Management に、 SitesSDK.getProperty('componentAssets', callback)および SitesSDK.setProperty('componentAssets', [assets])プロパティを使用して、これらのアセット の ID をサイトに格納するように指示する必要があります。そうしないと、サイトとと もに公開されず、削除される可能性があります。

### SitesSDK.publish(messageType, payload)

この関数は、メッセージをサーバーに送信します。ペイロードとして messageType お よび JSON オブジェクトを受け入れます。メッセージ・タイプは、ページで認識され て処理されます。ハンドラがない場合、渡されたメッセージ・タイプは無視されます。

### パラメータ

名前	タイプ	説明
messageType	文字列	メッセージのタイプ
payload	JSON オブジェク ト	メッセージ・ペイロード

### メッセージ・タイプ

この表では、ローカル(インライン・フレームを使用)またはリモート・コンポーネント に認識され、SitesSDK.publish()関数を使用して送信できる、メッセージのタイプを 示しています。

メッセージ・タイプ	説明
SETTINGS_UPDATED	設定の更新済メッセージをレンダリング・エンドポイン トに送信するために、「設定」パネルで使用します。



メッセージ・タイプ	説明
TRIGGER_ACTIONS	ホスト・サイトでのアクションをトリガーするために、 コンポーネント・レンダリング・エンドポイントで使用 します。

### SitesSDK.subscribe(messageType, callback)

この関数は、ホスト・サイトからディスパッチされるメッセージの効果的なメッセージ・リ スナーです。このコールは非同期です。

登録されたコールバックは、特定のタイプのメッセージを SDK が受信した場合にコールされ ます。コールバックによって値が返される場合、その値はページに返されます。この関数に コールバックが渡されない場合、その特定のメッセージ・タイプに登録されているリスナー が削除されます。

これらは JavaScript コールバックであるため、関数で JavaScript クロージャを使用するか、 関数を適切にバインドし、適切なコンテキストにアクセスできることを確認する必要があり ます。

名前	タイプ	説明
messageType	文字列	×ッセージのタイプ: SETTINGS_UPDATED TRIGGER_ACTIONS EXECUTE_ACTION GET_ACTIONS GET_TRIGGERS COPY_CUSTOM_DATA PASTE_CUSTOM_DATA
callback	関数	ホスト・サイトからメッセージを受信す る際の関数。

#### パラメータ

#### メッセージ・タイプ

メッセージ・タイプ	説明
SETTINGS_UPDATED	このメッセージをサブスクライブし、「設定」パネルで値が変 更されたときに通知を受信します。通常、コンポーネントは、 ユーザー・インタフェースでレンディションを更新すること によって応答します。



メッセージ・タイプ	説明
TRIGGER_ACTIONS	このメッセージを公開すると、トリガーが呼び出され、ペイ ロードが渡されます。
EXECUTE_ACTION	このメッセージをサブスクライブし、カスタム・アクション を処理します。リスナーは通常、特定のアクションを実行す ることによってこのメッセージを処理します。
GET_ACTIONS	このメッセージをサブスクライブし、コンポーネントが実行 できるアクションの配列を宣言します。アクションの配列を 返します。
GET_TRIGGERS	このメッセージをサブスクライブし、コンポーネントが呼び 出すことができるトリガーの配列を宣言します。トリガーの 配列を返します。
COPY_CUSTOM_DATA	このメッセージをサブスクライブし、クリップボードへのカ スタム設定データのコピーを処理します。クリップボードに 置くデータを表すオブジェクトを返します。
PASTE_CUSTOM_DATA	このメッセージをサブスクライブし、クリップボードからの カスタム設定データの貼付けを処理します。

一部のリスナーはデータが渡されることを想定しており、一部のリスナーはデータを 返すことを想定しています。

### 例 1:

メッセージ・タイプ EXECUTE_ACTION のメッセージ・リスナーを登録します:

SitesSDK.subscribe(SitesSDK.MESSAGE_TYPES.EXECUTE_ACTION, self.executeAction,self);

ここに示すように、登録された callback 関数が引数を受け入れる必要があります:

```
// 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
```

}

### 例 2:

COPY_CUSTOM_DATA リスナーは通常、次のようなコードを使用して実装されます:

// listen for COPY_CUSTOM_DATA request
SitesSDK.subscribe(SitesSDK.MESSAGE_TYPES.COPY_CUSTOM_DATA,



```
メソッド copyCustomDataListener()は、クリップボードにコピーするデータを表すオブジェ
クトを返します。次に例を示します:
```

```
// Handle Copy Style (save customSettingsData to the clipboard)
self.copyCustomDataListener = function() {
    return {
        prop1: this.prop1(),
        prop2: this.prop2()
    };
};
```

### 例 3:

PASTE_CUSTOM_DATA リクエストはデータを受け入れますが、何も返す必要がありません:

```
// listen for PASTE_CUSTOM_DATA request
SitesSDK.subscribe(SitesSDK.MESSAGE_TYPES.PASTE_CUSTOM_DATA,
pasteCustomDataListener);
```

次のコードを伴います:

```
// Handle Paste Style (apply customSettingsData from the clipboard)
self.pasteCustomDataListener = function(data) {
    ...
};
```

### SitesSDK.Utils.addSiteThemeDesign(cssUrl)

この関数は、現在のページの HTML <head>タグ内に link 要素を作成します。ソースは、 cssUrl のパスに設定されます。

### 🔺 注意:

Utils ネームスペースの 関数は、Oracle で公式にはサポートされていません。これら は、サンプル実装として役立つよう意図されています。自己責任において使用して ください。

名前	タイプ	説明
cssUrl	文字列	現在のテーマ・デザインの URL パス。リモート・ コンポーネントでのみ使用します。
		ローカル・コンポーネント(iframe 内で実現され る)は、URL のかわりにページからプロパティをフ ェッチする必要があります。



この関数は通常、このサンプル・コードに示されるとおり、ホスト・サイトからの現 在のテーマ・デザインのフェッチとともに使用されます:

```
// 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

この関数は、サイト SDK で使用されているロガー・オブジェクトを返します。

▲ **注意:** Utils ネームスペースの 関数は、Oracle で公式にはサポートされていません。 これらは、サンプル実装として役立つよう意図されています。Utils 関数は 独自のリスクで使用してください。

ログ・レベルは、debug、log、info、error または warn に設定できます。



# 28 トラブルシューティング

サイトの作成について質問が生じる可能性があります。次に、その回答を示します。

- サイトを作成しようとしていますが、テンプレートがありません
- サイトを削除できません
- サイト・ツリーを開くこともページを編集することもできません
- コンポーネントを追加しましたが、ページに表示されません
- フォルダ、ファイルおよび会話コンポーネントが動作しません
- イメージの新規バージョンをアップロードしましたが、ページに表示されません
- ページ・レイアウトを変更したら、コンテンツの一部が表示されなくなりました
- 別のサービスをベースとするコンポーネントを追加しましたが、機能しません
- エンタープライズ・サイトに警告が表示される

## サイトを作成しようとしていますが、テンプレート がありません

サイトを作成する前に、テンプレートをインストールして共有する必要があります。

Oracle Content Management に用意されているテンプレートを使用するには、管理者がそれ をインストールして共有する必要があります。管理者に連絡してください。

同様に、テンプレートを作成した場合、サイトからインポートしたか、コピーしたか、作成 したかには関係なく、自分でテンプレートを明示的に共有するまで他のユーザーは使用でき ません。

Administering Oracle Content Management のサイト設定の構成を参照してください。

## サイトを削除できません

ここでは、サイトを削除できない理由を示します。

• サイトがオンラインです。

最初にサイトをオフラインにしてから削除します。サイトをオフラインにするには、サイトに対してマネージャ・ロールを持っている必要があります。

• 必要な権限がありません。

サイトを削除できるのは、サイトを作成した(サイトの所有者である)場合か、他のユーザ ーがあなたとサイトを共有していて、あなたにコントリビュータまたはマネージャ・ロー ルを付与している場合です。

詳細は、サイトおよびサイト設定の管理を参照してください。



## サイト・ツリーを開くこともページを編集するこ ともできません

サイト・ビルダーは、デフォルトではプレビュー・モードで開き、サイトの表示はで きますが編集はできません。エディタが編集モードの場合にのみサイトを編集できま す。

## コンポーネントを追加しましたが、ページに表示 されません

これにはいくつかの理由が考えられます。

- リモート・コンポーネント(アプリケーション)に関連付けられている URL が有効 であることを確認してください。リモート・コンポーネントの登録を参照してくだ さい。
- エディタでは、インライン・フレーム(iframe タグ)と呼ばれる HTML 要素でリモート・コンポーネントを囲みます。すべてのリモート・コンポーネントがインライン・フレームで囲まれるわけではありません。

リモート・コンポーネントをインライン・フレームで囲むことができるかどうかを 知るには、コンポーネント・プロバイダに確認してください。

## フォルダ、ファイルおよび会話コンポーネントが 動作しません

Oracle Content Management とやり取りするコンポーネントには特定のリソースおよび設定が必要です。

次のコンポーネントは、Oracle Content Management のリソースへのアクセスを必要 とします:

- フォルダ・リスト
- ファイル・リスト
- ドキュメント・マネージャ
- 会話

フォルダおよびファイル・コンポーネントは、REST アプリケーション・プログラミ ング・インタフェース(API)へのアクセスを必要とします。

次を確認します。

コンポーネントは、フォルダおよびファイルの REST 操作にアクセスする API エンドポイントへのアクセス権を持っている必要があります。

デフォルトでは、サイトに指定された標準の URL を使用している場合、Oracle Content Management REST API エンドポイントは使用可能です。カスタム URL



を使用する場合、/documents REST API エンドポイントへのアクセスを明示的に指定す る必要があることがあります。

サイトの URL はどのようにしてマップするのですか。を参照してください

## ページ・レイアウトを変更したら、コンテンツの一 部が表示されなくなりました

より少ないまたは異なる名前付きスロットを含むレイアウトを選択する場合、他のスロット の既存のコンテンツは新しいレイアウトに表示されません。

コンテンツは削除されませんが、選択したレイアウトに同じ名前のスロットがないかぎり、 コンテンツを表示できません。

同じスロット数のレイアウトを選択するようにしてください。選択したレイアウトに同じ数 のスロットがあるにもかかわらず、コンテンツの一部が表示されない場合、スロットの名前 が異なっている可能性があります。テーマの設計者に連絡して、相違を解決してください。

## イメージの新規バージョンをアップロードしました が、ページに表示されません

コンポーネントとともに使用するためにイメージを選択すると、一意のコピーが作成され、 サイトとともに格納されます。

ファイルをアップロードすると、サイト・リポジトリに格納されます。既存のファイルと同じ名前でファイルをアップロードすると、ファイルの新しいバージョンを作成します。

ページに表示されるイメージ・ファイルは、リポジトリから選択されたファイルのコピーで、 自動的には更新されません。これは仕様で、リポジトリでイメージ・ファイルが更新される 際に、既存ページが不注意に変更されることを防止します。

コンポーネントで使用するイメージを更新するには、コンポーネントのプロパティ・ページ からイメージを明示的に選択する必要があります。詳細は、イメージを参照してください。

## 別のサービスをベースとするコンポーネントを追加 しましたが、機能しません

サイトは、複数の異なるプロセスやサービスと統合できます。

Process Cloud Service、Oracle Intelligent Advisor (以前の Oracle Policy Automation)およびその他のサービスに基づいてコンポーネントを作成した場合、両方のサービスで同じアイデン ティティ・ドメインを使用する必要があります。統合の設定を行ったユーザーに問い合せて、 2 つのサービスが同じアイデンティティ・ドメインを使用していることを確認してください。

## エンタープライズ・サイトに警告が表示される

エンタープライズ・サイトに v1.0 のコンテンツ・レイアウトが含まれていると、そのレイア ウトではレンダリングできないため、サイト・ビルダーでサイトを表示すると警告が表示さ れます。v1.1 のコンテンツ・レイアウトに更新する必要があります。



この問題は、次のいずれかの場合に発生する可能性があります。

- 既存のサイトを翻訳可能にした場合、そのサイトは、デフォルト言語およびローカ リゼーション・ポリシーが割り当てられてエンタープライズ・サイトに自動的に変 換されます。サイト・テンプレートに v1.0 のコンテンツ・レイアウトが含まれて いると、そのコンテンツ・レイアウトではレンダリングできません。
- v1.0のコンテンツ・レイアウトを使用する古いテンプレートからエンタープライズ・サイトを作成した場合、同様に、そのコンテンツ・レイアウトではレンダリングできません。

この問題を修正するには、コンテンツ・レイアウトを v1.1 のコンテンツ・レイアウト に更新します。コンポーネント・カタログに新しいコンテンツ・レイアウトを作成し て違いを確認できます。コンテンツ・レイアウトのバージョンを登録するための行を 追加して、data.fieldname を fields.fieldname に変更する必要があります。

