

Oracle® Cloud

通过 Oracle Content Management 构建站点



F26825-21
2021 June



Oracle Cloud 通过 Oracle Content Management 构建站点,

F26825-21

版权所有 © 2018, 2021, Oracle 和/或其附属公司。

第一作者: Bruce Silver

特约作者: Jean Wilson, Sarah Bernau, Kalpana N, Ron van de Crommert

撰稿人: 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.

目录

前言

目标读者	xvi
文档可访问性	xvi
多元化与包容性	xvi
相关资源	xvi
约定	xvii

第 I 部分 简介

1 Oracle Content Management 概览

访问 Oracle Content Management	1-1
了解角色	1-1
管理资产	1-2
管理内容	1-2
对内容开展协作	1-2
构建站点	1-3
集成和扩展 Oracle Content Management	1-3
入门	1-3
迁移至 Oracle Cloud Infrastructure	1-4

2 构建站点入门

站点入门	2-1
了解站点创建流程	2-2
了解站点开发	2-3
了解站点管理	2-5

3 创建第一个网站

开始之前	3-1
步骤 1: 设置环境	3-2

导入 Minimal-NavMenu 定制组件	3-2
发布 Minimal-NavMenu 定制组件	3-3
添加示例图像	3-4
步骤 2: 设置网站	3-5
创建网站	3-5
编辑网站	3-6
步骤 3: 发布网站	3-25
执行更多操作	3-27

第 II 部分 创建和编辑站点

4 创建站点

创建站点	4-1
复制站点	4-3
管理站点请求	4-4
更改站点请求详细信息	4-5
查看站点请求策略	4-5

5 编辑站点

初步了解“站点构建器”页	5-1
了解站点更新	5-3
使用更新	5-5
编辑提示和技巧	5-5
使用样式和格式设置	5-9
定制站点设置	5-10
添加定制站点属性	5-11
使用表	5-13
上传站点文件	5-14

6 在站点中使用模板和主题

了解模板	6-1
从站点创建模板	6-3
更改模板详细信息	6-5
更改模板策略	6-5
更改模板状态或受众	6-6
管理模板	6-7
导出和导入模板	6-9

了解主题	6-11
管理主题	6-12
发布主题	6-14

7 管理定制组件和布局

了解定制组件	7-1
了解布局	7-2
使用定制组件和布局	7-4
注册远程组件	7-6
创建本地组件、布局、内容字段编辑器或内容表单	7-7
导出或导入组件或布局	7-8

8 使用站点页

导航到页	8-1
查看页	8-1
添加页	8-3
移动页	8-5
删除页	8-5
更改页设置	8-6
更改页布局	8-8
更改背景或主题	8-9

9 排列页面内容

添加组件和部分布局	9-1
使用资产和内容项	9-3
使用触发器和操作	9-3
使用水平部分布局	9-5
使用两列或三列布局	9-6
使用垂直部分布局	9-7
使用选项卡式部分布局	9-8
使用滑块部分布局	9-8

10 使用内置组件

基本组件	10-2
标题	10-2
段落	10-4
纯文本	10-5

按钮	10-6
结构组件	10-7
分隔线	10-7
分隔符	10-8
媒体组件	10-8
图像	10-9
图库	10-11
图库网格	10-13
YouTube 视频	10-16
视频	10-17
文档组件	10-18
文档	10-18
文件夹列表	10-20
文件列表	10-20
文档管理器	10-22
项目库	10-23
社交组件	10-25
社交栏	10-25
Facebook 点赞和推荐	10-26
Twitter 分享和关注	10-26
对话组件	10-27
对话列表	10-29
流程组件	10-30
流程开始表单	10-30
处理任务列表	10-32
任务详细信息表单	10-33
内容项	10-35
内容项组件	10-35
内容占位符	10-36
内容列表	10-37
内容搜索	10-41
推荐	10-42
其他组件	10-43
地图	10-44
标题	10-44
文章	10-46
图像（具有文本）	10-47
组件组	10-48
在页上使用 Cobrowse	10-49
Oracle Intelligent Advisor	10-50

第 III 部分 发布和管理站点

11 管理站点

初步了解“站点”页	11-1
管理站点和站点设置	11-2
更改站点说明、徽标或嵌入属性	11-6
设置搜索引擎属性	11-6
启用预呈现服务以优化搜索引擎	11-7
设置“静态站点传送”选项	11-9
在发布时启用自动编译	11-9
覆盖编译站点的默认高速缓存控制标头	11-10
指定移动用户-代理以支持编译的自适应布局	11-10
指定和配置虚名 URL	11-10
启用 Cobrowse 集成	11-12
添加分析跟踪	11-14

12 发布站点

使站点联机或脱机	12-1
发布站点更改	12-1

13 保护站点

了解站点安全性	13-1
更改站点安全性	13-3

14 使用多语言站点

多语言站点概览	14-1
翻译站点	14-2
管理站点翻译作业	14-4
用于翻译的区域设置	14-4
用于翻译的定制区域设置	14-5
为 URL 重定向设置区域设置别名	14-5

15 使用站点重定向或 URL 映射

计划重定向	15-1
简单的字符串到字符串匹配	15-1
简化的通配符匹配	15-2
添加站点重定向	15-2
在 JSON 文件中指定重定向规则	15-2
将重定向规则文件上载到站点	15-8
映射站点 URL	15-8

16 提高站点性能

利用高速缓存提高性能	16-1
运行时高速缓存	16-1
站点构建器高速缓存	16-3
首屏 (Above the Fold, ATF) 呈现	16-3

第 IV 部分 开发站点

17 定制设计和样式

关于设计	17-1
设计文件	17-1
自适应表设计	17-4
定制对话列表样式	17-7
定制文件夹列表和文件列表样式	17-9
定制社交栏图标	17-10
为 Oracle Intelligent Advisor 配置交互式提问样式扩展	17-11

18 了解背景用法

关于背景和主题	18-1
如何实施背景	18-1
设置存储位置	18-2

19 设置触发器和操作

关于触发器和操作	19-1
设置触发器	19-1
设置操作	19-3

20 开发模板

关于模板	20-1
基本模板结构	20-2
创建模板	20-3
导出模板	20-4
导入模板	20-5
使用入门级模板	20-6
基于 Bootstrap 或网站设计模板创建站点模板	20-9
通过开发人员云服务开发模板	20-14
登录 Oracle Content Management 的开发人员云服务控制台	20-15
在开发人员云服务中创建项目	20-15
在开发人员云服务中创建模板	20-15
在开发人员云服务中复制模板	20-16
将模板导入到开发人员云服务中	20-16
合并更改	20-17
从开发人员云服务导出模板	20-17

21 开发主题

关于主题	21-1
基本主题结构	21-2
站点导航	21-4
创建主题	21-6
隐藏主题的组件和部分布局	21-11
隐藏主题中组件的对齐方式、宽度或间距选项	21-12
将组件与主题关联	21-13
站点呈现 API	21-15

22 开发布局

关于布局	22-1
搜索引擎优化 (Search Engine Optimization, SEO)	22-3
了解 components.json 文件和格式	22-4
在站点构建器中定制工具栏组	22-8
限制内容框中的组件	22-11
将布局内容设为可编辑	22-13
创建部分布局	22-14
创建支持延迟加载的部分布局	22-15
使用 API 开发定制部分布局	22-16
开发内容布局	22-19

通过 Oracle Content Management 创建内容布局	22-21
将布局视图传递到内容布局	22-27
通过 API 生成“站点详细信息”页 URL	22-27
通过开发人员云服务在本地开发内容布局	22-28
通过开发人员云服务创建内容布局	22-29
定义 RequireJS 模块	22-30
配置构造器函数参数	22-31
呈现内容布局	22-31
编辑 Mustache 模板中的内容布局	22-32
添加动态 DOM 操纵	22-32
在 design.css 文件中定义样式	22-32
获取引用项	22-32
获取媒体 URL	22-33
引发触发器	22-33
导航到包含搜索查询的搜索页	22-33
扩展宏和呈现多信息文本	22-34
指向“详细信息”页的链接	22-35
扩展内容列表查询中的宏	22-35
开发强健的内容布局	22-38
呈现内容项	22-39
标准化内容布局的数据结构	22-39
创建示例博客模板	22-43
将内容布局映射添加到模板	22-43
使用本地测试工具测试内容布局	22-44
使用本地测试工具进行测试	22-44
将具有内容布局的模板导入到 Oracle Content Management 中	22-44

23 开发组件

关于组件	23-1
关于开发组件	23-2
创建组件	23-5
通过开发人员云服务开发定制组件	23-6
为 Oracle Content Management 开发定制组件	23-6
开发定制组件	23-7
编写并运行单元测试	23-8
优化组件（缩小）以提高性能	23-9
运行连续集成作业	23-10
为多语言站点开发可翻译组件	23-10
构建包含“设置”面板的 H1 组件	23-10
创建新的本地组件	23-11

构建基本 H1 组件	23-11
为组件添加 CSS	23-15
添加“设置”面板以更改标题文本	23-16
更新主题以便其他人选取 H1 组件样式	23-17
使用 HTML 模板创建简单的组件	23-18
创建 HTML 组件	23-19
将用户提供的数据添加到 HTML 组件	23-22
比较本地组件与远程组件	23-26
呈现组件设置	23-27
本地组件实施	23-28
组件的样式类	23-29
如何为内置组件设置样式	23-31
组件样式设置基本信息	23-31
特定于组件的样式设置	23-32
设置组件属性	23-39
在内嵌框架中呈现的组件	23-40
关于在内嵌框架中呈现的组件的实例 ID 和结构	23-41
远程组件的安全性	23-42
注册远程组件	23-43
删除组件	23-44
Sites SDK	23-45

24 定制控制器文件

关于控制器文件	24-1
默认的控制文件	24-1
修改默认的 controller.html 文件	24-2
关于 SCS 对象	24-3
SCS.sitePrefix	24-3
SCS.preInitRendering	24-3
SCS.getDeviceInfo	24-3
不应定制的控制文件部分	24-4
使用标记来实现定制控制文件可移植性	24-4
定制控制文件示例	24-5
更改站点前缀	24-6
定制等待图	24-7
定制收藏夹图标	24-8
为非 JavaScript Crawler 定制 <noscript> 和 <meta> 标记	24-9
预取 JavaScript 文件	24-9
使用额外的标记来验证站点所有权	24-10
增加设备检测	24-11

第 V 部分 使用其他工具进行站点开发

25 使用 OCE 工具包进行开发

在本地计算机上设置 OCE 工具包	25-1
通过 npm 安装相关项	25-2
使用 cec 命令行实用程序	25-2
使用本地测试工具进行测试	25-67
升级到 jQuery 3.5.x	25-67
使用开发人员云服务为 Oracle Content Management 进行开发	25-68
关于使用开发人员云服务	25-69
登录 Oracle Content Management 的开发人员云服务控制台	25-69
在开发人员云服务中创建项目	25-70
使用 Oracle Content Management 模板创建开发人员云服务项目	25-70
通过从 Oracle Content Management 下载 OCE 工具包在开发人员云服务中创建项目	25-70
将 OCE 工具包添加到新的 Git 资料档案库中的项目代码	25-70
在本地测试工具中测试定制组件、模板和内容布局	25-71
合并更改	25-71
使用 OCE 工具包将更改从测试环境传播到生产环境	25-71
对密码进行加密	25-77
注册服务器	25-77
为站点创建使用情况和权限报告	25-77
下载和上载文档及文件夹	25-78
从模板创建站点并为内容保持相同 GUID	25-79
从标准站点创建企业模板	25-79
导入和导出分类	25-79
导入和导出推荐	25-81
添加或删除集合内容	25-82
使用 OCE 工具包开发定制字段编辑器	25-82
将站点从一个服务器传输或更新到另一个服务器	25-87
传输没有内容项的站点	25-87
分组下载或上载站点的内容项	25-88
使用 OCE 工具包为站点页编制索引	25-88
为站点页文本创建内容类型	25-89
使用 OCE 工具包创建页索引内容项	25-90
将内容搜索添加到 Oracle Content Management 中的站点	25-90
将搜索页添加到站点	25-91

将搜索字段添加到主题	25-91
使用 OCE 工具包为多语言站点编制索引	25-92
创建简化的组件以便于组件开发	25-94
设置站点编译服务	25-94
编译站点以提高站点页的运行时性能	25-95
站点编译概览	25-95
与预呈现的交互	25-96
控制器站点页呈现	25-96
已编译的站点页呈现	25-96
模板编译	25-96
设置	25-96
编译模板	25-97
站点编译	25-102
站点编译的先决条件	25-103
编译站点	25-103
定制编译器	25-104
约束	25-105
调试定制编译器	25-105
页布局编译器	25-105
组件编译器	25-106
内容布局编译器	25-108
SCSCompileAPI	25-109
组件水合	25-110
发布	25-112
静态站点传送优先级	25-113
高速缓存标头	25-113
详细信息页	25-114
将内容项添加到通道	25-114
为移动设备编译站点	25-115
站点生命周期和已编译页	25-116
在 Oracle Content Management 服务器中创建新站点或资产翻译作业	25-117
通过语言服务提供商翻译站点	25-119
使用 OCE 工具包创建翻译作业	25-120
列出翻译作业	25-121
创建翻译连接器	25-121
为多语言站点生成站点地图	25-122
将翻译作业提交给语言服务提供商	25-123
将翻译作业上传到服务器	25-124

26 教程：使用 Knockout 开发组件

使用 Knockout 进行组件开发的简介和先决条件	26-1
步骤 1：创建组件	26-2
步骤 2：查看本地组件呈现的结构	26-3
步骤 3：查看本地组件设置的结构	26-6
步骤 4：显示组件中的新属性	26-10
步骤 5：注册触发器	26-11
步骤 6：引发触发器	26-12
步骤 7：注册操作	26-14
步骤 8：执行操作	26-15
步骤 9：为组件的每个实例创建不同的标题	26-17
步骤 10：结合使用嵌套组件和内嵌编辑	26-18
步骤 11：支持不同的布局	26-19
步骤 12：定义定制样式	26-21
步骤 13：在内嵌框架中呈现组件	26-22
步骤 14：在内嵌框架中呈现组件时使用定制样式	26-25
步骤 15：与页撤消和重做行为集成	26-28
步骤 16：资产管理	26-29
教程回顾	26-33

27 Sites SDK 参考

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)	27-8
SitesSDK.openAssetPicker (可选)	27-9
SitesSDK.publish(messageType, payload)	27-9
SitesSDK.subscribe(messageType, callback)	27-10
SitesSDK.Utils.addSiteThemeDesign(cssUrl)	27-12
SitesSDK.Utils.Logger	27-12

28 故障排除

我尝试创建站点，但没有模板	28-1
我无法删除站点	28-1
无法打开站点树或编辑页	28-1
我添加了组件，但没有在页中显示	28-2
我的文件夹、文件和对话组件无法工作	28-2
我更改了页布局，但一些内容消失了	28-2
我上载了图像的新版本，但没有在页中显示	28-2
我基于另一个服务添加了组件，但它无法正常工作	28-3
我的企业站点显示一条警告	28-3

前言

本文档介绍如何使用 Oracle Content Management 创建和管理体验（包括站点、资产以及它们相关联的结构和策略）。

目标读者

此出版物面向想要使用 Web 浏览器、通过桌面同步应用程序或移动设备处理文件的 Oracle Cloud 用户。

文档可访问性

有关 Oracle 对可访问性的承诺，请访问 Oracle Accessibility Program 网站，网址为：<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>。

获得 **Oracle** 支持

购买了支持服务的 Oracle 客户可通过 My Oracle Support 获得电子支持。有关信息，请访问 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info>；如果您有听力障碍，请访问 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs>。

多元化与包容性

Oracle 致力于推动多元化与包容性。Oracle 推崇并重视员工多元化，这有助于提升思想领导力和创新能力。打造包容性更强的企业文化是我们行动计划的一部分，这将对我们的员工、客户与合作伙伴带来积极影响。因此，我们当前正在从产品和文档中删除那些敏感术语。我们也认识到，必须与我们客户的现有技术保持兼容，并且需要在 Oracle 产品和行业标准的发展过程中确保服务的连续性。由于这些技术制约因素，删除敏感术语的工作会持续进行。这项工作需要一定时日，也需要外部合作。

相关资源

有关详细信息，请参见以下 Oracle 资源：

- 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*

约定

本文档遵循以下文本约定。

惯例	含义
粗体	粗体表示与操作或文本/词汇表中定义的术语相关联的图形用户界面元素。
斜体	斜体表示书名、强调部分或您需要提供特定值的占位符变量。
等宽字符	等宽字符表示段落中的命令、URL、示例中的代码、屏幕上显示的文本或输入的文本。

第 I 部分

简介

此部分介绍了使用 Oracle Content Management 构建站点的相关概念信息，并包含易于使用的教程来帮助您创建首个站点。其中包括以下章节：

- [构建站点入门](#)
- [创建第一个网站](#)

1

Oracle Content Management 概览

Oracle Content Management 是基于云的内容中心，可实现全渠道内容管理并加快推出体验服务的速度。它提供的强大协作和 workflow 管理功能，既简化了内容的创建和交付，同时又促进了客户和员工的互动。

通过 Oracle Content Management，您可以在任何设备上快速进行内部和外部协作，以批准内容以及创建情境式体验。借助内置的适合业务用户的工具，您可以轻松地构建全新的 Web 体验与精彩内容。您可以使用同一内容平台和相同流程推动与所有利益相关方进行数字化互动。Oracle Content and Experience 消除了阻碍您的技术和组织瓶颈，因此，您可以创建吸引人的体验。



注：

Oracle Content Management 入门版的功能集有限。要利用完整的功能集，请升级到高级版。

访问 Oracle Content Management

在授予您对 Oracle Content Management 的访问权限后，您将收到一封欢迎电子邮件，其中包含有关实例 URL 和您的用户名的详细信息。登录服务需要此信息，因此建议您保留下来供将来参考。

有多种方式可与 Oracle Content Management 交互：

- 您可从您喜爱的 Web 浏览器通过 Web 界面轻松地进行访问。您可以管理您在云中的内容、与其他人共享文件和文件夹、启动和参与对话、创建网站（如果允许）等。
- 您可以通过桌面应用程序在云和您的计算机之间同步文件和文件夹。您可以同步自己的文件以及与您共享的文件，从而确保您始终访问最新版本。
- 您可以通过 Microsoft Office 插件直接从 Microsoft Word、Excel、PowerPoint 和 Outlook 访问 Oracle Content Management 功能。
- 您可以通过适用于 Android 和 iOS 的移动应用程序在手机或其他移动设备上轻松地进行访问。移动应用程序很容易使用，因为其外观和操作方式就像 Web 浏览器中的服务一样。您可以访问云内容，对文件及文件夹进行搜索和排序，共享内容以及处理对话。
- REST API 和 SDK 为开发人员提供了强大的工具，使他们可以通过编程方式将 Oracle Content Management 功能加入到 Web 应用程序和移动应用程序中。

了解角色

您可以访问的 Oracle Content Management 功能取决于分配给您的角色。您看到的选项会因您的用户角色而异。标准用户可以使用文档、对话和站点。企业用户还可以访问资产。开发人员可以看到用于构建和定制网站内容的选项，例如模板、主题、组件和布局。管理员可以看到用于配置服务、将服务与其他业务应用程序集成在一起以及设置资产资料档案库的选项。

Oracle Content Management 中存在不同类型的角色：

- **组织角色** — 您在组织中的角色决定了您需要执行的任务以及如何使用功能。
- **用户角色** — 用户角色控制您在 Oracle Content Management 中可看到的功能。
- **资源角色（权限）** — 您可以看到的内容以及可以对资源（例如文档、内容项、站点或模板）执行的操作取决于向您共享资源时为您分配的角色。

[了解详细信息...](#)

管理资产

Oracle Content Management 提供了强大的功能来帮助您管理所有资产，以便在不同通道中使用，包括网站、市场营销材料、电子邮件营销活动、在线商店和博客。它提供了一个包含所有资产的集中式内容中心，您可以在将资产划分到资料档案库和集合中，以及创建规则来定义资产使用方式和使用场合。

此外，还提供了大量管理和 workflows 功能来引导您完成资产的创建和审批流程，并确保您只能使用获得授权的版本。

您可以轻松地标记和筛选资产，以便快速查找所需资产，并在允许的场合使用它们。此外，智能内容功能会在您使用资产时自动标记资产并提供建议。

创建资产类型以定义在用户创建资产时需要收集哪些信息。数字资产类型定义数字资产（文件、图像和视频）所需的定制属性。内容类型可将不同的内容片段分组到可重用单元中。之后，用户可以根据这些资产类型创建数字资产和内容项，以在各通道之间一致地使用和表示它们。

[了解详细信息...](#)

管理内容

通过 Oracle Content Management，您可以在云中管理您的内容。内容全都在一个位置，并且可从任何位置访问。

您可以按文件夹分组文件，以及执行常见的文件管理操作（复制、移动和删除等），操作方式与在本地计算机上几乎一样。此外，由于所有文件都在云中，因此，无论您在何处都可以访问它们，也可以在移动设备上访问它们。如果您安装桌面应用程序，则您的所有内容都可以自动同步到您的本地计算机，因此，您的手边随时都有最新版本。

[了解详细信息...](#)

对内容开展协作

由于您的所有内容都在云中，因此可以轻松地共享您的文件或文件夹，以与您组织内部或外部的其他人进行协作。您与之共享内容的所有人都可以访问最新信息 — 无论其身在何处，也无论其何时需要。您可以授予对整个文件夹的访问权限，也可以提供指向特定项的链接。对共享项的所有访问都会记录，因此您可以监视对每个共享项的访问方式和时间。

借助 Oracle Content Management 中的对话，您可以通过实时讨论主题和发表评论来与其他人员进行协作。您可以启动有关任何主题的独立对话，同时根据需要添加文件。也可以启动有关特定文件、文件夹、资产或站点的对话，以便快速、轻松地获取反馈。

与对话关联的所有消息、文件和批注都会保留下来，因此，可以轻松跟踪和查看讨论。此外，您的对话是在云中进行的，因此，您还可以随时从移动设备查看并参与对话。

[了解详细信息...](#)

构建站点

通过 Oracle Content Management，您可以快速构建和发布市场营销和社区网站（从概念到发布）以提供吸引人的在线体验。该过程是完全集成的：内容、协作和创作都整合到单个创作和发布环境中。

要快速开始，请使用现成的模板、拖放组件、示例页布局和站点主题，以便利用预定义的构建块组装站点。开发人员也可以创建定制模板、定制主题或定制组件，以打造独一无二的在线体验。

只需将 YouTube 视频、流式视频、图像、标题、段落、社交媒体链接及其他站点对象拖放到页面上的指定内容框中，即可添加这些组件。只需按下按钮即可切换主题以及重新设计站点品牌形象，从而在组织中提供一致的优化外观。

您可以处理一项或多项更新，预览站点中的更新，然后（在准备就绪后）通过一次单击发布更新。

除了在站点构建器中创建和发布站点外，Oracle Content Management 还支持使用 REST API、React JS、Node JS 及其他 Web 技术进行“无头 (headless)”站点开发。

[了解详细信息...](#)

集成和扩展 Oracle Content Management

作为一项 Oracle 平台即服务 (Platform-as-a-Service, PaaS) 产品，Oracle Content Management 可与其他 Oracle Cloud 服务无缝协作。

您可以在 Web 应用程序中嵌入 Web UI，以便用户可以直接与内容交互。可使用应用集成框架 (Application Integration Framework, AIF) 通过定制操作将第三方服务和应用程序集成到 Oracle Content Management 界面中。或者开发内容连接器，将您已在其他地方创建的内容加入到 Oracle Content Management 中，对其进行集中管理，以及在多个通道中将其用于新体验。

借助一组用于内容和站点管理、传送和协作的丰富 REST API 和 SDK，您可以将 Oracle Content Management 功能加入到您的 Web 应用程序中。

创建与云中内容 SDK 和资产交互的客户端应用程序。开发与协作对象的定制集成，或检索资产以在任何需要它们的位置使用。您可以通过网站、内容传送网络 (Content Delivery Network, CDN) 以及移动应用程序访问和传送所有内容和资产，而且这些内容和资产针对每个通道进行了优化。

[了解详细信息...](#)

入门

为了帮助您开始使用 Oracle Content Management，请访问 [Oracle 帮助中心](#)，此处提供了大量资源，包括[文档](#)、[视频](#)、[指导教程](#)和[开发人员信息](#)。

此外，如果您需要，可从[技术支持](#)和[社区](#)获取帮助。

迁移至 Oracle Cloud Infrastructure

如果您尚未使用 Infrastructure 控制台在 Oracle Cloud Infrastructure (OCI) 上运行您的 Oracle Content Management 订阅，Oracle 建议您迁移到本机 OCI 环境。这将确保在将来能够利用 Oracle 云平台的优势和发展。

迁移不是自动进行的；您需要提交服务请求才能启动该过程。

[了解详细信息...](#)

2

构建站点入门

Oracle Content Management 是基于云的内容中心，可实现全渠道内容管理并加快推出体验服务的速度。通过 Oracle Content Management，您可以在任何设备上快速进行内部和外部协作，以批准内容、管理数字资产以及使用内置的适合业务用户的工具创建情境式体验。

- [Oracle Content Management 概览](#)
- [站点入门](#)
- [了解站点创建流程](#)
- [了解站点开发](#)
- [了解站点管理](#)
- [创建第一个网站](#)



视频

站点入门

具有适当权限的任何人都能够使用 Oracle Content Management 构建网站。您无需使用任何专有工具、代码或软件。采用友好直观的图形用户界面。



注：

对于 Oracle Content Management 入门版，您只能使用 1 个站点且无法使用站点监管。要利用完整的功能集，请升级到 Oracle Content Management 高级版。

能够构建站点的人员取决于多个因素：

- 服务管理员是否启用了站点创建。
- 服务管理员是否启用了站点管理。请参见[了解站点管理](#)
- 站点创建是否限于站点管理员（在站点管理已禁用时可用）。
- 站点管理员是否已使模板可用。

在创建站点时，可以从模板开始。模板包含您在开始构建站点时需要的全部内容，包括站点代码框架、带有示例页和内容的默认站点、具有样式设置的主题以及图像等资源，甚至包括定制组件。请参见[了解站点创建流程](#)。

如果您要创建企业站点，该站点将与资料档案库关联，并且必须定义了指定默认语言的本地化策略。在创建站点之前，必须先创建资料档案库和本地化策略。您将站点需要的资产和文档存储在资料档案库中，资料档案库策略指示通过资产可以实现的操作。有关如何使用资料档案库的详细信息，请参见[了解资产资料档案库](#)。

编辑站点时，您可以创建新的更新或者使用现有更新。在更新中，您可以编辑和添加内容、调整样式设置、添加和删除页、更改页布局以及排列页。请参见[初步了解“站点构建器”页](#)以了解编辑器的用途。

不必一次将所有更新都准备好。您和您的团队成员可以同时独立地处理多个更新。例如，您可能使用每周新闻处理更新，而其他团队成员正在为即将开始的销售会议添加页。您可以根据需要随时编辑、复查和保存更改到更新，以及随时将更新合并到基本站点。

在准备好之后，发布站点。只需单击一下即可将站点发布到 Web。简而言之，这就是从概念到发布的过程。

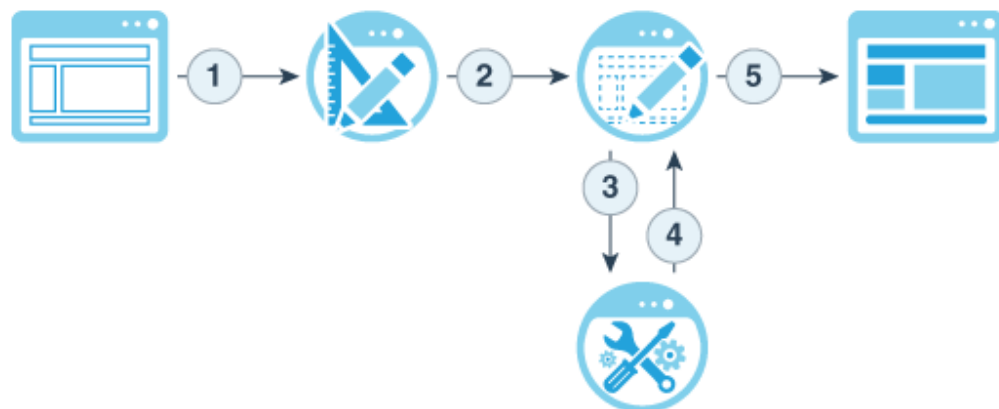
了解站点创建流程

接下来了解创建和发布网站的整体流程。

在您开始之前，服务管理员应启用了站点创建，配置了可以创建站点的人员，以及安装了模板并使其可用，以便您使用这些模板创建站点。服务管理员必须至少为您分配对模板的下载者权限。请参见[了解用户角色](#)、[站点入门](#)和[了解站点管理](#)。

如果您将要创建企业站点（支持使用资产和多语言站点），则其将与某个资料档案库关联。您将站点需要的资产和文档存储在资料档案库中，资料档案库策略指示通过资产可以实现的操作。资料档案库必须在创建企业站点之前创建。请参见[了解资产资料档案库](#)。

此处显示的是站点创建过程的概览，其中带有指向详细信息的链接。



1. 选择模板并为站点命名：如果启用了站点管理并且站点需要审批，则必须等到其获得批准后才能完成下一步。站点保持脱机，直至您准备好发布。
2. 创建站点：这将创建可定制的空白框架。
3. 创建新的更新或使用现有更新：更新是对当前基本站点的指定更改集合。每次在编辑器中查看或编辑站点时，您会使用某个更新。请参见[了解站点更新](#)。
4. 在编辑器中打开站点：使用拖放编辑器向站点中添加页和内容。请参见[了解站点更新](#)。
 - 将页添加到站点：选择预定义的布局以快速定义页类型。请参见[添加页](#)。
 - 将组件添加到页：将文本、图像、文档以及更多内容拖放到页上。请参见[使用内置组件](#)。

- **更改组件的内容或属性**：微调组件的间距、对齐方式和其他属性。请参见[使用样式和格式设置](#)。
 - **(可选) 切换页使用的布局**：实时更改页布局。请参见[更改页布局](#)。
 - **(可选) 切换站点使用的主题**：快速更改整个站点的外观。请参见[在站点中使用模板和主题](#)。
 - **设置搜索引擎属性**：帮助搜索引擎查找您的站点以增加流量。请参见[设置搜索引擎属性](#)。
5. **应用站点更新**：使用更新中的更改来更新站点。请参见[发布站点更改](#)。
- **与其他团队成员共享站点**：与指定的用户共享站点，向每位用户分配角色以确定他们能够在站点上执行的操作。有关共享的信息，请参见[了解站点安全性](#)。
 - **设置站点安全性**：选择哪些用户可以访问您的已发布站点。请参见[更改站点安全性](#)。
 - **发布站点**：发布对站点的更改。请参见[发布站点更改](#)。
 - **使站点联机**：使站点可供用户访问（访问基于站点安全性）。请参见[使站点联机或脱机](#)。

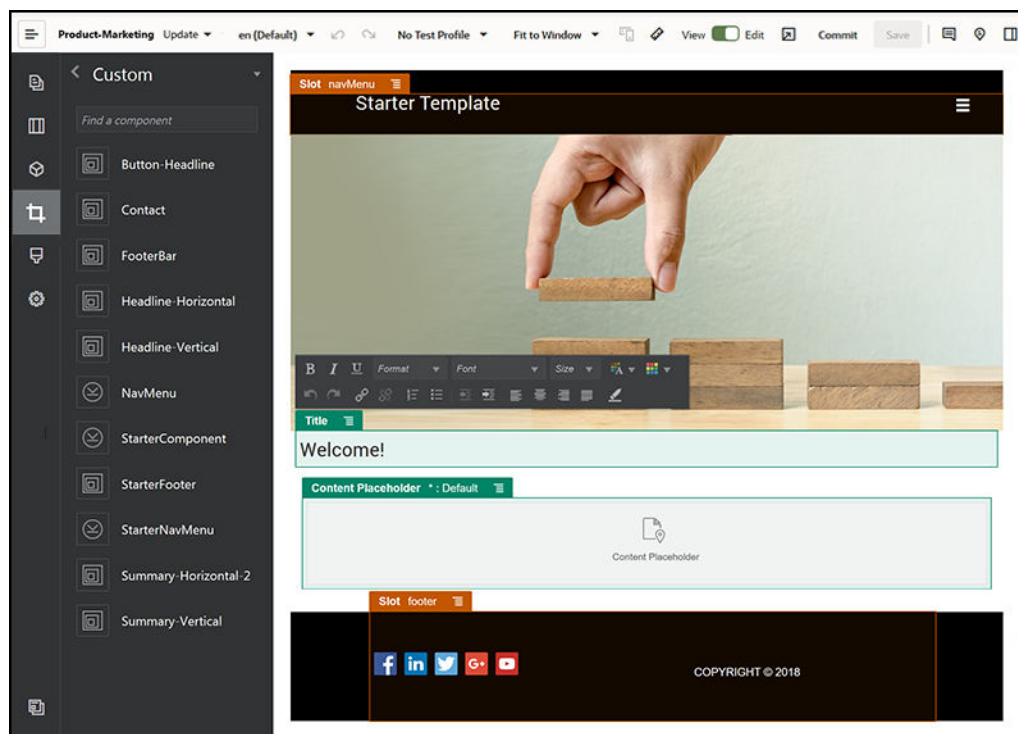
了解站点开发

作为站点开发人员，您负责定义站点创建者用于构建站点的框架，例如：

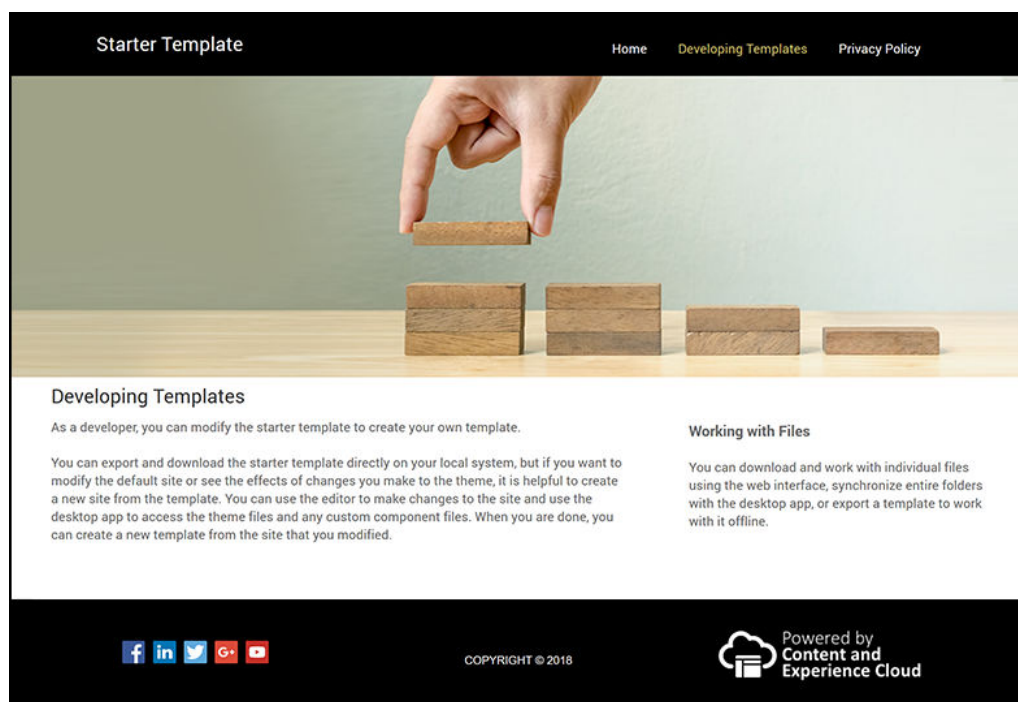
- **标准模板**，用作站点的框架，基于某个主题，带有示例页和内容、定制组件以及您在开始构建站点时需要的其他资源
- **主题**，用于定义站点的整体外观，包括徽标、样式表、配置文件和定义站点导航的背景代码
- **定制组件**，用于将特定类型的内容添加到站点页，使您可以使用任何选定的页技术开发可嵌入到站点页中的复合元素

通过将站点的显示形式与站点内容分开，可确保任何创建的站点都能维护您组织的标准和品牌，让内容创建者有时间专注于内容，使开发和站点创建更加高效。

下面是内容创建者在向站点添加内容时可使用的简单模板示例。请注意该页面上布置的通用占位符：



下面的示例是使用简单模板构建的站点在添加内容后，预览时可能呈现的外观。请注意使用特定内容和导航元素定制占位符所取得的效果：



此外，站点开发人员会创建和维护**样式表**，通过 Web 界面或使用 OCE 工具包**构建站点**，以及在 Oracle Content Management 和其他服务之间配置集成。此外，和任何其他员工一样，他们也可通过共享内容、启动或参与对话、或者使用桌面或移动应用程序与他人开展协作。

开发人员必须获得标准用户或企业用户角色，才能使用 Oracle Content Management。具有标准用户角色的开发人员可以创建组件、主题和标准模板。具有企业用户角色的开发人员也可以创建布局，并可以将站点另存为标准模板或企业模板。

除站点开发外，Oracle Content Management 作为云中一款强大且灵活的后端内容管理系统 (content management system, CMS)，也可在无头 (headless) 环境中使用。

了解站点管理

通过站点管理，业务用户可以轻松地创建符合公司策略的站点，而站点管理员可以轻松地从这个集中位置控制和跟踪站点。

注：

对于 Oracle Content Management 入门版，您只能使用 1 个站点且无法使用站点监管。要利用完整的功能集，请升级到 Oracle Content Management 高级版。

问题

目前，行业中存在一些问题。公司面临着体验迅猛发展的巨大挑战。公司必须能够快速创建和部署这些体验并提供集中可见性，以跟上公司品牌发展，确保及时的消息传送。企业往往会发现开发并提供新体验永无止境。要想获得成功，公司需具备适应能力出众的强大技术，以便您能够快速扩展，确保统一创建和管理众多体验，从而帮助推进产品或品牌计划。您必须能够快速、高效地推出新体验，否则就会错失市场机遇，无法能够很好地留住现有客户和吸引潜在客户。

体验迅猛发展通常是多年单独的孤立计划带来的结果；每个部门、品牌和团队根据自己的计划采购 Web 内容管理 (Web Content Management, WCM) 系统，但没有集中监督。由此产生的复杂环境不但进入市场的速度缓慢，而且很容易遭受安全威胁，其成本也是逐渐加速增长且不可预测。

由于缺乏一致的体验构建流程，再加上各种不同的 WCM 技术的混用，导致组织面临着巨大的安全风险。当多个利益相关方各自存在一些网站问题时，核心职责（例如安全性）变得非常分散，这将使企业处于危险境地。如果存在多个不同的 WCM 系统，它们分别采用了不同的 Web 应用程序和 IT 基础结构监督措施，通过它们来交付这些体验就会导致很难（甚至是不可能）确保所有内容都得到适当的保护。

此外，维护多个价格昂贵的商业 WCM 系统还会导致重叠成本，也就是说，可能要白白付出额外的许可证年度续订费用和持续的内部或外部合作伙伴支持成本。

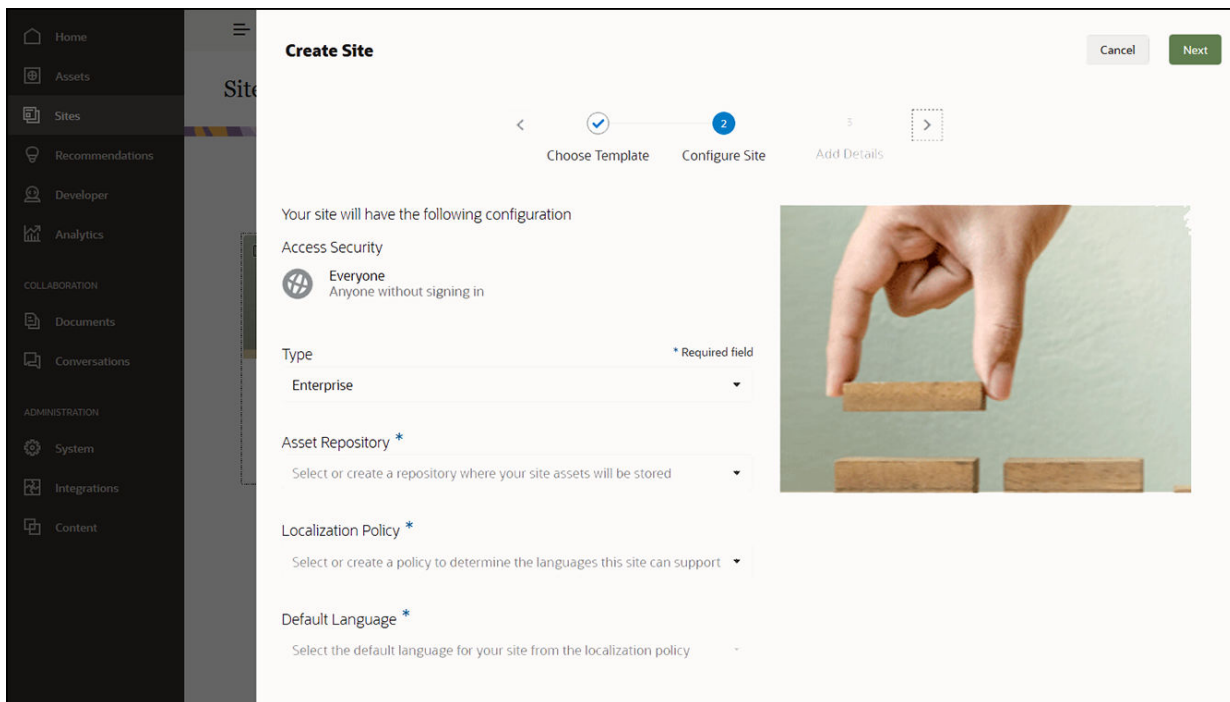
这导致企业各方面都受到约束：

- 对企业的约束：
 - 没有自助服务；依赖 IT 或昂贵的外包
 - 在没有技术帮助的情况下就无法进行更新
 - 没有适合业务用户的工具来管理工作
 - 无法集中查看所有体验和活动
- 对 IT 的约束：
 - 所有体验都需要 IT 参与其中
 - 对企业创建的体验缺乏管理，包括外包体验不受监视

- 体验构建架构分散堆叠
- 必须管理和部署数百个体验中对内容和布局所做的更改
- 对用户的约束：
 - 各通道的消息传送不一致
 - 信息过时
 - 性能低下，导致通道废弃

解决方案

Oracle Content Management 的管理功能可简化并加速向业务用户提供体验的过程，同时让 IT 部门可以轻松地从一个集中位置控制和跟踪体验，同时能够完全管理体验的整个生命周期，从而降低了创建和维护公司所需的每个新体验的成本。管理功能内置在 Oracle Content Management 核心中，只需管理员启用即可。请参见 *Administering Oracle Content Management* 中的“配置站点和资产设置”。



功能

- **批准的模板目录** — IT 开发人员可以在一个模板目录中填充一组站点模板以满足不同业务线的需求。他们可以限制哪些模板可用以及这些模板对于哪些人员可用。他们可以应用模板策略来指定新站点必须遵循的安全类型以及新站点是否需要审批。这样，就可以快速简单地申请新站点，同时又能确保业务用户遵循品牌和安全准则。请参见 [更改模板状态或受众](#)和[更改模板策略](#)。
- **简化的申请、审批和预配** — 让业务用户可以快速申请需要审批和采用自动预配的新站点。请参见[创建站点](#)和[管理站点请求](#)。
- **站点管理** — 无论站点由谁创建和部署，IT 部门都可以从一个位置集中管理所有站点。IT 用户可以监视任何已部署站点的站点状态以及更改其状态。请参见[管理站点和站点设置](#)。

优势

使用管理时，您将看到创建和维护每个新体验的成本降低。

- 为企业带来的优势：
 - 无需依赖 IT 即可快速预配
 - 内容与设计分开，方便重用
 - 在一个控制台中全局管理体验、用户和权限
 - 通过报告深入了解体验操作
- 为 IT 带来的优势：
 - 没有复杂的实施（云原生解决方案）
 - 自动扩展以应对增长和季节性高峰
 - 对体验进行管理，确保安全且满足公司品牌要求和遵从性标准
- 为用户带来的优势：
 - 各通道的消息一致
 - 最新信息
 - 最佳性能
 - 安全体验

流程

下面是使用站点管理时创建和管理站点涉及的步骤：

1. 管理员启用管理。请参见 *Administering Oracle Content Management* 中的“配置站点和资产设置”。
2. 站点管理员能使批准后的模板可供用户用来创建站点。请参见[更改模板状态或受众](#)和[更改模板策略](#)。
3. 用户从批准的模板创建站点请求。请参见[创建站点](#)。
4. 站点管理员批准站点请求，然后系统自动创建站点。站点管理员也可以拒绝站点请求，并添加注释来解释拒绝请求的原因，然后站点创建者可以更正问题并再次提交请求。请参见[管理站点请求](#)。
5. 站点创建者继续执行站点创建过程，编辑并发布站点。请参见[了解站点创建流程](#)。
6. 可以在站点页上进行持续的站点管理和更新。站点管理员可以在站点页上查看所有站点；其他用户可以查看自己创建或作为其成员的站点。请参见[管理站点和站点设置](#)。

3

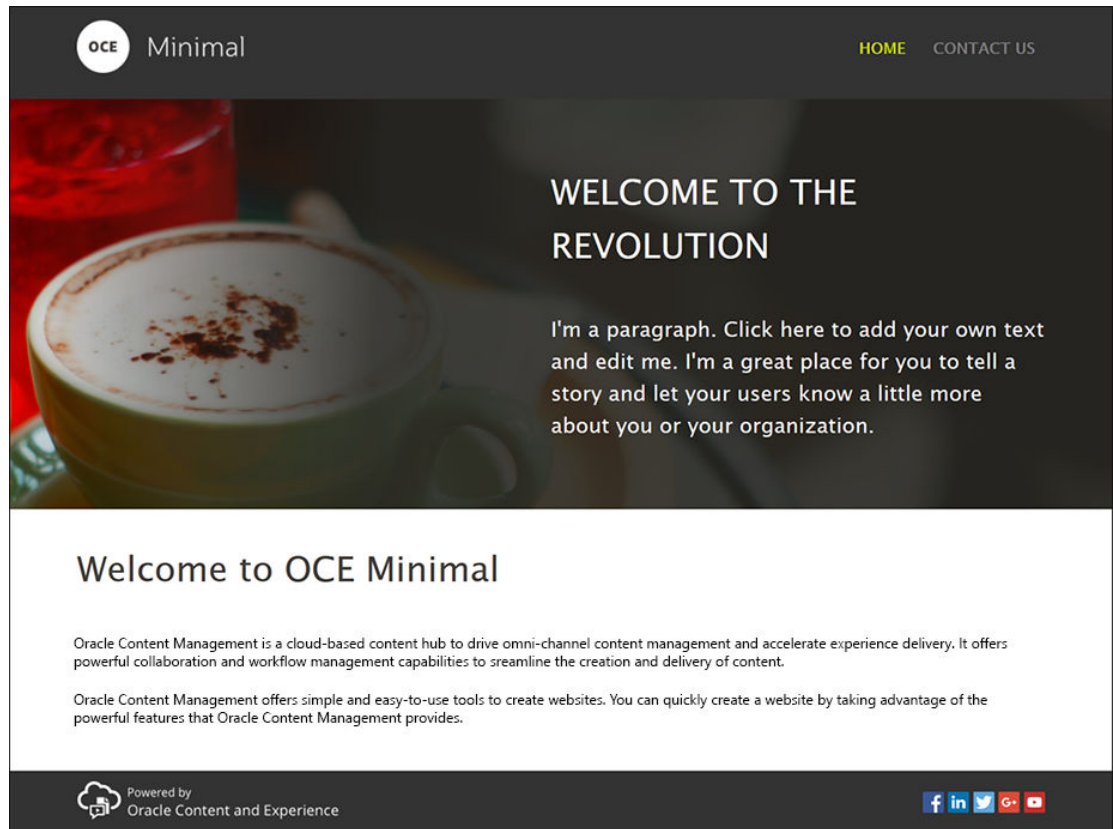
创建第一个网站

使用 Oracle Content Management 轻松快速地构建第一个网站。

在本教程中，您将了解如何使用 Oracle Content Management 中名为 **Blank-Template** 的现成模板设置和发布网站。您将看到无需任何额外编码即可使用站点构建器轻松构建自己的第一个网站。

本教程由三个基本步骤组成：

1. [设置环境](#)
2. [设置网站](#)
3. [发布网站](#)



但在开始之前，我们先[检查一些事项](#)。

开始之前

要开始使用，您将需要使用下面的应用程序角色访问 Oracle Content Management 实例：

- **CECContentAdministrator**

- **CECDeveloperUser**

如果您没有上述角色，可以请服务管理员为您分配这些角色。

有权访问 Oracle Content Management 实例后，以内容管理员身份登录。

我们开始吧。

步骤 1: 设置环境

首先，需要设置环境，方法是导入 Minimal-NavMenu 定制组件并上传示例图像。

因此，下载 <https://www.oracle.com/middleware/technologies/content-experience-downloads.html> 上提供的资产包 **OCESamplesAssetPack.zip**。在下载资产包中，找到 **OCECreateYourFirstSite_data.zip** 文件，此文件包含以下两个 zip 文件：

- **Minimal-NavMenu.zip**，其中包含 Minimal-NavMenu 定制组件。
- **Minimal-Images.zip**，其中包含网站中使用的示例图像。

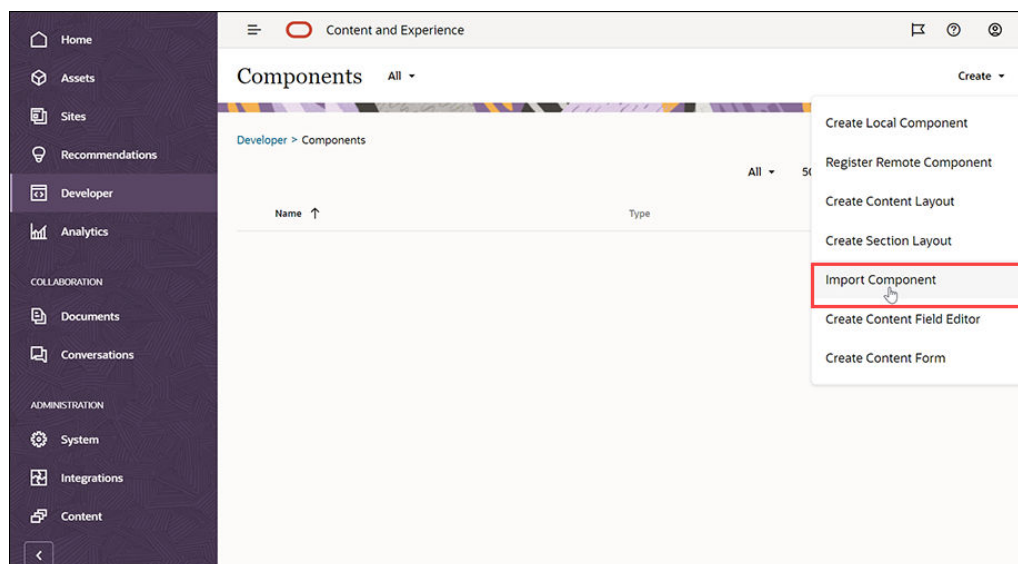
下面开始设置环境：

1. 导入 Minimal-NavMenu 定制组件
2. 发布 Minimal-NavMenu 定制组件
3. 添加示例图像

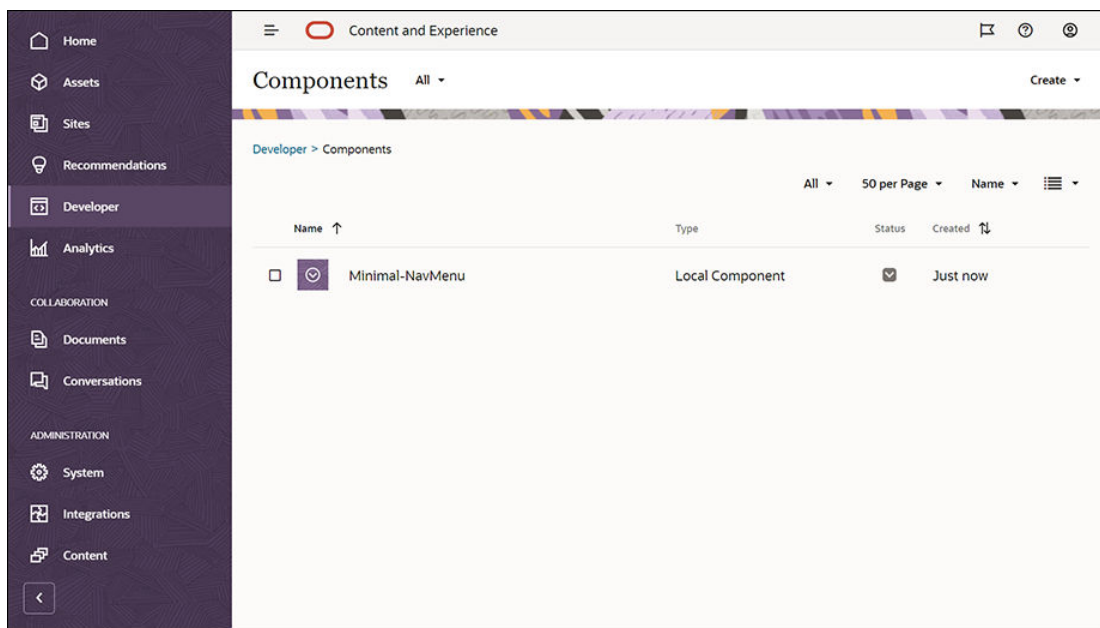
导入 Minimal-NavMenu 定制组件

登录 Oracle Content Management Web 界面后，可单击左侧导航菜单中的开发人员，然后单击查看所有组件。如果您看不到开发人员选项，表明您没有所需的用户角色。

在“组件”页上，单击创建，然后选择导入组件。上传 **Minimal-NavMenu.zip** 文件（在 **OCECreateYourFirstSite_data.zip** 文件内提供）并选择此文件。



现在，Minimal-NavMenu 组件列在“组件”页上。

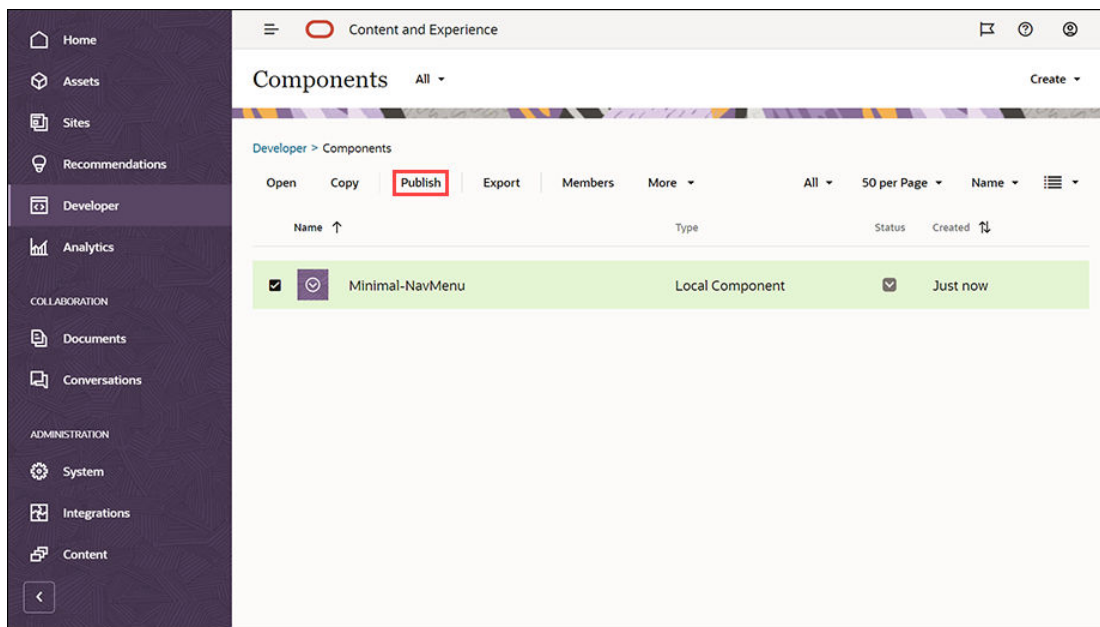


下一步: 发布 Minimal-NavMenu 定制组件

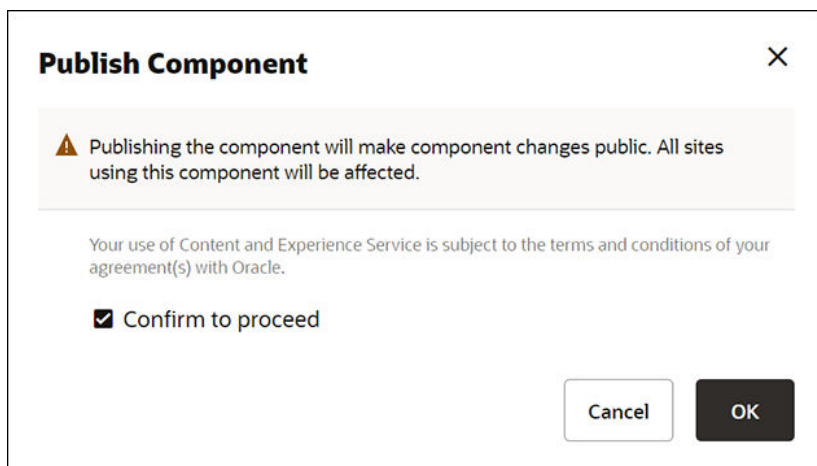
发布 Minimal-NavMenu 定制组件

现在, 需要发布导入的 Minimal-NavMenu 组件。

在“组件”页上, 选择 Minimal-NavMenu 组件, 然后单击菜单栏或右键单击菜单中的发布。



在“发布组件”对话框中, 选择确认以继续, 然后单击确定。



发布 Minimal-NavMenu 组件后，页面顶部会显示一则通知，提示已发布组件。

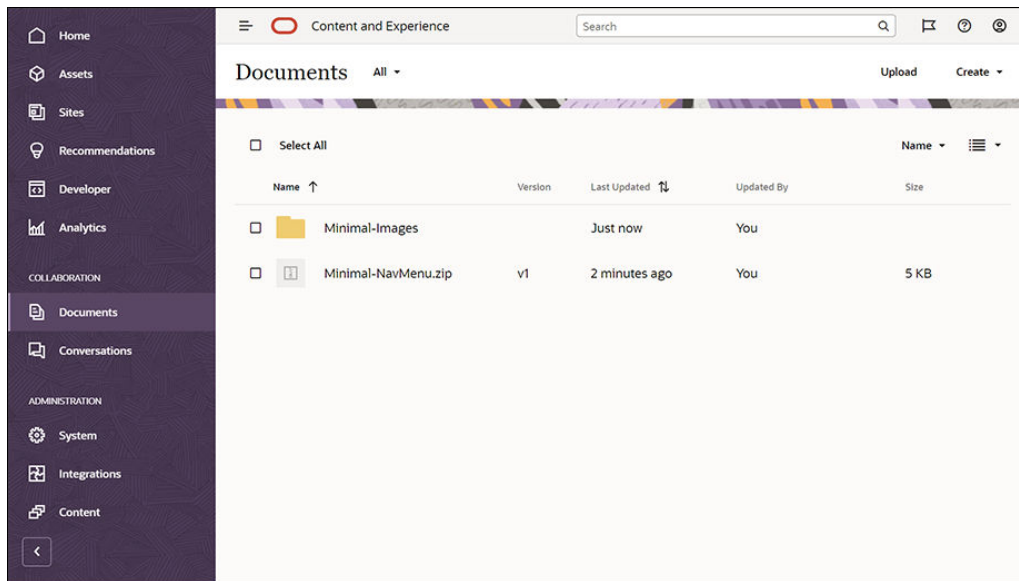
下一步：添加示例图像

添加示例图像

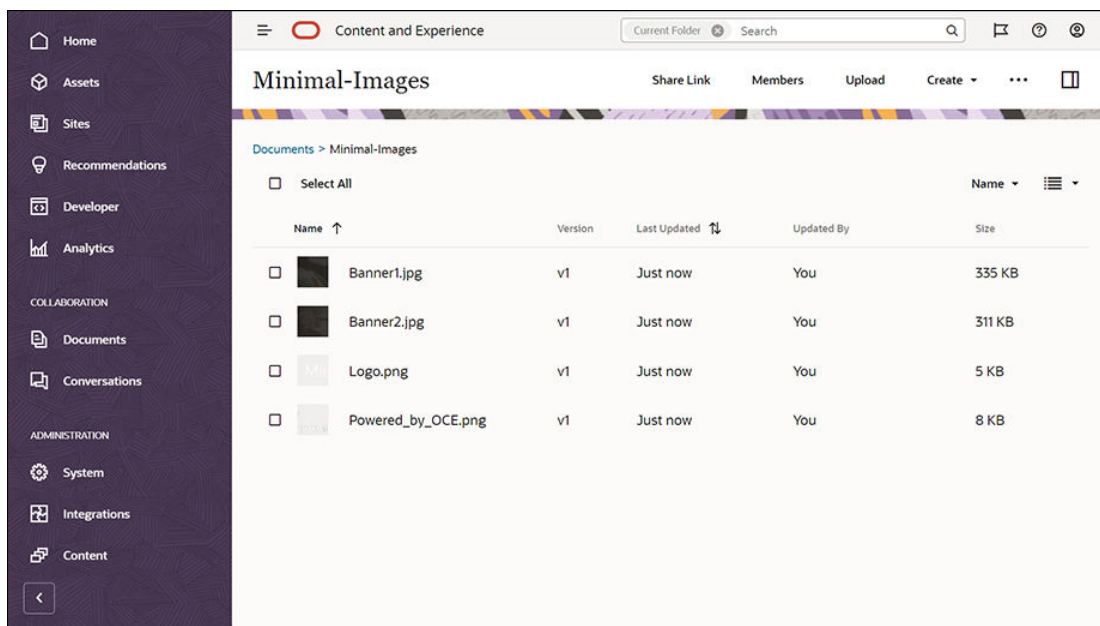
现在，需要将这些示例图像（将在创建的网站中使用它们）添加到文档中。

将 **Minimal-Images.zip** 文件（在 [OCECreateYourFirstSite_data.zip](#) 文件内提供）的内容提取到本地计算机上名为 **Minimal-Images** 的文件夹中。

在左侧导航菜单中，单击文档，然后单击创建以创建名为 **Minimal-Images** 的文件夹。



将本地计算机上 **Minimal-Images** 文件夹中的所有图像上传到 Oracle Content Management 中新创建的 **Minimal-Images** 文件夹中。



注:

模板和网站中使用的图像是 Shutterstock 的图像。如果您要重用这些图像，必须获得 Shutterstock 的授权。您也可以将这些图像替换为自己的图像。

下一步: 设置网站

步骤 2: 设置网站

现在，您已有创建网站所需的全部内容，可以开始设置网站了。

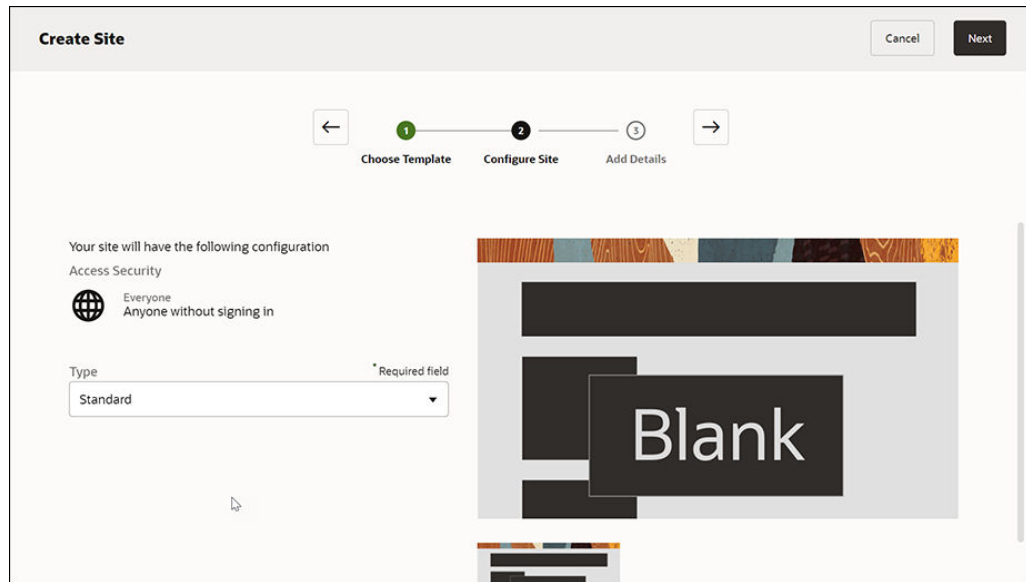
1. 创建网站
2. 编辑网站

创建网站

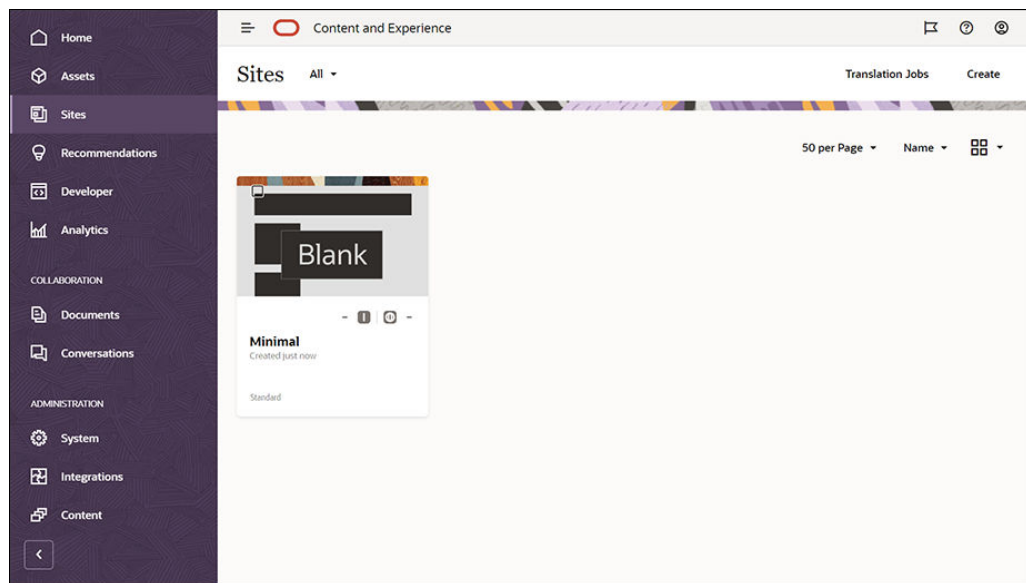
在左侧导航菜单中，依次单击站点和创建。选择 **Blank-Template**，然后单击下一步。

如果未看到 **Blank-Template**，请与服务管理员联系。服务管理员将需要使现成的模板可供使用。预配服务时，服务管理员通常会 [安装现成的模板](#)（该模板由 Oracle Content Management 提供）。

请确保选择创建标准网站，然后单击下一步。输入网站的名称（例如 **Minimal**）。单击完成。




将创建新网站并在“站点”页上列出新网站。

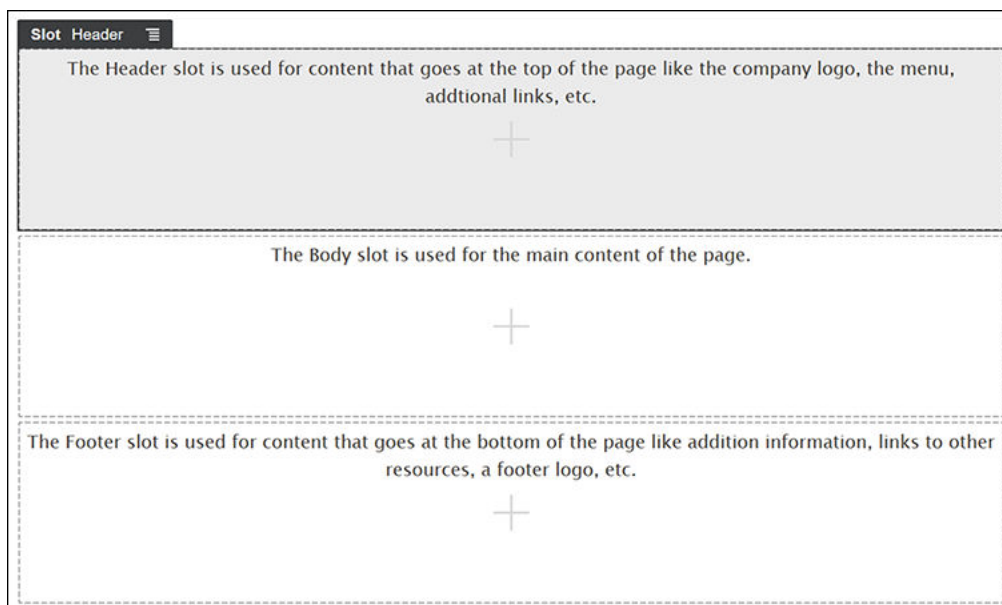


下一步：编辑网站

编辑网站

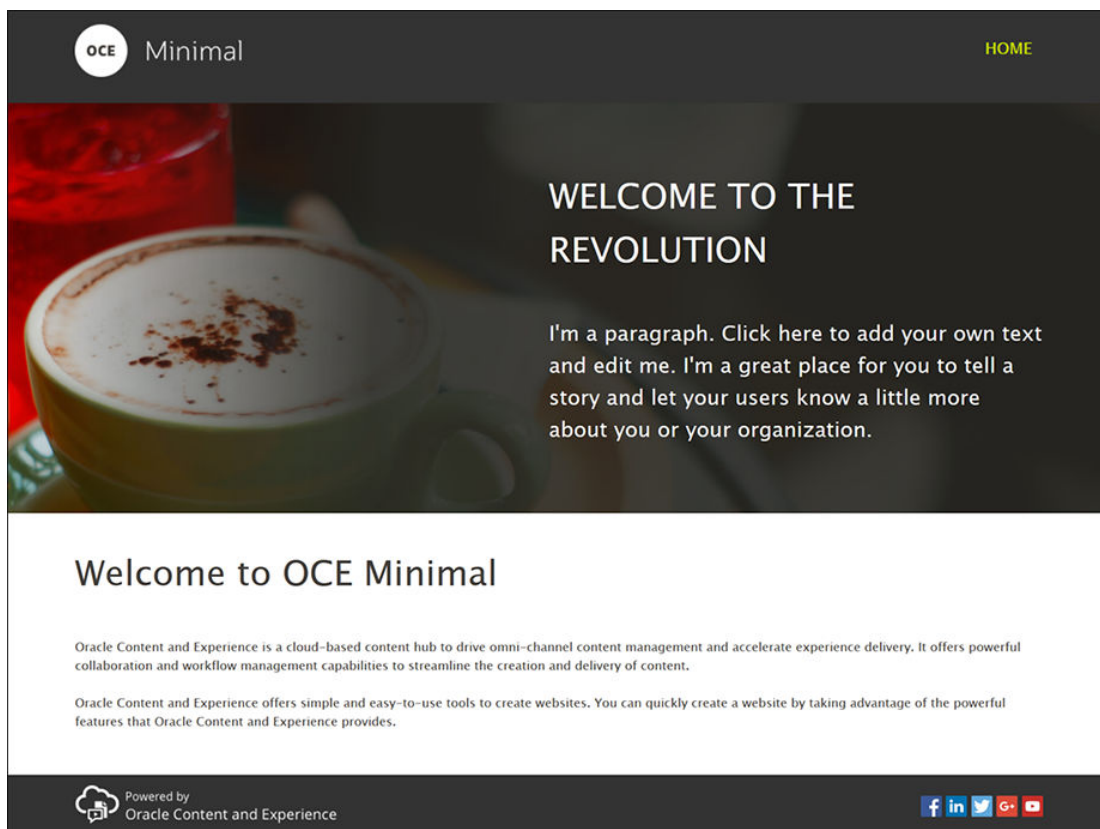
在站点构建器中打开新创建的网站，方法是选择该网站并从菜单栏或右键单击菜单中选择打开。在站点构建器中，将  开关设置为编辑模式。输入更新名称，然后单击确定。

在编辑模式下，您将看到该网站有三个内容框，这些内容框是页面上可用的区域（具体取决于页面布局）。将鼠标悬停在页面上的每个 + 上可查看页眉、正文和页脚等内容框。




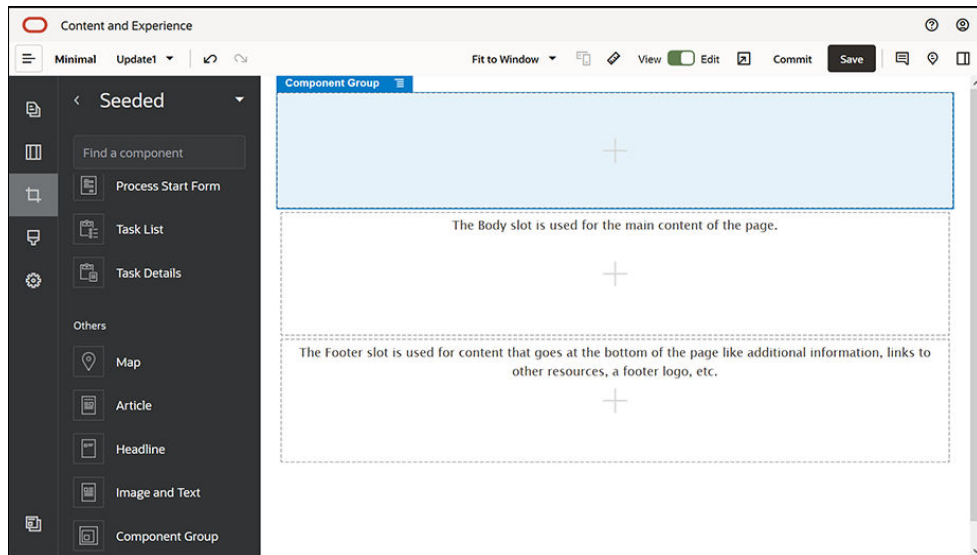
通常，我们使用“页眉”内容框显示公司的标识、导航菜单等。我们使用“正文”内容框显示页面的主要内容，使用“页脚”内容框显示版权信息、社交媒体链接以及任何其他信息。


首先，我们来构建主页。下面是已完成的主页的外观：

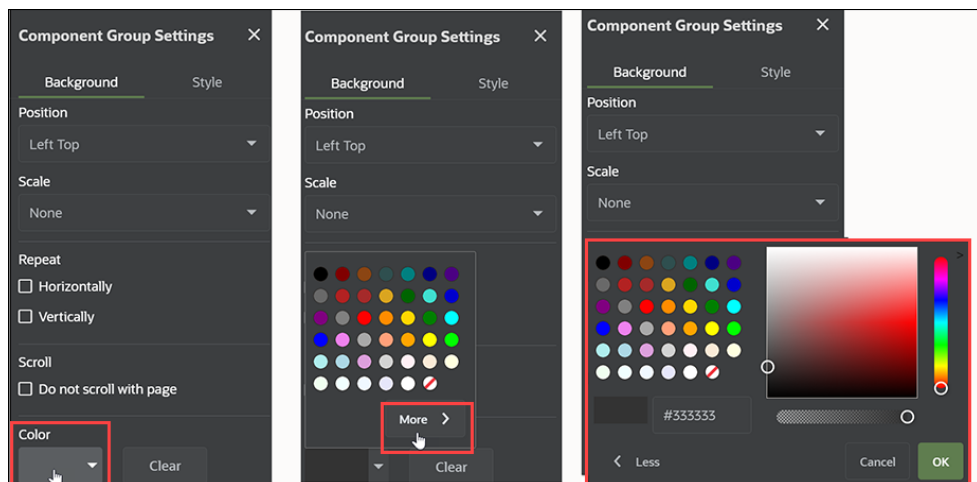


下面使用现成的组件填充“页眉”内容框：

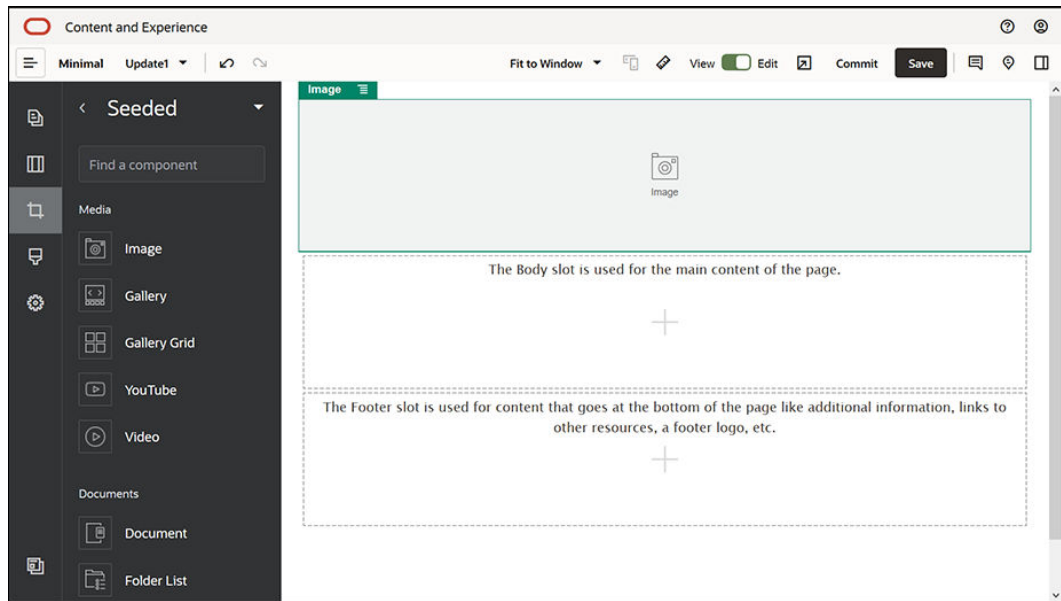
1. 在左侧子工具栏中，单击 ，然后单击已植入以显示随 Oracle Content Management 提供的现成组件的列表。
2. 在左侧子工具栏中，查找名为“组件组”的现成组件。将其拖放到“页眉”内容框中。



3. 依次单击组件组的菜单图标  和设置。在设置中，单击颜色下拉列表（位于设置列表底部），然后单击更多。输入 #333333 并单击确定。

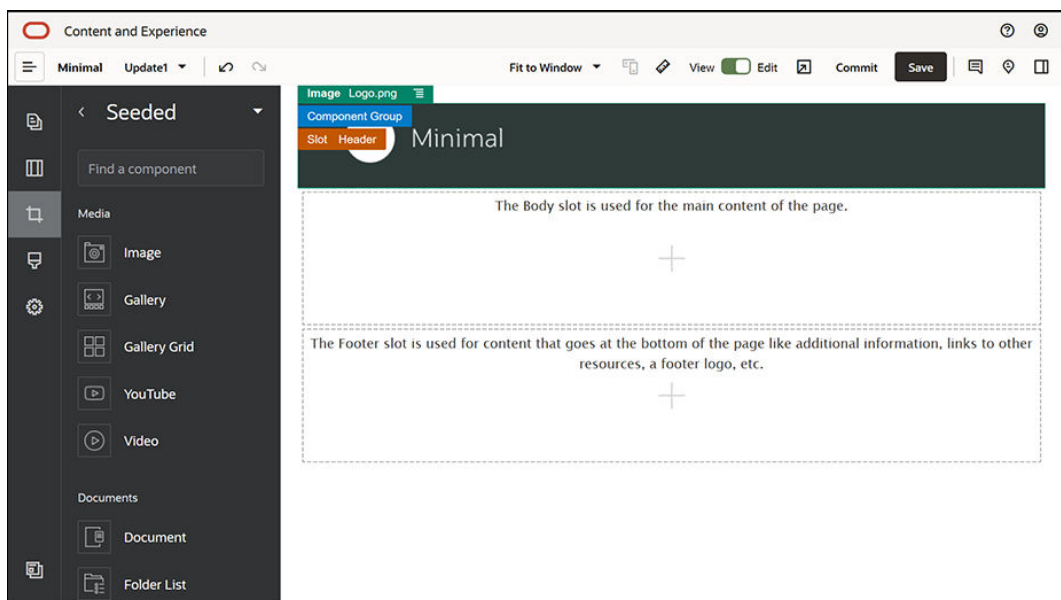


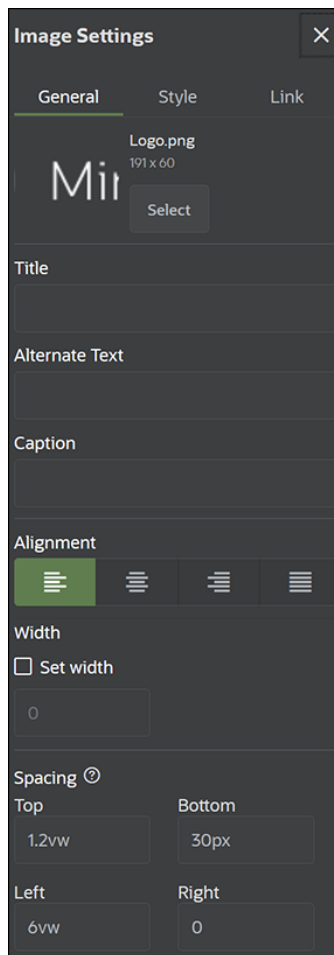
4. 将图像组件拖放到组件组中。



- 依次单击图像组件的菜单图标  和设置。在一般信息选项卡中完成设置。


属性	值
选择	Minimal-Images 文件夹中的 Logo.png
对齐方式	左对齐
宽度	取消选择设置宽度
上对齐	1.2vw
下对齐	30px
左对齐	6vw
右对齐	0

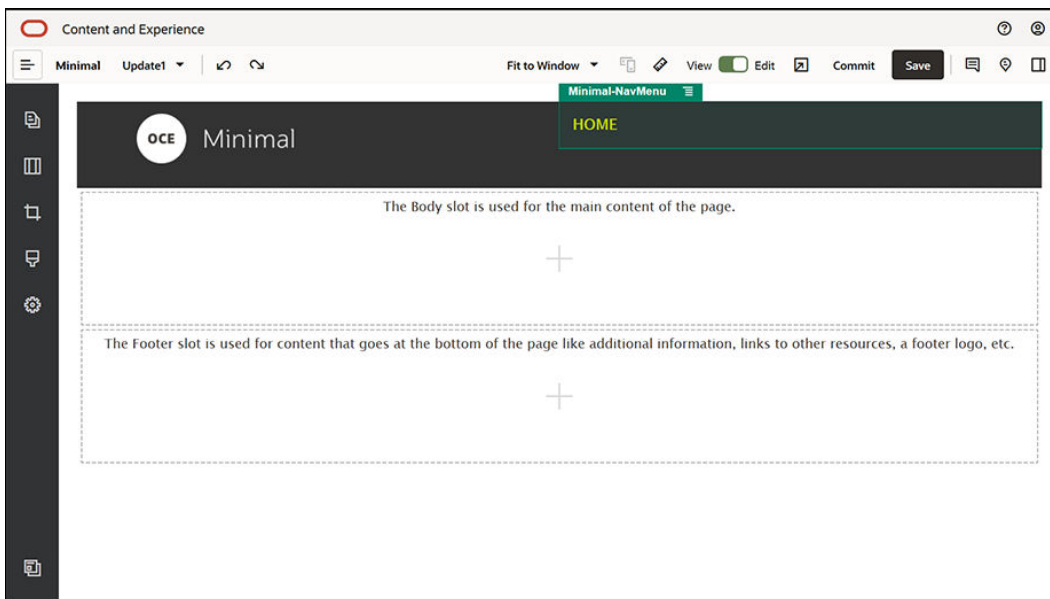




6. 下面将此标识图像链接到主页。在链接选项卡中完成图像组件的设置。


属性	值
选择链接类型	站点页
页	主页
目标	在相同的窗口中打开

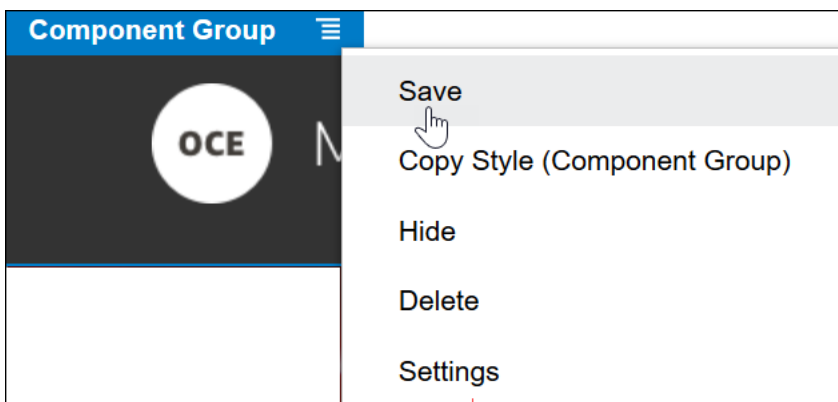
7. 在左侧子工具栏中，单击 ，然后单击定制以显示定制组件的列表。
8. 现在，我们来使用 Minimal-NavMenu 定制组件向主页中添加导航菜单。将 Minimal-NavMenu 组件拖放到组件组中图像组件右侧。单击 Minimal-NavMenu 组件的标题以确保其父项是先前添加的组件组。可使用此方式查看任何组件在 Web 页结构中的位置。



在一般信息选项卡中完成相关设置。

属性	值
对齐方式	右对齐
上对齐	1.2vw
下对齐	0
左对齐	0
右对齐	6vw

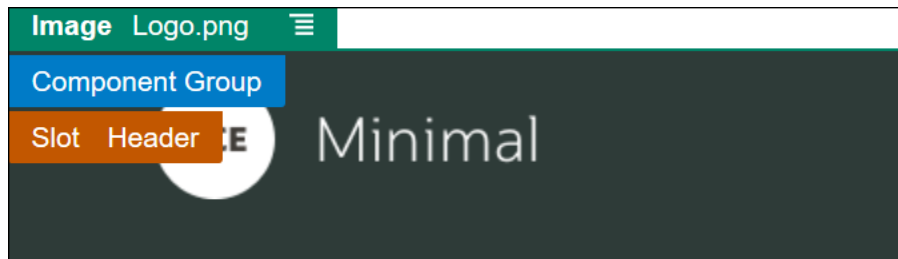
- 现在，页眉已就绪。我们将此组件组保存为定制组件组，以便以后可以在其他网站页面中使用它。依次单击组件组的标题、它的菜单图标  和保存。在“保存组件组”对话框中的名称字段中，输入 "Minimal-Header"，然后单击保存。



 **提示:**

如果您单击组件组时看到的是图像或 Minimal-NavMenu 组件标题（而不是组件组标题），再次单击图像或 Minimal-NavMenu 组件标题，将会看到组件组的标题。现在，可以单击组件组的标题并执行上述步骤。



下图显示了“页眉”内容框中图像组件的父结构：

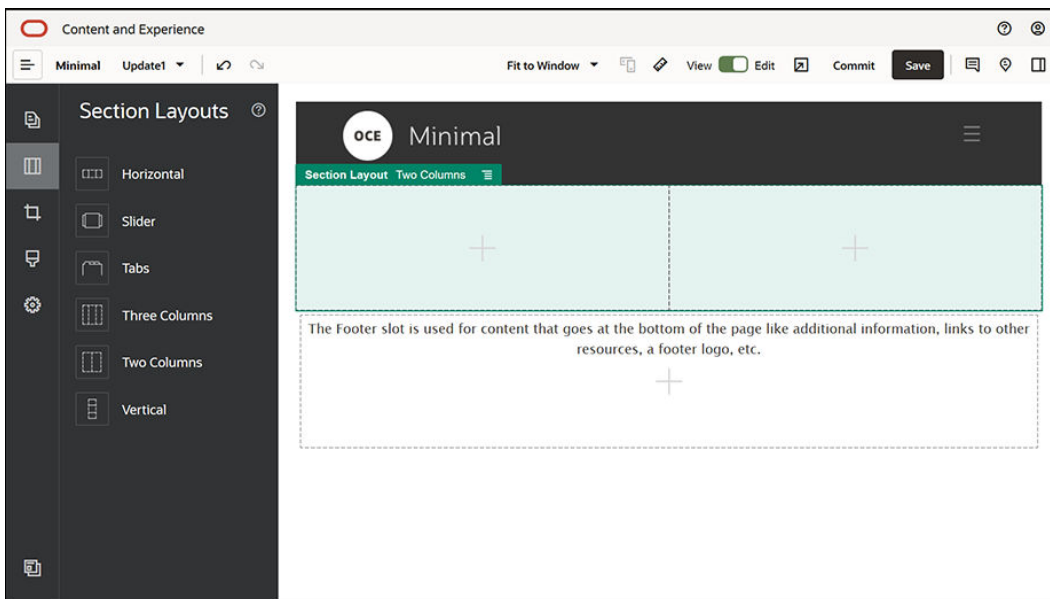


10. 单击站点构建器右上方的保存以保存所做更改。现在，“页眉”内容框看上去应如下图所示所示：



下面转到“正文”内容框：

1. 在左侧子工具栏中，依次单击  和已植入。
2. 从左侧子工具栏中，将组件组拖放到“正文”内容框中。我们将使用此组件组以及要添加到其中的组件创建横幅。
3. 在左侧子工具栏中，单击 .
4. 将“两列”部分布局拖放到组件组中。




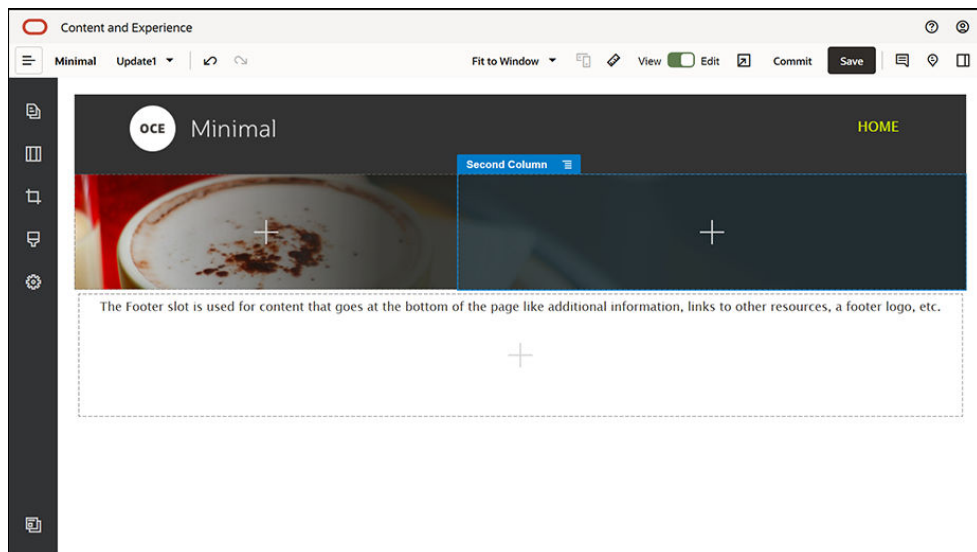
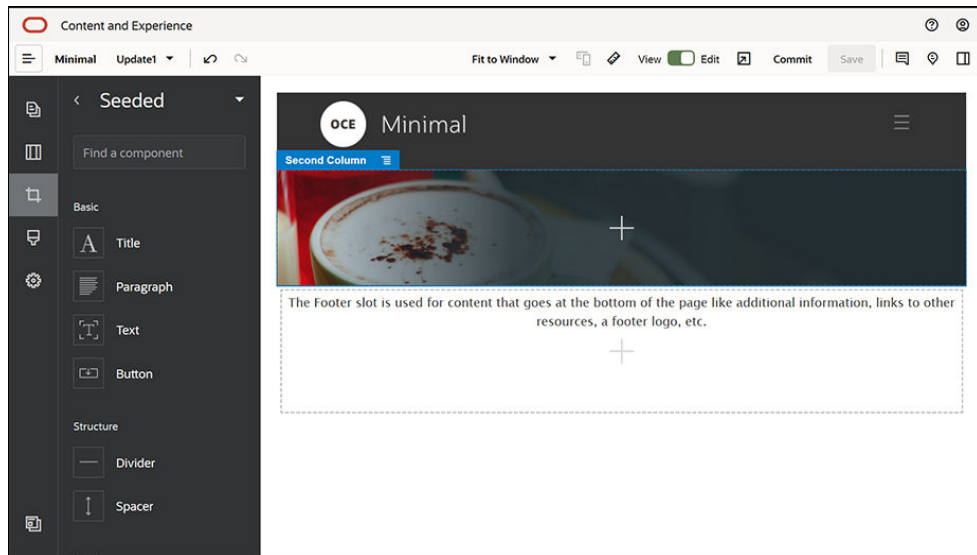
5. 在一般信息选项卡中完成部分布局的设置。单击定制设置以指定以下设置。

属性	值
第一列宽度 (%)	43
第二列宽度 (%)	57
响应断点 (像素)	1,023
响应行为	隐藏第一列

在背景选项卡中完成设置：

属性	值
图像	Minimal-Images 文件夹中的 Banner1.jpg
位置	中心居中
缩放	拉伸

6. 在左侧子工具栏中，单击  以查看植入组件的列表。
7. 从植入组件列表中，将标题组件拖放到“两列”布局的第二列中。



- 在标题组件中单击并输入 "WELCOME TO THE REVOLUTION"。选择该文本，然后在文本编辑器中将其字体颜色设置为白色。在一般信息选项卡中完成标题组件的设置。

属性	值
上对齐	6vw
下对齐	1.8vw
左对齐	6vw
右对齐	6vw

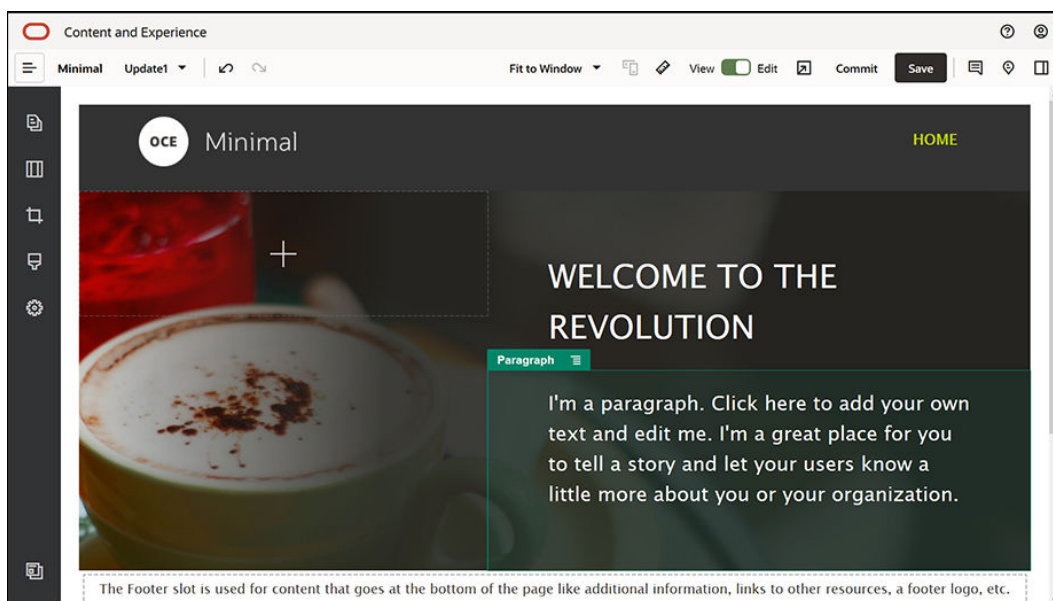
- 从左侧子工具栏中的植入组件列表中，将段落组件拖放到标题组件（位于“两列”布局的第二列中）下方。在一般信息选项卡中完成相关设置。


属性	值
上对齐	1.8vw
下对齐	6vw
左对齐	6vw
右对齐	6vw

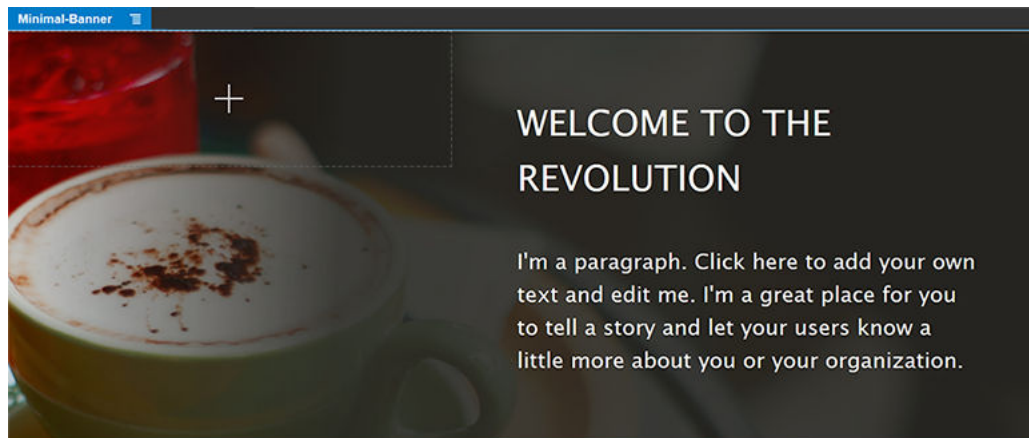
10. 在段落组件中单击并输入以下文本:

"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."

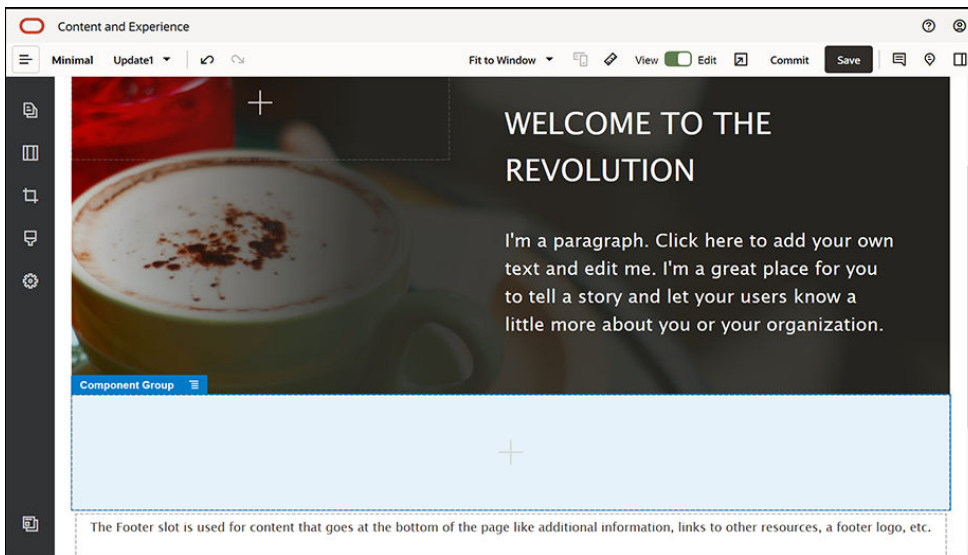
选择该文本, 然后在文本编辑器中将其大小设置为 24。另外, 在文本编辑器中将其字体颜色设置为白色。



11. 现在, 横幅已就绪。我们将此组件组保存为定制组件组, 以便以后可以在其他网站页面中使用它。依次单击组件组的菜单图标  和保存。在“保存组件组”对话框中的名称字段中, 输入 "Minimal-Banner", 然后单击保存。您将注意到现在显示组件组的名称 (Minimal-Banner)。

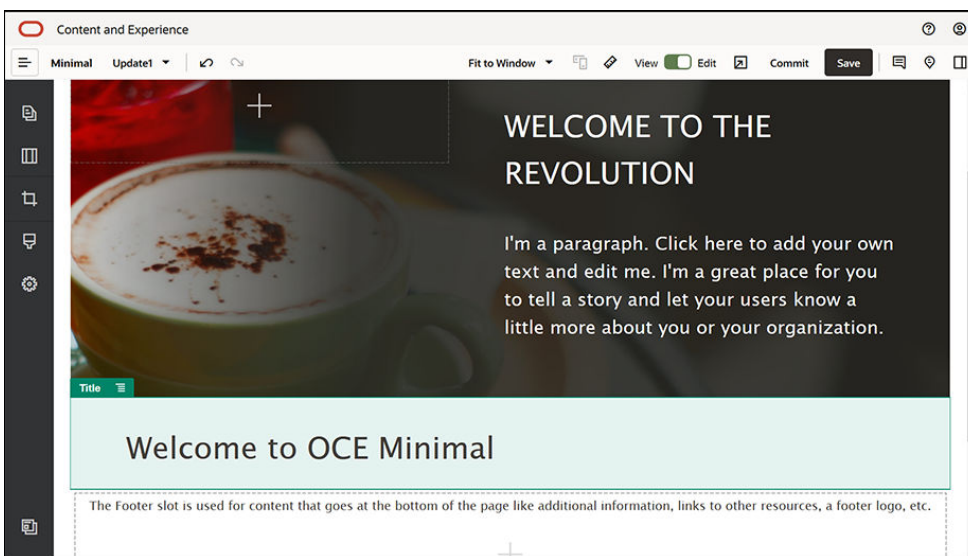


12. 从左侧子工具栏中，将另一个组件组拖放到“正文”内容框中已添加的 Minimal-Banner 组件组下方。



13. 将标题组件拖放到组件组中。
14. 在标题组件中单击并输入 "Welcome to OCE Minimal".
15. 在一般信息选项卡中完成标题组件的设置。

属性	值
上对齐	3vw
下对齐	1.8vw
左对齐	6vw
右对齐	6vw



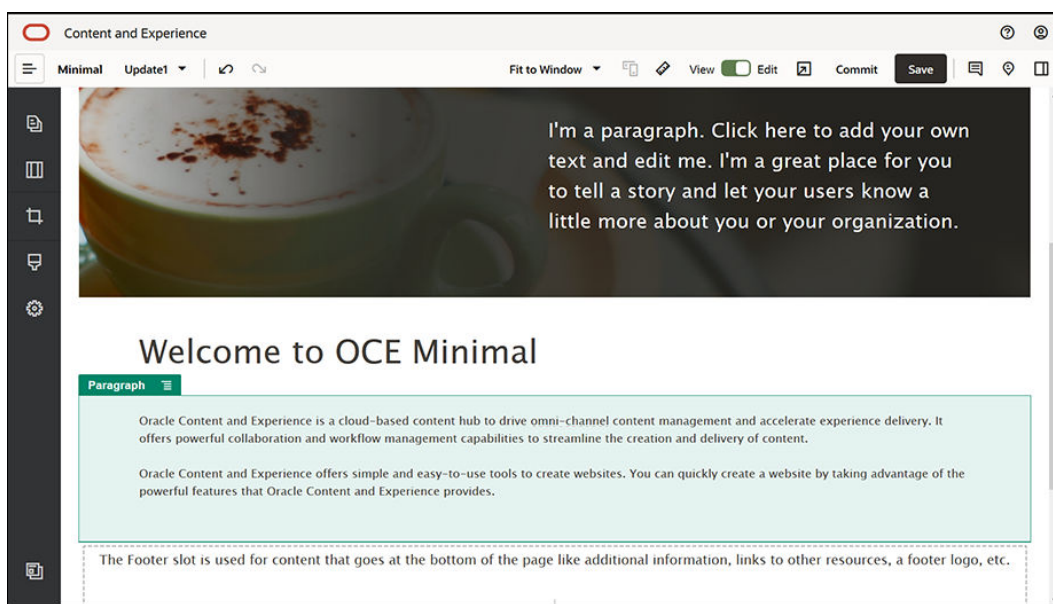
16. 从左侧子工具栏中，将段落组件拖放到组件组中的标题组件下方。在段落组件中单击并输入以下文本：


"Oracle Content Management is a cloud-based content hub to drive omni-channel content management and accelerate experience delivery.It offers powerful collaboration and workflow management capabilities to streamline the creation and delivery of content."

"Oracle Content Management offers simple and easy-to-use tools to create websites.You can quickly create a website by taking advantage of the powerful features that Oracle Content Management provides."

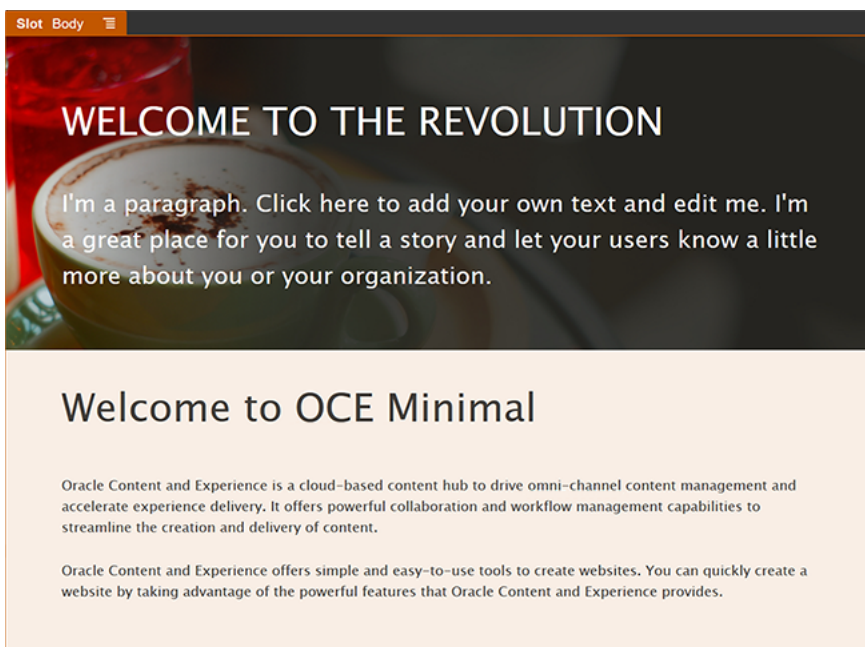
17. 在一般信息选项卡中完成段落组件的设置。

属性	值
上对齐	20px
下对齐	50px
左对齐	6vw
右对齐	6vw



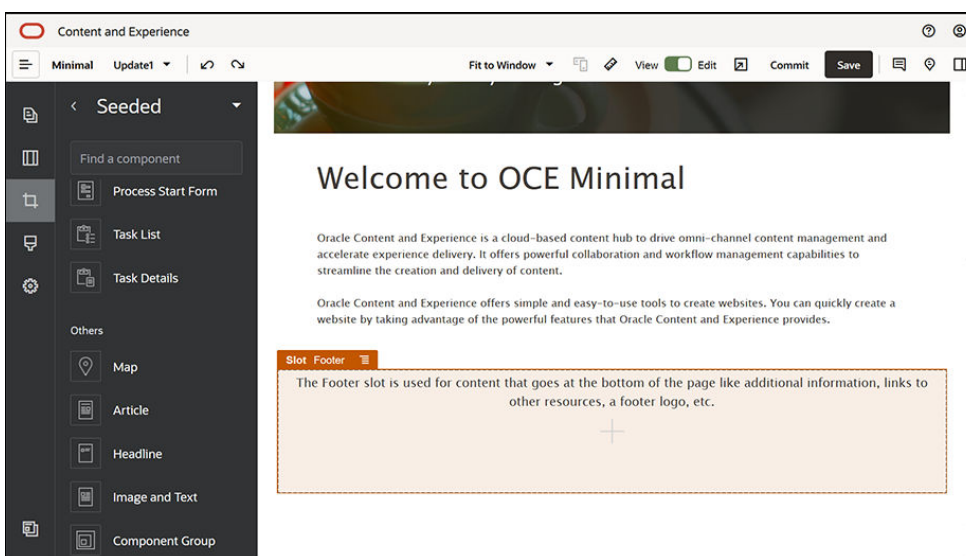
18. 我们完成了“正文”内容框。我们将组件组保存为定制组件组，以便以后可以在其他网站页面中使用它。依次单击组件组的菜单图标  和保存。在“保存组件组”对话框中的名称字段中，输入 "Minimal-Body"，然后单击保存。

19. 单击站点构建器右上方的保存以保存所做更改。现在，“正文”内容框看上去应如下图所示：



我们来完成“页脚”内容框：

1. 从左侧子工具栏中，将组件组拖放到“页脚”内容框中。在组件组的设置中，将颜色字段设置为 #333333。

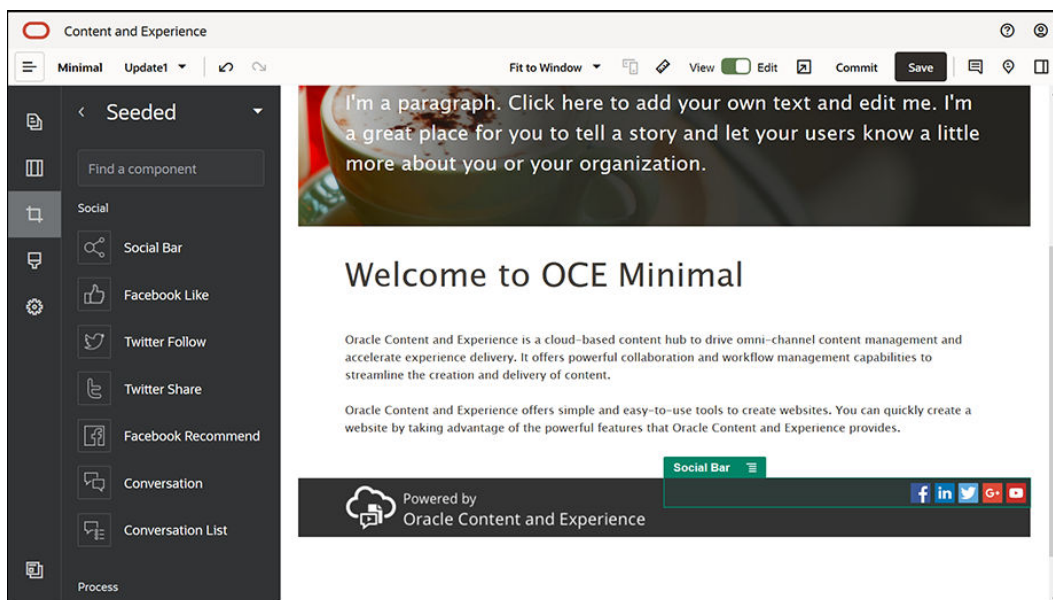


2. 将图像组件拖放到组件组中，并在一般信息选项卡中完成其设置。

属性	值
选择	Minimal-Images 文件夹中的 Powered_by_OCE.png
对齐方式	左对齐
宽度	取消选择设置宽度
上对齐	0.9vw

属性	值
下对齐	0.9vw
左对齐	6vw
右对齐	0

3. 从左侧子工具栏中，将社交栏组件拖放到组件组中图像组件右侧。




在一般信息选项卡中完成社交栏组件的设置。

属性	值
上对齐	1.8vw
下对齐	1.8vw
左对齐	0.3vw
右对齐	6vw


在一般信息选项卡中，依次图标和图标名称以完成设置。

属性	值
URL	<ul style="list-style-type: none"> • https://www.facebook.com/Oracle/ (对于 Facebook) • https://www.linkedin.com/company/oracle/ (对于 LinkedIn) • https://twitter.com/Oracle (对于 Twitter) • https://www.youtube.com/oracle/ (对于 YouTube)
目标	在新窗口中打开

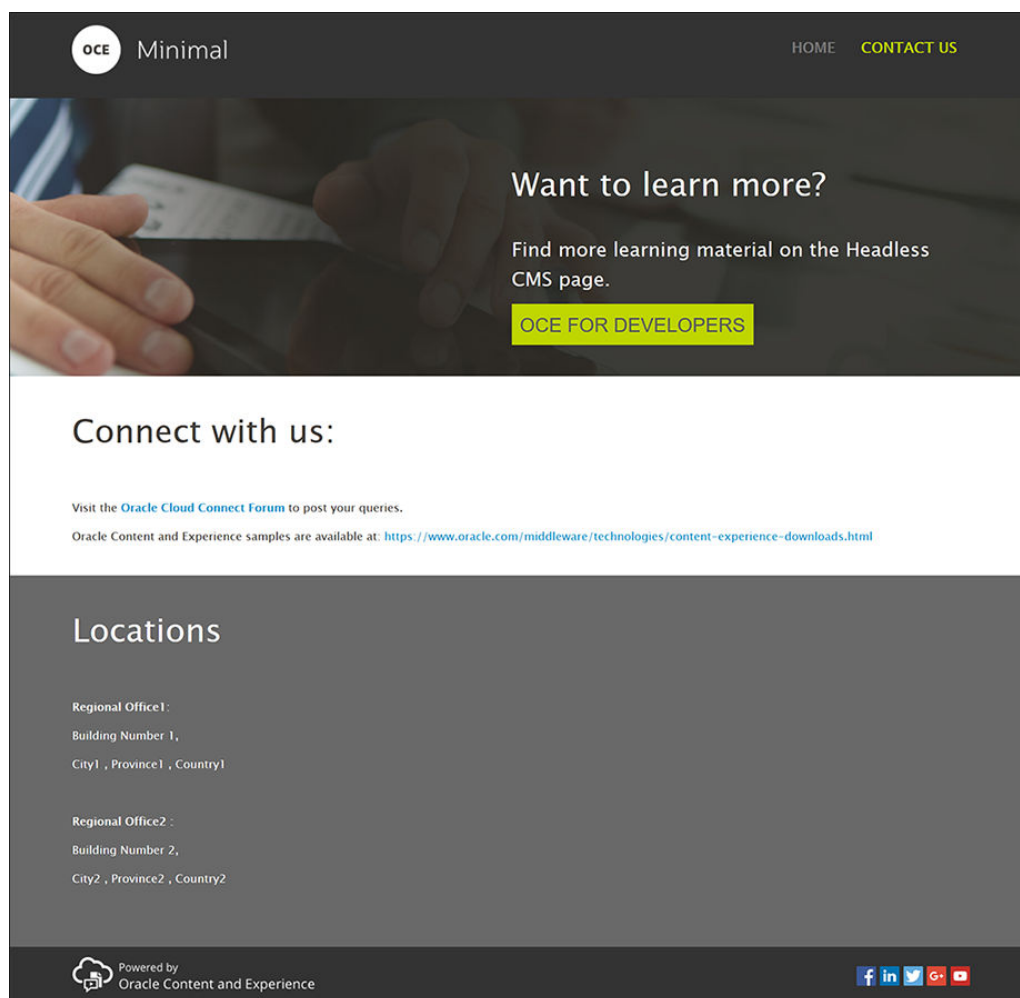
4. 现在，页脚已就绪。我们将此组件组保存为定制组件组，以便以后可以在其他网站页面中使用它。依次单击组件组的菜单图标  和保存。在“保存组件组”对话框中的名称字段中，输入 "Minimal-Footer"，然后单击保存。组件组看上去应如下所示：





5. 单击站点构建器右上方的保存以保存所做更改。

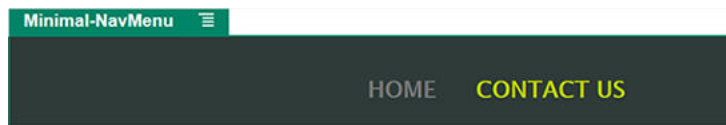
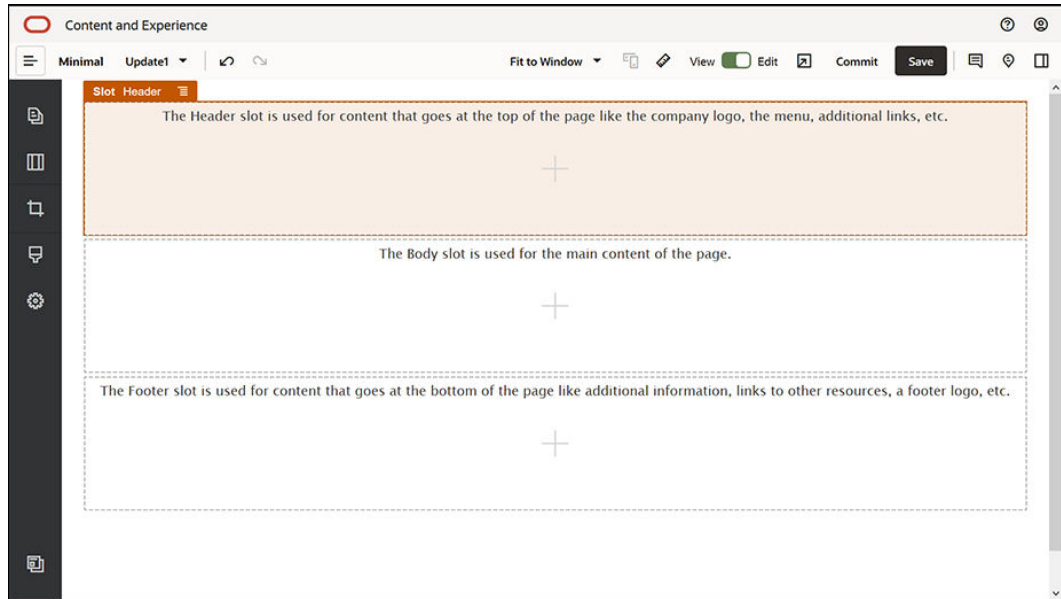
单击站点构建器右上方的  来预览网站的第一个页面。现在，该网站仍未发布，因此其他人现在无法查看。

您已完成了 "HOME" (主页) 的创建。下面构建 "CONTACT US" (联系我们) 页。下面是已完成的联系页的外观：

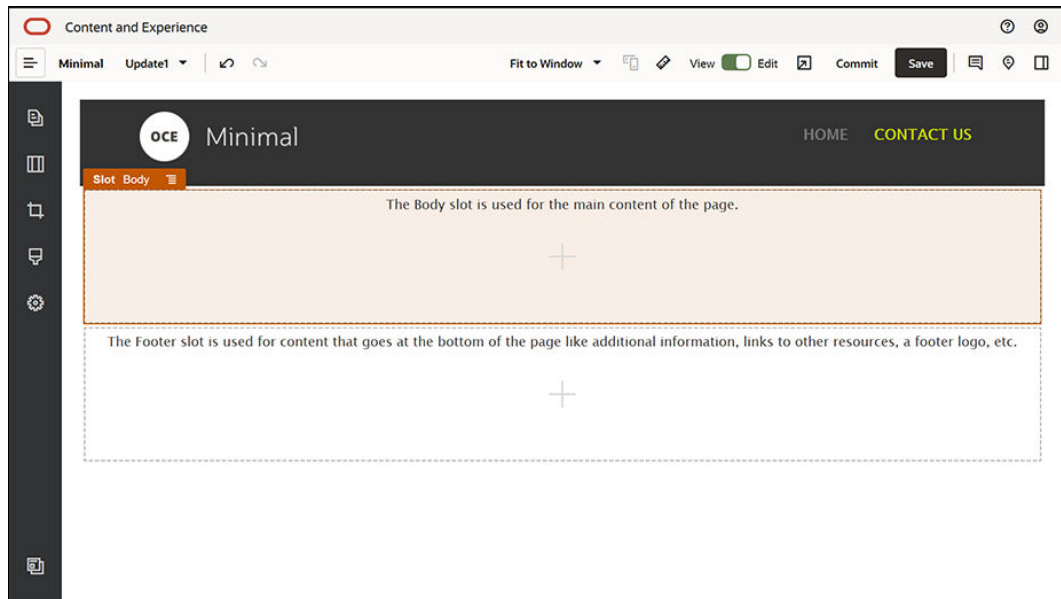


我们在各个内容框中添加组件：

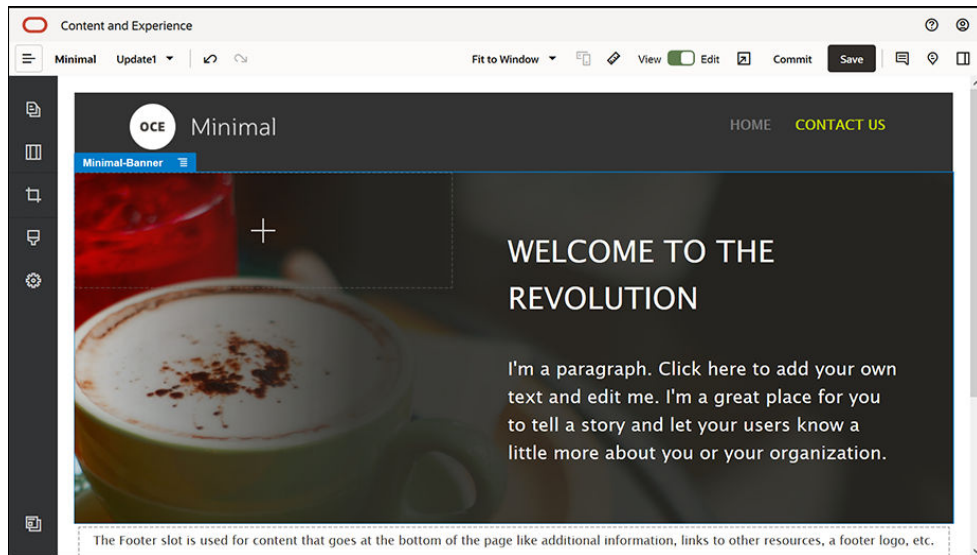
1. 在左侧子工具栏中，依次单击  和添加页。
2. 在页名字段中输入 "CONTACT US" (联系我们)，然后单击关闭。即向网站添加了一个新页。
3. 在左侧子工具栏中，依次单击  和定制。
4. 将 (先前创建并保存的) Minimal-Header 组件拖放到“页眉”内容框中。



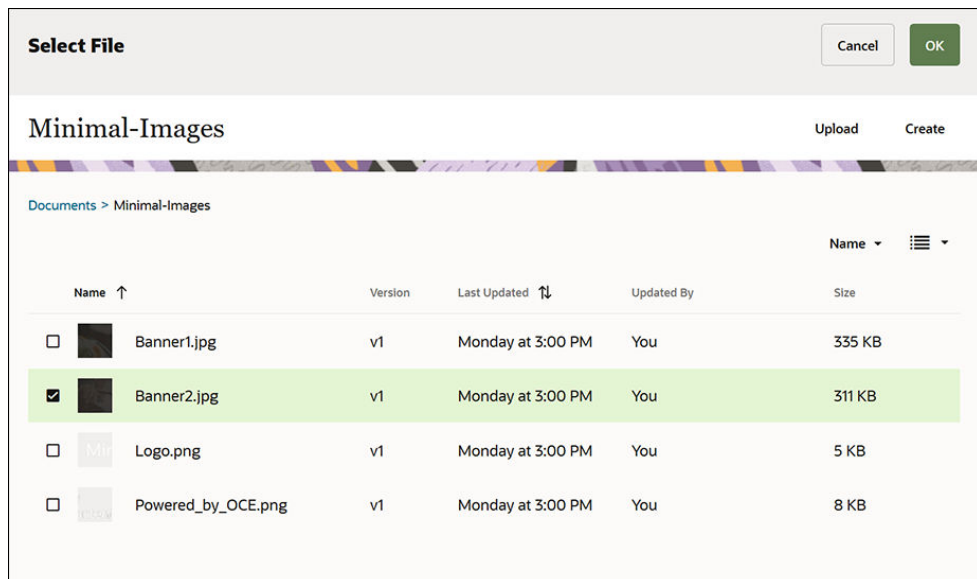
请注意，Minimal-NavMenu 组件已自动选取刚才新建的 "CONTACT US"（联系我们）页。




5. 从左侧子工具栏中，将 Minimal-Banner 组件拖放到“正文”内容框中。



6. 我们来修改横幅，使其看上去与 "HOME"（主页）的横幅不同，并适合 "CONTACT US"（联系我们）页。在组件组中，修改“两列”部分布局的设置：
在背景选项卡中的图像字段中，单击选择图像，然后从您先前在文档中创建的 **Minimal-Images** 文件夹中选择 **Banner2.jpg**。



7. 在“两列”部分布局中，修改标题组件和段落组件中的文本。
 - 在标题组件中输入 "Want to learn more?"。
 - 在段落组件中输入 "Find more learning material on the Headless CMS page."。
8. 在左侧子工具栏中，依次单击  和已植入。
9. "CONTACT US"（联系我们）页的图像有一个名为 **OCE FOR DEVELOPERS**（面向开发人员的 OCE）的按钮作为横幅的一部分，因此，我们将此按钮添加到横幅

中。在 Minimal-Banner 组件组中，将按钮组件拖放到“两列”部分布局的第二列中（位于最近添加的段落组件下方）。在一般信息选项卡中完成按钮组件的设置。

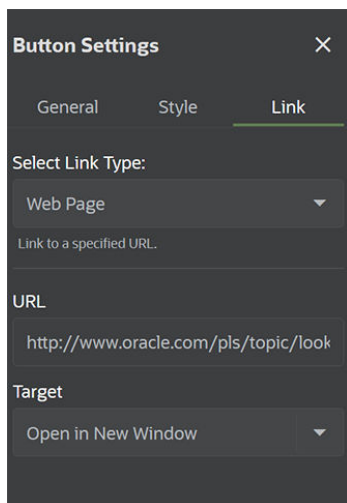
属性	值
标签	OCE FOR DEVELOPERS
上对齐	0.3vw
下对齐	3vw
左对齐	6vw
右对齐	0.3vw

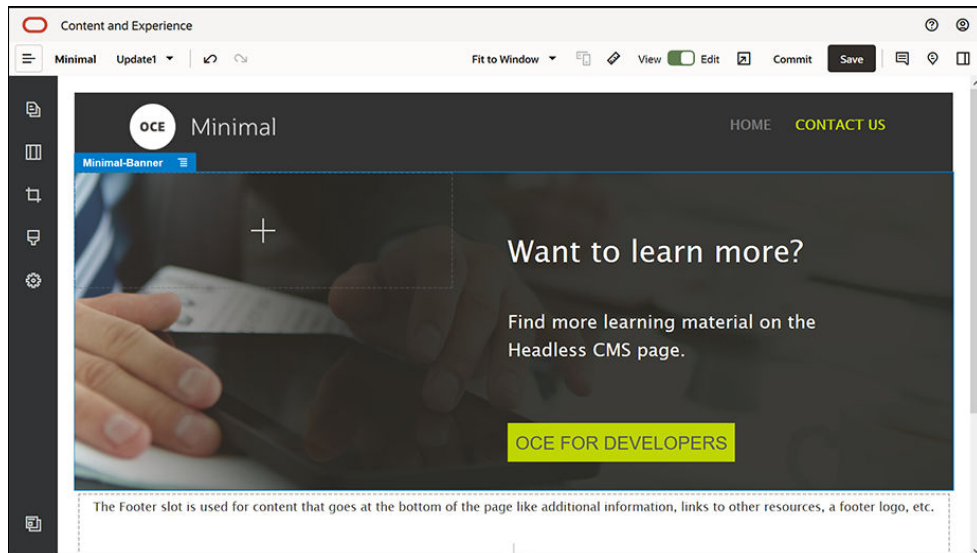
在样式选项卡中，选择定制并完成设置。


属性	值
背景颜色	#c0d600
字体	<ul style="list-style-type: none"> 输入 24 作为大小。 输入颜色 #58595b。
边框	无
悬停颜色	<ul style="list-style-type: none"> 将背景设置为 #e1fa00。 将字体设置为 #58595b。 将边框设置为 #2222dd。
角	0

在链接选项卡中完成设置。


属性	值
选择链接类型	Web 页
URL	http://www.oracle.com/pls/topic/lookup?ctx=cloud&id=content-cloud-headless
目标	在新窗口中打开





10. 在左侧子工具栏中, 依次单击  和定制。
11. 将 Minimal-Body 组件拖放到“正文”内容框中的 Minimal-Banner 组件下方。
12. 在 Minimal-Body 组件中, 修改标题组件和段落组件中的文本。
 - a. 在标题组件中输入 "Connect with us:"。
 - b. 在段落组件中输入详细信息 (例如, 电子邮件地址和与支持相关的其他链接) :
"Visit the [Oracle Cloud Connect Forum](#) to post your queries."
"Oracle Content Management samples are available at: <https://www.oracle.com/middleware/technologies/content-experience-downloads.html>"
13. 从左侧子工具栏中, 将另一个 Minimal-Body 组件拖放到“正文”内容框中先前添加的 Minimal-Body 组件下方。
14. 在组件组设置中的背景选项卡中, 将颜色字段设置为 #696969。
15. 在 Minimal-Body 组件中, 修改标题组件和段落组件。
 - a. 在标题组件中输入 "Locations"。选择该文本, 然后在文本编辑器中将其字体颜色设置为白色。
 - b. 在段落组件中输入以下文本。
"Regional Office1:
Building Number 1,
City1, Province1, Country1
Regional Office2:
Building Number 2,
City2, Province2, Country2"
选择该文本, 然后在文本编辑器中将其字体颜色设置为白色。
16. 现在, 我们向 "CONTACT US" (联系我们) 页添加页脚。将 Minimal-Footer 组件拖放到“页脚”内容框中。

17. 单击站点构建器右上方的保存以保存所做更改。

单击站点构建器右上方的  来预览网站，确保一切看上去正常。应确保单击 "CONTACT US" (联系我们) 页上的标识图像时会返回主页。应测试菜单以确保可在网站页面之间正确导航。

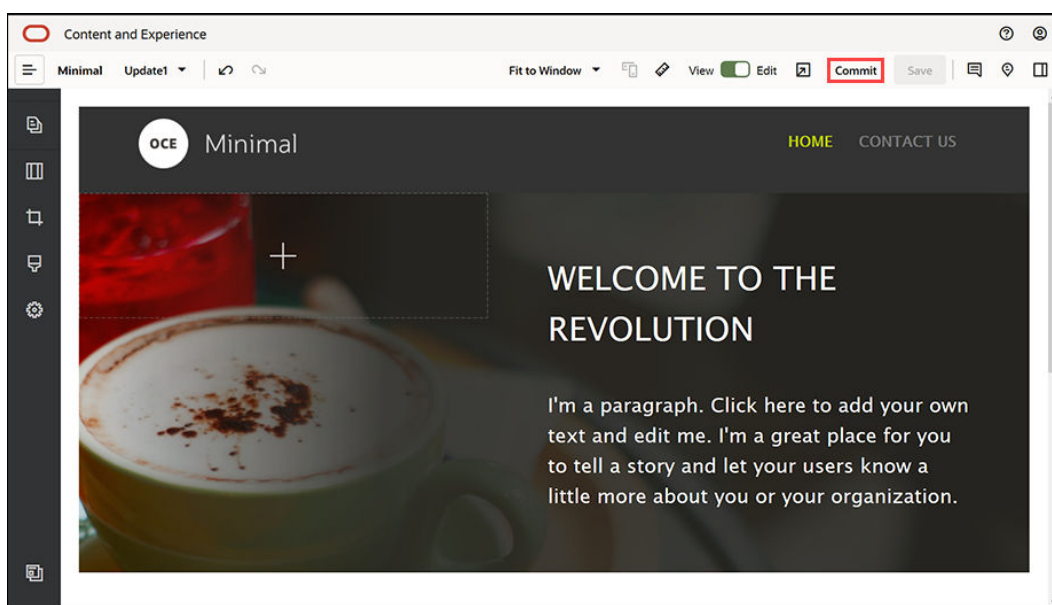
您的网站现已就绪，可以发布了。

[下一步：发布网站](#)

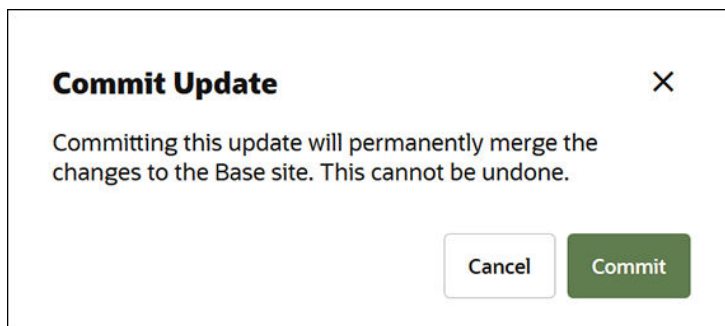
步骤 3: 发布网站

现在，您已成功创建了您的网站，可以将其发布，使其联机以供您的用户查看。

1. 一切看上去正常后，可以单击站点构建器右上方的提交来提交对基本网站所做更改。



2. 在“提交更新”对话框中，单击提交。

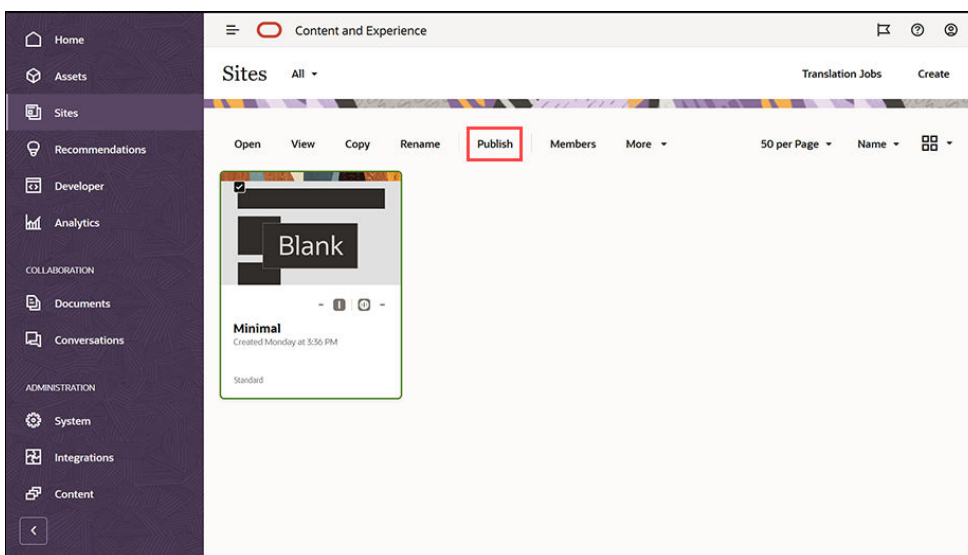


3. 提交所做更改后，该网站就已就绪，可以发布。

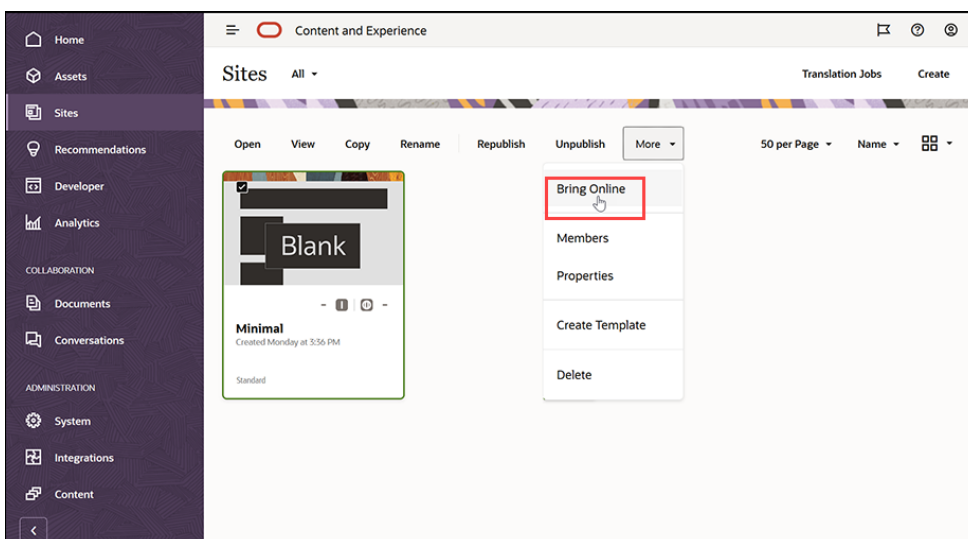
4. 关闭站点构建器。

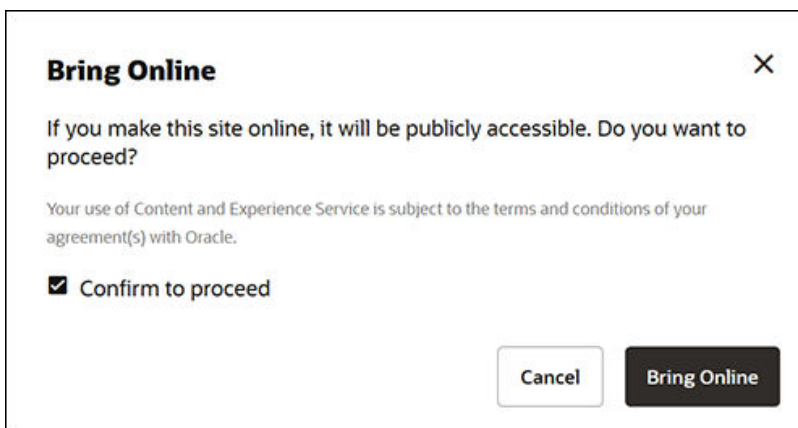
5. 单击 Oracle Content Management Web 界面中左侧导航菜单中的站点，然后选择该网站。

6. 从菜单栏或右键单击菜单中选择发布。

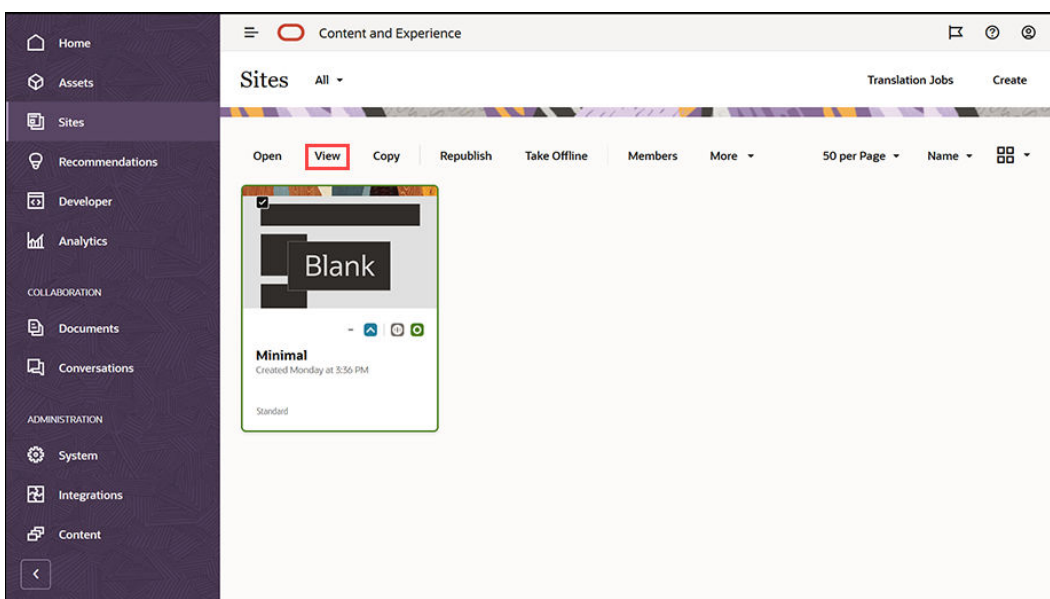


7. 发布网站后，页面顶部会在显示一则通知，提示已发布网站。
8. 在“站点”页上，再次选择该网站，然后从菜单栏或右键单击菜单中选择联机。在“联机”对话框中，选择确认以继续，然后单击联机。





9. 大功告成！您的网站已联机，可供其他人查看了。
要查看您的公共网站，请从菜单栏或右键单击菜单中选择查看。



执行更多操作

您可以根据组织的需求定制网站。下面提供了一些可帮助您入门的有用链接：

- [使用样式和格式设置](#)
- [编辑提示和技巧](#)
- [管理站点和站点设置](#)
- [上传站点文件](#)
- [添加页](#)
- [移动页](#)
- [排列页面内容](#)

- 更改页设置
- 更改页布局
- 使用表
- 设置搜索引擎属性

第 II 部分

创建和编辑站点

此部分详细介绍了如何使用模板和主题开始创建站点，以及如何使用组件和布局来组织和添加内容，从而编辑站点。其中包括以下章节：

- [创建站点](#)
- [编辑站点](#)
- [在站点中使用模板和主题](#)
- [管理定制组件和布局](#)
- [使用站点页](#)
- [排列页面内容](#)
- [使用内置组件](#)

4

创建站点

具有适当权限的任何人都能够使用 Oracle Content Management 构建网站。您无需使用任何专有工具、代码或软件。采用友好直观的图形用户界面。

在创建站点时，可以从模板开始。模板包含您在开始构建站点时需要的全部内容，包括站点代码框架、带有示例页和内容的默认站点、具有样式设置的主题以及图像等资源，甚至包括定制组件。请参见[了解站点创建流程](#)。

Oracle Content Management 的站点管理功能可简化并加速向业务用户提供体验的过程，同时让 IT 部门可以轻松地从某个集中位置控制和跟踪体验，同时能够完全管理体验的整个生命周期，从而降低了创建和维护公司所需的每个新体验的成本。管理功能内置在 Oracle Content Management 核心中，只需管理员启用即可。请参见 *Administering Oracle Content Management* 中的“配置站点和资产设置”。

- [了解站点创建流程](#)
- [创建站点](#)
- [复制站点](#)
- [管理站点请求](#)
- [更改站点请求详细信息](#)
- [查看站点请求策略](#)

创建站点

要创建站点，请选择一个模板，命名站点，然后添加内容。如果启用了管理，必须先提交站点请求，在其获得批准后才能添加内容。

管理员必须先启用站点创建并使模板可供您使用，然后您才能创建站点。如果您在站点页上未看到创建选项，或者没有任何模板可用，请与管理员联系。请参见[站点入门](#)和[了解站点管理](#)。

如果您要创建企业站点（支持使用资产和多语言站点），则该站点将与资料档案库、本地化策略和（可能）默认语言关联。如果没有可用的资料档案库和本地化策略，请创建它们，或者请内容管理员创建它们。



注：

在 Oracle Content Management 入门版中，翻译不可用且无法启用监管，只能创建一个站点。要利用完整的功能集并消除站点限制，请升级到 Oracle Content Management 高级版。

1. 单击站点。
2. 单击创建。
3. 在选择模板页上，为您的站点选择模板。

- 如果启用了管理，该模板将确定您创建的站点是标准站点（不使用资产资料档案库或本地化策略）还是企业站点（使用资产资料档案库和本地化策略）。
4. 在配置站点页上，您将看到在创建站点之前需要什么样的审批，站点所需的最低安全级别，以及站点页面的图像。如果限制为只有特定人员才能进行审批，请单击显示审批者以查看谁可以批准您的站点请求。
- 如果未启用管理，请为要创建的站点类型选择标准或企业。
 - 如果是创建标准站点，请单击下一步转至下一页。
 - 如果是创建企业站点，请完成以下步骤：
 - a. 选择用于管理站点内容的默认资料档案库。如果您是资料档案库管理员，而且没有可用的资料档案库，或者您希望使用新的资料档案库，请选择创建新资料档案库并按照步骤创建资料档案库。有关在站点中使用多个资料档案库的信息，请参见《通过 Oracle Content Management 管理资产》中的 授予站点访问多个资料档案库的权限。
 - b. 为站点选择本地化策略。如果您是资料档案库管理员，而且没有可用的策略，或者您希望使用新策略，请选择创建新的本地化策略并按照步骤创建策略。
 - c. 如果没有为模板的本地化策略设置默认语言，现在请选择一种语言。您只会看到本地化策略所需的语言。
 - d. 单击下一步可转到下一页。
5. 在添加详细信息页上，输入以下信息。
- 输入站点的名称。站点 URL 中将使用此名称。可以使用字母、数字、下划线 (_) 和连字符 (-)。URL 区分大小写。如果输入空格，则使用连字符自动取代空格。不要使用具有不同大写形式的现有站点名称。例如，如果存在名为 ABC 的站点，则不要创建名为 Abc 的另一个站点。

 **注：**

不要将以下名称用于模板、主题、组件、站点或站点页：authsite、content、pages、scstemplate_*、_comps、_components、_compsdelivery、_idcservice、_sitescloud、_sitesclouddelivery、_themes、_themesdelivery。下面的名称虽然可用于站点页，但不要将其用于模板、主题、组件或站点：documents、sites。

- 如果是创建企业站点，模板策略可能允许您编辑友好 URL 值的默认站点前缀。此前缀将附加到内容项缩略名值（特定于页或资产的 URL 的一部分）。
 - （可选）输入站点的说明。
 - （可选）输入此站点请求的理由。这将帮助站点管理员确定是否应批准请求。
6. 单击完成。

如果站点请求需要审批，则会向需要对其进行审批的人员发送电子邮件。如果您的站点请求得到批准，您将收到一封电子邮件，通知您该站点请求已得到批准而且将创建站点。如果您的请求遭到拒绝，您将收到一封电子邮件，通知您该请求已遭到拒绝。您可以在站点页上查看您的待处理请求，方法是在筛选器菜单中选择请求。如果该请求遭到拒绝，您应查看有关其被拒绝的原因的消息，以便您可以更正问题并再次提交请求。

如果站点请求设置为自动批准，则将自动创建站点。进度栏显示新站点名称和创建状态。创建站点后，其名称将显示在站点列表中，并且其处于脱机状态。

如果您创建了企业站点，则会在您选择的资料档案库中创建对应的站点集合。如果您与某个用户共享站点，该用户对关联的集合具有相同的权限。


创建站点后，您可以共享站点、更改安全性、添加和编辑内容、发布站点以及使站点联机。请参见[管理站点和站点设置](#)和[编辑站点](#)。

复制站点

您可以复制站点，以便有个好的开头。原始站点上的所有内容（包括主题、所有未完成更新、页、页内容、推荐以及所有其他资产（例如图像和策略））都将复制到您提供的新名称下的新站点中。

注：

如果您要复制使用来自多个资料档案库的内容的站点，您需要使用 OCE 工具包来完成。请参见[使用 OCE 工具包进行开发](#)和[使用 cec 命令行实用程序](#)

1. 单击站点。
2. 选择要复制的站点，然后从右键单击菜单中选择复制，或单击操作栏中的 。
3. 在配置站点页上，您将看到在创建站点之前需要什么样的审批，站点所需的最低安全级别，以及站点页面的图像。请完成以下步骤：
 - a. 在复制下拉列表中，选择是只复制基本站点还是包括更新。如果包括更新，复制的站点将包括原始站点的未完成更新。如果要创建标准站点，请单击箭头转至下一页，然后跳至步骤 4。
 - b. 如果要复制企业站点，则根据原始站点设置默认资产资料档案库、本地化策略、必需语言以及默认语言。还可以选择其他资料档案库来管理站点的内容（可选）。
 - c. 单击下一步可转到下一页。
4. 在添加详细信息页上，输入以下信息。
 - 输入站点的名称。站点 URL 中将使用此名称。可以使用字母、数字、下划线 (_) 和连字符 (-)。URL 区分大小写。如果输入空格，则使用连字符自动取代空格。不要使用具有不同大写形式的现有站点名称。例如，如果存在名为 *ABC* 的站点，则不要创建名为 *Abc* 的另一个站点。

注：

不要将以下名称用于模板、主题、组件、站点或站点页：authsite、content、pages、scstemplate_*、_comps、_components、_compsdelivery、_idcservice、_sitescloud、_sitesclouddelivery、_themes、_themesdelivery。下面的名称虽然可用于站点页，但不要将其用于模板、主题、组件或站点：documents、sites。

- 如果是复制企业站点，还可以编辑友好 URL 值的默认站点前缀（可选）。此前缀将附加到内容项缩略名值（特定于页或资产的 URL 的一部分）。
- 输入站点的可选说明。
- 输入此站点请求的可选理由。这将帮助站点管理员确定是否应批准请求。

5. 单击完成。

如果站点请求需要审批，则站点将在站点管理员批准后创建。您可以在站点页上查看您的待处理请求，方法是在筛选器菜单中选择请求。如果您的请求被拒绝，您应查看有关其被拒绝的原因的消息，以便您可以更正问题并再次提交请求。

如果站点请求设置为自动批准，则将自动创建站点。进度栏显示新站点名称和创建状态。创建站点后，其名称将显示在站点列表中，并且其处于脱机状态。

如果您复制了企业站点，则会在您选择的资料档案库中创建对应的站点集合。如果您与某个用户共享站点，该用户对关联的集合具有相同的权限。

创建站点后，您可以共享站点、更改安全性、添加和编辑内容、发布站点以及使站点联机。如果您要共享有权访问多个资料档案库的站点，只会共享默认资料档案库。任何其他资料档案库都需要单独共享。请参见[管理站点和站点设置](#)和[编辑站点](#)。

管理站点请求

如果启用了站点管理，则创建站点可能需要审批。您可以在“站点”页上使用“请求”筛选器查看待处理的站点请求。

注：

Oracle Content Management 入门版无法启用监管。要利用完整的功能集并消除站点限制，请升级到 Oracle Content Management 高级版。

请参见[了解站点管理](#)。

要查看待处理的请求，请在站点页上的筛选器菜单中选择请求。如果您是站点管理员，您可看到自己系统中的所有站点请求，包括需要其他人员审批的请求。如果您是审批者，您会看到您作为审批者的所有站点请求。所有用户都能看到他们自己提交的站点请求。

在请求列表中，您可以查看站点的缩略图、站点名称、请求者、请求时间、站点说明、请求状态以及显示站点是否要求登录的图标。

根据您的角色，您可以执行以下操作：

- 要查看请求的更多详细信息（例如，所需的最低安全性、所使用的模板、站点的可选理由以及缩略图），请单击站点名称或选择请求，然后单击查看。
- 如果您是站点管理员或审批者，您可以通过选择请求，然后单击批准来批准请求。站点在获得批准后将会自动创建。
- 如果您是站点管理员或审批者，您可以通过查看详细信息，然后单击拒绝来拒绝请求。输入拒绝请求的原因，然后单击拒绝。尽管您也可以通过在请求列表中选择请求，然后单击拒绝来拒绝请求，但您无法添加拒绝请求的原因。
- 如果您的请求失败或者遭到拒绝，您可以查看详细信息，根据需要编辑请求，然后重新提交请求。
- 如果您是站点创建者，您可以通过选择请求，然后单击删除来删除请求。



注：

如果站点请求者已被删除，则站点管理员可以删除站点请求。

创建站点后，您可以共享站点、更改安全性、添加和编辑内容、发布站点以及使站点联机。请参见[管理站点和站点设置](#)和[编辑站点](#)。

更改站点请求详细信息

“站点请求详细信息”选项卡显示站点名称、说明、所使用的模板、理由、所有者（发出请求的人）和站点图像。

站点请求的创建者（即站点请求的所有者）可以更新站点请求详细信息。

要查看或更改站点请求详细信息，请执行以下操作：

1. 单击站点，然后在“站点”菜单中单击请求。
2. 单击要查看或编辑的请求。
3. 如果您是审批者，可以批准或拒绝请求。如果您是请求者，则可以根据需要编辑请求信息，然后重新提交请求。

查看站点请求策略

“站点请求策略”选项卡显示站点的类型（标准或企业）、谁可以访问所发布的站点以及在创建站点之前所需的任何审批。对于企业站点，还可以查看资产资料档案库、本地化策略、必需语言以及用于站点的默认语言。

要查看站点请求策略，请执行以下操作：

1. 单击站点，然后在“站点”菜单中单击请求。
2. 单击要查看或编辑的请求，然后单击策略选项卡。
3. 如果您是审批者，可以批准或拒绝请求。如果您是请求者，则可以根据需要更改资产资料档案库，然后重新提交请求。

5

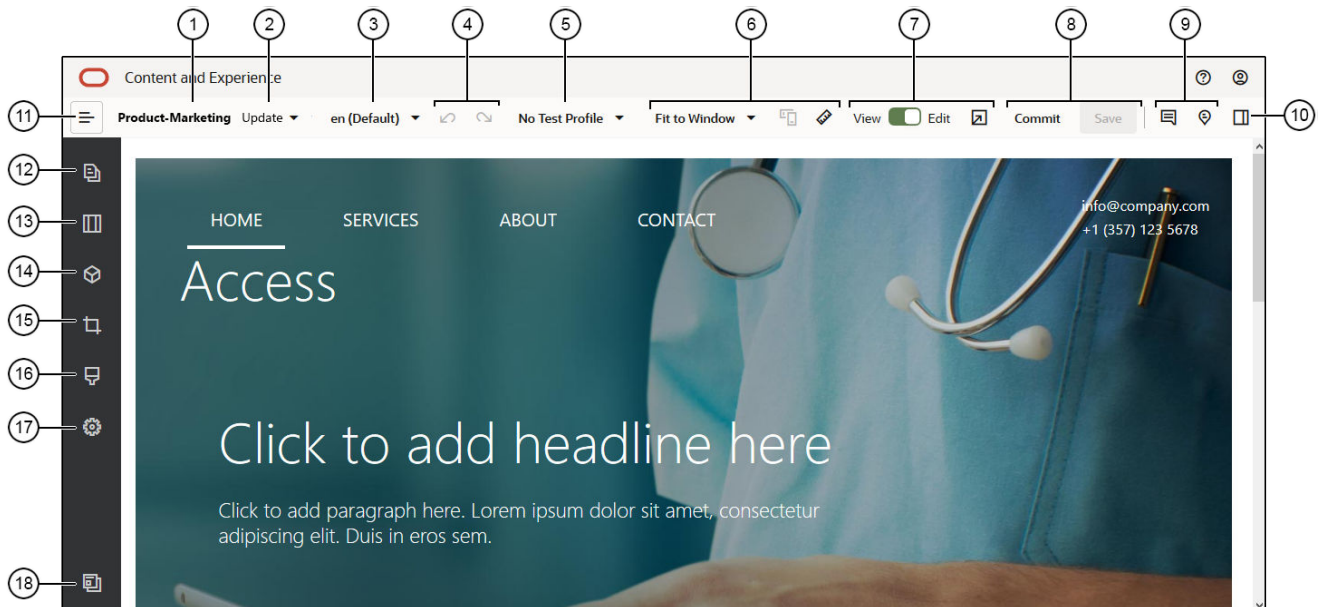
编辑站点

现在来初步了解编辑器以及通过编辑器能够完成的操作。










- 初步了解“站点构建器”页
- 了解站点更新
- 使用更新
- 编辑提示和技巧
- 使用样式和格式设置
- 添加定制站点属性
- 定制站点设置
- 使用表
- 上载站点文件


初步了解“站点构建器”页

为站点编辑现有更新或创建更新时，该更新将在站点构建器中打开。用一点时间来熟悉页的布局以及可用的工具。




下面提供了一些注意事项：







标注	说明
1	列出站点的名称。
2	查看更新以了解当前正在处理的更新的名称。如果有多个更新可用，您可以切换到不同更新。
3	检查语言以查看您当前处理的版本。
4	使用撤消  可还原上次在编辑器中所做的编辑或更改。使用重做  可再次应用最近使用撤消操作还原的更改。您可以多次使用撤消操作，来撤消当前更新中的一系列更改，包括在内容、样式和页组织方面的更改。
5	测试概要信息让您可以选择创建包含各种受众属性的概要信息，并使用概要信息表示不同的站点访问者来测试推荐。测试概要信息是在站点管理员创建建议时设置的。
6	使用预览选项 可查看内容在不同环境下的外观。您可以导航到任意页，并查看应用了来自当前更新中更改的基本站点。单击调整到窗口大小并选择一个尺寸，查看页在具有该屏幕尺寸的设备上的外观。提供了多种大小，您也可以创建自己的设备大小。单击  以查看标记。单击标尺上的间隔可快速查看站点在不同大小下的外观。您还可以选择  以查看站点在移动设备上不同方向时外观如何。具有响应式设计的主题会自动排列页内容，以在所选屏幕大小上实现最佳效果。
<div style="border: 1px solid #0070c0; padding: 10px; background-color: #e6f2ff;"> <p> 注： 在预览模式下，指向站点页的链接不起作用。</p> </div>	
7	使用页面顶部的开关可在查看和编辑之间切换。单击  可预览站点。这显示站点访问者能够看到的页外观，没有内容框和组件边框以及在编辑时使用的其他视觉辅助工具。您可以使用页上的链接在站点中移动，包括导航菜单中的链接、文本中的链接等等。出于安全考虑，指向其他站点的链接会在新窗口中打开。指向当前站点中页的链接将在同一个窗口中打开，也可能在新窗口中打开，具体取决于您指定的链接目标。
8	使用可用的选项来处理更改。单击提交将更改合并到基本站点，或单击保存将更改保存到更新。
9	使用批注控件可添加批注或查看现有批注。 <ul style="list-style-type: none"> 要添加批注，请单击 ，再单击您要添加批注的组件，然后在文本框中输入注释，再单击发布。请参见 <i>Collaborating on Documents with Oracle Content Management</i> 中的“添加批注”。 要查看批注，请单击 。单击批注时，将会在对话中突出显示该批注。
10	单击  可显示（或隐藏）对话面板，可以在其中创建或参与关于此站点的讨论。请参见 <i>Collaborating on Documents with Oracle Content Management</i> 中的“使用对话”。

标注	说明
11	单击  可显示（或隐藏）子工具栏中的工具。

注：

您必须将  设置为编辑，然后才能使用子工具栏中的工具。

单击子工具栏中的图标 (12–18) 可管理页、添加组件、更改主题等等。

标注	单击...	操作
12		编辑和添加页，管理嵌套页结构，重新排列页，以及更改页设置。
13		添加部分布局，例如水平或垂直布局。
14		将资产添加到站点。可以选择创建站点时选择的资料档案库中的任何资产。
15		将组件插入站点。可以从不同类型的组件中选择，例如主题化、定制、植入组件，或者查看所有可用组件。
16		为站点选择其他主题或者更改页或内容框的背景。
17		调整站点的设置，例如搜索引擎优化 (search engine optimization, SEO) 和站点重定向的关键字。
18		返回到“站点”页以管理站点。

了解站点更新

更新是对当前基本站点的指定更改集合。在您提交更改并永久更新基本站点之前，更改会一直保留在更新中。

每次在编辑器中查看或编辑站点时，您会使用某个更新。在编辑器中进行的任何更改是该更新的一部分。您可以有一个或多个更新，并可以随时间继续对更新添加更改。您可以通过更新灵活地管理对站点的编辑。例如，您可以让多个人对站点的不同部分同时进行自己的更新。您可以检查和修改各个更新，在准备好之后，可将更新提交到基本站点。您仍需要发布站点，以使更新可供联机访问。

您可以通过多种方式组织更新：








- 在以页命名的更新中特定于该页的更改
- 在以用户名命名的更新中由该特定用户进行的更改
- 在指定日期或者针对特定项目里程碑进行的更改

 注:

显示该更新中对基本站点的更改的更新。虽然在编辑器中可以很轻松地在可用更新之间切换，不过您一次只能查看一个更新。如果指定页有多个更新，您可能无法知道是否存在对相同内容区域的多个更改。为了防止对页的更改发生冲突，请将单独的更新定位到具体页或站点区域。

提交当前更新中的更改后，将对基本站点进行更改并删除更新。您需要发布站点（使站点联机或重新发布站点），以使这些更改对可访问该站点的任何人都联机可见。

要编辑站点，请执行以下操作：

1. 在列表中选择站点，然后在右键单击菜单中选择打开，或选择操作栏中的 。
2. 编辑器将在预览模式中打开。要进行更改或者使用子工具栏中的导航选项，请确保“编辑”开关  设置为编辑。
3. 如果这是首次对站点进行更新，请输入更新的名称和可选说明，然后单击确定。可以在名称中使用字母、数字、下划线 (_) 和连字符 (-)。如果输入空格，则使用连字符自动取代空格。如果已具有站点的更新，请从列表中选择此更新，然后单击 。
4. 要编辑特定页，请通过使用左侧子工具栏中的站点树、搜索该页或者使用站点自己的导航找到该页。要显示站点树，请单击左侧的 ，然后单击 。
5. 根据需要添加和更改页内容。选择调整到窗口大小以使用布局选项来查看该页在不同大小的不同设备上外观如何。提供了多种大小，您也可以创建自己的设备大小。单击  以查看标记。单击标尺上的间隔可快速查看站点在不同大小下的外观。您还可以选择  以查看站点在移动设备上不同方向时外观如何。
6. 在编辑完站点后，通过下列方式之一来保存更改：
 - 单击保存可将更改保存到当前更新。可以继续使用当前更新或稍后返回到更新。
 - 单击提交可将您的更改应用于基本站点。

您仍需要发布站点，然后您的网站用户才能看到这些更改。请参见[使站点联机或脱机](#)或[发布站点更改](#)。

提交当前更新中的更改后，将对基本站点进行更改并删除更新。每次在编辑器中查看或编辑站点时您都必须使用当前更新，因此您将返回到可以创建新更新站点的列表中。

在更新中，过程是线性的：

- 原始站点（基本站点）已存在。我们可以将它称为版本 1。
- 您创建了更新。合并更新（使用“提交”）时，该更新将永久写入现有站点。现在您的基本站点有了新版本（版本 2）。
- 如果您创建并合并另一个更新，则更新将会永久写入现有站点。现在您的基本站点有了新版本（版本 3）。



处理更新时，请记住：

当前基本站点 + 合并的更新 = 基本站点的新版本

使用更新

每次在编辑器中编辑站点时，您会使用某个更新。在编辑器中进行的任何更改是该更新的一部分。

要创建和使用更新，请执行以下操作：


1. 在“站点”页上，选择站点，然后在右键单击菜单中选择打开，或单击操作栏中的 。
2. 将“编辑”开关  设置为“编辑”。
3. 如果这是首次对站点进行更新，请输入更新的名称和可选说明，然后单击确定。可以使用字母、数字、下划线 (_) 和连字符 (-)。如果输入空格，则使用连字符自动取代空格。要使用现有更新，请在列表中单击该更新。
4. 站点的名称和当前更新将显示在编辑器的顶部栏中。如果您具有多个更新，可从更新列表中选择其他更新来切换更新。
5. 根据需要添加和更改页内容。使用不同的显示大小选项可查看页在不同设备上的外观。
6. 完成编辑后，单击保存。在您准备好将更改与基本站点合并后，即可提交更新。您仍需要发布站点，然后网站用户才能看到这些更改。请参见 [发布站点更改](#)。



编辑提示和技巧

下面提供了一些可帮助您入门的编辑器相关知识。

- [隐藏或显示子工具栏](#)
- [开始](#)
- [管理页](#)
- [拖放编辑](#)
- [组件](#)
- [调整组件的大小和间距](#)
- [可编辑的布局内容](#)
- [样式和格式设置](#)
- [撤消所做更改](#)

隐藏或显示子工具栏

单击  以显示用于管理和编辑页及页内容的选项。













将  设置为编辑以访问子工具栏。在预览页时，单击  可隐藏子工具栏并增大可视区域大小。

开始


要使用站点树选择其他页，请单击子工具栏中的 。使用页列表顶部的搜索框搜索页。您还可以在页本身上使用站点导航或链接。

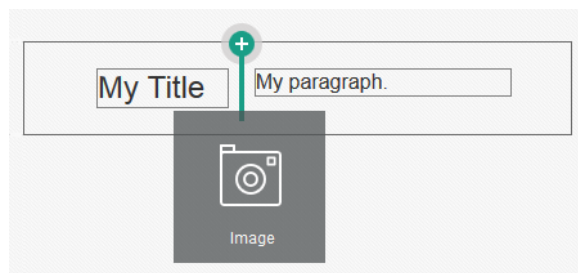
对页进行的任何更改在切换到其他页时保存。您还可以单击保存以将更改保存在当前更新中。

管理页

- 要添加页，请选择要在其中添加页的级别或分支，然后单击添加页。或者，要添加子页，请依次单击  和 。“新页”会添加到站点树的底部，并且将提示您为该页命名并指定其他设置。
- 要删除页，请选中它，然后单击 。
- 要移动页，请选中它，然后将它拖到新位置。或者，可以剪切该页并将其粘贴到新位置。要剪切页，请依次单击  和 。要粘贴页，请选择要将其粘贴到的分支，然后依次单击  和 。
- 要复制页，请选中它，然后依次单击  和 。要粘贴页，请选择要将其粘贴到的分支，然后依次单击  和 。
- 要更改页设置（如元数据、页眉、页脚和其他选项），请单击 。请参见[更改页设置](#)。

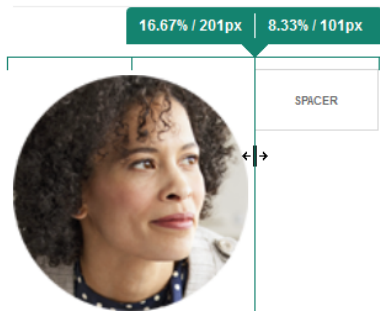
拖放编辑

要从子工具栏中添加组件，或者在页中移动组件，请单击并将项拖放到页上的位置。将项拖动到页上时，将显示可用内容框以及任何现有项的边框。放置栏  指示新内容将放到的位置（上、下、左、右）：




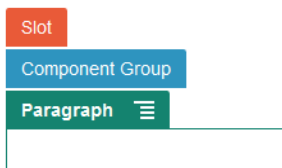
一个内容框中可以有多项，只需将项拖放到新位置即可将其移动到页中。您还可以通过单击并拖动内容框中两个组件之间的边界，来调整两个组件的相对宽度。组件对齐到在组件上方显示的“标尺”中指示的下一个网格线。


每个组件的大小会同时显示像素数以及相对于内容框中可用空间的百分比。要将宽度调整为由网格定义的值之外的值，请在单击并拖动组件边界的同时按住 **Ctrl** 键。



组件

在将组件放到页面上之后，您可以选择项，单击项的菜单图标  并选择设置来调整对齐方式、间距和其他属性。如果您单击组件名称而不是菜单图标，则可以查看并选择内容框和组件组（如果当前组件属于组件组）的菜单图标。如果选择了其他选项卡之一，则可以看到菜单图标并且可为该元素设置属性：



主题设计人员可指定随主题附带的组件，以使对一个站点可用的主题对其他站点不可用。主题设计人员还能够针对给定页布局中的给定内容框，指定其中允许哪些组件。如果特定内容框中不允许某个组件，则放置栏会更改颜色和符号（减号） ，并将显示类似于下文的消息：



调整组件的大小和间距

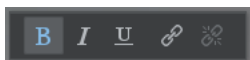
可以将许多组件（例如图库或图像）的大小调整为不同的 CSS 单位，还可以调整组件周围的间距。例如：


- px（像素）：默认值。如果仅指定数值，则假定为像素。
- %（百分比）：将项设置为其父 HTML 元素大小的百分比。示例：25%
- em：以 em 间距设置组件的大小。示例：20em
- vw：将大小设置为可视区域宽度的百分比，与可视区域的大小相应。示例：10vw

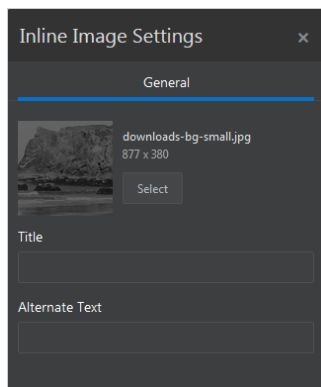
可编辑的布局内容

主题设计人员可以将将在编辑器中无法更改的“内置”内容（例如版权声明）添加到页布局。主题设计人员还可以将简单文本和图像内容指定为可编辑，包括数字资产。这使得内容提供者可以

更改文本或图像内容，但不会变更位置或其他布局属性。可编辑文本元素有一个菜单，其中提供了若干选项用于指定加粗、斜体和下划线文本，并可用于更改或删除链接：



可编辑图形元素包含“设置”图标  和面板，在其中可以指定图像、标题和替代文本。



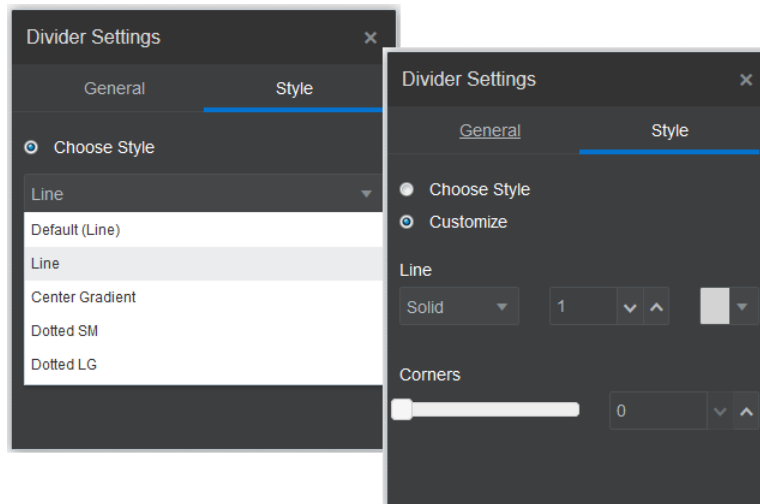
围绕着文本或图像的框架会做出调整，以适应文本的长度或图像的大小。

注：


所进行的更改只应用于当前页。初始内容随主题中的布局一起存储，在将该布局首次应用于页时，初始内容是默认内容。


样式和格式设置

大部分组件具有一个或多个由主题定义的基本样式，该主题指定组件外观的各个部分。您可以轻松切换样式或覆盖样式。要在可用样式之间选择，请打开组件的“设置”面板，依次单击样式选项卡和选择样式，然后从菜单中选择样式。要为在样式定制中指定的属性指定自己的值，请单击定制并指定格式设置选项。



撤消所做更改

使用  可还原上次在编辑器中所做的编辑或更改。还可以撤消多次以便还原一系列更改。





使用  可再次应用最近使用撤消操作所还原的更改。如果您连续使用了多次撤消操作，则可以使用多次重做操作。


您可以撤消在当前更新中所做的内容、样式和页组织结构更改。一些操作不包括在撤消链中：

- 如果您切换到其他更新，则撤消链将重置，您无法撤消之前在更新中所做的更改。在一个更新内，即使更改已保存，您仍可以撤消这些更改。
- 如果您在编辑器中更改视图，例如切换页或者更改给定页的大小，则必须手动还原这些类型的更改。
- 如果您编辑了文本组件，例如标题或段落，文本编辑器有自己的撤消链。在您退出文本编辑器之后，就无法再撤消这些更改。

使用样式和格式设置

大部分组件具有一个或多个由主题定义的基本样式，该主题指定组件外观的各个部分。您可以轻松地切换样式或者使用所选选项覆盖样式。

1. 导航到您要编辑的页，并确保  设置为编辑。
2. 要设置组件的基本样式，请单击组件的菜单图标  并选择设置。单击样式选项卡。
 - 要使用站点主题中提供的样式，请单击选择样式，然后从菜单中选择样式。样式是为单个组件定义的，因此样式列表可能会有所不同。例如，图像的样式设置不同于段落的样式设置。
 - 要指定自己的基本格式设置选项，请单击定制并指定格式设置选项。
3. 要复制基本样式并粘贴到一个或多个类似组件，请单击组件的菜单图标 ，然后选择复制 **component** 样式。单击类似组件的菜单图标 ，然后选择粘贴 **component** 样式。



4. 要设置标题或段落组件中文本的格式，请在文本组件中单击。此时将显示带格式设置选项的工具栏。选择要设置格式的文本，然后选择任意选项，例如字体、颜色或对齐方式。您所做的更改将立即应用。要删除格式，请选择文本并单击 。格式设置更改在应用时会覆盖基本样式。在更改基本样式时，格式覆盖保持不变。

定制站点设置

您可以定制站点设置，以添加站点图标、控制器文件、站点地图、机器人文件、辅助文件，以及指定地图提供方。

这些设置存储在更新中，直至您提交更新。发布之后，文件存储在主题的根本文件夹中，这样使用该主题的任何站点将使用这些文件。

要更改整个站点的图标和链接行为，请执行以下操作：

1. 打开站点进行编辑。
2. 单击子工具栏中的 ，然后单击  站点。
3. 在收藏图标部分中，选择在浏览器或其他平台（例如移动设备）中最小化站点时用于站点的图像。图标必须另存为您可以访问的数字资产。单击选择要上载的文件，导航到图标并选择该文件，然后单击确定。收藏图标需要进行定制设置以在所有浏览器中正常工作：
 - 对于 **Chrome** 和 **Safari**，必须在控制器文件中引用收藏图标，包括站点前缀，如以下示例所示：

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

对于 **Internet Explorer 11** 和 **Firefox**，必须在页模板中包含类似条目。但是，页模板属于主题，它可能用在多个站点中。因此，它无法使用固定站点前缀，并且必须改用令牌。请参见此示例：

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

提交页面时，该令牌会更换为站点前缀。

请注意，Internet Explorer 和 Firefox 从控制器加载收藏图标，并再次从页面模板加载它。因此，图标将刷新，除非控制器和页面模板引用的是同一个图标。Chrome 和 Safari 仅加载控制器引用的图标。

4. 在控制器文件部分中，可以添加文件来更改浏览器处理链接请求的方式。除了引用收藏图标，您还可以添加 OpenGraph 标记或元数据标记以进行 Web 管理员站点验证，或者用于在社交媒体上共享站点。可以下载默认控制器文件并编辑它，或者上载您自己的文件。此文件必须存储为您可以访问的数字资产。单击选择要上载的文件，导航到文件并选择，然后单击确定。

例如，如果站点包含“Facebook 分享”按钮，您可能希望提供元数据供 Facebook 用于显示 Facebook 中站点的详细信息，如以下示例所示：

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

```
<meta property="og:url" content="https://my.domain.com"/>  
<meta property="og:site_name" content="My Site Name on FB"/>
```

可以添加 Google Web 管理员验证标记，类似于以下示例：

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



5. 在站点地图和机器人文件部分中，可以上传定制的站点地图和机器人文件。站点地图是一个 XML 文件，可用于列出网站的 URL 以及有关各 URL 的信息，例如上次更新时间。机器人文件是一个创建用于指示搜索引擎机器人如何对网站的页编制索引的文本文件。这些文件必须存储为您可以访问的数字资产。单击选择要上传的文件，导航到文件并选择，然后单击确定。
6. 在辅助文件部分中，可以根据需要上传更多文件，例如验证网站所有权所需的那些文件。此文件必须存储为您可以访问的数字资产。单击选择要上传的文件，导航到文件并选择，然后单击确定。
7. 可以为地图组件和使用的链接选择提供方。选择 **Oracle Maps** 或 **Google Maps**。
8. 发布更新时，更改也将发布并投入使用。

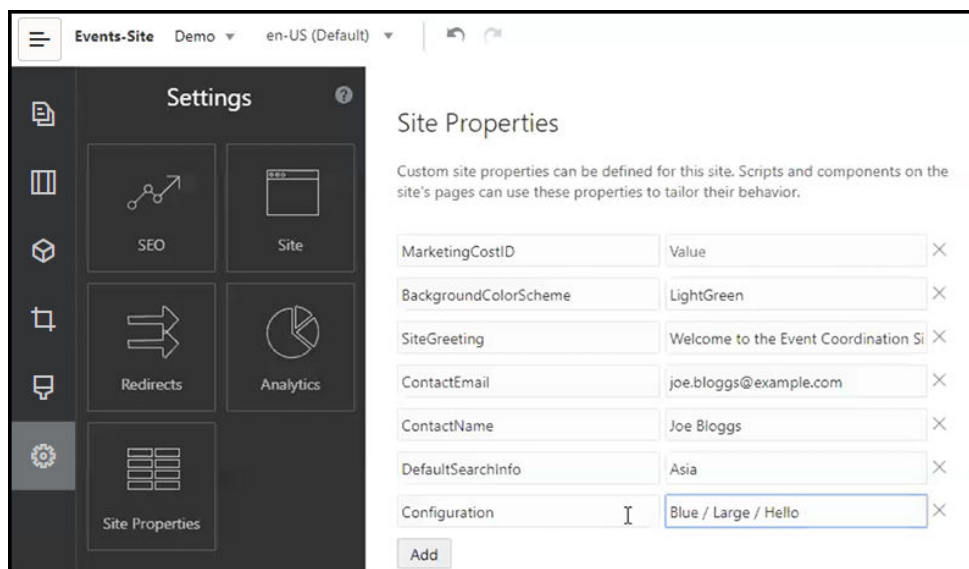
添加定制站点属性

您可以采用名称/值对的形式将定制属性添加到站点。这些属性会存储在站点中，供站点页上的脚本和组件使用。这些属性有助于对站点进行参数化或自定义，而不必更改底层脚本和组件代码。

例如，可以使用定制属性更改页背景颜色，细化搜索结果，填充列表，并且通常还能控制依赖站点的变量。

编辑站点时，可使用设置面板添加定制站点属性。

1. 打开站点进行编辑。
2. 单击子工具栏中的 ，然后单击  站点属性。
3. 单击添加。
4. 输入定制站点属性的名称和值。您最多可以添加 50 个定制站点属性。名称字段有 200 个字符的限制，值字段有 2000 个字符的限制。



5. 单击名称/值对旁边的 X 可将其删除。
6. 添加或删除定制站点属性后，请单击提交。

注：

这些更改不会与现有定制站点属性合并。提交对定制站点属性的更改将覆盖原始站点中任何现有的定制站点属性。

定义定制站点属性后，可在整个站点和站点组件的脚本中使用。例如，用于页脚，支持 SCSSMacro 扩展的组件的“其他查询字符串”字段（例如内容列表），或者标题和段落组件中的标记。

以下脚本在运行和设计时使用了 SCSSRenderAPI。此外，通过 SCSSCompileAPI 在模板编译器中针对布局和组件代码提供了定制站点属性值。此 API 具有一个类似于 SCSSRenderAPI 的新功能 getCustomSiteProperty，允许布局和组件代码读取定制部分属性的值。

例如，比如说您想定义一个定制部分布局，让它使用 SCSSRenderAPI 为标头、联系人姓名和联系人电子邮件调用和检索定制站点属性。以下脚本会使用获取的附加到 DOM 的定制站点属性构建 HTML 字符串。

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

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

      try {
        html += '<div>';
```

```



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

```

基于添加了定制站点属性的站点创建模板时，或者基于具有定制站点属性的模板创建站点时，会保留其中的定制属性。

使用表

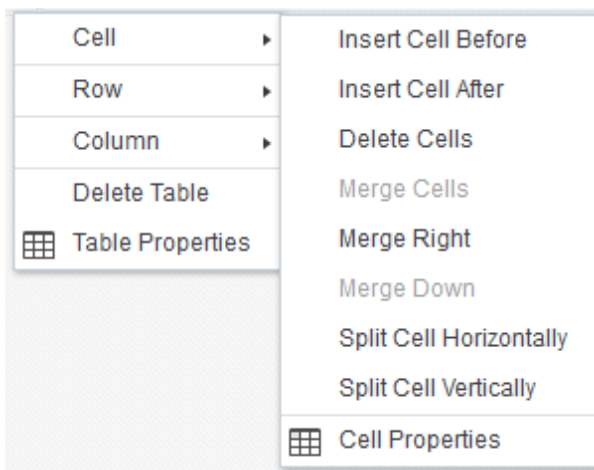
在段落组件中，可以包含您创建的表或从现有 HTML 源代码粘贴的表。

1. 转到您要编辑的页，并确保  设置为编辑。
2. 向页中添加组件，或者单击现有的段落组件。
3. 要从其他 HTML 页添加表，只需单击并拖动以选择表，然后将它复制并粘贴到段落组件。
要在当前光标位置创建新表，请单击 。选择行数、列数、宽度和其他格式设置选项。您可以添加在表上方居中的可选标题以及可选的概要，这可以为自动屏幕阅读器等辅助技术提供附加上下文。
4. 完成后，单击确定以关闭窗口。要在以后更改这些表设置，请右键单击表并选择表属性。要删除表，请右键单击表并选择删除表。

注：

如果针对不同的显示大小（如手机屏幕）开发，则可以使用在 StarterTheme design.css 文件中找到的代码 [创建自适应表](#)，该表允许堆叠移动设备上显示的行数据。

5. 要添加、删除或修改特定的行、列或单元，请右键单击行、列或单元，并从菜单选项中进行选择。例如：



 **注：**

一次只能更改一个单元的单元属性。

6. 您可以合并和拆分单元以创建复杂的表布局。例如：

		Split Cells	
Merged cells			

- 要将一个单元拆分为两个单元，请右键单击该单元，选择单元，然后选择水平拆分单元或垂直拆分单元。
 - 要合并两个水平单元，请在左侧单元中右键单击，选择单元，然后选择向右合并。
 - 要合并两个垂直单元，请在上方单元中右键单击，选择单元，然后选择向下合并。
7. 可以使用常规格式菜单选项更改单元内文本的格式和对齐方式。

上载站点文件

您可以随时使用 Oracle Content Management 界面上载图像和文档以用于您的站点。您也可以在处理背景图像、图像、图库和文档组件时，通过站点构建器上载文件。

在站点中使用图像和文档文件时，您可以使用站点中或其他您可以访问的位置中存储的图像。您也可以使用与您共享或者您从本地或网络文件位置上载的图像。

上载文件

要从本地或网络位置上载一个或多个文件，请执行以下操作：


1. 单击文档并转至您要存储文件的位置。单击创建可在当前位置中添加新文件夹。

2. 单击上传。
3. 查找并选择一个或多个文件，然后单击打开。

在编辑器中上传文件

在处理背景图像或者组件（例如文档或图像）时，如果要使用的文件不在 Oracle Content Management 位置上，您可以直接上传文件。

例如，要从本地或网络位置上传一个或多个文件以用于图库组件，请执行以下操作：

1. 要将图像添加到图库中，请单击菜单图标 ，选择设置，然后单击一般信息选项卡上的图像。
2. 单击添加图像。
3. 转到您要存储文件的位置，或者单击创建以将新文件夹添加到当前位置。
4. 单击上传。
5. 查找并选择一个或多个文件，然后单击打开。
6. 在资料档案库中选择一个或多个图像，然后单击确定。

注：

此时窗口中将显示所有可用文件。选择适合上下文的文件类型。例如，如果选择图像文件，请选择具有有效图像格式（GIF、JPG、JPEG、PNG 或 SVG）的文件。要链接到文件，请选择使用对原始文件的引用而不是将文件复制到站点。如果没有选择此选项，文件副本将随站点一起存储，并从站点中引用。

6

在站点中使用模板和主题

模板包含您在开始构建站点时需要的内容，包括站点代码框架、带有示例页和内容的默认站点、具有样式设置的主题以及图像等资源，甚至包括定制组件。主题定义站点的一般外观（即整体样式），包括颜色方案、字号、字体类型以及页背景。

模板

- [了解模板](#)
- [从站点创建模板](#)
- [更改模板详细信息](#)
- [更改模板策略](#)
- [更改模板状态或受众](#)
- [管理模板](#)
- [导出和导入模板](#)

主题

- [了解主题](#)
- [管理主题](#)
- [发布主题](#)

了解模板

模板包含您在开始构建站点时需要的全部内容，包括站点代码框架、带有示例页和内容的默认站点、具有样式设置的主题以及图像等资源，甚至包括定制组件。

默认模板

Oracle Content Management 提供了各种您可用于创建站点的模板。只需要选择模板、命名站点，然后您就可以立即开始添加内容。这些模板通常由服务管理员在初始化服务时安装。

模板	说明
BlankTemplate	BlankTemplate 是包含页眉、正文和页脚内容框的单个页面，让您完全自由地使用自己的设计。
营销材料展示	“抵押品展示”模板采用右对齐的水平菜单样式，简单的页布局，顶级页列出相关的视频资产，后跟嵌套的详细信息页，每页演示一个特定视频。它采用标准菜单组件，显示当前已登录用户。
新产品启动	“新产品启动”模板包含右对齐的横向菜单以及用于案例研究、功能详细信息、定价、公司简介和联系信息的布局。主页包含滚动的横幅图像以及文本。
产品和服务概览	“产品和服务概览”模板包含用于案例研究、产品详细信息、公司简介、隐私政策等的布局。主页包含滚动的横幅图像以及文本。该模板是完全响应式的。

模板	说明
入门级模板	使用入门级模板可以创建您自己的现成站点解决方案。入门级模板提供了一个简单而功能齐全的示例，您可以使用组件和交互来探索和扩展该模板。它包括站点代码框架，带有示例页和内容的默认站点，具有样式设置的主题，图像等资源，以及带有触发器和操作功能的定制组件。示例页包括有关创建模板的信息，这些模板包含指向提供详细信息的资源的链接。
学习演示显示	这些模板彼此之间使用颜色区别。它们是响应式模板，在组中有定制的 JavaScript 菜单组件。模板配有定制组件组，其中包含多种标准组件。它们还显示已登录用户。
访问知识关联	这些模板彼此之间使用颜色区别。它们是响应式模板，在组中有定制的 JavaScript 菜单组件。模板配有定制组件组，其中包含多种标准组件。
共享	此模板采用单个很长的页面。它是响应式模板，具有定制的 JavaScript 菜单，可以导航到页上的位置而非单独的页。
JET 入门级模板	Oracle JET (JavaScript 扩展工具包) 是一种模块化的开源工具包，它使用开源 JavaScript 库的集合。有关完整的详细信息，请参见 Oracle Jet 。

还可以从现有站点创建模板，或者导出现有模板，脱机修改，然后再将其导入为新模板。组织可能已具有可供您使用的模板。

模板工作原理

创建站点时，模板的用法如下：

- 如果启用了站点管理，则创建站点可能需要审批。请参见 [了解站点管理](#)。
- 模板中的默认站点复制到新站点，为您提供创建页的起点。
- 所有必需的支持文件复制到新站点。
- 如果主题文件夹中尚不存在模板主题，则将主题复制到主题文件夹。站点从主题文件夹中的主题位置引用主题。如果主题存在，则新站点只需引用现有主题。
- 如果有尚不在组件文件夹中的定制组件，则将定制组件复制到组件文件夹。站点从组件文件夹中的组件位置引用组件。如果任何组件已存在，则新站点将引用现有组件。

创建定制模板

如果您是 Web 开发人员，模板将您构建网站所需的全部内容收集在一个程序包中，包括站点、布局、导航、示例内容等等。您可以向站点添加组件和交互，提供现成的站点解决方案来满足您的业务需求。

模板以文件夹结构的形式表示，您可以像处理其他文件夹一样处理。模板的一些元素（例如主题和定制组件）从其在 Oracle Content Management 中的关联位置引用。例如，模板从可用主题列表中引用关联主题的方式与站点引用主题的方式相同。

在您从现有站点创建模板时，新模板使用该站点的副本作为默认站点。模板引用站点使用的主题以及站点页中使用的任意定制组件。主题和定制组件不复制到模板，而是通过与站点相同的方式来引用。

 注：

模板反映了在创建该模板时使用的站点。以后对创建模板时所用站点进行的更改不会反映在存储于模板中的站点上。

您可以创建用于共享内容模型的内容模板，这包括支持内容模型所需的内容布局、内容项和数字资产（例如示例内容）。应该从站点使用已发布的内容项和数字资产创建内容模板。请注意，在从带有内容的程序包导入站点模板时并未创建内容类型。它们在从内容模板创建站点时创建。因此，内容类型不由导入站点模板的用户拥有。相反，内容类型由从该模板创建第一个站点的用户拥有。然后，用户可以根据需要与其他用户共享模板。根据您具有的 Content Management Cloud 订阅类型以及服务的开始日期，此功能可能不可用。

在导出模板时，模板的所有元素（包括主题和任意组件的副本）收集到模板程序包内，您可以下载并脱机处理该程序包。

如果导入脱机修改的模板，或者如果模板、主题或定制组件的名称或 ID 已存在，则系统将提示您解决冲突。系统可能提供相应的选项，供您用来创建新模板、主题或定制组件；或者在某些情况下，您可以使用导入的版本覆盖现有模板、主题或定制组件。请参见[开发模板](#)。

您还可以将模板导入到特定的资料档案库。执行此操作时，您可以选择更新或复制现有资产。请参见[将模板导入到特定的资料档案库中](#)。

共享模板

在创建模板时，不论是通过导入、复制还是从某个站点创建的方式，除非您共享，否则任何其他人都无法使用该模板。

 注：

随 Oracle Content Management 提供并由管理员安装的模板也是如此。如果您未看到任何模板，请与管理员联系。可能尚未与您共享它们。

首次与用户共享模板时，将自动与该用户共享关联的主题和任何关联的定制组件，并提供下载者角色，确保用户在从模板创建站点时这些内容可用。接下来在模板中对该用户角色的更改不会更新关联主题或定制组件的共享信息。

如果启用了站点管理，您可通过模板策略使模板可用。请参见[了解站点管理](#)和[更改模板策略](#)。

从站点创建模板

如果您希望使用某个站点作为其他站点的起点，您可以从该站点创建模板。

 注：


要从站点创建模板，您必须具有站点的下载者、内容提供者或管理者的角色。此外，管理员必须在创建菜单中启用选项。如果在模板页上未看到创建菜单，请与管理员联系。

这些步骤显示如何从站点创建模板。您还可以导入脱机创建或修改的模板程序包。请参见[导出和导入模板](#)。

 注：

如果您要从使用来自多个资料档案库的内容的站点创建模板，您需要使用 OCE 工具包来完成。请参见[使用 OCE 工具包进行开发](#)和[使用 cec 命令行实用程序](#)。

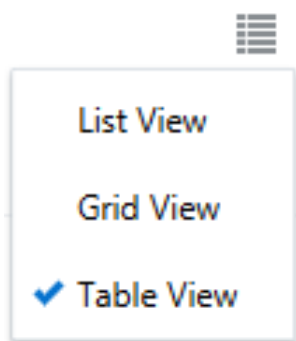
要使用 Oracle Content Management Web 界面从站点创建模板，请执行以下操作：



1. 单击站点，并选择要使用的站点。
2. 在右键单击菜单中选择创建模板，或单击操作栏中的 。
3. 输入模板的名称。可以使用字母、数字、下划线 (_) 和连字符 (-)。如果输入空格，则使用连字符自动取代空格。

不要将以下名称用于模板、主题、组件、站点或站点页：authsite、content、pages、scstemplate_*、_comps、_components、_compsdelivery、_idcservice、_sitescloud、_sitesclouddelivery、_themes、_themesdelivery。下面的名称虽然可用于站点页，但不要将其用于模板、主题、组件或站点：documents、sites。

4. 为模板输入可选的说明。
5. 选择您是否将在模板中包括未发布的内容项和数字资产。
6. 准备好之后，单击创建模板。

创建了模板之后，该名称显示在模板的列表中。要查看所有模板，请单击开发人员，然后单击查看所有模板。您可以通过单击“视图”图标并选择列表中的某个选项来控制模板的显示方式。



要查看与模板关联的文件夹和文件，请从右键单击菜单中选择打开，或单击操作栏中的 。要查看或更改模板的名称、说明和其他详细信息，请单击模板名称，然后在右键单击菜单中选择详细信息，或在操作栏中单击 。

新模板使用该站点的副本作为其默认站点。模板引用站点使用的主题以及站点页中使用的任意定制组件。主题和定制组件不复制到模板，而是通过与站点相同的方式来引用。

如果模板是从企业站点创建的，默认情况下从该模板创建的所有新站点将选择原始站点使用的本地化策略和默认语言。

模板反映了在创建该模板时所使用的站点。如果未来对创建模板时所用的站点执行任何更改，则这些更改都不会体现在随模板存储的站点中。

更改模板详细信息




“模板详细信息”选项卡显示模板名称、作者、说明、模板主题、模板中包含的定制组件以及模板预览图像。如果启用了站点管理，您还可以查看模板状态（创建站点时是否可以使用该模板）。

在拥有合适权限的情况下，您可以更改或更新模板属性，例如名称和说明。您还可以添加或删除模板的预览图像。

如果模板由您创建（您是模板所有者）、您是模板管理员或者其他人将模板共享给您并为您授予了“内容提供者”或“管理者”角色，则您可以更新模板详细信息。

如果您添加预览图像，文件将存储在模板的资产文件夹中。这些文件不会显示在任何关联的网站中，但模板所有者可以提供有关模板本身的信息。

要查看或更改模板详细信息，请执行以下操作：


1. 单击开发人员，然后单击查看所有模板。
2. 选择模板，然后在右键单击菜单中选择详细信息，或单击操作栏中的 。
3. 根据需要编辑信息。如果您不能编辑信息，说明您没有“内容提供者”或“管理者”角色。
4. 要添加预览图像，请单击 。找到您要使用的图像，或者单击上载以上本地存储的图像。图像应至少为 180px X 135px（纵横比为 4:3）。较小图像可能无法达到最佳显示效果，较大图像可能会影响性能。
如果所有的预览图像点都已填充，您将需要先删除一个图像，才能添加新图像。要删除预览图像，请单击图像上的 。
5. 单击保存。

更改模板策略

“模板策略”选项卡显示模板的类型（标准或企业）、所需的审批以及从模板创建的站点的最低安全性。您还可以查看模板状态（创建站点时是否可以使用该模板）。

仅当启用了站点管理而且您是站点管理员时，您才能看到“模板策略”选项卡。请参见[了解站点管理](#)。

要查看或更改模板策略，请执行以下操作：

1. 单击开发人员，然后单击查看所有模板。
2. 选择模板，然后在右键单击菜单中选择详细信息，或单击操作栏中的 。
3. 选择策略选项卡。
4. 选择从该模板创建的站点请求是需要站点管理员审批、自动审批还是需要特定人员审批。如果您希望限制为只有特定人员才能进行审批，请首先输入您希望添加为审批者的人员或组的名称或电子邮件，然后从搜索结果中选择该人员或组。要删除审批者，请单击其名称旁边的 X。

- 选择从此模板创建的站点所需的最低安全性。站点创建者可以根据需要为其站点选择更高的安全级别。
 - 特定服务用户 — 仅可以登录此 Oracle Content Management 实例的选定用户可以访问站点。站点创建者在站点创建后选择 Oracle Content Management 用户。请参见[更改站点安全性](#)。
 - 特定云用户 — 仅可以登录您的域的选定用户可以访问站点。站点创建者在站点创建后选择云用户。请参见[更改站点安全性](#)。
 - 服务用户 — 可以登录此 Oracle Content Management 实例的任何用户都可以访问站点。
 - 云用户 — 可以登录您的域的任何用户可以访问站点。
 - 每个人 — 任何人无需登录即可访问站点。
- 选择失效策略以确定使用模板的站点何时失效。站点管理员可以指定是将失效的站点脱机还是删除。站点所有者和管理者会在站点失效之前获得通知，他们可以选择是否延迟失效日期。如果他们未延迟失效日期，则当站点被置于脱机状态或者被删除时，他们将收到电子邮件通知。在那时，他们可以延迟失效日期并将站点重新联机，或者在必要时从回收站中还原站点。失效策略选项包括：
 - 从不
 - 1 年
 - 2 年
 - 定制通过选择定制，可以设置短于 1 年或长于 2 年的有效期。

 注：

在创建站点之后，站点管理员可以在站点属性对话框中更改站点的失效策略。


- 如果要编辑企业模板，您可以选择如何为友好 URL 值创建站点前缀。此前缀将附加到内容项缩略名值（特定于页或资产的 URL 的一部分）。您可以让系统根据站点名称自动生成前缀，也可以让用户输入前缀。
- 如果要编辑企业模板，则选择用来确定站点需要哪些语言的本地化策略。
- 完成后，单击保存。

更改模板状态或受众

“模板受众”选项卡显示模板的状态（创建站点时是否可以使用该模板）以及谁可以使用模板创建站点。

仅当启用了站点管理而且您是站点管理员时，您才能看到“模板受众”选项卡。请参见[了解站点管理](#)。

要查看或更改模板受众，请执行以下操作：





- 单击开发人员，然后单击查看所有模板。
- 选择模板，然后在右键单击菜单中选择详细信息，或单击操作栏中的 。
- 选择受众选项卡。





4. 要更改模板状态，请单击状态开关。您必须使模板处于活动状态，用户才能基于该模板创建站点。默认情况下，当您激活模板时，它将对于所有的站点创建者可用。如果要更改谁可以使用模板，请更改此选项卡上的设置。
5. 选择谁可以使用此模板创建站点。
 - 特定人员 — 只有所选人员才能使用此模板创建站点。首先输入您希望能够使用此模板的人员或组的名称或电子邮件，然后从搜索结果中选择该人员或组。要删除人员或组，请单击其名称旁边的 X。
 - 每个人 — 所有能够创建站点的人均可以使用此模板。
6. 完成后，单击保存。

管理模板

您可以复制、重命名和删除模板文件夹，就像处理任何其他文件夹一样，但在导入或共享模板时，有一些特殊注意事项。

选择开发人员，然后单击查看所有模板以打开“模板”页。使用操作栏上的选项或“模板”页上的右键单击菜单可执行以下任一任务。

任务	说明
创建模板	如果您希望使用某个站点作为其他站点的起点，您可以从该站点创建模板。
	<div style="border: 1px solid #ccc; padding: 10px; background-color: #e6f2ff;"> <p> 注：</p> <p>您必须具有站点的“下载者”、“内容提供者”或“管理者”角色，并且您的管理员必须启用该选项。如果在模板页上未看到创建菜单，请与管理员联系。此外，如果您要从使用来自多个资料档案库的内容的站点创建模板，您需要使用 OCE 工具包来完成。请参见使用 OCE 工具包进行开发和使用 cec 命令行实用程序。</p> </div>
	<p>在模板页上，单击创建，然后选择从现有站点创建。选择站点，命名模板，然后单击创建。请参见从站点创建模板。</p>
查看和编辑模板文件	<p>可以选择模板并在右键单击菜单中选择打开或单击操作栏中的 ，来浏览构成模板的文件夹和文件。</p> <p>如果您是 Web 开发人员，则可以下载和修改单个模板文件，也可以使用桌面应用程序，并同步您在本地系统上所做的更改。您还可以导出模板程序包，其中包括关联的主题和任何定制组件，并在您偏好的开发环境中脱机使用它。</p> <p>模板以文件夹结构的形式表示，您可以像处理其他文件夹一样处理。</p> <p>在导出模板时，模板的所有元素（包括主题和任意组件的副本）收集到模板程序包内，您可以下载并脱机处理该程序包。请参见开发模板。</p>
创建站点	<p>要从模板创建站点，请选择模板，然后在右键单击菜单中选择创建站点，或单击操作栏中的 。请参见创建站点。</p>
重命名模板	<p>选择要重命名的模板，然后从右键单击菜单中选择重命名，或单击操作栏中的 。输入模板的名称，该名称应该与同一个服务器上的任何其他模板名称都不同。有关命名指南，请参见从站点创建模板。</p>

任务	说明
复制模板	<p>您可以复制现有模板并对副本进行更改来创建模板。</p> <div data-bbox="824 373 919 409"> 注：</div> <div data-bbox="868 426 1365 457">复制模板时，不复制与该模板关联的共享信息。</div> <p>选择要复制的模板，然后从右键单击菜单中选择复制，或单击操作栏中的 。输入模板的名称，该名称应该与同一个服务器上的任何其他模板名称都不同。有关其他命名指南，请参见从站点创建模板。进度栏中显示新模板名称和复制状态。</p>
删除或还原模板	<p>如果您有合适的权限，则可以删除模板文件夹及其内容。删除模板时，模板文件夹及其关联的所有文件夹和文件将移到回收站中。</p> <p>如果您创建了模板（您是模板所有者）或者其他人与您共享了模板并授予您内容提供者或管理者角色，则您可以删除或还原该模板。</p> <div data-bbox="824 877 919 913"> 注：</div> <div data-bbox="868 930 1365 961">删除模板时，不会删除关联的主题和定制组件。</div> <p>要删除模板，请选择要删除的模板，然后从右键单击菜单中选择删除，或单击操作栏中的 。系统会提示您将模板移到回收站。已删除的模板会保留在回收站中，直至：</p> <ul style="list-style-type: none">• 您还原模板。• 您永久地删除该模板。• 达到您的回收站限额。• 回收站会根据服务管理员设置的间隔自动清空。默认值为 90 天。 <p>要还原模板，请在“模板”下拉列表中选择回收站，然后选择模板，并从菜单栏或右键单击菜单中选择还原。</p>
导出或导入模板	<p>您可以导出模板以脱机修改，然后将其作为新模板导入或者替换现有模板。您也可以导出模板以将其移动到其他实例并在该处导入。</p> <p>导出模板时，您将模板作为单个 .zip 文件复制到文件夹中。您可以直接从文件夹下载模板程序包，将其解压缩并处理单个文件。完成文件的处理之后，创建包含模板程序包的 .zip 文件，导入它并覆盖原始模板或者创建新模板。</p> <p>请参见导出和导入模板。</p>

任务	说明
共享模板和管理成员角色	<p>如果您是模板的所有者或管理者，则可以与其他用户共享模板并分配用来定义用户可以对模板所执行操作的角色。</p> <p>如果未启用管理，则为用户分配任何角色就会赋予用户从模板创建站点的权限。如果启用了管理，则必须激活模板并确保用户包含在模板受众中，以便授予用户从模板创建站点的权限。请参见更改模板状态或受众。</p> <p>模板的主题和定制组件存储为不同的对象，而且只是在模板中引用。因此，主题和定制组件成员资格将分开管理。当您与某个用户共享模板时，无论为用户提供的模板角色如何，都会自动为用户授予对任何相关主题的下载者角色，确保用户在从模板创建站点时相关主题可用。</p> <p>选择要共享的模板，然后在右键单击菜单或操作栏中选择成员。</p> <p>输入一个或多个用户名或电子邮件地址，并分配下列角色之一：</p> <ul style="list-style-type: none"> 查看者：查看者可以查看模板的文件夹和文件，但无法进行更改。 下载者：下载者还可以下载文件并将文件保存到自己的计算机中。 内容提供者：内容提供者还可以编辑模板详细信息和模板文件、上载新文件、删除文件和删除模板本身。 管理者：管理者也可以添加用户并分配用户的角色。将自动为模板创建者（所有者）分配管理者角色。
查看或编辑模板详细信息	<p>要查看模板详细信息，请选择模板，然后在右键单击菜单中选择详细信息，或单击操作栏中的 。模板详细信息包括模板名称、作者、支持文件、说明、主题、定制组件和缩略图。如果您具有适当的模板角色，则可以编辑详细信息。请参见更改模板详细信息。</p>
查看或编辑模板策略（在启用了管理时）	<p>如果启用了管理而且您是站点管理员，则您可以查看模板策略，例如，必需的审批、最低安全性、站点失效和本地化。选择模板，然后在右键单击菜单中选择详细信息，或单击操作栏中的 ，然后选择策略选项卡。请参见更改模板策略。</p>
使模板可用于站点创建，管理谁可以使用模板创建站点（在启用了管理时）	<p>如果启用了管理而且您是站点管理员，则可以使模板可用于创建站点，并管理谁可以使用模板创建站点。处于活动状态的模板（可用于创建站点）的旁边显示 。</p> <p>要更改状态或受众，请选择模板，然后在右键单击菜单中选择详细信息，或单击操作栏中的 。</p> <ul style="list-style-type: none"> 要使模板可用于创建站点，请单击状态开关将其激活。 要更改谁可以使用模板创建站点，请单击受众选项卡。请参见更改模板状态或受众。

导出和导入模板

您可以导出模板以脱机修改，然后将其作为新模板导入或者替换现有模板。也可以导出模板，将其移动到其他 Oracle Content Management 实例并在该处导入

导出模板

导出模板时，您实际上是将模板作为单个 .zip 文件复制到 Oracle Content Management 中的文件夹内。您可以直接从文件夹下载模板程序包，将其解压缩并处理单个文件。完成模板文件的处理后，创建包含模板程序包的 .zip 文件，将它导入到站点并覆盖原始模板或者创建新模板。

 注：

导出模板时，不包括模板的共享信息。

要导出模板，请执行以下操作：

1. 单击开发人员，然后单击查看所有模板。
2. 选择模板，然后从菜单栏或右键单击菜单中选择导出。
3. 导航到现有文件夹，或者通过单击创建，提供名称和可选的说明来创建新文件夹。
4. 通过选中复选框，然后单击确定来选择文件夹。

导入模板

 注：

如果您将模板导入到其他服务器，默认站点中的一些链接在新服务器上下文中可能无效。如果站点使用图像或其他内容的引用链接而非直接复制内容到站点，则该内容在新服务器上不可用。即使您复制内容到新服务器，该内容仍将具有不同的内部 ID，链接将无效。导入模板时，您会收到页中包含无效引用链接的通知。

要导入模板程序包，请执行以下操作：

1. 单击开发人员，然后单击查看所有模板。
2. 单击创建并选择导入模板程序包。
3. 上载模板程序包之后，导航到包含模板程序包的文件夹并打开该文件夹。如果您尚未上载模板程序包，请转到要上载到的文件夹，或者创建新文件夹。单击上载，找到并选择模板程序包，然后单击打开。
4. 要使用模板，请单击模板文件名旁边的复选框，然后单击确定。为模板、关联主题和任何定制组件创建了新文件夹。如果模板、主题或定制组件的名称或 ID 已经存在，系统将提示您解决冲突。您可能需要创建新的模板、主题或组件，或者可以使用导入的版本覆盖现有项。

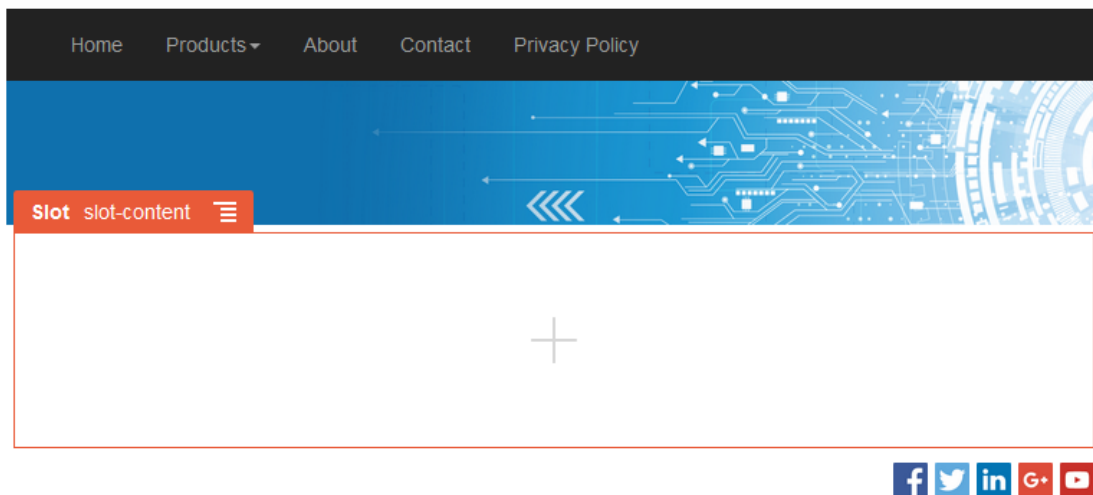
将模板导入到特定的资料档案库中

您可以选择资料档案库，并将模板程序包专门导入其中。

1. 单击内容，然后从菜单中选择资料档案库。
2. 选择要在其中导入模板程序包的资料档案库，然后单击导入内容。
3. 上载模板程序包之后，导航到包含模板程序包的文件夹并打开该文件夹。如果您尚未上载模板程序包，请转到要上载到的文件夹，或者创建新文件夹。单击上载，找到并选择模板程序包，然后单击确定。
4. 对于资料档案库中已存在的资产，选择是要使用新的修订版更新，还是要创建这些资产的复制版，然后单击导入。
5. 单击详细信息以查看导入的资产和内容类型列表。完成后单击确定。

了解主题

主题定义站点的一般外观（即整体样式），包括颜色方案、字号、字体类型以及页背景。主题为站点中的页提供一致的视觉效果。您可以调整设计，添加内容来创建站点，用于推销您的风格、品牌和愿景。



主题包括：

- 页布局
- 级联样式表 (CSS)
- 样式表上的变量
- 配置文件
- 定义站点导航的背景代码

每个网站使用一个主题。从模板创建站点时，您从模板继承主题。您可以随时更改站点的主题。

Oracle Content Management 提供了多种带有主题的模式，可供您在入门时使用。要创建新主题，请复制现有主题。您可以下载和修改主题文件，也可以使用桌面应用程序并同步在本地系统上进行的更改。有关其他主题创建方式的信息，请参见[开发主题](#)。

开发人员还可以创建使用特定组件子集的主题。使用这些主题之一创建站点时，将仅向您显示该主题允许的那些组件。这样，您就不必担心组件是否对您选择的主题有效。您仍可以将任何植入组件与主题化组件配合使用。

如果站点使用新的未发布主题，则在您首次使站点联机时，主题会自动随站点一起发布。如果您对主题进行了更改并希望更新联机站点以显示更改，则必须明确发布主题。只有主题所有者或具有管理者权限的用户可以明确发布主题。

 注：

如果您发布对某个主题的更改，则所有使用该主题的联机站点将会反映这些更改。例如，如果您更改了在主题中的默认字体并发布主题，则使用该主题的所有站点都将使用新的默认字体。

管理主题

您可以像复制、重命名和删除任何其他文件夹一样对主题文件夹执行这些操作，但由于主题可由多个站点共享，因此主题具有特殊注意事项。

您可以单击开发人员，然后单击查看所有主题来查看所有主题。您使用菜单栏上的选项或“主题”页上的右键单击菜单来执行这些任务。



任务	说明
创建新主题	要创建新主题，请复制现有主题。有关创建主题的其他信息，请参见 开发主题 。
复制主题	您可以复制现有主题并对副本进行更改来创建新主题。

 注：

复制主题时，不复制与该主题关联的共享信息。

选择要复制的主题，然后从菜单栏或右键单击菜单中选择复制。输入未由另一个主题使用的名称。可以使用字母、数字、下划线 (_) 和连字符 (-)。如果输入空格，则使用连字符自动取代空格。单击复制。

不要将以下名称用于模板、主题、组件、站点或站点页：authsite、content、pages、scstemplate_*、_comps、_components、_compsdelivery、_idcservice、_sitescloud、_sitesclouddelivery、_themes、_themesdelivery。下面的名称虽然可用于站点页，但不要将其用于模板、主题、组件或站点：documents、sites。

任务	说明
共享主题	<p>如果管理员已启用共享而且您是该主题的所有者或管理者，则您可以与其他用户共享您的主题。在共享主题时，您分配一个角色来定义用户可对主题执行的操作。</p> <div style="border: 1px solid #0070c0; padding: 10px; margin: 10px 0;"> <p> 注：</p> <p>当有人与某个用户共享模板时，无论为用户提供的模板角色如何，都会自动为用户授予对任何相关主题的下载者角色，确保用户在从模板创建站点时相关主题可用。</p> </div> <p>右键单击要共享的主题，选择共享，然后单击添加成员。输入一个或多个用户名或电子邮件地址，并分配下列角色之一：</p> <ul style="list-style-type: none"> 查看者：查看者可以查看主题的文件夹和文件，但无法进行更改。 下载者：下载者还可以下载文件并将文件保存到自己的计算机中。 内容提供者：内容提供者还可以编辑主题、上载新文件、删除文件和删除主题本身。 管理者：管理者也可以添加用户并分配用户的角色。将自动为主题创建者（所有者）分配管理者角色。
编辑主题	<p>如果您是 Web 开发人员，则可以下载和修改单个主题文件，也可以使用桌面应用程序并同步您在本地系统上所做的更改。</p> <p>您还可以导出模板程序包（包含关联的主题和任何定制组件）并在首选开发环境中脱机使用模板程序包。请参见开发主题。</p> <p>如果您对主题进行了更改，则必须发布该主题以便使用该主题的站点反映这些更改。</p>
发布主题	<p>如果站点使用新的未发布主题，则在您首次使站点联机时，主题会自动随站点一起发布。如果您对主题进行了更改并希望更新联机站点以显示更改，则必须明确发布主题。如果您是所有者或者如果向您分配了管理者角色，则可以发布主题。</p> <p>在列表中选择主题，然后单击菜单栏中的发布（对于上一个已发布主题，单击重新发布）。会在列表中的主题旁边添加一个“已发布”图标 。请参见发布主题。</p>

任务	说明
删除或还原主题	如果您有合适的权限，则可以删除主题文件夹及其内容。删除主题时，主题文件夹及其关联的所有文件夹和文件将移到回收站中。 如果您创建了主题（您是主题所有者）或者其他人与您共享了主题并授予您内容提供者或管理者角色，则您可以删除或还原该主题。

 **注：**

如果主题已由任何站点使用，则无法删除。

要删除主题，请右键单击要删除的主题并选择删除。系统将提示您将主题移到回收站。已删除的主题会保留在回收站中，直至：

- 您还原主题。
- 您永久地删除该主题。
- 达到您的回收站限额。
- 回收站会根据服务管理员设置的间隔自动清空。默认值为 90 天。

要还原主题，请在“主题”页菜单栏中单击回收站，再从列表中右键单击主题，然后选择还原。

发布主题

主题定义了站点的一般外观。您可以更新主题来更改使用主题的任何站点的外观。

如果站点使用新的未发布主题，则在您首次使站点联机时，主题会自动随站点一起发布。如果您对主题进行了更改并希望更新联机站点以显示更改，则必须明确发布主题。要发布对主题的更改，您必须是主题所有者或者具有管理者角色。

 **注：**

如果您发布对某个主题的更改，则所有使用该主题的联机站点将会反映这些更改。确保已对所做的更改进行脱机测试，并且您已了解了对关联站点的影响，然后再发布对主题的更新。

要发布主题，请执行以下操作：

1. 单击开发人员，然后单击查看所有主题。
2. 从主题列表中选择现有主题。
3. 在菜单栏或右键单击菜单中单击发布。
4. 单击确认以继续，然后单击确定。会在列表中的主题旁边添加一个“已发布”图标



7

管理定制组件和布局

作为开发人员，您可以创建和管理定制组件和布局。作为站点内容提供者，您可以注册第三方组件（应用程序）和组件组。

定制组件包括您在编辑器中创建的组件组，以及使用此部分中所述选项创建的本地和远程组件。定制布局包括部分布局（用于在页上的内容框中排列组件）和内容布局（用于在内容项中排列字段）。

- [了解定制组件](#)
- [了解布局](#)
- [使用定制组件和布局](#)
- [注册远程组件](#)
- [创建本地组件、布局、内容字段编辑器或内容表单](#)
- [导出或导入组件或布局](#)

要了解如何在站点中使用组件，请参见[排列页面内容](#)。

有关使用单个组件的信息，请参见[使用内置组件](#)。

有关如何创建自己的组件的详细信息，请参见[开发组件](#)。

了解定制组件

组件是 Web 页中的单独部件。查看某个 Web 页时，您看到了什么？您可能会看到一些标题、一些文本段落以及站点中其他页的链接。还可能看到图像、按钮、分隔线、地图和图库。所有这些项都是组件。

要将组件添加到页，请确保将  设置为编辑，单击  并选择要使用的组件类型。

在面板中拖动组件并将其放到页中的内容框。就这样。将标题、段落、图像和其他组件拖放到您希望放在页中的位置。

您可以使用组件管理器中的选项，轻松地注册和集成远程组件（应用程序），甚至创建自己的组件。单击开发人员，然后单击[查看所有组件](#)。

单击创建并选择关联的选项以创建新的本地组件或注册远程组件。通过这种方法创建和共享的组件在编辑器的定制组件面板中列出。


有关如何创建自己的组件的详细信息，请参见[开发组件](#)。

远程组件

如果您有要使用的第三方组件（应用程序），只需注册它即可在站点中使用。非常简单。

在注册远程组件时，您先为远程组件本身指定一个 URL，然后再为 Web 作者可为远程组件指定的任何设置指定一个 URL。

注册远程组件之后，您可以将它与其他用户共享。对于您拥有的任何已注册远程组件或与您共享的远程组件，它们会在站点构建器的“定制组件”面板中列出。

在编辑器中更改远程组件属性的方式与更改任何其他组件属性的方式相同。只需单击按钮的菜单图标  并选择设置。注册远程组件时，定制按钮打开您指定的设置 URL。

注：

由于远程组件和 Oracle Content Management 分别托管在不同的服务器上，出于安全考虑，将前者封装在内嵌框架中（使用 `iframe` 元素）。并非所有远程组件都可以封装在内嵌框架中。向提供者确认它是否可以封装。

本地组件

开发人员可以创建组件来访问 Oracle Content Management 提供的相同特性和功能。您可以直接将组件插入到页中，或者将其封装在内嵌框架中（使用 `iframe` 元素）。

单击创建来创建本地组件时，会将具有全部功能的示例组件添加到组件列表中，该组件具有您指定的名称和唯一标识符。作为开发人员，您可以修改示例以创建自己的解决方案。

有关如何创建自己的组件的详细信息，请参见[开发组件](#)。

了解布局

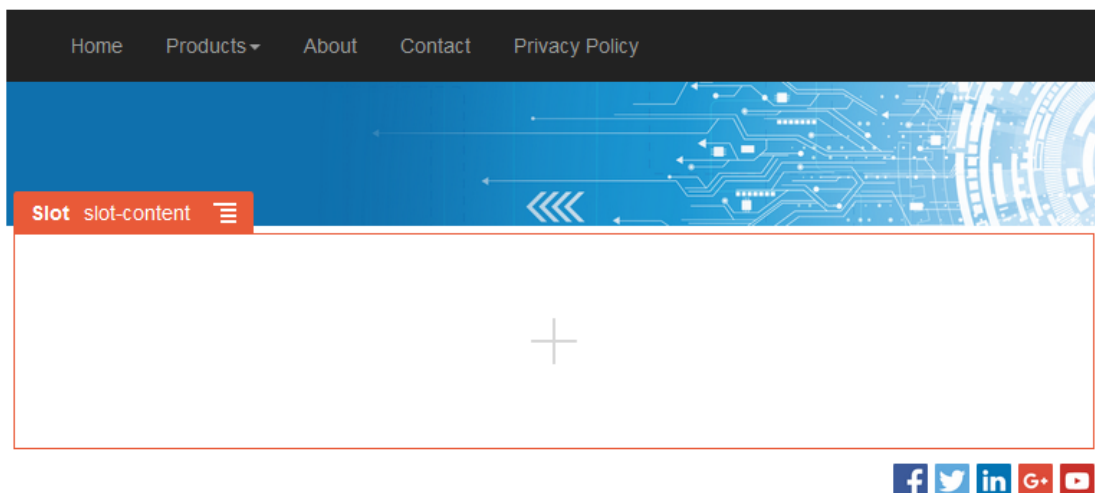
页布局在页上排列内容框和内容。部分布局在内容框内排列内容。内容布局在内容项中排列字段。

通常，布局指定内容的表示形式而不是内容自身。将内容与其表示形式分开后，更易于以不同的方式表示相同的内容或更改表示形式而不必触碰内容。

页布局

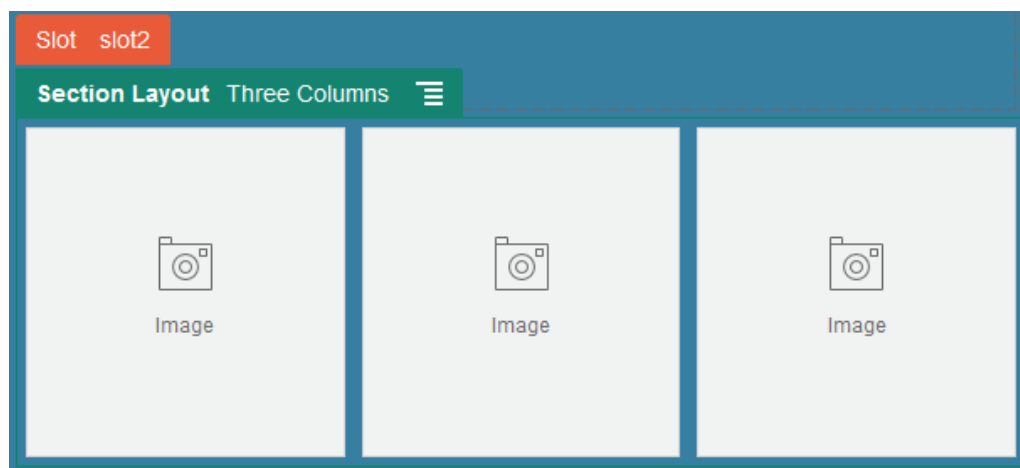
将页添加到站点之后，您选择用于该页的布局。每个布局在页上都有称为“内容框”的区域，内容提供者在其中可以拖放内容。页布局定义页上内容框的数量和位置。布局还可以包括位于页上的预定义内容。此内容可以是不可编辑的静态内容（例如，公司徽标），也可以是最低限度可编辑但无法更改位置或外观的内容（例如，内容提供者可以更改的标题文本）。

页布局存储在主题中。主题可以具有一个或多个页布局。作为开发人员，您可以复制并修改现有主题以创建新主题。请参见[开发布局](#)。



部分布局

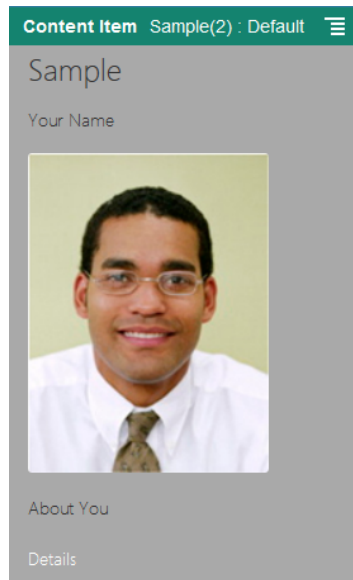
部分布局自动组织添加到它的内容，从而使内容提供者易于添加内容，而无需花费时间在页上设置内容格式。例如，部分布局可以自动将内容组织到多个列或组织到一个垂直列表。站点内容提供者可以将一个或多个部分布局添加到内容框来组织内容。



除提供的布局之外，开发人员还可以创建其他部分布局，以解决特定的布局问题或简化内容提供者的创作。主题设计人员甚至可以将部分布局构建到页布局中的内容框。请参见[开发布局](#)。

内容布局

如果您是 enterprise 用户，则可以基于为您提供的内容类型和布局创建和使用内容项。如果以这种方式构造的内容，内容提供者可以在站点构建器外部组合内容项的内容。与内容类型关联的多个内容布局使站点设计者可以在不同的上下文中显示内容项，而不必触碰组合的内容。作为开发人员，您可以根据提供的默认布局创建新布局。请参见[开发布局](#)。



使用定制组件和布局


作为开发人员，您可以创建和管理定制组件和布局。作为站点内容提供者，您可以注册第三方组件（应用程序）和组件组。


定制组件包括您在编辑器中创建的组件组，以及使用以下所述选项创建的本地和远程组件。如果您有要使用的第三方应用程序（远程组件），只需注册它即可在站点中使用。开发人员还可以创建本地组件来访问 Oracle Content Management 提供的相同特性和功能。

定制布局包括部分布局（用于在页上的内容框中排列组件）和内容布局（用于在内容项中排列字段）。

使用菜单栏上的选项，或者右键单击“组件”页上的菜单以执行以下任务。

任务	说明
注册远程组件	<p>在注册远程组件时，您先为远程组件本身指定一个 URL，然后再为 Web 作者可为远程组件指定的任何设置指定一个 URL。</p> <p>管理员必须在创建菜单中启用选项，然后才能创建组件。如果在组件页上未看到创建菜单，请与管理员联系。</p> <p>单击创建并选择关联的选项以注册远程组件。请参见注册远程组件。</p>
创建本地组件或布局	<p>创建本地组件或布局时，向您提供具有全部功能的示例组件或布局，您可以此为基础来创建自己的组件或布局。</p> <p>管理员必须在创建菜单中启用选项，然后才能创建组件或布局。如果在组件页上未看到创建菜单，请与管理员联系。</p> <p>单击创建并选择关联的选项以创建新的本地组件或布局。请参见创建本地组件、布局、内容字段编辑器或内容表单。</p> <p>有关其他组件创建方式的信息，请参见开发组件。</p>

任务	说明
复制组件或布局	<p>您可以复制现有组件或布局并对副本进行更改来创建新组件或布局。</p> <div data-bbox="803 373 899 415"> 注：</div> <p data-bbox="849 430 1341 575">在创建或注册组件或布局之后，无法更改该组件的名称。您可以复制组件或布局并为副本指定其他名称。包括远程组件关键字值在内的所有其他注册信息将会保留。共享信息独立于注册信息，不会复制。</p> <p>右键单击要复制的组件或布局，然后选择复制。输入名称，然后单击复制。可以使用字母、数字、下划线 () 和连字符 (-)。如果输入空格，则使用连字符自动取代空格。</p>
创建内容字段编辑器	<p>您可以 创建 内容字段编辑器，之后可以提升它并在创建用来控制数据字段外观的内容类型时使用它。支持除媒体和引用以外的所有数据类型。要使组件在创建内容类型时可用，必须先提升组件。</p>
共享组件或布局	<p>您可以与其他 Oracle Content Management 用户共享您的组件或布局。如果您是所有者或者分配了管理者角色，则您可以共享组件或布局。在共享组件或布局时，您必须分配一个角色给用户来定义用户可对组件或布局执行的操作。</p> <p>右键单击要共享的组件或布局，选择共享，然后单击添加成员。</p> <p>输入一个或多个用户名或电子邮件地址，并分配下列角色之一：</p> <ul style="list-style-type: none">查看者：查看者在编辑器中可以查看和使用定制组件和布局列表内的组件或布局。还可以查看主题的文件夹和文件，但无法进行更改。下载者：下载者还可以下载文件并将文件保存到自己的计算机中。内容提供者：内容提供者还可以编辑组件或布局、上载新文件、删除文件和删除组件或布局本身，前提是该组件或布局未在站点中使用。管理者：管理者也可以添加用户并分配用户的角色。将自动为组件或布局创建者（所有者）分配管理者角色。
编辑组件或布局	<p>如果您是 Web 开发人员，您可以下载和修改单个组件或布局文件，或者使用桌面应用程序并同步您在本地系统上进行的任何更改。</p> <p>您还可以单独导出各个组件或布局，或在模板程序包中导出（模板程序包中包含任何定制组件和布局），并在您的首选开发环境中脱机使用它。</p> <p>如果您对组件或布局进行更改，则必须发布组件或布局以使这些更改反映在使用该组件或布局的站点中。</p>
导出或导入组件或布局	<p>您可以导出组件或布局以脱机修改，然后将其作为新组件或布局导入或者替换现有组件或布局。也可以导出组件或布局以将其移动到其他实例并在该处导入。可以单独导出各个组件或布局，或在模板程序包中导出（模板程序包中包含任何定制组件和布局）。</p> <p>导出组件或布局时，您是将组件或布局作为单个 .zip 文件复制到文件夹内。您可以直接从文件夹下载组件或布局程序包，将其解压缩并处理单个文件。完成组件或布局文件的处理之后，创建包含组件或布局程序包的 .zip 文件，导入它并覆盖原始组件或布局或者创建新组件或布局。</p> <p>请参见 导出或导入组件或布局。</p>

任务	说明
发布组件或布局	<p>如果站点使用新的未发布组件或布局，则在您首次使站点联机时，组件或布局会自动随站点一起发布。如果您对组件或布局进行了更改并希望更新联机站点以显示更改，则必须明确发布组件或布局。如果您是所有者或者分配了管理者角色，则您可以发布组件或布局。</p> <p>在列表中选择组件或布局，然后在菜单栏中单击发布（对于以前发布的组件或布局使用重新发布）。会在列表中的主题旁边添加一个“已发布”图标 。</p>
删除或还原组件或布局	<p>如果您有合适的权限，则可以删除组件或布局文件夹及其内容。删除组件或布局时，组件或布局文件夹及其关联的所有文件夹和文件将移到回收站中。</p> <p>如果您创建了组件或布局（您是组件或布局所有者）或者其他人与您共享了组件或布局并授予您内容提供者或管理者角色，则您可以删除或还原该组件或布局。</p>

 **注：**

如果组件或布局由任何站点或更新使用（包括回收站中的站点或更新），则无法删除。

要删除组件或布局，请右键单击要删除的组件或布局，然后选择删除。系统会提示您将组件或布局移动到回收站。删除的组件或布局会保留在回收站中，直至：

- 您还原组件或布局。
- 您永久地删除该组件或布局。
- 达到您的回收站限额。
- 回收站会根据服务管理员设置的间隔自动清空。默认值为 90 天。

要还原组件或布局，请单击回收站，然后从列表中右键单击组件或布局，并选择还原。

注册远程组件

要在站点中使用远程组件，必须先在 Oracle Content Management 中注册它。

您可以注册第三方远程组件以及您开发的远程组件。

 **注：**

管理员必须在创建菜单中启用选项，然后才能注册远程组件。如果在组件页上未看到创建菜单，请与管理员联系。

要注册远程组件以便在 Oracle Content Management 中使用，请执行以下操作：

1. 单击开发人员，然后单击查看所有组件。
2. 单击创建并选择注册远程组件。
3. 在“注册远程组件”窗口中，输入或选择下列信息：
 - 名称：用户将看到的组件的名称。

- **说明：**用户将看到的组件的说明。
- **组件 URL：**在 iframe 中使用的端点，用于在页中呈现组件内容。必须为 HTTPS。
- **设置 URL：**在 iframe 中使用的端点，用于呈现添加到页的远程组件的设置。必须为 HTTPS。
- **设置宽度：**设置组件设置面板的默认宽度，以像素为单位。
- **设置高度：**设置组件设置面板的默认高度，以像素为单位。
- **密钥：**与远程组件关联的 192 位 AES 密钥，用于在预配组件时创建签名散列令牌。它用于加密，确保安全地读取和写入组件设置。

4. 单击注册。

创建了远程组件之后，该名称显示在组件的列表中。在组件列表中单击组件名称可以浏览用于注册组件的文件。

组件注册信息存储在由站点使用的目录中，该站点是在同一 Oracle Content Management 实例中创建的，但组件保持为远程服务。

对于组件所有者来说，组件图标会添加到编辑器的“定制组件”面板中，使用您分配给组件的名称。您可以将组件与其他用户共享，对方可在编辑器的“定制组件”面板中看到该组件。

创建本地组件、布局、内容字段编辑器或内容表单

作为开发人员，在创建本地组件或布局时，系统将向您提供具有全部功能的示例，您可以进行修改来创建自己的组件或布局。

管理员必须在创建菜单中启用选项，然后才能创建组件或布局。如果在组件页上未看到创建菜单，请与管理员联系。

作为开发人员，您可以创建组件来访问 Oracle Content Management 提供的相同特性和功能。

要创建示例本地组件或布局，请执行以下操作：

1. 单击开发人员，然后单击查看所有组件。
2. 单击创建，然后选择要创建的组件或布局类型。
3. 输入组件或布局的名称。您无法使用已由另一个组件或布局使用的名称。

可以使用字母、数字、下划线 (_) 和连字符 (-)。如果输入空格，将使用下划线自动取代空格。

不要将以下名称用于模板、主题、组件、站点或站点页：authsite、content、pages、scstemplate_*、_comps、_components、_compsdelivery、_idcservice、_sitescloud、_sitesclouddelivery、_themes、_themesdelivery。下面的名称虽然可用于站点页，但不要将其用于模板、主题、组件或站点：documents、sites。

4. (可选) 输入组件或布局的说明。
5. 对于组件类型，选择默认值以直接将组件插入页中，选择沙盒以将组件封装到内嵌框架中 (使用 iframe 元素)，或者选择模板以仅使用模板呈现。

注：

如果选择模板，则本地组件不基于 Knockout，因此当放置在页上时，无法使用嵌套组件。例如，这意味着编辑标题或正文文本必须通过组件设置面板中的定制设置来完成，而不是按照页面上显示的那样选择组件文本。

6. 单击创建。

进度栏显示创建状态。创建了组件、布局或表单之后，该名称显示在组件的列表中。可以单击组件列表中的组件或布局名称来浏览构成组件或布局的文件夹和文件。

7. 要选择分配给组件或布局的默认图标之外的图标，请执行以下操作：

- a. 从列表中选择组件或布局。
- b. 单击属性。
- c. 单击组件徽标选项卡。
- d. 在徽标库中单击某个徽标，然后单击完成。

有关如何创建自己的组件的详细信息，请参见[开发组件](#)。

在对组件、布局、内容字段编辑器或内容表单进行定制之后，可以将其与其他人共享或者对其进行提升，以便其他人可以按如下方式使用它：

- 定制组件：在使用站点构建器时，组件图标将添加到编辑器的“定制组件”面板中，并显示您分配给组件的名称。
- 部分布局：在使用站点构建器时，部分布局图标将添加到编辑器的“部分布局”面板中，并显示您分配给布局的名称。
- 内容布局：内容管理员可以将布局分配给一个或多个内容类型，作为默认视图或者添加到布局列表中。站点设计师可以在站点构建器中选择该布局，以指定如何在页面上显示该类型的内容项。
- 内容字段编辑器：要使内容字段编辑器在创建内容类型时可用，请选择它并单击提升，然后进行确认并单击确定。在提升内容字段编辑器之后，即可在创建内容类型时，使用内容字段编辑器控制除媒体和引用以外的所有数据字段的外观。
- 内容表单：要使内容表单在创建内容类型时可用，请选择它并单击提升，然后进行确认并单击确定。在提升内容表单之后，可以将其用于所有内容类型和资产类型。

导出或导入组件或布局

您可以导出组件以脱机修改，然后将其作为新组件导入或者替换现有组件。也可以导出组件，以将其移动到其他 Oracle Content Management 实例并在该处导入。

导出

导出组件时，您实际上是将组件作为单个 .zip 文件复制到 Oracle Content Management 中的文件夹内。您可以直接从文件夹下载组件，将其解压缩并处理单个文件。完成组件文件的处理之后，创建包含组件文件夹和文件的 .zip 文件，将其导入并覆盖原始组件或者创建新组件。



注：

导出组件时，不包括组件的共享信息。

要导出组件，请执行以下操作：

1. 单击开发人员，然后单击查看所有组件。
2. 选择组件或布局，然后从菜单栏或右键单击菜单中选择导出。

3. 导航到文件夹，或者单击新建，提供名称和可选的说明，然后单击创建来创建新文件夹。要打开文件夹，请单击文件夹图标或文件夹名称。
4. 单击关联文件夹的复选框，然后单击确定来选择文件夹。
使用组件名称或布局名称和 .zip 扩展名在选定文件夹中创建组件或布局程序包文件。

导入

管理员必须在创建菜单中启用选项，然后才能导入或创建组件、布局或内容字段编辑器。如果在组件页上未看到创建菜单，请与管理员联系。

要导入组件、布局或内容字段编辑器，请执行以下操作：

1. 单击开发人员，然后单击查看所有组件。
2. 单击创建并选择导入组件。选择此选项可导入组件、部分布局或内容布局。
3. 如果您上传了组件或布局程序包，请导航到包含组件或布局的文件夹并打开此文件夹。如果您尚未上传程序包，请转至要将组件或布局上传到的文件夹或创建新文件夹。单击上传，查找组件或布局程序包，然后单击打开。
4. 单击组件或布局文件名旁边的复选框，然后单击确定。创建新组件或布局并将其添加到组件列表。如果已存在组件或布局的名称或 ID，则系统将提示您解决冲突。您可能需要创建新的组件或布局，也可以使用导入的版本覆盖现有组件或布局。

8





使用站点页

现在来创建页、选择布局并确定页在站点中的位置。

- [导航到页](#)
- [查看页](#)
- [添加页](#)
- [移动页](#)
- [删除页](#)
- [更改页设置](#)
- [更改页布局](#)
- [更改背景或主题](#)

导航到页



要查看页的层次并在其中导航，请执行以下操作：

1. 打开站点进行编辑。请确保  设置为“编辑”。
2. 单击  以显示用于管理和编辑页及页内容的选项。
3. 单击  可列出第一级页。
4. 单击某个页以查看该页。
带有箭头 (>) 的页具有嵌套的页层。单击该页可以显示嵌套的页。
还可以使用页列表顶部的搜索框查找所需的页。
5. 要查看或更改特定页的设置，请单击该页的 。请参见[更改页设置](#)。

查看页

要查看站点中的页，请在编辑器中打开站点，然后使用不同的预览、大小和方向选项。

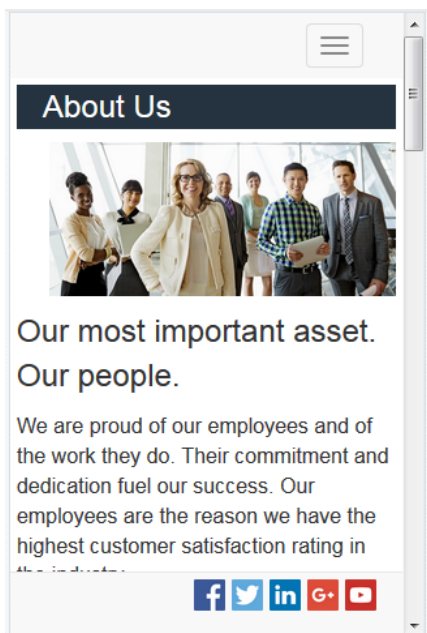
要更改使用编辑器时查看页的方式，请执行以下操作：




1. 打开站点进行编辑。
2. 导航到所需的页。
3. 要在编辑器中预览页，请确保  设置为“预览”。这将显示站点访问者能够看到的页外观，但没有在编辑时提供的视觉辅助工具。
4. 要在单独的浏览器窗口内预览应用到基本站点的当前更新，请单击 。

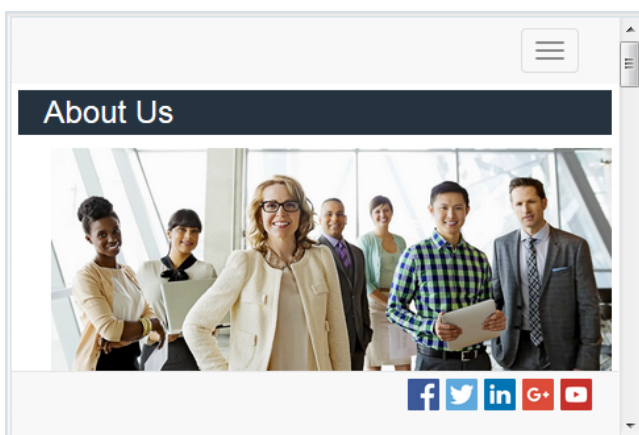
 注:

在预览模式下，指向站点页的链接不起作用。

5. 要预览页在具有特定屏幕大小的设备上的外观，请单击顶部菜单栏中的调整到屏幕大小，然后从列表中选择一组屏幕尺寸。具有响应式设计的主题会自动排列页内容，以在所选屏幕大小上实现最佳效果。



6. 要创建定制大小，请单击调整到窗口大小，然后选择创建设备预设...。填写必需的字段。要删除定制大小，请单击大小旁的 。您还可以激活标尺  并在标尺上选择任意间隔来快速查看不同大小。
7. 要在纵向和横向之间切换页方向，请单击 。



添加页

站点中的页按照类似于文件夹的层次进行组织，这种层次也称为站点树。您可以添加页、定义页的设置并决定页在站点中的位置。



您可以选择添加页或者添加外部页的连接。

- **Web 页：**该页和内容位于站点内。您可以命名页并定义其内容，指定页在站点中的位置，并且指定该页在当前站点上下文中的行为。
- **外部链接：**该页引用自通过 URL 指定的其他位置。您可以命名页，指定页在站点中的位置，并且指定该页在当前站点上下文中的行为。由于您在使用来自活动站点的页，因此无法更改链接页的内容。

在站点树中，对于链接到外部 URL 的页，在页名前有一个 。

添加 Web 页

要将 Web 页添加到站点，请执行以下操作：

1. 打开站点进行编辑。
2. 选择要在其中添加页的级别或分支，然后单击添加页。或者，要添加子页，请单击 ，然后单击 。“新页”会添加到站点树的底部，并且将提示您为该页命名并指定其他设置。
3. 选择 **Web 页** 作为页类型。如果要管理页上的内容而非重用其他站点的页，则使用此选项。
4. 为页指定名称。可以使用字母、数字、下划线 (_) 和连字符 (-)。不要将以下名称用于模板、主题、组件、站点或站点页：authsite、content、pages、scsTEMPLATE_*、_comps、_components、_compsdelivery、_idcSERVICE、_sitesCLOUD、_sitesCLOUDdelivery、_themes、_themesdelivery。下面的名称虽然可用于站点页，但不要将其用于模板、主题、组件或站点：documents、sites。
5. 指定页 URL。默认情况下，页名用于页 URL。空格将自动替换为连字符。要使用默认值之外的 URL，请单击覆盖，然后添加 URL 中使用的文件名。可以使用字母、数字、下划线 (_) 和连字符 (-)。请确保包括文件扩展名。默认文件扩展名是 .html。
6. 选择页布局。它定义页的一般结构，但不定义内容。布局包含一个或多个命名内容框，在其中可以放置内容。页布局的数量和类型取决于与您的站点关联的主题。
7. 选择移动页布局。这会定义在移动设备上查看时页的一般结构。
8. 指定页标题。它不在页本身上显示，而是在显示页时出现在浏览器标题栏或浏览器选项卡中。要在页上添加标题，请使用编辑器添加标题组件。
9. 添加可选说明。此说明不在页上显示，而是让您向其他内容提供者添加有关页的信息，或者供您自己使用。
10. 指定以逗号分隔的可选关键字以帮助搜索引擎识别页的内容。关键字可用于识别不在页的文本中显示或在图像中显示的术语或概念。关键字不显示在页上，但搜索引擎使用它们查找和识别您的站点。适当的说明、关键字和同义词可以增加网站的流量。
11. 添加可选的页眉脚本或标记以用于分析或跟踪。验证在页眉或页脚中使用的任意代码，确保代码正常工作并且不会给站点带来任何安全风险。
12. 添加可选的页脚脚本或标记以用于分析或跟踪。默认情况下，页脚包含所显示弹出窗口的文本，这些文本与在站点上使用 cookie 相关，并提供指向隐私政策的链接。如果您是开发

人员，则可以编辑显示的文本。验证在页眉或页脚中使用的任意代码，确保代码正常工作并且不会给站点带来任何安全风险。

13. 可选：您可以使用任何可用页选项：
 - **错误页**：如果某个错误导致所请求的页无法显示，则显示此页而不是默认错误页。默认情况下，选择此选项时，也会选定从导航中隐藏页和所有搜索引擎选项。如果希望在导航或搜索引擎结果中包括错误页，则可以取消选择这些选项。
 - **在导航中隐藏页**：在为站点自动生成的导航中不包括该页。
 - **详细信息页**：使用此页可以显示在其他页上选择的内容项的详细信息。结构化内容仅可供企业用户使用。默认情况下，选择此选项时，也会选定从导航中隐藏页。如果您希望在导航中包括详细信息页，则可以取消选择此选项。当结构化内容项配置为使用详细信息页并且用户单击结构化内容项的详细信息链接时，将显示详细信息页，其中包含内容项的详细信息。
 - **搜索页**：使用此页可以显示搜索的结果。可以在内容搜索组件的“链接”设置中选择此页。请参见[内容搜索](#)。默认情况下，选择此选项时，也会选定从导航中隐藏页。如果希望在导航中包括搜索页，可以取消选择该选项。
14. 可选：选择搜索引擎优化 (SEO) 选项：
 - **在搜索引擎中隐藏页**：通知搜索引擎不索引页的内容，这样该页不显示在搜索结果中。
 - **在搜索引擎中隐藏页链接**：通知搜索引擎不跟踪页上的链接，这样也就不索引链接目标。
 - **禁用搜索引擎页高速缓存**：通知搜索引擎不缓存此页。
 - **在搜索引擎中隐藏页说明**：通知搜索引擎在搜索结果中页的后面不包括说明（在上面指定）。
15. 可选：选择此页是否具有 "Cobrowse" 按钮用于浏览会话。Cobrowse 会话使用 Oracle Cobrowse Cloud Service 来管理与站点访问者的屏幕共享体验。请参见[启用 Cobrowse 集成](#)和[在页上使用 Cobrowse](#)。
16. 可选：覆盖站点的分析跟踪片段，并添加特定于此页的片段。添加 JavaScript 跟踪代码的片段进行 Web Analytics 跟踪，以便更轻松地与外部分析提供方（例如 Google、Adobe 或 Oracle Infinity）集成。请参见[添加分析跟踪](#)。
17. 要保存当前更新中所有待处理更改，请单击**保存**。

添加链接页

要将外部链接页添加到站点，请执行以下操作：

1. 打开站点进行编辑。
2. 转到站点树中您将在其中添加页的级别。页将添加到当前级别，不过您可以轻松地将它们移动到站点树中的其他位置。
3. 单击添加页。“新页”会添加到站点树的底部，并且将提示您为该页命名并指定其他设置。
4. 选择外部链接作为页类型。
5. 为页指定名称。可以使用字母、数字、下划线 (_) 和连字符 (-)。可以在站点树中通过名称来识别该页。空格将自动替换为连字符。
6. 指定页的完整 URL。例如：`https://www.example.com/sharedpage.htm`

7. 如果希望在新浏览器窗口或选项卡中打开页，请选择在新窗口中打开页。如果未选择此选项，则将在当前窗口中打开页，并替换您的站点页和导航。
8. (可选) 选择任一可用选项：
 - 错误页：如果某个错误导致所请求的页无法显示，则显示此页而不是默认错误页。默认情况下，选择此选项时，也会选定从导航中隐藏页。如果希望在导航中包括错误页，可以取消选择该选项。
 - 在导航中隐藏页：在为站点自动生成的导航中不包括该页。
 - 详细信息页：使用此页可以显示在其他页上选择的内容项的详细信息。结构化内容仅可供企业用户使用。默认情况下，选择此选项时，也会选定从导航中隐藏页。如果您希望在导航中包括详细信息页，则可以取消选择此选项。当结构化内容项配置为使用详细信息页并且用户单击结构化内容项的详细信息链接时，将显示详细信息页，其中包含内容项的详细信息。
 - 搜索页：使用此页可以显示搜索的结果。可以在内容搜索组件的“链接”设置中选择此页。请参见[内容搜索](#)。默认情况下，选择此选项时，也会选定从导航中隐藏页。如果希望在导航中包括搜索页，可以取消选择该选项。
9. 要保存当前更新中所有待处理更改，请单击保存。

移动页






您可以在站点树中将页拖放到当前级别中的其他位置，或者剪切并将页粘贴到其他级别。

层次中始终有一个顶级页，即站点的主页，该页的下方将列出所有其他页。您可以将页拖放到指定级别以更改其顺序。您也可以剪切页并粘贴到层次中的其他位置。

页可以有嵌套页。移动具有嵌套页的某个页时，嵌套页也会移动，并保持嵌套在所移动页之下。如果您使用的主题包括自动导航，则更改页顺序还会更改页在导航菜单中列出的顺序。

要在站点树中将页移到其他位置，请执行以下操作：

1. 打开站点进行编辑。
2. 转到要移动的页。
3. 在站点树中将页拖放到当前级别中的其他位置，请单击该页的名称并将其拖动到页列表中的其他位置，然后放下该页。

4. 要将页移动到站点树的不同级别，请单击要移动的页，然后单击该页的 。单击  可从当前位置剪切页，或者单击  可在当前位置保留页的副本。转到希望放置页的级别，然后单击 。如果您选择该级别中的页，则粘贴的页将嵌套到该页下方。单击  可将页粘贴到当前位置。要重新排序页，请在站点树中将页拖放到当前级别的其他位置。

删除页


在更新过程中，您可以删除页。

单击保存时，将记录在更新过程中删除的页。在您发布更新之前，该页不会从基本站点中删除。具有内容提供者或管理者角色的所有用户都可以删除页。

 注:



如果您发布用于删除某页的更新，则该页将永久删除，无法恢复。如果从多语言站点中删除了某页，则在发布站点时也会删除该页的翻译版本。如果希望管理页添加和删除操作，则在站点中添加或删除页时，您可以使用专用的更新来隔离这些操作。

要从站点中删除页，请执行以下操作：

1. 打开站点进行编辑。
2. 导航到您要删除的页，然后单击 。
3. 单击确定以确认删除。单击保存以保存当前更新中的所有待处理更改。

更改页设置

可以更改页设置，如名称、标题、URL、页眉、页脚和其他选项。

1. 打开站点进行编辑。
2. 单击要更改其设置的页，然后单击 。
3. 根据需要更改设置：
 - 页类型：
 - **Web 页**：该页和内容位于站点内。您可以命名页并定义其内容，指定页在站点中的位置，并且指定该页在当前站点上下文中的行为。
 - **外部链接**：该页引用自通过 URL 指定的其他位置。您可以命名页，指定页在站点中的位置，并且指定该页在当前站点上下文中的行为。由于您在使用来自活动站点的页，因此无法更改链接页的内容。
在站点树中，对于链接到外部 URL 的页，在页名前有一个 。

如果您选择“Web 页”，您将看到以下设置：

- **页名**：可以使用字母、数字、下划线 (_) 和连字符 (-)。不要将以下名称用于模板、主题、组件、站点或站点页：authsite、content、pages、scstemplate_*、_comps、_components、_compsdelivery、_idcservice、_sitescloud、_sitesclouddelivery、_themes、_themesdelivery。下面的名称虽然可用于站点页，但不要将其用于模板、主题、组件或站点：documents、sites。
- **页 URL**：默认情况下，页名用于页 URL。空格将自动替换为连字符。要使用默认值之外的 URL，请单击覆盖，然后添加 URL 中使用的文件名。可以使用字母、数字、下划线 (_) 和连字符 (-)。请确保包括文件扩展名。默认文件扩展名是 .html。
- **页布局**：页布局定义页的一般结构，但不定义内容。布局包含一个或多个命名内容框，在其中可以放置内容。页布局的数量和类型取决于与您的站点关联的主题。
- **移动页布局**：移动页布局定义在移动设备上查看时页的一般结构。
- **页标题**：页标题不在页本身上显示，而是在显示页时出现在浏览器标题栏或浏览器选项卡中。要在页上添加标题，请使用编辑器添加标题组件。

- **页说明：** 页说明不在页上显示，而是让您向其他内容提供者添加有关页的信息，或者供您自己使用。
- **关键字：** 指定以逗号分隔的可选关键字以帮助搜索引擎识别页的内容。关键字可用于识别不在页的文本中显示或在图像中显示的术语或概念。关键字不显示在页上，但搜索引擎使用它们查找和识别您的站点。适当的说明、关键字和同义词可以增加网站的流量。
- **页眉：** 添加可选的页眉脚本或标记以用于分析或跟踪。验证在页眉或页脚中使用的任意代码，确保代码正常工作并且不会给站点带来任何安全风险。
- **页脚：** 添加可选的页脚脚本或标记以用于分析或跟踪。默认情况下，页脚包含所显示弹出窗口的文本，这些文本与在站点上使用 cookie 相关，并提供指向隐私政策的链接。如果您是开发人员，则可以编辑显示的文本。验证在页眉或页脚中使用的任意代码，确保代码正常工作并且不会给站点带来任何安全风险。
- **页选项：**
 - **错误页：** 如果某个错误导致所请求的页无法显示，则显示此页而不是默认错误页。默认情况下，选择此选项时，也会选定从导航中隐藏页和所有搜索引擎选项。如果希望在导航或搜索引擎结果中包括错误页，则可以取消选择这些选项。
 - **在导航中隐藏页：** 在为站点自动生成的导航中不包括该页。
 - **详细信息页：** 使用此页可以显示在其他页上选择的内容项的详细信息。结构化内容仅可供企业用户使用。默认情况下，选择此选项时，也会选定从导航中隐藏页。如果您希望在导航中包括详细信息页，则可以取消选择此选项。当结构化内容项配置为使用详细信息页并且用户单击结构化内容项的详细信息链接时，将显示详细信息页，其中包含内容项的详细信息。
 - **搜索页：** 使用此页可以显示搜索的结果。可以在内容搜索组件的“链接”设置中选择此页。请参见[内容搜索](#)。默认情况下，选择此选项时，也会选定从导航中隐藏页。如果希望在导航中包括搜索页，可以取消选择该选项。
- **SEO 选项：**
 - **在搜索引擎中隐藏页：** 通知搜索引擎不索引页的内容，这样该页不显示在搜索结果中。
 - **在搜索引擎中隐藏页链接：** 通知搜索引擎不跟踪页上的链接，这样也就不索引链接目标。
 - **禁用搜索引擎页高速缓存：** 通知搜索引擎不缓存此页。
 - **在搜索引擎中隐藏页说明：** 通知搜索引擎在搜索结果中页的后面不包括说明（在上面指定）。
- **Cobrowse：** 如果您的管理员为您的系统启用了 Cobrowse，而且已经为该站点启用了 Cobrowse，您将看到一个选项，可供您选择此页是否将有一个用于浏览会话的 Cobrowse 按钮。Cobrowse 会话使用 Oracle Cobrowse Cloud Service 来管理与站点访问者的屏幕共享体验。请参见[启用 Cobrowse 集成](#)和[在页上使用 Cobrowse](#)。
- **分析选项：** 如果站点包括分析跟踪片段，您可以在此页面上覆盖该片段。单击覆盖开关，然后编辑脚本。有关详细信息，请参见[添加分析跟踪](#)。如果您编辑站点片段，则会显示一则消息，指出脚本已修改。要删除定制内容，请单击还原为最新的站点脚本。

如果您选择“外部链接”，您将看到以下设置：

- **页名：** 可以使用字母、数字、下划线 (_) 和连字符 (-)。可以在站点树中通过名称来识别该页。空格将自动替换为连字符。

- **链接 URL：**指定页的完整 URL。例如：`https://www.example.com/sharedpage.htm`。
- **在新窗口中打开链接：**如果希望在新浏览器窗口或选项卡中打开页，请选择此选项。如果未选择此选项，则将在当前窗口中打开页，并替换您的站点页和导航。
- **页选项：**
 - **错误页：**如果某个错误导致所请求的页无法显示，则显示此页而不是默认错误页。默认情况下，选择此选项时，也会选定从导航中隐藏页。如果希望在导航中包括错误页，可以取消选择该选项。
 - **在导航中隐藏页：**在为站点自动生成的导航中不包括该页。
 - **详细信息页：**使用此页可以显示在其他页上选择的内容项的详细信息。结构化内容仅可供企业用户使用。默认情况下，选择此选项时，也会选定从导航中隐藏页。如果您希望在导航中包括详细信息页，则可以取消选择此选项。当结构化内容项配置为使用详细信息页并且用户单击结构化内容项的详细信息链接时，将显示详细信息页，其中包含内容项的详细信息。
 - **搜索页：**使用此页可以显示搜索的结果。可以在内容搜索组件的“链接”设置中选择此页。请参见[内容搜索](#)。默认情况下，选择此选项时，也会选定从导航中隐藏页。如果希望在导航中包括搜索页，可以取消选择该选项。

4. 要保存当前更新中所有待处理更改，请单击保存。

更改页布局

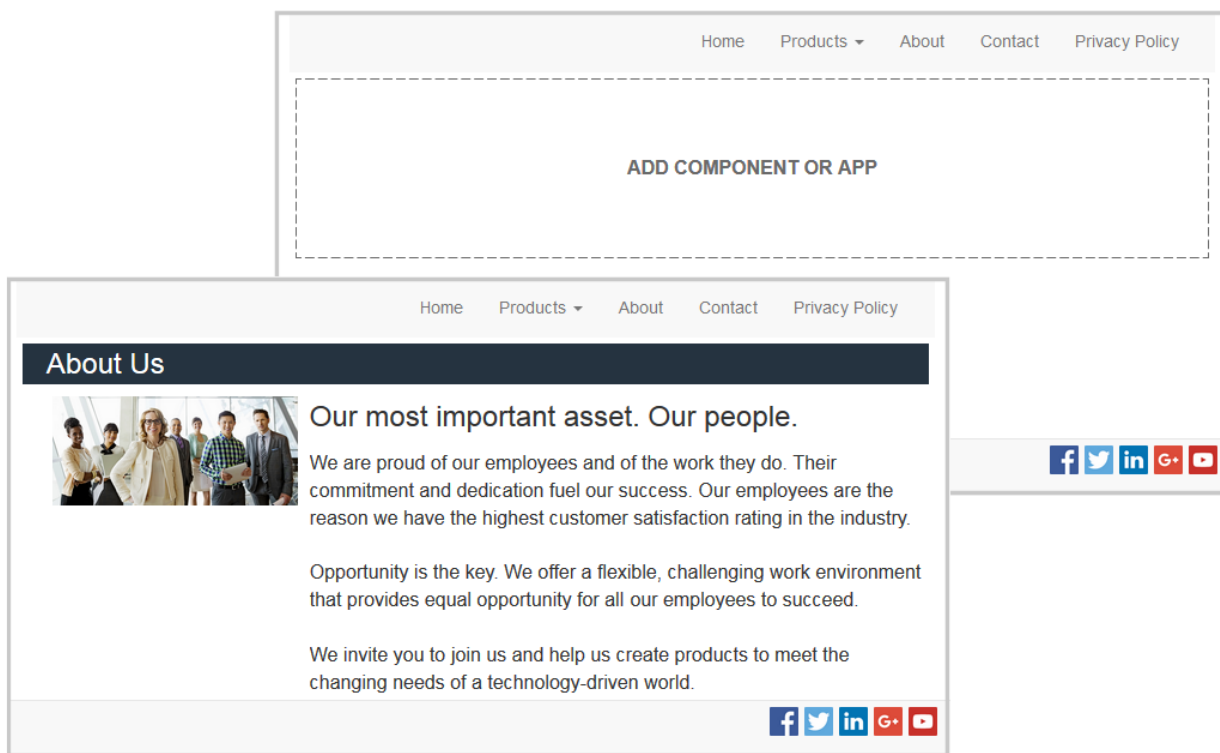
布局定义内容如何在页上排列。不同布局可以包含不同数量的命名内容框，该框是跨整个页面宽度的区域。内容框可以包含一种或多种内容。

每个主题有多个页布局。将页添加到站点之后，您选择用于该页的布局。每个布局都具有可以在其中拖放内容的内容框。这些内容框中的内容由您决定。这可以是标题、文本和分隔线乃至多媒体、图库和社交媒体的任何内容。您可以排列内容框中的内容，但是无法更改页上内容框的数量或排列。为此，您必须选择新的页布局。


注：

您可以将一个布局更换为另一个。不过请谨慎。如果选择数量较少或者不同名称的内容框，则其他内容框中的现有内容将不会显示在新布局中。内容不会被删除，但仅当您选择的布局具有同名内容框时才能够显示。

以下插图展示了页的布局示例。您可以看到页布局中的空内容框，以及将标题、图像和文本添加到内容框后完成的页。



要为页选择其他布局，请执行以下操作：

1. 打开站点进行编辑。
2. 导航到该页并单击 .
3. 转到“页布局”字段，然后从菜单中选择其他布局。页布局的数量和类型取决于与您的站点关联的主题。
4. 要保存当前更新中所有待处理更改，请单击保存。

更改背景或主题

您可以为页、页上的各个内容框以及内容框中的部分布局指定背景色和图像。您还可以更改站点的整个主题。







内容框是一个横跨页的区域，其中可以容纳一种或多种类型的内容。为内容框指定的背景应用到整个内容框以及内容框中的所有组件。

更改背景

背景层位于其他层之上。当您为内容框指定背景时，它“叠加在”为页指定的背景之上。对于大多数图像和颜色，较高的层能够有效地覆盖较低的层。如果在背景色或图像中使用透明度，则较低层中的颜色可能会显示或者与较高层中使用的颜色混合。

一些组件（例如，段落和标题）可以作为预定义样式或定制样式的一部分提供背景色。如果您为组件指定背景色，它也会叠加在任何部分布局、内容框或页背景之上。请参见[使用样式和格式设置](#)。

对于页、内容框和部分布局，背景选项相同，不过访问的方式略有不同：

1. 打开站点进行编辑。
2. 导航到您要编辑的页，并确保  设置为编辑。
3. 要更改页或内容框的背景，请执行以下操作：
 - 单击子工具栏中的 ，然后单击 。
 - 要为整页指定背景，请单击完整页，然后单击 。要为页上的内容框指定背景，请选择内容框并单击 。
4. 要更改部分布局的背景，请执行以下操作：
 - 单击部分布局的“菜单”图标 ，然后选择设置。
 - 单击背景。对于页、内容框和部分布局，背景选项相同。
5. 要为背景使用图像，请单击选择图像。可以使用来自您可以访问的任何位置的图像。您也可以使用与您共享或者您从本地或网络文件位置上载的图像。
6. 选择图像，然后单击确定。此时窗口中将显示所有可用文件。选择适合上下文的文件类型。例如，如果选择图像文件，请选择具有有效图像格式（GIF、JPG、JPEG、PNG 或 SVG）的文件。
7. 要调整图像设置，请执行以下操作：
 - 使用位置来将图像放在页上或内容框中。
 - 选择缩放选项以调整图像的显示形式：
 - 适合：缩放图像，使得整个图像适合可用空间，而不会导致图像变形。
 - 拉伸：两个尺寸（宽度或高度）中较大的那个将缩放以适合可用空间，较小的尺寸将拉伸以填充可用空间。
 - 无：使用图像的完整像素分辨率，如果无法容纳在可用空间中，则会按照可用空间来剪裁。
 - 选择重复选项可平铺太小的图像来填充可用空间。如果您选择了拉伸作为缩放选项，则不应用此选项。
 - 选择不在页中滚动可在用户滚动页时保持图像静止。
8. 要为背景选择颜色，请从颜色菜单中选择颜色，或者单击更多来从完整的颜色范围中选择颜色。
 - 单击右侧的色谱栏以选择颜色并在颜色范围显示中显示该颜色的变化。屏幕上的右上角会显示选定的颜色，所选颜色明暗度向左白色比例增加，向下黑色比例增加。
 - 要选择颜色的变体，请在颜色范围显示中单击，或者单击并将选择点拖动到新的颜色位置。当前颜色框以及 6 位数颜色代码显示了选定颜色。
 - 要调整透明度（阿尔法通道），请单击透明度滑块并向左拖动以增加透明度。透明度值 0% 使颜色透明，允许完全显示较低层的颜色和內容。透明度值 100% 使得颜色完全不透明，不显示较低层中的所有颜色和內容。其他值允许较低层的图像和颜色显示，其颜色与较高层的颜色以不同的级别混合。
 - 要应用当前颜色选择，请单击选择。

更改主题



单击保存时，将记录在更新过程中对主题进行的更改。在您发布更新之前，主题更改不会应用到站点。

注：

并非所有主题都彼此兼容。不同主题可能包含不同导航、样式名称或页布局信息。例如，如果选择具有不同布局名称的主题，现有页可能会使用无法在该主题中使用的布局。这些页显示为空白。页内容不会被删除，但仅当您选择的主题具有同名布局和内容框时才能够显示。如果选择不兼容的主题，可以将该主题更改为原始主题以还原页内容。

请确保您切换到了兼容主题。如果不确定，请与主题开发人员商议。

要更改站点使用的主题，请执行以下操作：

1. 打开站点进行编辑。
2. 单击子工具栏中的 ，然后单击 。
3. 从可用主题列表中选择主题，然后单击确定。主题应用到当前更新中。要保存当前更新中所有待处理更改，请单击保存。

9

排列页面内容

组件和内容项提供用户所需的功能和内容，而使用部分布局可以自动排列它们。

- [添加组件和部分布局](#)
- [使用资产和内容项](#)
- [使用触发器和操作](#)
- [使用水平部分布局](#)
- [使用垂直部分布局](#)
- [使用两列或三列布局](#)
- [使用选项卡式部分布局](#)
- [使用滑块部分布局](#)

有关使用单个组件的信息，请参见[使用内置组件](#)。

要了解如何管理组件和布局，请参见[管理定制组件和布局](#)。





有关如何创建自己的组件的详细信息，请参见[开发组件](#)。


添加组件和部分布局

组件是 Web 页中的单独部件。部分布局自动组织添加到它的内容，从而使内容提供者易于添加内容，而无需花费时间在页上设置内容格式。

添加组件

要将组件添加到页中，请执行以下操作：

1. 导航到您要编辑的页，并确保  设置为编辑。
2. 单击 ，然后选择以下组件类型之一：
 - 单击主题化来显示已选择用于与站点关联的主题的组件列表。
 - 单击定制来显示已与您共享的定制组件列表。
 - 单击已植入来显示可用于服务的默认组件列表。
 - 单击全部来显示已与您共享的所有组件。
3. 单击并按住某个特定图标，并将其拖动到要放在页上的位置。当您项拖动到页时，将显示可用内容框、部分布局和任意现有项的边框。“放置”图标  或垂直放置栏 （放在左侧或右侧）指示新内容将放到的位置。部分布局或内容项周围的实线边框指示您可在其中放置项，项将自动放置。一个内容框中可以有多项，只需将项拖放到新位置即可将其移动到页中。
4. 移动到合适的位置时，将项放到页上。

- 要调整项的属性，请单击项的菜单图标  并选择设置。根据项，您需要添加文本、指定图像的链接、指定其他站点或地图的 URL、调整项的大小或者其他执行操作。



每个组件都有诸如大小、对齐方式、间距、颜色和边框等设置。这些定义了组件的外观和行为。例如，段落设置包括字体样式、字号以及可确定文本呈现方式的其他功能。

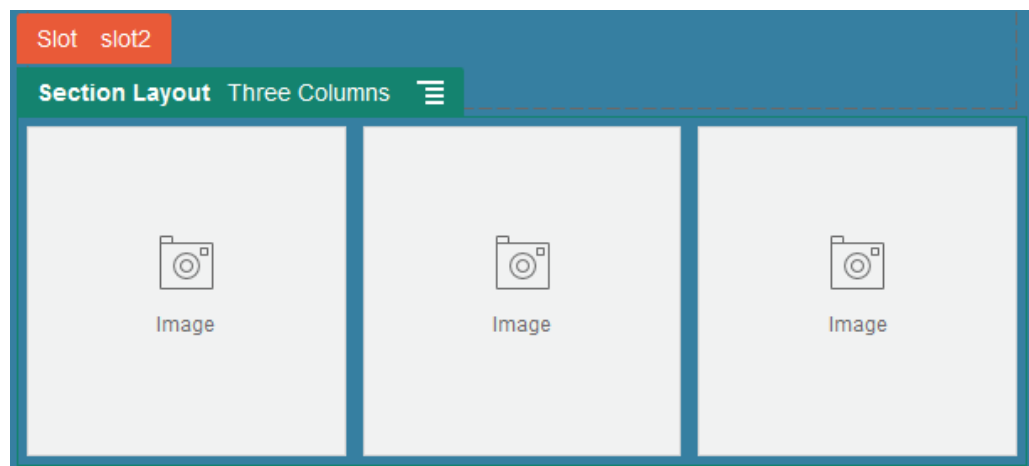
要调整组件的属性，请选择该组件，然后单击  菜单并选择设置。不同的组件具有不同类型的设置：


设置	说明
一般信息	常规设置包括间距、对齐方式以及组件之间通用的设置。
样式	样式是指定的一组默认设置值，可用于控制外观。样式在主题中定义。不同的主题可以具有不同的样式，在主题中，不同的组件可以具有不同的样式。您还可以为组件的特定实例手动指定样式设置。
链接	链接设置包括图像的位置、文档或组件使用的其他资源。对于可以执行操作的按钮和其他组件，链接设置还包括组件支持的触发器和操作。
组件	对于包括其他标准组件（例如文章组件）的组件，它们可提供单独组件的列表，让您可以分别访问这些组件的设置。
定制	定制设置对于组件唯一，在标准设置组之外单独提供。例如，远程组件可以将首选项存储为唯一 URL 并将其作为定制设置。

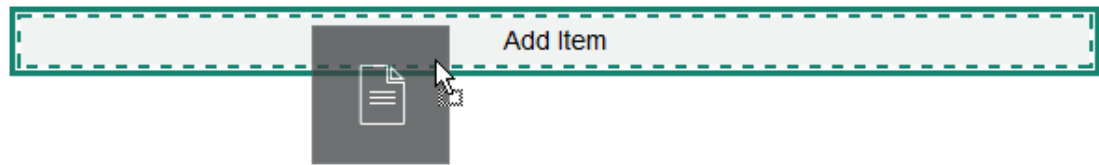
使用部分布局

部分布局可以自动将内容组织到多个列、一个垂直列表或一组选项卡式区域中。站点内容提供者可以将一个或多个部分布局添加到内容框来组织内容。要将布局添加到页中，请执行以下操作：

- 导航到您要编辑的页，并确保  设置为编辑。
- 单击  以在页上插入部分布局。选择要使用的布局类型并遵循您用于放置组件的相同指南来放置项。



位置图标  指示部分布局将置于的位置（上、下、左、右）。部分布局或内容项周围的实线边框指示您可在其中放置项，项将自动放置：






如果您是使用结构化内容项的企业用户，可以在向页中添加内容项列表时分配部分布局，部分布局将自动设置页上项的格式。

除提供的布局之外，开发人员还可以创建其他部分布局，以解决特定的布局问题或简化内容提供者的创作。主题设计人员甚至可以将部分布局构建到页布局中的内容框。请参见[开发布局](#)。

使用资产和内容项

如果您是 enterprise 用户，您的站点可以包括存储在站点集合中的数字资产和内容项或关联的资料档案库。集合是资料档案库中可以在站点上使用的资产子集。

要将数字资产或内容项添加到页，请确保将  设置为编辑，然后单击 。如果您的站点使用多个资料档案库，请选择要使用的资料档案库。有关在站点中使用多个资料档案库的信息，请参见《通过 Oracle Content Management 管理资产》中的“授予站点访问多个资料档案库的权限”。

对资产进行筛选以精确查找所需的内容。单击  并选择如何缩小选择范围。如果您没有看到任何资产，则该资产可能不是站点集合的一部分。更改筛选器以显示资料档案库中的所有集合，而不只是站点集合，看看这样是否有帮助。有关完整的详细信息，请参见对资产进行搜索、筛选和排序。

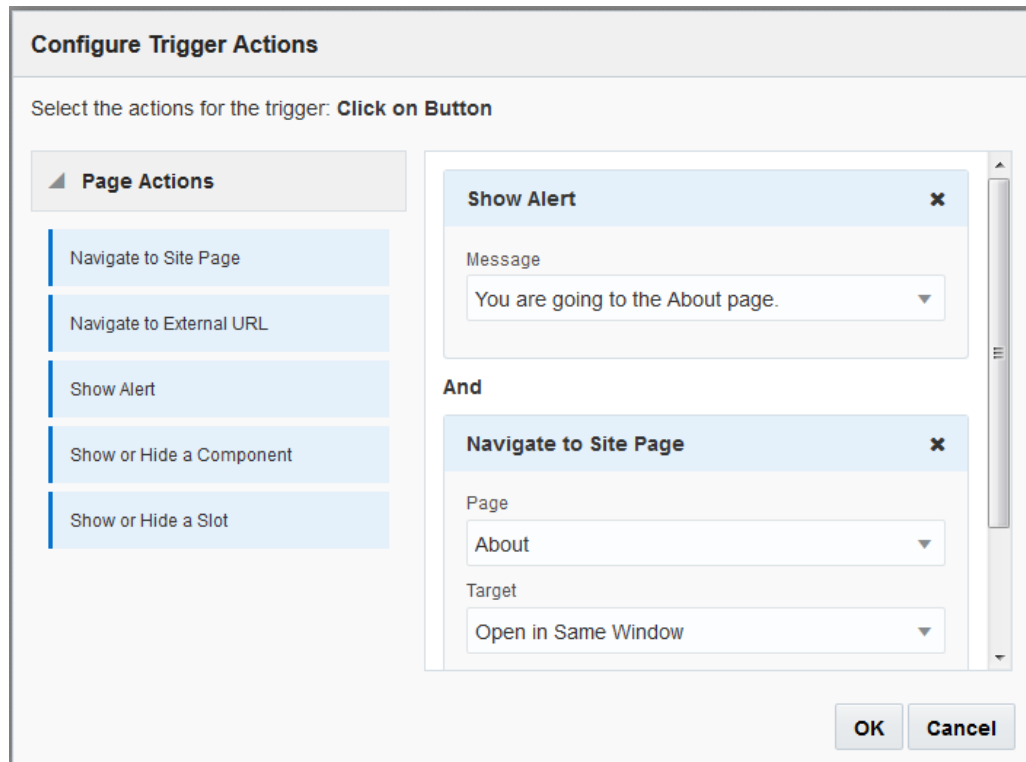
将数字资产或内容项从面板拖放到页上的内容框中。可以将图像直接嵌入段落组件的光标位置，其中有一些选项可用于输入替代文本、设置图像高度和宽度以及设置对齐方式。

如果您将项从站点集合添加到页，它会自动放在具有合适类型的组件中。例如，如果您添加图像数字资产，它会自动放在图像组件中。如果您添加内容项，它会自动放在内容项组件中。此外，您可以先添加图像或内容项组件，以后再将内容项从“内容”面板拖放到组件上。或者，您可以为组件选择设置，然后单击选择以从资产或文档列表中选择图像。

使用触发器和操作

按钮组件可用于启动一个或多个操作，例如显示或隐藏页组件和显示消息。特定组件，例如文件夹和文件列表，可以根据用户进行的选择在配套组件中启动操作。

例如，您可以配置一个按钮，在按下按钮时（触发）会将用户引导到其他页或外部 URL，并显示预警以通知用户这一更改（操作）。根据您同时使用的组件种类，您看到的显示信息将有所变化。



使用多个操作时，请考虑操作的顺序并将操作按照所希望的执行顺序排列。在以上示例中，首先列出预警操作。这让用户有时间先阅读和关闭消息，然后再重定向到页。如果先列出重定向操作，则用户可能还没有来得及阅读消息，新页就已经取代了消息。

页操作

支持触发器和操作的所有组件都支持页操作：

- 导航到站点页：选择当前站点上的页。
- 导航到外部 URL：指定外部页或站点的完整 URL。
- 显示预警：在窗口中显示指定的消息。
- 显示或隐藏组件：在当前页的组件列表中选择要显示、隐藏或切换的组件。
- 显示或隐藏内容框：在当前页的内容框列表中选择要显示、隐藏或切换的内容框。


组件特定的操作

除了页操作之外，组件还可以定义自己的操作。通过这些操作，组件可以与其他组件通信并在配套组件中启动操作。例如，当用户在文件夹列表组件中选择文件夹时，文件列表组件可以显示所选文件夹中的文件。在这种情况下，文件夹列表组件支持选择文件夹触发器，文件列表组件支持显示文件操作。

在您添加支持操作的组件到页时，该组件以及所支持的任意操作将添加到可用操作列表中。一些组件只支持操作或特定操作。一些组件提供触发器，但本身不支持任何操作。

要了解如何创建自己的组件，请参见[开发组件](#)。

例如，要为按钮组件指定一个或多个操作，请执行以下操作：

1. 单击按钮的菜单图标  并选择设置。


2. 在“设置”面板中，单击链接。
3. 单击选择链接类型，然后选择触发操作。
4. 在“可用触发器”下，单击单击按钮。
5. 在“配置触发操作”窗口中，单击并将操作从左侧的列拖放到标签为执行操作的内容框。

使用水平部分布局

您可以使用部分布局来自动确定添加到布局的组件的间隔和排列。

水平布局将添加的项按水平方向一个接一个排列。随着页宽度的增加或减少，布局按比例更改。默认情况下，项将适合一行并分配均匀的水平间隔。

要将布局添加到页中，请执行以下操作：

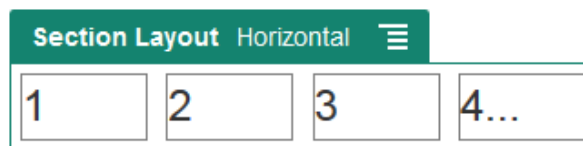
1. 导航到您要编辑的页，并确保  设置为编辑。
2. 向页中添加布局。
3. 要将其他内容放置在部分布局中，请将该内容拖放到布局上。


布局使用实线边框突出显示，并带有显示“添加项”的横幅。



您可以继续将项添加到布局，布局将相应对其设置格式。甚至可以添加其他部分布局以创建复杂的布局。

以下为水平布局，带有编号文本组件，用于显示布局中项的顺序：




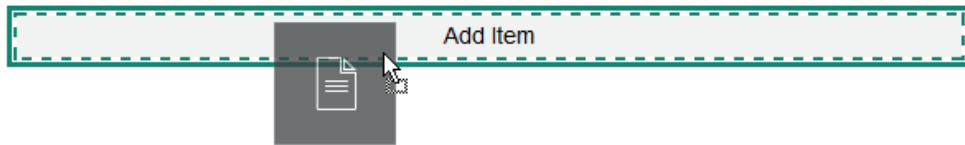
4. 要编辑组件及其外观，请单击其菜单图标 ，然后选择设置。如果不希望使用默认的按比例调整大小，您可以设置单个区域的宽度。您还可以设置对齐（左对齐、居中对齐或右对齐）。
5. 使用“一般信息”选项卡可以修改布局中单独组件的设置。
单击组件名称可查看该组件的设置。
6. 使用“背景”选项卡可修改布局的背景设置。
请参见[更改背景或主题](#)。

使用两列或三列布局

您可以使用多列部分布局来自动确定添加到布局的组件的间隔和排列。

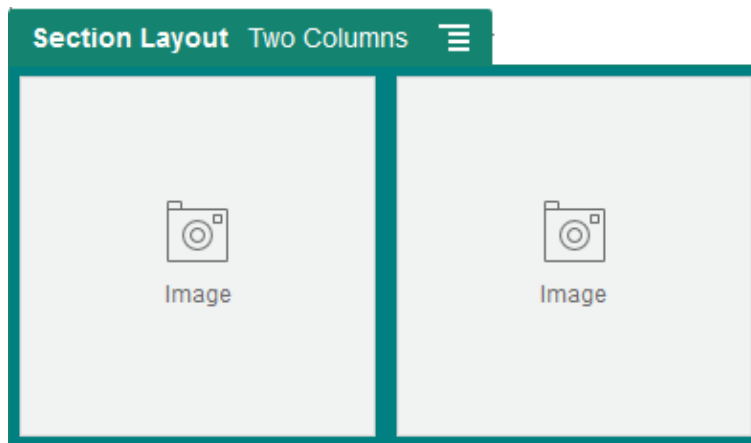
要将布局添加到页中，请执行以下操作：

1. 导航到您要编辑的页，并确保  设置为编辑。
2. 向页中添加布局。
3. 要将其他内容放置在部分布局中，请将该内容拖放到布局上。
布局使用实线边框突出显示，并带有显示“添加项”的横幅。




您可以继续将项添加到布局，布局将相应对其设置格式。您甚至可以添加其他部分布局以创建复杂的布局，或者添加组件组。

下面是图像组件作为占位符的两列布局：



如果添加更多项，则这些项显示在额外的行中，每一行最多显示两个项。

4. 要编辑组件及其外观，请单击其菜单图标 ，然后选择设置。
5. 使用“一般信息”选项卡可以修改布局中单独组件的设置。
单击组件名称可查看该组件的设置。
6. 使用“背景”选项卡可修改布局的背景设置。
7. 选择定制设置可设置有关所显示内容的其他默认设置。
 - 第一列宽度 (%)：以布局可用空间百分比形式指定列宽度。
 - 第二列宽度 (%)：以布局可用空间百分比形式指定列宽度。
 - 第三列宽度 (%)：以布局可用空间百分比形式指定列宽度。


- 响应断点（以像素为单位）：对于在可用显示大小改变时自动重新设置内容格式的响应页设计，指定部分布局在标准两列布局和您下面指定的响应行为选项之间切换的宽度（以像素为单位）。
- 响应行为：选择在可用显示大小小于响应断点值时布局如何更改。
 - 无操作：不调整布局行为。
 - 堆叠列：在单一列中从上到下排列项，第二列中的项排在第一列中的所有项之后，以此类推。
 - 隐藏第一列：隐藏第一列中的内容以便为其余列提供更多空间。
 - 隐藏第二列：隐藏第二列中的内容以便为其余列提供更多空间。
 - 隐藏第三列：隐藏第三列中的内容以便为其余列提供更多空间。
 - 隐藏这两列：隐藏布局中的所有内容。
 - 将第二列移动到第一列之下：在单一列中排列项，第二列中的所有项排在第一列中的所有项之后。
 - 将第一列移动到第二列之下：在单一列中排列项，第一列中的所有项排在第二列中的所有项之后。

使用垂直部分布局

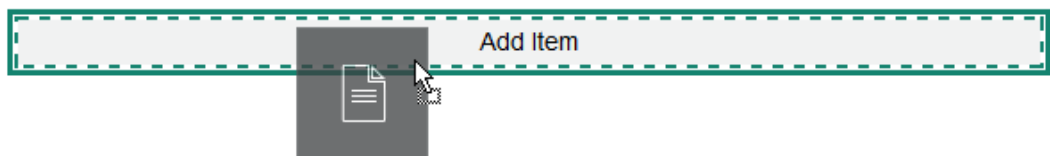
您可以使用部分布局来自动确定添加到布局的组件的间隔和排列。

垂直布局将添加的项按垂直方向一个接一个排列。

要将布局添加到页中，请执行以下操作：

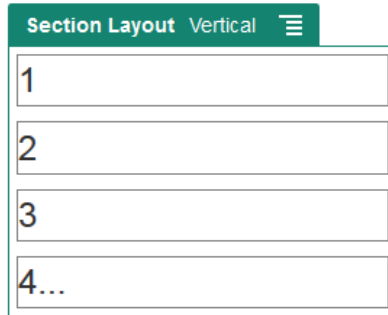
1. 导航到您要编辑的页，并确保  设置为编辑。
2. 向页中添加布局。
3. 要将其他内容放置在部分布局中，请将该内容拖放到布局上。


布局使用实线边框突出显示，并带有显示“添加项”的横幅。



您可以继续将项添加到布局，布局将相应对其设置格式。甚至可以添加其他部分布局以创建复杂的布局。

下面是一个垂直布局，带有编号文本组件，用于显示布局中项的顺序：



4. 要编辑组件及其外观，请单击其菜单图标 ，然后选择设置。
5. 使用“一般信息”选项卡可以修改布局中单独组件的设置。
单击组件名称可查看该组件的设置。
6. 使用“背景”选项卡可修改布局的背景设置。

使用选项卡式部分布局

您可以使用选项卡式部分布局，对添加到布局中的组件创建间隔和排列。


要将布局添加到页中，请执行以下操作：

1. 导航到您要编辑的页，并确保  设置为编辑。
2. 向页中添加布局。默认情况下，会随布局添加单个选项卡。单击新建选项卡可添加额外的选项卡。
3. 要在选项卡中放置其他内容，请将内容拖放到选项卡上。
您可以继续将项添加到布局，布局将相应对其设置格式。您甚至可以添加其他部分布局以创建复杂的布局，或者添加组件组。
4. 要编辑选项卡及其外观，请单击其菜单图标 ，然后选择设置。
5. 使用“背景”部分修改选项卡的背景设置。
您可以为选项卡使用图像、更改其位置等。请参见[更改背景或主题](#)。
6. 使用“样式”部分修改选项卡的其他外观设置。选择某个样式（例如细线、框架等），或者定制选项卡的边框和圆角。

使用滑块部分布局


可以使用滑块部分布局创建内容，该内容保持不变，直到它“滑”到一侧，以便为新内容腾出空间。用户通过单击幻灯片部分下面的导航点，可以在幻灯片之间导航。

要将滑块布局添加到页中，请执行以下操作：

1. 导航到您要编辑的页，并确保  设置为编辑。
2. 向页中添加布局。默认情况下，随布局添加单个幻灯片。单击新建幻灯片可添加额外的幻灯片。

3. 要将内容放在特定幻灯片上，可在滑块中显示该幻灯片，然后将该组件或内容项拖放到幻灯片上。

您可以继续将项添加到幻灯片并根据需要来定位它们。您可以添加其他部分布局以创建复杂的布局，或者添加组件组，让您根据需要来定位内容。

4. 要编辑幻灯片布局及其外观，请单击菜单图标 ，然后选择设置。
5. 使用“背景”部分修改整个幻灯片区域的背景设置。

请参见[更改背景或主题](#)。您可以使用图像、更改其位置等。

6. 使用“样式”部分修改其他外观设置。选择某个样式（例如细线、框架等），或者定制幻灯片的边框和圆角。
7. 要修改每个单独幻灯片的设置，请单击相应幻灯片的名称以打开其设置。根据需要为每个幻灯片更改背景、样式等。

10

使用内置组件

组件、部分布局、数字资产和结构化内容让您灵活地提供用户所需的内容和功能。

本节提供有关使用单个组件的信息。有关组件的其他信息，请参见以下内容：

- 要了解如何在站点中使用组件，请参见[排列页面内容](#)。
- 要了解如何管理组件，请参见[管理定制组件和布局](#)。
- 要了解如何创建自己的组件，请参见[开发组件](#)。

基本组件

- [标题](#)
- [段落](#)
- [纯文本](#)
- [按钮](#)

结构组件

- [分隔线](#)
- [分隔符](#)

媒体组件

- [图像](#)
- [图库](#)
- [图库网格](#)
- [YouTube 视频](#)
- [视频](#)

文档组件

- [文档](#)
- [文件夹列表](#)
- [文件列表](#)
- [文档管理器](#)
- [项目库](#)

社交组件

- [社交栏](#)
- [Facebook 点赞和推荐](#)
- [Twitter 分享和关注](#)

- [对话组件](#)
- [对话列表](#)

流程组件

- [流程开始表单](#)
- [处理任务列表](#)
- [任务详细信息表单](#)

内容项

- [内容项组件](#)
- [内容占位符](#)
- [内容列表](#)
- [内容搜索](#)
- [推荐](#)

其他组件

- [地图](#)
- [标题](#)
- [文章](#)
- [图像（具有文本）](#)
- [组件组](#)
- [在页上使用 Cobrowse](#)
- [Oracle Intelligent Advisor](#)
- [Oracle Visual Builder](#)


基本组件

基本组件为文本内容提供构建块。


- [标题](#)
- [段落](#)
- [纯文本](#)
- [按钮](#)

标题

要添加标题组件并设置格式，请执行以下操作：

1. 导航到您要编辑的页，并确保  设置为编辑。
2. 向页中添加组件。
3. 单击标题组件以输入标题文本。文本采用组件的默认样式的格式设置。


4. 要在标题文本中添加链接，请执行以下操作：

- a. 输入和选择要用作链接文本的文本，然后单击 .
- b. 单击选择链接类型并选择以下选项之一：
 - **Web 页**：指定外部页或站点的完整 URL，并选择在何处打开链接。
 - **站点页**：使用页选择器选择当前站点上的页，并选择在何处打开链接。可以用 `key1=value1&key2=value2` 格式指定其他 URL 参数。系统支持空值；例如 `key1=&key2=value2`。此外，您还可以指定 URL 锚定点，但需要将一个特殊的“锚定点”部分布局添加到目标站点页上您希望解析锚定点链接的位置，并在布局设置中指定在定义触发操作时使用的同一锚定点名称。

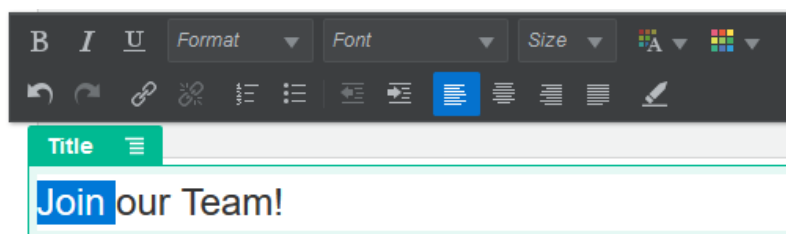
 注：

OCE 工具包中包含使用 URL 锚定点所需的“锚定点”部分布局。有关如何获取该工具包的信息，请参见使用 OCE 工具包进行开发。


- **文件下载**：从资料档案库下载所选文件。选择文件。
如果选择数字资产，则可以选择特定重现。如果未选择重现，则将使用原始大小。如果要在发布站点时发布最新的资产版本，请选择使用最新的资产版本。如果未选择使用最新的资产版本，则将使用发布的最新版本，而不会使用更新的草稿版本（如果存在）。
- **文件预览**：在暗灰色不活动的页版本上以重叠方式预览所选文件并选择性地下载此文件。
- **内容项**：从关联的资产资料档案库中选择内容项，选择要显示的详细信息页，以及将页定向到在同一窗口或新窗口中打开。
- **电子邮件**：指定有效的电子邮件地址，并可指定主题（可选）。生成的消息将在默认电子邮件客户端中打开并通过它来发送。
- **地图**：输入有效的地址或坐标，并选择您要在桌面和移动浏览器中的何处打开地图。
- **电话**：输入有效的电话号码。

要删除链接，请在链接文本中的任意位置单击，然后单击 .

5. 如果要更改默认格式设置，请选择要设置格式的文本，然后在格式设置工具栏中选择任意选项，例如字体、颜色或对齐方式。根据您的主题，这些选项可能并非所有都可见。








您进行的更改将立即应用。这些格式设置更改在应用时会覆盖基本样式。在更改基本样式时，格式覆盖保持不变。

6. 要删除使用这些选项应用的格式设置，请选择文本，然后单击 。

段落

要添加段落组件并设置格式，请执行以下操作：


1. 导航到您要编辑的页，并确保  设置为编辑。
2. 向页中添加组件。
3. 单击段落组件以输入文本。文本采用组件的默认样式的格式设置。
4. 要向段落添加视频 ()、图像 () 或内容项 ()，请单击相应图标。选择要添加的资产并根据需要调整设置。例如，插入内容项，并选择用来控制段落中复杂布局的不同视图。
5. 要在段落组件中添加额外的段落，只需按 Enter。
6. 要在段落中添加链接，请执行以下操作：
 - a. 输入并选择您要用作链接文本的文本，然后单击 。
 - b. 单击选择链接类型并选择以下选项之一：
 - **Web 页**：指定外部页或站点的完整 URL，并选择在何处打开链接。
 - **站点页**：使用页选择器选择当前站点上的页，并选择在何处打开链接。可以用 `key1=value1&key2=value2` 格式指定其他 URL 参数。系统支持空值；例如 `key1=&key2=value2`。此外，您还可以指定 URL 锚定点，但需要将一个特殊的“锚定点”部分布局添加到目标站点页上您希望解析锚定点链接的位置，并在布局设置中指定在定义触发操作时使用的同一锚定点名称。

注：

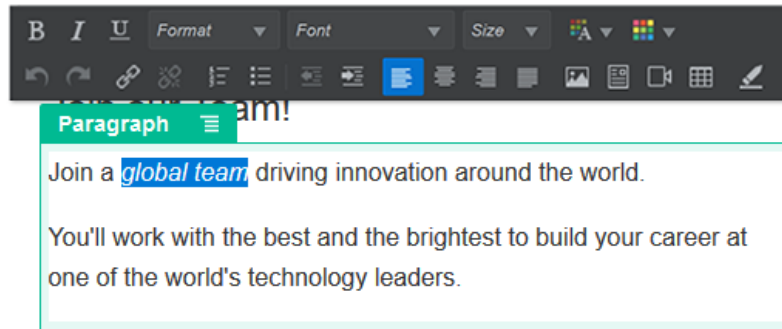
OCE 工具包中包含使用 URL 锚定点所需的“锚定点”部分布局。有关如何获取该工具包的信息，请参见使用 OCE 工具包进行开发。

- **文件下载**：从资料档案库下载所选文件。选择文件。


如果选择数字资产，则可以选择特定重现。如果未选择重现，则将使用原始大小。如果要在发布站点时发布最新的资产版本，请选择使用最新的资产版本。如果未选择使用最新的资产版本，则将使用发布的最新版本，而不会使用更新的草稿版本（如果存在）。
- **文件预览**：在暗灰色不活动的页版本上以重叠方式预览所选文件并选择性地下载此文件。
- **内容项**：从关联的资产资料档案库中选择内容项，选择要显示的详细信息页，以及将页定向到在同一窗口或新窗口中打开。
- **电子邮件**：指定有效的电子邮件地址，并可指定主题（可选）。生成的消息将在默认电子邮件客户端中打开并通过它来发送。
- **地图**：输入有效的地址或坐标，并选择您要在桌面和移动浏览器中的何处打开地图。
- **电话**：输入有效的电话号码。

要删除链接，请在链接文本中的任意位置单击，然后单击 。

7. 如果要更改任何文本部分的默认格式设置，请选择要设置格式的文本，然后在格式设置工具栏中选择任意选项，例如字体、颜色或对齐方式。根据您的主题，这些选项可能并非所有都可见。



您进行的更改将立即应用。这些格式设置更改在应用时会覆盖基本样式。在更改基本样式时，格式覆盖保持不变。


8. 要删除应用到这些选项的格式设置，请选择文本，然后单击 。

纯文本

使用文本组件可以将文本添加到页面，并可以使用主题中定义的样式专门设置其格式。编辑内容时，格式设置工具栏不可用，因此您无法使用定制格式覆盖所选样式。

如果您为标题等元素使用标准化格式，此功能非常有用。您可以更新主题中定义的样式，并且可以在整个站点内自动更新相关文本，而无需使用会阻止更新的格式覆盖。


要添加文本组件，请执行以下操作：

1. 导航到您要编辑的页，并确保  设置为编辑。
2. 向页中添加组件。
3. 单击段落组件以输入文本。文本采用组件默认样式的格式（<p> 或段落元素）。

要再添加一个段落，只需按 Enter。

注：

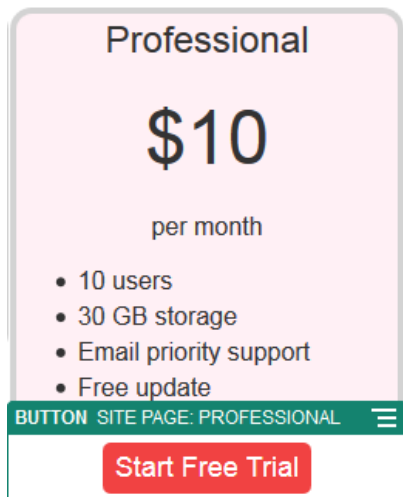
您可以将内容从其他源复制并粘贴到文本组件，但图像和基础 HTML 标记会被删除。

4. 要更改文本组件的基本样式，请执行以下操作：
 - a. 单击组件的菜单图标  并选择设置。
 - b. 单击选择样式，然后从菜单中选择样式。



列表包括段落和标题的标准 HTML 标记。您选择的样式会向内容分配相关标记。组件反映了所选基本样式。

按钮

使用按钮可使链接或其他功能在页上更明显。



要将按钮添加到页，请执行以下操作：

1. 导航到您要编辑的页，并确保  设置为编辑。
2. 向页中添加组件。
3. 要调整按钮的属性，请单击其菜单图标 ，然后选择设置。

您可以指定按钮上的文本（标签）、大小、对齐方式和按钮的其他显示选项。

注：

如果将宽度和高度字段设置为 0（零），则按钮将自动调整大小以适合您为标签指定的文本。

要指定背景色、字体、边框和其他设置，请单击样式选项卡。您可以从当前主题中的预定义样式中进行选择。根据您的主题，可能还可以单击自定义来指定自己的值。

4. 要将链接或其他操作与按钮关联，请执行以下操作：
 - a. 在“设置”面板中，单击链接。
 - b. 单击选择链接类型并选择以下选项之一：
 - **无链接**：用户单击按钮时，按钮不执行任何操作。
 - **Web 页**：指定外部页或站点的完整 URL，并选择在何处打开链接。
 - **站点页**：使用页选择器选择当前站点上的页，并选择在何处打开链接。可以用 `key1=value1&key2=value2` 格式指定其他 URL 参数。系统支持空值；例如 `key1=&key2=value2`。此外，您还可以指定 URL 锚定点，但需要将一个特殊的“锚定点”部分布局添加到目标站点页上您希望解析锚定点

链接的位置，并在布局设置中指定在定义触发操作时使用的同一锚定点名称。

 **注：**

OCE 工具包中包含使用 URL 锚定点所需的“锚定点”部分布局。有关如何获取该工具包的信息，请参见使用 OCE 工具包进行开发。

- **文件下载：**从资料档案库下载所选文件。选择文件。
如果选择数字资产，则可以选择特定重现。如果未选择重现，则将使用原始大小。如果要在发布站点时发布最新的资产版本，请选择使用最新的资产版本。如果未选择使用最新的资产版本，则将使用发布的最新版本，而不会使用更新的草稿版本（如果存在）。
- **文件预览：**在暗灰色不活动的页版本上以重叠方式预览所选文件并选择性下载此文件。
- **内容项：**从关联的资产资料档案库中选择内容项，选择要显示的详细信息页，以及将页定向到在同一窗口或新窗口中打开。
- **电子邮件：**指定有效的电子邮件地址，并可指定主题（可选）。生成的消息将在默认电子邮件客户端中打开并通过它来发送。
- **触发操作：**选择在按下按钮时执行的一个或多个页操作。页操作包括：
 - **导航到站点页：**选择当前站点上的页。
 - **导航到外部 URL：**指定外部页或站点的完整 URL。
 - **显示预警：**在窗口中显示指定的消息。
 - **显示或隐藏组件：**在当前页上选择要显示、隐藏或切换的组件。
 - **显示或隐藏内容框：**在当前页上选择要显示、隐藏或切换的内容框。
- **地图：**输入有效的地址或坐标，并选择您要在桌面和移动浏览器中的何处打开地图。
- **电话：**输入有效的电话号码。

请参见[使用触发器和操作](#)。

结构组件


使用结构组件可以分隔页上的内容。

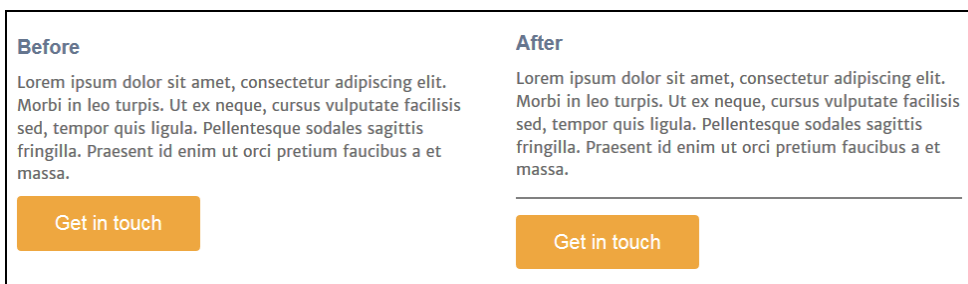
- [分隔线](#)
- [分隔符](#)


分隔线

使用分隔线（水平线条）可以在栏之间或整个页中创建视觉上的分隔效果。

要将分隔线添加到页，请执行以下操作：

1. 导航到您要编辑的页，并确保  设置为编辑。
2. 向页中添加组件。




3. 要调整分隔线的属性，请单击其菜单图标 ，然后选择设置。

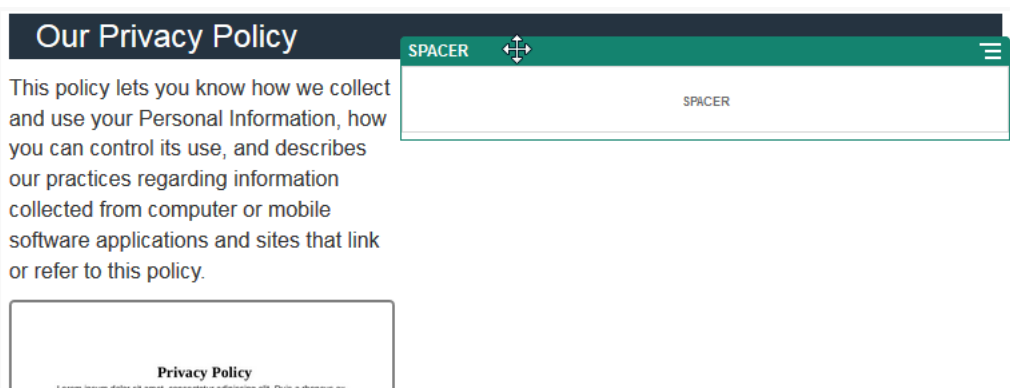
您可以通过调整左右间距选项，指定分隔线的水平位置和长度。线条颜色、宽度以及其他设置由组件样式定义。您可以在当前主题的预定义样式中选择，也可以单击定制以指定自己的值。


分隔符

管理页上的“空白”可以使页更具视觉吸引力且更容易阅读。使用分隔符组件可以添加空白，而无需覆盖在样式或其他组件中定义的间距。

要在页中的组件之间添加垂直空白间隔，请执行以下操作：

1. 导航到您要编辑的页，并确保  设置为编辑。
2. 向页中添加组件。



3. 要调整分隔符的高度，请单击其菜单图标 ，然后选择设置。

媒体组件

媒体组件在页上添加图像和视频。

- [图像](#)
- [图库](#)
- [图库网格](#)
- [YouTube 视频](#)

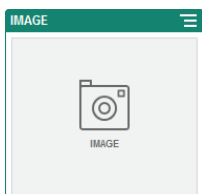
- 视频


图像

图像可以吸引查看者的注意力，邀请查看者浏览页面上的不同内容区域。

要将图像添加到页，请执行以下操作：

1. 导航到您要编辑的页，并确保  设置为编辑。
2. 向页中添加组件。在您选择要使用的图像之前，图像组件显示占位符图像。















3. 要选择图像，请单击菜单图标 ，选择设置，然后单击“图像”字段旁的选择。
4. 可从站点资料档案库中选择图像，从与您共享的文档文件夹中选择图像，或将图像上载到文档文件夹。

注：

此时窗口中将显示所有可用文件。您必须选择适合上下文的文件类型。例如，如果选择图像文件，则必须选择具有有效图像格式（GIF、JPG、JPEG、PNG 或 SVG）的文件。

- a. 找到并选择要使用的图像。

如果您未看到任何数字资产，请单击 ，然后将集合筛选器更改为全部。
 - b. 如果从文档文件夹中选择了图像，则可以链接到文件，而不必将它复制到站点。要链接到文件，请选择使用对原始文件的引用而不是将文件复制到站点。如果没有选择此选项，文件副本将随站点一起存储并可供从站点中引用。链接到原始文件可以避免复制内容。即使文件的权限发生更改或者限制查看，站点访问者仍可以通过链接来查看内容。
 - c. 单击确定。
 - d. 如果选择了数字资产，则可以选择特定重现。如果未选择重现，则将使用原始大小。如果要在发布站点时发布最新的资产版本，请选择使用最新的资产版本。
5. 要编辑图像，请单击 ，并通过以下任一操作编辑图像：
 - 要剪裁图像，请单击  剪裁。在剪裁工具栏中选择预定义的图像纵横比之一，或根据需要在图像上拖动剪裁手柄。满意后，在剪裁工具栏中单击剪裁。
 - 要旋转或翻转图像，请单击  旋转。在旋转工具栏中，输入定制旋转角度，使用按钮向左或向右旋转图像，或者选择水平或垂直翻转图像。

- 要将水印添加到图像，请单击  水印。将文本添加到图像，然后使用水印工具根据需要更改文本大小、样式、颜色和不透明度。
 - 要更改图像格式，请单击  选项，然后从格式下拉列表中选择新格式。
 - 要更改背景色，请单击  选项，然后从背景色下拉菜单中选择一个选项。
 - 如果编辑的是 .jpg 或 .webp（在 Google Chrome 浏览器中可用）格式的图像，您可以更改图像质量以创建较小的文件。请单击  选项，然后在质量框中输入新的百分比。
 - 要撤消或重做更改，请单击  或 。要删除所做的所有更改，请单击重置。
 - 要更改图像的缩放比例，请使用缩放控件 ()。
6. 使用设置菜单  的一般信息面板添加标题或修改间距、对齐方式、样式及其他选项。
7. 可以将链接或其他操作与图像相关联：
- a. 在“设置”面板中，单击链接。
 - b. 选择以下选项之一：
 - **无链接**：用户单击图像时，图像不执行任何操作。
 - **Web 页**：指定外部页或站点的完整 URL，并选择在何处打开链接。
 - **站点页**：使用页选择器选择当前站点上的页，并选择在何处打开链接。可以用 `key1=value1&key2=value2` 格式指定其他 URL 参数。系统支持空值；例如 `key1=&key2=value2`。此外，您还可以指定 URL 锚定点，但需要将一个特殊的“锚定点”部分布局添加到目标站点页上您希望解析锚定点链接的位置，并在布局设置中指定在定义触发操作时使用的同一锚定点名称。

 注：

OCE 工具包中包含使用 URL 锚定点所需的“锚定点”部分布局。有关如何获取该工具包的信息，请参见使用 OCE 工具包进行开发。

- **文件下载**：从资料档案库下载所选文件。选择文件。



如果选择数字资产，则可以选择特定重现。如果未选择重现，则将使用原始大小。如果要在发布站点时发布最新的资产版本，请选择使用最新的资产版本。如果未选择使用最新的资产版本，则将使用发布的最新版本，而不会使用更新的草稿版本（如果存在）。
- **文件预览**：所选文件将以重叠形式显示在页上。
- **内容项**：从关联的资产资料档案库中选择内容项，选择要显示的详细信息页，以及将页定向到在同一窗口或新窗口中打开。
- **电子邮件**：指定有效的电子邮件地址，并可指定主题（可选）。生成的消息将在默认电子邮件客户端中打开并通过它来发送。

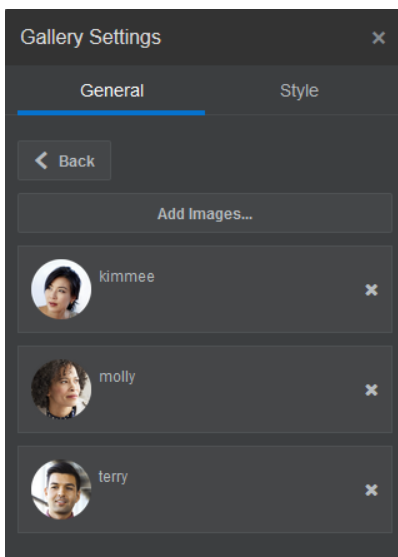
- **图像预览：**所选图像将以重叠形式显示在页上。
- **地图：**输入有效的地址或坐标，并选择您要在桌面和移动浏览器中的何处打开地图。
- **电话：**输入有效的电话号码。

图库

使用图像库可以表示一系列图像。您可以选择让图像自动循环或者让用户手动向前通过图像。

要将图库添加到页，请执行以下操作：

1. 导航到您要编辑的页，并确保  设置为编辑。
2. 向页中添加组件。在您选择要使用的图像之前，图库组件显示占位符图像。
3. 要将一个或多个图像添加到图库中，请单击其菜单图标 ，选择设置，然后单击一般信息选项卡上的图像。
4. 单击添加图像。



5. 选择一个或多个图像。

注：

此时窗口中将显示所有可用文件。您必须选择适合上下文的文件类型。例如，如果选择图像文件，则必须选择具有有效图像格式（GIF、JPG、JPEG、PNG 或 SVG）的文件。

- a. 找到并选择要使用的图像。

如果您未看到任何数字资产，请单击 ，然后将集合筛选器更改为全部。

- b. 如果从文档文件夹中选择了图像，则可以链接到文件，而不必将它复制到站点。要链接到文件，请选择使用对原始文件的引用而不是将文件复制到站点。如果没有选择此选项，文件副本将随站点一起存储并可供从站点中引用。链接到原始文件可以避免复

制内容。即使文件的权限发生更改或者限制查看，站点访问者仍可以通过链接来查看内容。

- c. 单击确定。
- d. 如果选择了数字资产，则可以选择特定重现。如果未选择重现，则将使用原始大小。如果要在发布站点时发布最新的资产版本，请选择使用最新的资产版本。

所选图像将添加到图像列表中。拖放图像以在列表中重新排序。各个图像的默认标题是不带扩展名的文件名。

6. 要更改特定图像的标题、说明或其他选项，请单击列表中的图像并进行更改。

您还可以将链接或其他操作与图库中的图像关联：


- a. 在特定图像的“设置”面板中，单击“链接”字段。
- b. 选择以下选项之一：
 - 无链接：用户单击图像时，图像不执行任何操作。
 - **Web 页**：指定外部页或站点的完整 URL，并选择在何处打开链接。
 - **站点页**：使用页选择器选择当前站点上的页，并选择在何处打开链接。可以用 `key1=value1&key2=value2` 格式指定其他 URL 参数。系统支持空值；例如 `key1=&key2=value2`。此外，您还可以指定 URL 锚定点，但需要将一个特殊的“锚定点”部分布局添加到目标站点页上您希望解析锚定点链接的位置，并在布局设置中指定在定义触发操作时使用的同一锚定点名称。

 **注：**

OCE 工具包中包含使用 URL 锚定点所需的“锚定点”部分布局。有关如何获取该工具包的信息，请参见使用 OCE 工具包进行开发。

- **文件下载**：从资料档案库下载所选文件。选择文件。



如果选择数字资产，则可以选择特定重现。如果未选择重现，则将使用原始大小。如果要在发布站点时发布最新的资产版本，请选择使用最新的资产版本。如果未选择使用最新的资产版本，则将使用发布的最新版本，而不会使用更新的草稿版本（如果存在）。
 - **内容项**：从关联的资产资料档案库中选择内容项，选择要显示的详细信息页，以及将页定向到在同一窗口或新窗口中打开。
 - **电子邮件**：指定有效的电子邮件地址，并可指定主题（可选）。生成的消息将在默认电子邮件客户端中打开并通过它来发送。
 - **地图**：输入有效的地址或坐标，并选择您要在桌面和移动浏览器中的何处打开地图。
 - **电话**：输入有效的电话号码。
- c. 单击返回可返回到图像设置面板。再次单击返回可返回到图像列表，以选择其他图像进行更新。
 - d. 当您完成更新单个图像时，单击返回可指定图库选项。
7. 要对图库中的图像调整大小和缩放，请执行以下操作：
 - a. 使用宽度指定内容框中图库的宽度，以像素为单位。单击填充之外的对齐选项来指定宽度。设置宽度之后，您可以使用填充将图像拉伸到指定宽度。

- b. 选择缩放选项以调整图像在图库中的显示形式：
 - 剪裁：两个尺寸（宽度或高度）中较小的那个尺寸将缩放以适合可用空间，较大的尺寸将剪裁以避免拉伸图像。
 - 适合：缩放各个图像，使得整个图像适合可用空间，而不会导致图像变形。
 - 拉伸：两个尺寸（宽度或高度）中较大的那个将缩放以适合可用空间，较小的尺寸将拉伸以填充可用空间。
 - 无：使用图像的完整像素分辨率，如果无法容纳在可用空间中，则会按照可用空间来剪裁。
8. 要帮助用户浏览图库中的图像，请执行以下操作：
 - a. 选择导航方法：
 - 缩略图：在图库下方按顺序显示图库中的图像列表。用户可以单击缩略图图像转到图库中的关联图像。
 - 索引器：在图库下方显示一系列按钮  来表示图库中的每个图像。用户可以单击按钮转到图库中的关联图像。
 - 无：不提供直观导航。用户可以左右划动以显示相邻图像。此手动方法始终可用。
 - b. 单击显示上一个/下一个可以在每个图像上包括箭头图标，以转到图库中的上一个或下一个图像。
9. 您可以通过选择显示标题框，选择显示或隐藏图像的标题。
10. 要使图库自动循环通过图像，请选择自动播放并指定转换时间和显示时间。
用户仍可以使用您设置的所有手动浏览功能。

图库网格

使用图库网格可以在行和列中一次提供一组图像。

要将图库添加到页，请执行以下操作：

1. 导航到您要编辑的页，并确保  设置为编辑。
2. 向页中添加组件。在选择要使用的图像之前，图库网格组件会显示占位符图像。
3. 要将一个或多个图像添加到图库中，请单击其菜单图标 ，选择设置，然后单击一般信息选项卡上的图像。
4. 单击添加图像。
5. 选择一个或多个图像。

注：

此时窗口中将显示所有可用文件。您必须选择适合上下文的文件类型。例如，如果选择图像文件，则必须选择具有有效图像格式（GIF、JPG、JPEG、PNG 或 SVG）的文件。

- a. 找到并选择要使用的图像。

如果您未看到任何数字资产，请单击 ，然后将集合筛选器更改为全部。

- b. 如果从文档文件夹中选择了图像，则可以链接到文件，而不必将它复制到站点。要链接到文件，请选择使用对原始文件的引用而不是将文件复制到站点。如果没有选择此选项，文件副本将随站点一起存储并可供从站点中引用。链接到原始文件可以避免复制内容。即使文件的权限发生更改或者限制查看，站点访问者仍可以通过链接来查看内容。
- c. 单击确定。
- d. 如果选择了数字资产，则可以选择特定重现。如果未选择重现，则将使用原始大小。如果要在发布站点时发布最新的资产版本，请选择使用最新的资产版本。

所选图像将添加到图像列表中。拖动图像以在列表（和网格）中重新排序。各个图像的默认标题是不带扩展名的文件名。

6. 要更改特定图像的标题、说明或其他选项，请单击列表中的图像并进行更改。

您还可以将链接或其他操作与图库中的图像关联：

- a. 在特定图像的“设置”面板中，单击“链接”字段。
- b. 选择以下选项之一：
 - **无链接**：用户单击图像时，图像不执行任何操作。
 - **Web 页**：指定外部页或站点的完整 URL，并选择在何处打开链接。
 - **站点页**：使用页选择器选择当前站点上的页，并选择在何处打开链接。可以用 `key1=value1&key2=value2` 格式指定其他 URL 参数。系统支持空值；例如 `key1=&key2=value2`。此外，您还可以指定 URL 锚定点，但需要将一个特殊的“锚定点”部分布局添加到目标站点页上您希望解析锚定点链接的位置，并在布局设置中指定在定义触发操作时使用的同一锚定点名称。

 **注：**

OCE 工具包中包含使用 URL 锚定点所需的“锚定点”部分布局。有关如何获取该工具包的信息，请参见使用 OCE 工具包进行开发。

- **文件下载**：从资料档案库下载所选文件。选择文件。

如果选择数字资产，则可以选择特定重现。如果未选择重现，则将使用原始大小。如果要在发布站点时发布最新的资产版本，请选择使用最新的资产版本。如果未选择使用最新的资产版本，则将使用发布的最新版本，而不会使用更新的草稿版本（如果存在）。
- **内容项**：从关联的资产资料档案库中选择内容项，选择要显示的详细信息页，以及将页定向到在同一窗口或新窗口中打开。
- **电子邮件**：指定有效的电子邮件地址，并可指定主题（可选）。生成的消息将在默认电子邮件客户端中打开并通过它来发送。
- **图像预览**：所选图像将以重叠形式显示在页上。
- **地图**：输入有效的地址或坐标，并选择您要在桌面和移动浏览器中的何处打开地图。
- **电话**：输入有效的电话号码。

- c. 单击返回可返回到图像设置面板。再次单击返回可返回到图像列表，以选择其他图像进行更新。
 - d. 当您完成更新单个图像时，单击返回可指定图库选项。
7. 使用布局选项来排列网格中的图像。
- 砖石结构
 - 列
 - 定制
- 每个这些选项在后面的步骤中说明。
8. 选择砖石结构布局可自动在可用空间中图像排成一行。
- 生成的行具有一致的高度，但没有定义列。



- a. 指定高度可等比例地将所有图像缩放到指定的高度，以像素为单位。
 - b. 指定图像间距可增加或减少图像的行之间的间距。
9. 选择列布局可将图像按行和列排列。
- a. 选择缩放选项以调整图像在网格中的显示形式：
 - 剪裁：两个尺寸（宽度或高度）中较小的那个尺寸将缩放以适合可用空间，较大的尺寸将剪裁以避免拉伸图像。
 - 适合：缩放各个图像，使得整个图像适合可用空间，而不会导致图像变形。

例如，以下网格使用四列并缩放六个图像以适合大小：



下面是同一个网格，其中对图像进行了剪裁：



- b. 指定纵横比可确定网格中的单元形状。
 - 正方形：纵横比 1:1。
 - 横向：纵横比 16:9。
 - 纵向：纵横比 9:16。
 - 定制：指定自己的纵横比值。
 - c. 指定列数。
网格会自动调整以创建等宽的列。
 - d. 指定图像间距以增加或减少图像的行和列之间的间距。
10. 选择定制布局可根据您指定的图像大小和宽度，按行和列排列图像。
- a. 选择缩放选项以调整图像在网格中的显示形式：
 - 剪裁：两个尺寸（宽度或高度）中较小的那个尺寸将缩放以适合可用空间，较大的尺寸将剪裁以避免拉伸图像。
 - 适合：缩放各个图像，使得整个图像适合可用空间，而不会导致图像变形。
 - b. 指定图像高度和图像宽度可确定网格中的单元形状。
网格会自动调整以创建具有您指定尺寸的单元。
 - c. 指定图像间距以增加或减少图像的行和列之间的间距。
11. 指定对齐方式、宽度和间距选项以在内容框中放置网格。
- 使用宽度指定内容框中图库的宽度，以像素为单位。单击填充之外的对齐选项来指定宽度。设置宽度之后，您可以使用填充将图像拉伸到指定宽度。


YouTube 视频

在页中包括 YouTube 流视频可以增添活力，实现互动效果。

要使用 YouTube 所托管视频之外的视频，请参见[视频](#)

要在页中添加 YouTube 视频，请执行以下操作：

1. 导航到您要编辑的页，并确保  设置为编辑。
2. 向页中添加组件。在您选择要使用的视频之前，该组件显示占位符图像。

3. 要指定所使用的 YouTube 视频并调整其显示属性，请单击其菜单图标 ，然后选择设置。
4. 找到要使用的视频，然后复制其 URL 并粘贴到 **YouTube URL** 字段中。
5. 指定任意显示选项：
 - 显示控件：启用屏幕和设备选项，以允许用户手动控制视频回放。
 - 显示信息：在视频的左上角临时包括视频说明。
 - 自动播放：自动开始视频回放。
 - 循环：在视频播放完成后自动重复。
6. 使用为视频定义的纵横比（自动），或者选择其他纵横比以确定所显示视频的形狀。
7. 指定任意对齐方式或间距选项以放置视频。


视频

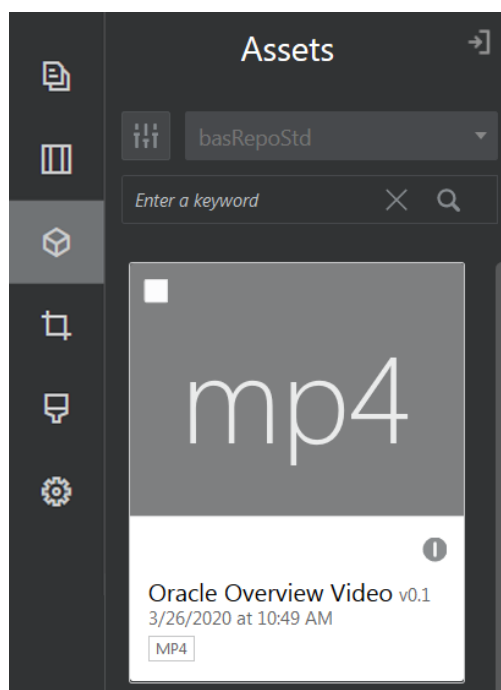
包括来自 Oracle Content Management 的视频以将运动和视觉体验添加到页中。

Oracle Content Management 提供了多个用来向页中添加视频的选项。您可以嵌入 YouTube 站点中托管的 [YouTube 视频](#)，也可以添加 Oracle Content Management 存储、管理和传送的视频，以利用增强视频提供的自动转码和优化的流式处理。这在站点为具有不同功能的多种设备服务时非常有用。如果系统管理员尚未启用增强视频，则可以选择标准视频。

要使用 YouTube 托管的视频，请参见 [YouTube 视频](#)

要使用 Oracle Content Management 文档或资产资料档案库中存储的视频，请执行以下操作：

1. 导航到您要编辑的页，确保“编辑”开关 () 设置为编辑。
 - a. 要使用存储为数字资产的视频，请打开资产面板，在该面板中搜索要使用的视频，然后选中它并将其拖放到该页上。





要将文档添加到页中，请执行以下操作：


1. 导航到您要编辑的页，并确保  设置为编辑。
2. 向页中添加组件。在您指定要显示的文档之前，文档组件显示占位符图像。
3. 要选择文档，请单击菜单图标 ，选择设置，然后单击“文档”字段旁的选择。

该文档必须存储在站点资料档案库或其他您有权访问的资料档案库中。您也可以使用与您共享或者您从本地或网络文件位置上载的文档。

4. 选择文档并单击确定。

注：

此时窗口中将显示所有可用文件。您必须选择适合上下文的文件类型。例如，如果您要选择文档文件，则必须选择具有有效文档格式（例如 TXT 或 DOC）的文件。



5. 使用“设置”面板可以添加标题或修改间距、对齐方式、样式及其他表示选项。
6. 要帮助用户浏览图库中的文档，请执行以下操作：
 - a. 单击显示页码以在每页下显示页码。
 - b. 选择导航方法：
 - 缩略图：按顺序在文档下显示文档中各页的列表。用户可以单击缩略图图像以转到关联的页。
 - 索引器：在文档下显示一系列按钮  以表示文档中的每一页。用户可以单击按钮以转到关联的页。
 - 无：不提供直观导航。用户可以左右划动以显示相邻页。此手动方法始终可用。
 - c. 单击显示上一个/下一个可在每一页上包括箭头图标以切换到文档中的上一页或下一页。

文件夹列表

您可以使用文件夹列表来列出 Oracle Content Management 账户的指定文件夹中的文件夹。

如果将此组件与页上的一个或多个文件列表或文档管理器组件一起使用，则这些组件可以自动显示在文件夹列表中选定文件夹的内容。

要将文件夹列表组件添加到页，请执行以下操作：

1. 导航到您要编辑的页，并确保  设置为编辑。
2. 向页中添加组件。
3. 要编辑组件及其外观，请单击其菜单图标 ，然后选择设置。
4. 选择定制设置可设置有关所显示内容的默认详细信息。
5. 单击选择文件夹旁边的选择可更改要显示的文件夹。您必须选择文档资料档案库主页之外的文件夹。完成后单击返回。

注：

文件夹列表向所有用户授予下载者访问权限。无论是什么角色，用户都可以查看和下载文件。如果站点访问者拥有的权限超过为组件指定的权限，则个别权限会覆盖在组件上设置的权限。



6. 选择以下选项可为所显示内容设置其他默认值。
 - 选择默认选定内容：如果您选择了 **Oracle Documents** 文件夹，则选择要在列表中显示为处于选定状态的文件夹（如果有）。
 - 显示文件夹名称标题：选择此项可在嵌入式组件的标题中显示文件夹名称。
 - 文件夹排序：选择最初如何显示项，可以按名称的字母顺序排序，也可以按照上次更新项的时间排序。
 - 显示子文件夹：使用滑动栏来限制显示的子文件夹数。
7. 使用“一般信息”选项卡可以修改间距、对齐方式以及其他表示形式选项。
8. 使用“样式”选项卡可以使用预定义样式或者您自己的定制选项，设置包含组件的框架格式。

文件列表

您可以使用文件列表提供 Oracle Content Management 账户中指定文件夹内文件的视图。

在将此组件与页上的一个或多个文件夹列表组件联合使用时，文件列表组件可以自动显示在文件夹列表中选定文件夹的内容。您也可以配置组件以在用户单击列表中的文件时执行一个或多个操作。例如，您可以在单独的浏览器窗口、“Lightbox”重叠或页面的“文档管理器”组件中预览所选文件。

要将文件列表组件添加到页，请执行以下操作：

1. 导航到您要编辑的页，并确保  设置为编辑。
2. 向页中添加组件组。
3. 要编辑组件及其外观，请单击其菜单图标 ，然后选择设置。
4. 选择定制设置可设置有关所显示内容的默认详细信息。
5. 单击选择文件夹旁边的选择可更改要显示的文件夹。您必须选择文档主页之外的文件夹。完成后单击返回。

 注：

文件列表向所有用户授予下载者访问权限。无论是什么角色，用户都可以查看和下载文件。如果站点访问者拥有的权限超过为组件指定的权限，则个别权限会覆盖在组件上设置的权限。

6. 选择以下选项可为所显示内容设置其他默认值。
 - 显示选项：选择随列出的文件显示哪些详细信息。
 - 文件夹名称标题：在标题中显示文件夹名称。
 - 文件说明：显示文件说明（如果提供）。
 - 文件分隔符：使用线条分隔各个文件。
 - 下载图标：包括一个下载图标，这样用户可以在需要时下载文件。
 - 上次更新时间：显示上次更新文件的日期。
 - 文件大小：显示文件的大小。
 - 图像：显示文件内容的缩略图。
 - 触发器和操作：选择是否使用在“文件夹列表”组件中选择的文件夹内容自动刷新文件列表。在选定了文件时，您还可以选择启用“选择文件”触发器。使用“链接”选项卡将操作与选择文件触发器关联。
 - 文件排序：选择最初如何显示项，可以按名称的字母顺序排序，也可以按照上次更新项的时间排序。
 - 显示文件：选择是显示所有文件还是使用滑动条来限制文件数量。
7. 使用“一般信息”选项卡可以修改间距、对齐方式以及其他表示形式选项。
8. 使用“样式”选项卡可以使用预定义样式或者您自己的定制选项，设置包含组件的框架格式。
9. 使用“链接”选项卡将操作与选择文件触发器关联。例如，要配置文件列表组件来在 Lightbox 重叠中预览所选文件，请执行以下操作：
 - a. 请确保在“文件列表设置”窗口中选择了选择文件时激活触发器。
 - b. 在文件列表设置的“链接”选项卡中，单击选择文件触发器。
 - c. 单击并将 **Lightbox** 预览页面操作拖动到操作列表。
 - d. 在“文件 ID”或“URL”字段中，选择文件链接。



文件链接使用提供下载者权限的引用链接，这样所有站点访问者可以预览文件并选择下载。如果您使用文件 ID，则使用成员链接。成员链接仅允许已注册用户预览和选择下载所选文件。

当用户单击文件列表中的某个文件时，文件预览将在暗灰色不活动的页版本上打开。

文档管理器

您可以使用文档管理器提供 Oracle Content Management 中主页或文件的视图。

要向页面中添加文档管理器组件，请执行以下操作：

1. 导航到您要编辑的页，并确保  设置为编辑。
2. 向页中添加组件。
3. 要编辑组件及其外观，请单击其菜单图标 ，然后选择设置。
4. 选择定制设置可设置有关所显示内容的默认详细信息。
5. 单击选择文件夹旁边的选择可更改要显示的文件夹。

要选择文件夹，请选中文件夹名称旁边的框。要打开文件夹，请单击文件夹名称。单击路径中的某个文件夹名称以返回到该文件夹，或单击导航到主目录以返回到主目录文件夹。完成后单击返回。

6. 单击选择文件夹访问权限并选择要授予访问者的访问角色。

访问者将能够基于其角色以及您在下一步中启用的浏览选项查看和使用文件夹内容。


- **成员访问权限：**访问者将能够使用可供文件夹成员使用的所有功能，例如，查看对话、注释或各项的定制属性。
- **查看者：**查看者可以查看文件和文件夹，但不能更改它们。
- **下载者：**下载者还可以下载文件并将文件保存到自己的计算机中。
- **内容提供者：**内容提供者还可以修改文件，更新文件，上载新文件，以及删除文件。

设置文件夹访问权限时，请考虑以下几点：

- 站点作者不能授予大于自己所拥有的访问权限的文件夹访问权限。例如，如果作者具有文件夹的下载者访问权限，则他们不能向站点访问者授予内容提供者权限。
 - 对组件中文件夹设置的权限可以提升访问者的权限。例如，如果访问者具有文件夹的查看者权限（或没有权限），则组件可以根据选定角色授予更大的权限。这些增强权限仅在组件本身内有效。
 - 如果站点访问者拥有的权限超过为组件指定的权限，则个别权限会覆盖在组件上设置的权限。
 - 对文件夹授予的权限适用于该文件夹中嵌套的文件夹和文件。
7. 选择以下选项可为所显示内容设置其他默认值。
 - **布局：**为文件夹和文件选择初始网格、列表或简洁列表布局。用户可以在查看已完成的嵌入式组件时更改布局。
 - **颜色方案：**为嵌入式文件夹列表选择可用颜色方案之一。
 - **排序顺序：**选择最初将如何显示各项（按名称的字母顺序，或者按各项的上次更新时间）。

- **浏览选项：**选择在用户选择项时可供其使用的选项。例如，您可以选择允许用户查看文件，下载文件，共享或复制文件和文件夹，或删除文件。如果您希望限制用户可以对文件和文件夹执行的操作，请取消选择此列表中的选项。如果您在上一步中选择了“成员访问权限”，则可以选择显示侧窗格，将在此窗格中显示所有对话、注释或定制属性。
 - **查看者选项：**选择用户将如何查看文件。您可以允许在嵌入式框架内或在其他选项卡（或窗口，具体取决于您的浏览器设置）中查看文件。您还可以隐藏或显示缩略图，以及定制视频的查看方式。
 - **显示缩放控件：**选择是否在文件夹的嵌入式视图中显示滑块栏或缩放控件。
 - **查看器适应模式：**可以选择以填充页面、填充页面宽度或原始大小的形式显示文件。
 - **触发器和操作：**选择刷新同时用于文件夹列表组件的文件列表。
8. 使用“一般信息”选项卡可以修改间距、对齐方式以及其他表示形式选项。
 9. 使用“样式”选项卡可以使用预定义样式或者您自己的定制选项，设置包含组件的框架格式。

添加文档管理器之后，查看者将看到嵌入站点上框架中的所选文件夹的视图。用户可以使用提供的显示选项更改文件夹和文件的列出方式。如果用户选择文件或文件夹，则可以选择菜单栏上提供的选项，或者使用右键单击菜单并执行其角色以及您所指定浏览选项允许的任意操作。

如果对话与某一项相关联，则您将会看到与该项一起显示 。单击此图标可以打开对话窗格，您可以在该窗格中查看批注和注释。必须使用对话组件添加独立对话（不与文件夹关联的对话）。

如果项具有定制属性，您可以在窗格中查看这些属性。单击**更多**，然后选择定制属性以打开属性窗格。



项目库

您可以使用项目库将不同位置的文件夹装配到资料档案库中，而不必更改原始文件夹或其位置。您还可以使用项目库组件为每个文件夹分配不同的权限级别，以容纳不同的项目小组角色。

例如，如果为项目库中的某个文件夹分配访问者角色之一（查看者、下载者或内容提供者），则访问者可以使用与相应角色关联的权限查看该文件夹内容并与之交互。如果指定成员访问权限，则只有成员才会看到项目库中列出的文件夹。成员使用为其指定的对原始文件夹的权限与文件夹内容交互。

如果您将此组件与页面上的一个或多个文件列表或文档管理器组件配合使用，则这些组件可以自动显示项目库中所选文件夹的内容。

要向页面中添加项目库组件，请执行以下操作：

1. 导航到您要编辑的页，并确保  设置为编辑。
2. 向页中添加组件。
3. 要编辑组件及其外观，请单击其菜单图标 ，然后选择设置。
4. 选择定制设置以选择对话并选择显示选项。
5. 指定标题。默认标题为“项目库”。您可以通过选择或取消选择以下标题显示选项，来选择显示或隐藏标题。
6. 选择颜色方案。

 **注:**

如果您使用触发器和操作将项目库与文档管理器组件相关联，则为项目库选择的颜色方案也适用于显示在文档管理器组件中的所选文件夹。这不适用于将项目库与文件夹列表或文件列表组件相关联的情况，因为这些组件不支持颜色方案。

7. 单击文件夹列表旁的添加以添加现有文件夹或创建新文件夹。
 - a. 导航到一个或多个文件夹并进行选择，或单击创建创建文件夹。您在此窗口中选择的每个文件夹都将添加到项目库中。

您可以使用工具栏中列出的任何选项（包括查看和排序选项）。
 - b. 完成后单击返回。
8. 要组织列表，请选择一个文件夹并单击上移或下移以更改其在列表顺序中的位置，或单击删除从列表中删除该文件夹。
9. 要根据用户角色指定对某个文件夹的访问权限，请从文件夹列表中选择该文件夹，单击选择文件夹权限并选择访问角色。
 - 成员访问权限：只有具有文件夹权限的已注册用户才会看到项目库中列出的文件夹。成员使用为其指定的对原始文件夹的权限与文件夹内容交互。
 - 查看者：查看者可以查看文件和文件夹，但不能更改它们。
 - 下载者：下载者还可以下载文件并将文件保存到自己的计算机中。
 - 内容提供者：内容提供者还可以修改文件，更新文件，上载新文件，以及删除文件。设置文件夹访问权限时，请考虑以下几点：
 - 站点作者不能授予大于自己所拥有的访问权限的文件夹访问权限。例如，如果作者具有文件夹的下载者访问权限，则他们不能向站点访问者授予内容提供者权限。
 - 对组件中文件夹设置的权限可以提升访问者的权限。例如，如果访问者具有文件夹的查看者权限（或没有权限），则组件可以根据选定角色授予更大的权限。这些增强权限仅在组件本身内有效。
 - 如果站点访问者拥有的权限超过为组件指定的权限，则个别权限会覆盖在组件上设置的权限。
 - 对文件夹授予的权限适用于该文件夹中嵌套的文件夹和文件。
10. 对于具有成员访问权限的文件夹，您可以通过单击在文档管理器中显示对话框，显示与文件夹或内容关联的对话。

如果您选择此选项并在页面上配置文档管理器组件以显示选定文件夹，则用户可以单击对话图标以显示与文件夹关联的任何对话。如果不选择此选项，则不显示对话图标。
11. 完成后，关闭窗口。
12. 使用“一般信息”选项卡可以修改间距、对齐方式以及其他表示形式选项。
13. 使用“样式”选项卡可以使用预定义样式或者您自己的定制选项，设置包含组件的框架格式。
14. 使用“链接”选项卡将操作与已选择文件夹触发器相关联。例如，如果您还将文档管理器组件添加到页面中，则可以使用文档管理器组件显示项目文件夹中选定文件夹的内容：

- a. 在项目库设置的“链接”选项卡中，单击已选择文件夹触发器。
- b. 在“配置触发操作”窗口中，单击“文档管理器”。
- c. 单击并将显示文档操作拖动到操作列表。
- d. 在“文件夹 ID”或“URL”字段中，选择所选文件夹。

当用户单击项目库中的文件夹时，文件夹内容将显示在页面上的文档管理器组件中。

社交组件

社交组件可帮助用户保持联系和交流。



- [社交栏](#)
- [Facebook 点赞和推荐](#)
- [Twitter 分享和关注](#)
- [对话组件](#)
- [对话列表](#)

社交栏



轻松地添加流行社交媒体的图标和链接，例如 Facebook 和 Twitter。社交栏默认情况下包括一些目标，不过您可以在社交栏中添加和删除项。



要添加和修改社交栏，请执行以下操作：

1. 导航到您要编辑的页，并确保  设置为编辑。
2. 向页中添加组件。默认情况下社交栏显示所包括的图标。
3. 要在社交栏中添加或更改图标，请单击菜单图标 ，选择设置，然后单击位于一般信息顶部的图标。

要删除社交图标，请单击名称旁边的 x。要添加图标，该图标必须存储在站点资料档案库或者其他您有权访问的资料档案库中。您也可以使用与您共享或者您从本地或网络文件位置上载的图像。

4. 要从本地或网络位置上载一个或多个图标，请执行以下操作：
 - a. 单击位于面板顶部的添加图标。
 - b. 导航到资料档案库中您要存储图像的位置，或者单击  以在当前位置添加新文件夹。
 - c. 单击 。
 - d. 查找并选择一个或多个图像文件，然后单击打开。
此时图像文件将上载到资料档案库中的当前位置。
5. 在资料档案库中选择一个或多个图像，然后单击确定。

- a. 找到并单击要使用的一个或多个图像。
- b. 要链接到资料档案库中的文件，请选择使用对原始文件的引用而不是将文件复制到站点。如果没有选择此选项，文件副本将随站点一起存储，并从站点中引用。链接到原始文件可以避免复制内容。即使文件的权限发生更改或者限制查看，站点访问者仍可以通过链接来查看内容。
- c. 单击确定。

所选图像将添加到图像列表中。拖动图像以在列表中重新排序。各个图像的默认标题是不带扩展名的文件名。

6. 要更改特定图像的目标 URL、标题、说明或其他选项，请单击列表中的图像并进行更改。
当您完成更新单个图标时，单击返回可指定一般选项。
7. 指定社交栏中所有图标的大小、间距、方向和对齐方式。

Facebook 点赞和推荐

您可以在页上添加 Facebook 的“点赞”按钮，让查看者轻松地在 Facebook 上分享对您站点的喜爱。

1. 导航到您要编辑的页，并确保  设置为编辑。
2. 向页中添加组件。
3. 要编辑应用程序及其外观，请单击其菜单图标 ，然后选择设置。
4. 选择 **Facebook“点赞”** 设置或 **Facebook“推荐”** 设置可以设置 URL 并选择性地添加共享按钮，这样您可以轻松地在 Facebook 页上发布您的站点的链接。
5. 使用“一般信息”选项卡可以修改间距、对齐方式以及其他表示形式选项。
6. 使用“样式”选项卡可以使用预定义样式或者您自己的定制选项，设置包含应用程序的框架格式。

Twitter 分享和关注

您可以在页上添加 Twitter 的“共享”按钮，让查看者在 Twitter 账户上快速共享链接。

1. 导航到您要编辑的页，并确保  设置为编辑。
2. 向页中添加组件。
3. 要编辑应用程序及其外观，请单击其菜单图标 ，然后选择设置。
4. 选择 **Twitter 关注** 设置可设置 Twitter 用户名并选择是否显示用户名，还可以选择关注按钮的大小。选择 **Twitter 共享** 设置以设置下列默认值。用户可以在使用按钮时更改值。
 - **共享 URL**：用户可以共享的站点的 URL。
 - **推文文本**：介绍该页的推文文本。
 - **推文用户**：推文使用的 Twitter 用户账户。
 - **推荐用户**：用于 Twitter 推荐功能的 Twitter 用户账户。
 - **主题标签 #**：用于帖子的主题标签。

- 计数：显示共享计数，可以采用纵向或横向。
 - 大按钮：选择为应用程序使用大按钮还是小按钮。
5. 使用“一般信息”选项卡可以修改间距、对齐方式以及其他表示形式选项。
 6. 使用“样式”选项卡可以使用预定义样式或者您自己的定制选项，设置包含应用程序的框架格式。


对话组件

您可以使用对话直接从您的站点提升有关某个主题的讨论。

注：

要在某个站点上使用对话，该站点必须是安全站点，仅允许指定的用户或具有 Oracle Content Management 用户角色的用户访问。请参见[更改站点安全性](#)。

要将对话组件添加到页中，请执行以下操作：

1. 导航到您要编辑的页，并确保  设置为编辑。
2. 向页中添加组件。
3. 要编辑组件及其外观，请单击其菜单图标 ，然后选择设置。
4. 选择定制设置以选择对话和表示形式颜色方案。
5. 单击选择对话旁边的选择，以选择现有对话或者创建新对话。
 - a. 从可用对话列表中选择对话，或者单击创建以创建新对话并命名。

该列表包含您具有访问权限的所有对话。
可以筛选列表：

 - 全部：显示除标记为静音的那些对话外的其他所有对话。
 - 收藏：显示已加入收藏夹的对话。
 - 静音：显示已静音并已从对话列表中排除的对话。
 - 关闭：显示标记为关闭的对话。

您可以对该列表进行排序：

 - 上次更新时间：按从最近更新到最早更新的顺序列出对话。
 - 名称：按从最低到最高的字母数字顺序列出对话。
 - 未读：按具有最多未读评论到具有最少未读评论的对话顺序先列出未读对话。
 - b. 完成后单击返回。
 - c. 如果您选择了独立对话（不基于文件或文件夹的对话），则可以设置用户查看该对话时需要的访问权限。
 - d. 选择颜色方案。
 - e. 要在页上将对话组件与对话列表组件联合使用，请选择根据在对话列表组件中的选择自动刷新对话。

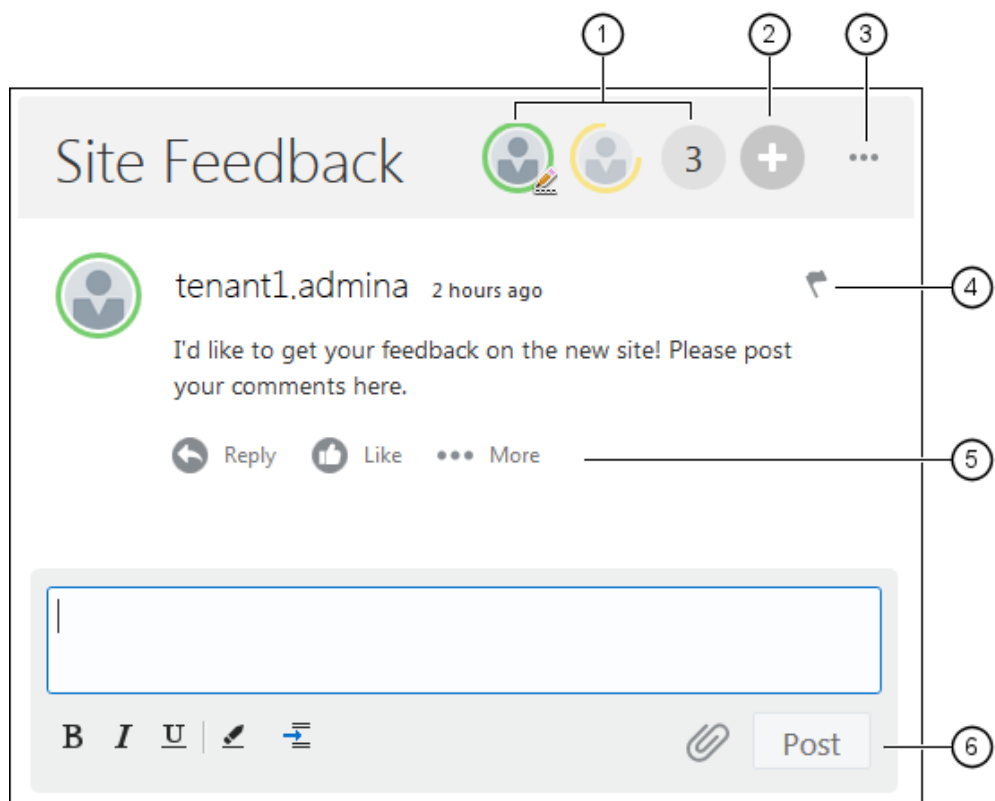
用户从列表中选择对话时，该对话将显示在对话组件中。

- f. 完成后，关闭窗口
6. 使用“一般信息”选项卡可以修改间距、对齐方式以及其他表示形式选项。
7. 使用“样式”选项卡可以使用预定义样式或者您自己的定制选项，设置包含组件的框架格式。

在已发布站点上，访问者将在站点的一个框架内看到对话。用户可以在对话内导航，以阅读和回复评论。

 注：

如果访问者尚未作为对话成员明确添加，他们能够阅读和回复评论，但看不到下图中的 1 至 4 项。



- 菜单栏 (1) 包含对话中各个参与者的相关信息。单击用户图标可获取状态信息以及查看使用用户的选项。数字图标显示参与者总数。单击此图标可查看完整列表。
- 单击添加用户 (2) 可将用户添加到对话。
- 单击更多选项 (3) 可查看使用对话的选项列表。选项列表因用户角色而异。对于与文件夹和文件关联的对话，选项列表还取决于对文件夹或文件设置的权限。
- 单击标记 (4) 可为标记分配通知标记，以便提醒特定用户。将通过标记首选项中指定的电子邮件来提醒用户。
- 使用评论选项 (5) 可在对话中的特定评论上执行操作，例如回复、链接、编辑或删除评论。

- 添加或编辑评论时，使用编辑选项 (6) 可添加或删除基本格式（例如粗体或下划线）、添加附件以及将评论发布到对话。除非站点访问者是对话成员，否则他们无法添加附件。

对话列表



您可以使用对话直接从您的站点提升有关某个主题的讨论。

在将此组件与页面上的一个或多个对话列表组件一同使用时，对话列表组件可以自动显示在对话列表中选定对话的内容。

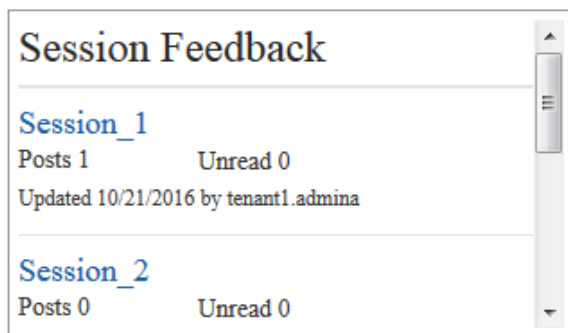
注：

要在某个站点上使用对话或对话列表，该站点必须是安全站点，仅允许指定的用户或具有 Oracle Content Management 用户角色的用户访问。请参见[更改站点安全性](#)。

要将对话列表组件添加到页，请执行以下操作：

1. 导航到您要编辑的页，并确保  设置为编辑。
2. 向页中添加组件。
3. 要编辑组件及其外观，请单击其菜单图标 ，然后选择设置。
4. 选择定制设置以选择对话并选择显示选项。
5. 指定标题。默认标题为“对话列表”。您可以通过选择或取消选择以下标题显示选项，来选择显示或隐藏标题。
6. 单击对话列表旁边的添加可添加现有对话或者创建新对话。
 - a. 从可用对话列表中选择一个或多个对话，或者单击创建以创建新对话并命名。
 - b. 完成后单击返回。
 - c. 为查看对话列表的用户选择允许的权限。
7. 要组织对话列表，请选择一个对话，然后单击上移或下移以更改该对话在列表顺序中的位置，或者单击删除将该对话从列表中删除。
8. 选择颜色方案并选择额外的显示选项。您选择的显示选项将显示在列表中该对话名称的下面。
9. 完成后，关闭窗口。
10. 使用“一般信息”选项卡可以修改间距、对齐方式以及其他表示形式选项。
11. 使用“样式”选项卡可以使用预定义样式或者您自己的定制选项，设置包含组件的框架格式。

以下图像显示标题为会话反馈且选中了所有显示选项的对话列表。



流程组件

站点用户可使用流程组件启动和管理预定义流程的任务。

注：

要使用流程组件，您必须使用 Oracle Process Cloud Service 发行版 17.1.3 或更高版本。

- [流程开始表单](#)
- [处理任务列表](#)
- [任务详细信息表单](#)



流程开始表单

可以使用流程开始表单来启动使用 Oracle Process Cloud Service 定义的流程。

要在站点上使用流程开始表单，必须满足以下条件：

- 关联的流程和流程开始表单必须使用 Oracle Process Cloud Service 定义，然后才能显示在此组件中。请参见《*Using Processes in Oracle Integration*》中的 "Developing Structured Processes"。
- 要使用流程开始表单，必须向用户分配与包含开始表单的流程泳道关联的角色。
- 管理员必须设置 Oracle Process Cloud Service 与 Oracle Content Management 之间的集成。请参见 *Administering Oracle Content Management* 中的“与 Oracle Process Cloud Service 集成”。这两个服务之间的集成需要 SSO 登录，因此这两个服务必须位于同一个身份域中。

要将流程开始表单组件添加到页中，请执行以下操作：

1. 导航到您要编辑的页，并确保  设置为编辑。
2. 向页中添加组件。
3. 要编辑组件及其外观，请单击其菜单图标 ，然后选择设置。
4. 选择定制设置以选择流程开始表单并设置表单默认值。
 - a. 如果您的站点将是公共站点，请选择一个代理服务。

- b. 为开始表单选择分区。您可以使用“测试”分区验证流程是否按计划工作，或使用“生产”分区部署流程供一般使用。
- c. 如果您希望始终使用被选作默认流程的流程版本，请选择使用默认流程版本。如果您未选择此选项，则需要选择特定版本；如果流程进行了更新，您将需要更新这些设置中的选择项。
- d. 选择流程。
- e. 选择开始表单。表单具有以下语法：流程类型:版本:流程名称:开始。例如，Basic Approval:1.0:Process:Start Basic Approval。

如果窗口顶部的消息表明“无 Process Cloud Service 连接”，则可能未配置 Oracle Process Cloud Service 与 Oracle Content Management 之间的集成。请与管理员联系。

流程作者必须将您添加为流程的启动者才能在列表中查看它。流程作者必须将所有站点访问者添加为流程的启动者，否则这些访问者将能够完成表单，但不能启动流程。

- f. 选择以下选项以便为表单设置其他默认值。
 - 表单标题：（可选）使用您自己的标题替换默认表单标题。
 - “提交”按钮名称：（可选）使用您指定的值重命名提交按钮。
 - 显示“提交”按钮：（可选）显示或隐藏组件上的提交按钮（默认情况下显示）。如果表单本身提供了类似按钮，则可以隐藏组件的提交按钮。
 - 提交确认：（可选）使用您自己的消息替换默认确认消息。
 - 显示提交确认：（可选）在单击提交按钮时显示或隐藏确认消息。默认情况下显示确认消息，并且该消息显示在流程开始表单组件中。
 - 显示“保存”按钮：（可选）显示或隐藏组件上的保存按钮（默认情况下隐藏）。如果关联的流程允许您保存工作并在以后返回，则可以显示组件保存按钮。
 - 显示“放弃”按钮：（可选）显示或隐藏组件上的放弃按钮（默认情况下隐藏）。如果希望允许用户放弃表单内容并重新开始，则可以显示放弃按钮。
 - 显示附件：（可选）在表单上显示或隐藏附件区域，提供上载一个或多个文件的功能（默认情况下隐藏）。所上载文件作为流程的一部分存储在 Oracle Process Cloud Service 中。
 - 定制默认值：可以设置用来预填充开始表单的名称-值对。单击添加字段，然后输入字段名称和值。名称是表单字段之一而不是在表单上使用的标签，值是该字段所允许的内容。要插入当前已登录用户的数据，请使用特殊值 %username% 和 %userid%。
 5. 使用“一般信息”选项卡可以修改间距、对齐方式以及其他表示形式选项。
 6. 使用“样式”选项卡可以使用预定义样式或者您自己的定制选项，设置包含组件的框架格式。
 7. 使用“链接”选项卡以将操作分配到组件提供的触发器：
 - 已提交开始表单：用户单击提交按钮时执行此触发器。
 - 已保存开始表单：用户单击保存按钮时执行此触发器。
 - 已放弃开始表单：用户单击放弃按钮时执行此触发器。
- 单击触发器以分配操作。有关分配触发器和操作的详细信息，请参阅[使用触发器和操作](#)。



处理任务列表

可以使用流程任务列表来有选择地列出使用 Oracle Process Cloud Service 定义的流程。您可以显示任务列表组件中任务的详细信息，或将流程任务列表组件与任务详细信息组件配合使用，以简化列表并仅显示所选任务的详细信息。

对于在站点上使用的流程任务列表，必须符合以下条件：

- 关联的流程必须使用 Oracle Process Cloud Service 定义，然后才能显示在此组件中。请参见《*Using Processes in Oracle Integration*》中的 "Developing Structured Processes"。
- 站点必须是安全站点，仅允许指定的用户或具有 Oracle Content Management 用户角色的用户访问。请参见[更改站点安全性](#)。
- 管理员必须设置 Oracle Process Cloud Service 与 Oracle Content Management 之间的集成。请参见 *Administering Oracle Content Management* 中的“与 Oracle Process Cloud Service 集成”。这两个服务之间的集成需要 SSO 登录，因此这两个服务必须位于同一个身份域中。

要将流程任务列表组件添加到页，请执行以下操作：

1. 导航到您要编辑的页，并确保  设置为编辑。
2. 向页中添加组件。
3. 要编辑组件及其外观，请单击其菜单图标 ，然后选择设置。
4. 选择定制设置以设置显示选项并筛选可用任务。
5. 从以下项中选择以设置任务列表的显示选项。

注：

您可以设置组件的显示选项，但任务本身的表示形式由 Oracle Process Cloud Service 中的设计决定。

- **显示详细信息：**选择以在任务列表中包含任务详细信息。将流程任务列表组件与任务详细信息组件配合使用，以简化列表并仅显示所选任务的详细信息。
 - **显示搜索：**选择以在任务列表顶部包含搜索栏。如果您选择显示搜索，则还可以选择显示筛选器以在搜索栏中包含筛选器选项。使用下面的筛选器选项设置默认筛选器值。
 - **显示全选：**选择以允许用户选择所有显示的任务以进行处理。
 - **页大小：**调整以指定要显示的任务的最大数目。如果拥有的任务数小于指定的最大值，则该组件会自动调整到更小的数目。如果拥有的任务数大于指定的最大值，则该组件会显示最大数目并添加指向更多页的链接。
6. 从以下项中选择以筛选列表中的可用任务。

您应用的筛选器将决定初始任务列表。如果您选择上述显示搜索和显示筛选器选项，则用户可调整设置以修改所有设置（自用户除外）的筛选器。

- **搜索关键字：**（可选）指定一个或多个搜索词。搜索结果仅包含其标题包含所有指定关键字的任务。如果您选择显示搜索，这些关键字将显示在搜索栏中，并且可由用户删除或修改。
 - **状态：**选择可用的状态值之一。默认值为已分配。
 - **分配对象：**选择可用的分配对象值之一。默认值为我和我的组全部，这包括对用户及其组可用的所有任务（包括已申请但不可行的任务）。
 - **自用户：**（可选）选择任务源自的一个或多个用户。开始键入用户名以启动对关联的 Oracle Process Cloud Service 上的可用用户的搜索。您指定的值不会显示给最终用户，且无法删除。用户可添加其他用户名以进一步扩展用户列表。
 - **到期日期：**（可选）选择一个到期日期，该日期等于、早于、晚于从日历中选择的日期或介于所选日期之间。
 - **应用程序：**（可选）选择特定应用程序的名称。可以从对用户可用的应用程序列表中选择一个或多个应用程序。
7. 使用“一般信息”选项卡可以修改间距、对齐方式以及其他表示形式选项。
 8. 使用“样式”选项卡可以使用预定义样式或者您自己的定制选项，设置包含组件的框架格式。

 **注：**

如果您在页面上具有流程任务列表和任务详细信息表单，则任务详细信息组件自动显示从列表中选择的任务的详细信息。有关触发器和操作的详细信息，请参见[使用触发器和操作](#)。



任务详细信息表单

您可将任务详细信息表单与流程任务列表配合使用来显示所选任务的详细信息。如果您在页面上具有流程任务列表和流程详细信息表单，则流程任务详细信息组件自动显示从列表中选择的任务的详细信息。

对于在站点上使用的任务详细信息表单，必须符合以下条件：

- 关联的流程必须使用 Oracle Process Cloud Service 定义，然后才能显示在此组件中。请参见 *Using Processes in Oracle Integration* 中的 "Developing Structured Processes"。
- 站点必须是安全站点，仅允许指定的用户或具有 Oracle Content Management 用户角色的用户访问。请参见[更改站点安全性](#)。
- 管理员必须设置 Oracle Process Cloud Service 与 Oracle Content Management 之间的集成。请参见 *Administering Oracle Content Management* 中的“与 Oracle Process Cloud Service 集成”。这两个服务之间的集成需要 SSO 登录，因此这两个服务必须位于同一个身份域中。

要将任务详细信息组件添加到页，请执行以下操作：

1. 导航到您要编辑的页，并确保  设置为编辑。
2. 向页中添加组件。
3. 要编辑组件及其外观，请单击其菜单图标 ，然后选择设置。
4. 选择定制设置以设置显示选项。

5. 从以下项中选择以设置任务详细信息表单的显示选项。

 注：

您可以设置组件的显示选项，但任务详细信息的表示形式由 Oracle Process Cloud Service 中的设计决定。

- 显示操作：选择以显示对用户可用的操作，例如批准、拒绝等。
 - 显示保存：选择以显示保存按钮。
 - 显示关闭：选择以显示关闭按钮。
 - 显示附件：选择以显示详细信息表单中的附件部分。
 - 显示评论：选择以显示详细信息表单中的评论部分。
 - 显示历史记录：选择以显示详细信息表单中的历史记录部分。
 - 显示更多信息：选择以显示详细信息表单中的更多信息部分。
 - 显示链接：选择以显示详细信息表单中的链接部分。
6. 使用“一般信息”选项卡可以修改间距、对齐方式以及其他表示形式选项。
- 默认情况下，任务详细信息组件将展开以显示所有指定的详细信息。单击设置高度并调整高度值以设置特定高度。
7. 使用“样式”选项卡可以使用预定义样式或者您自己的定制选项，设置包含组件的框架格式。
8. 使用“链接”选项卡以将操作分配到组件提供的触发器：
- 已提交任务详细信息：此触发器在用户单击提交按钮时出现。
 - 任务已批准：此触发器在用户单击批准按钮时发生。
 - 任务已拒绝：此触发器在用户单击拒绝按钮时发生。
 - 任务已关闭：此触发器在用户单击关闭按钮时发生。
 - 任务已保存：此触发器在用户单击保存按钮时发生。
 - 任务评论已添加：此触发器在用户添加评论文本并单击发表评论按钮时发生。
- 单击触发器以分配操作。

 注：

流程任务列表组件不支持手动触发器或操作，但如果您向页面添加任务详细信息组件，则它会自动显示从列表中选择的任务的详细信息。有关触发器和操作的详细信息，请参见[使用触发器和操作](#)。

有关使用 Oracle Process Cloud Service 的信息，请参见《Getting Started with Process》。

内容项

对于企业用户，每个站点具有集合，其中包含与站点关联的数字资产和内容项。通过内容项组件可以轻松地将项添加到站点。

您可以在编辑器中的“内容”面板直接拖放数字资产和内容项，相应的组件将自动使用，不论是数字资产图像还是结构化内容项。

从“组件”面板中选择的其他组件可以基于内容类型动态显示内容项。




- 内容项组件
- 内容占位符
- 内容列表
- 内容搜索
- 推荐

内容项组件

作为企业用户，您可以使用内容项组件来帮助布局页和设置页交互，直至您准备好添加内容项本身。


将内容项从“内容”面板拖动到页上时，它会自动插入内容项组件来容纳项，除非内容项是定制数字资产类型。将定制数字资产拖到某页上会导致该资产作为图像或视频组件插入。您需要将其删除并先添加内容项组件，然后将定制数字资产或标准内容项从“内容”面板拖到该组件上以便正确显示。


要将内容项组件添加到页，请执行以下操作：

1. 导航到您要编辑的页，并确保  设置为编辑。
2. 向页中添加组件。
3. 要编辑内容项组件内容，请单击其菜单图标 ，然后选择编辑，或者选择设置以打开设置面板并单击编辑图标 。

注：

如果内容项组件中使用的内容类型是定制数字资产，而不是标准内容类型，则您无法编辑该定制数字资产类型。组件菜单中将没有编辑选项，设置面板中将没有任何编辑图标。

4. 更改内容项，然后单击保存。内容项将在内容资料档案库中另存为新修订版。
5. 要编辑组件外观，请单击其菜单图标 ，然后选择设置。
6. 使用“一般信息”选项卡可以修改间距、对齐方式以及其他表示形式选项。
如果某个内容项分配到了组件，则将显示内容项的缩略图视图。如果尚未分配内容项，则将显示占位符图像和消息。
7. 选择以下选项可为所显示内容设置其他默认值。

- **要使用的版本：**如果选择了使用最新的资产版本并且内容项存在较新的未发布版本，则在发布当前站点更新时将自动发布较新的版本。除非明确请求，否则所有内容项将为草稿或最新版本。
- **项视图：**选择用于显示内容项的布局。提供的默认布局显示内容项中的所有字段。如果为内容项设计了其他定制布局，则可以选择任意可用布局。如果选择启用了添加对于站点中所用定制设置的支持的定制布局，则可以通过单击  并在定制数据字段中输入数据，向布局添加定制数据，然后单击返回以返回到“一般信息”选项卡。
- **用于显示单个项的页：**如果您指定了一页或多页作为详细信息页，则会在此处列出这些页。选择当用户单击内容项上的链接以查看详细信息时，用于显示详细信息的页。

注：

如果您没有创建详细信息页，则不为默认布局中的内容项显示用于显示详细信息的链接。



8. 使用“样式”选项卡可以使用预定义样式或者您自己的定制选项，设置包含组件的框架格式。

内容占位符

作为企业用户，您可以使用内容占位符组件来动态显示一个或多个类型的内容项。

例如，可以在指定的详细信息页上使用内容项占位符，当用户单击链接以获取特定内容项更详细的信息时，它可以自动加载关联内容项的详细信息视图。如果，如果某页具有多篇文章，每篇文章都有标题和图像，在用户单击特定文章时，不论选择哪篇文章，详细信息页都会显示完整的文章。

要将内容占位符组件添加到页，请执行以下操作：

1. 导航到您要编辑的页，并确保  设置为编辑。
2. 向页中添加组件。
3. 要编辑组件及其外观，请单击其菜单图标 ，然后选择设置。
4. 使用“一般信息”选项卡可以修改间距、对齐方式以及其他表示形式选项。
5. 选择以下选项可为所显示内容设置其他默认值。
 - **内容类型：**选择可用的一个或多个内容类型。内容类型是站点集合中的内容项的类型并且包括定制数字资产类型。
 - **项视图：**选择用于显示内容项的布局。提供的默认布局显示内容项中的所有字段。如果为内容项设计了其他定制布局，则可以选择任意可用布局。
 - **用于显示单个项的页：**如果您指定了一页或多页作为详细信息页，则会在此处列出这些页。选择当用户单击内容项上的链接以查看详细信息时，用于显示详细信息的页。

 注:

如果您没有创建详细信息页，则不为默认布局中的内容项显示用于显示详细信息的链接。

6. 使用“样式”选项卡可以使用预定义样式或者您自己的定制选项，设置包含组件的框架格式。

内容列表

作为企业用户，您可以使用内容列表组件动态显示特定类型的内容项。

例如，可以在指定的详细信息页上使用内容项列表，当用户单击链接以获取特定内容项更详细的信息时，它可以自动加载关联内容项的详细信息视图。如果，如果某页具有多篇文章，每篇文章都有标题和图像，在用户单击特定文章时，不论选择哪篇文章，详细信息页都会显示完整的文章。

要将内容列表组件添加到页，请执行以下操作：

1. 导航到您要编辑的页，并确保  设置为编辑。
2. 向页中添加组件。
3. 要编辑组件及其外观，请单击其菜单图标 ，然后选择设置。
4. 使用“一般信息”选项卡可以修改间距、对齐方式以及其他表示形式选项。
5. 选择以下选项可为所显示内容设置其他默认值。
 - **内容类型：**选择可用内容类型之一。内容类型是站点集中的内容项的类型并且包括定制数字资产类型。
 - **最大项数和起始项：**指定要显示的最大项数以及显示开始位置。不显示额外的项。
 - **分页：**指定您是否希望在列表中包括分页选项。如果选中，您随后可以根据需要使用按钮或页码以及不同标签定制分页的显示方式。
 - **日期：**使用提供的选项可根据项创建日期早于、晚于、介于或位于选定日期范围，来选择要显示的内容项。
 - **类别：**选择类别来筛选内容项列表，以仅显示某个或某些特定类别中的项。单击选择类别可打开滑出面板，可以使用此面板从站点资料档案库中选择类别。如果在同一分类中选择了多个类别，为其分配了选定类别之一的项就会显示。如果从不同分类中选择了多个类别，则为其分配了所有选定类别的项才会显示。例如，如果从目标分类中选择了表示城市和公园的类别，则将显示属于城市或公园的项。如果从目标分类中选择类别城市，并从区域分类中选择类别欧洲，则将显示属于欧洲境内城市的项。

默认情况下，选定类别的所有子级都将可用。如果您希望将列表范围限定于某个特定类别节点中的项，请选择该节点并取消选择包括子类别。
 - **语言：**从与资料档案库关联的那些语言中选择一种语言。
 - **其他查询字符串（可选）：**指定其他查询参数以使用类似于 `field.dept eq "Finance"` 的语法进一步细化显示的项列表。有关可用运算符列表，请参见下表。
 - **排序方式：**按名称或日期的升序或降序顺序对项排序。如果内容类型包含其他日期、数字或小数字段，则还可以按这些字段排序。

还可以选择定制，然后输入定制表达式。例如，要按部门编号的升序排序，则可以输入 `fields.deptno:asc`。您还可以定义用来排序的 URL 参数。例如，您可以输入

{{URLParams.sortBy}}, 然后将以下参数添加到站点 URL 的末尾: ?
sortBy=fields.deptno:asc。

- **项视图**: 选择用于显示内容项的布局。提供的默认布局显示内容项中的所有字段。如果为内容项设计了其他定制布局, 则可以选择任意可用布局。如果选择启用了添加对于站点中所用定制设置的支持的定制布局, 则可以通过单击



并在定制数据字段中输入数据, 向布局添加定制数据, 然后单击返回以返回到“一般信息”选项卡。

- **用于显示单个项的页**: 如果您指定了一页或多页作为详细信息页, 则会在此处列出这些页。选择当用户单击内容项上的链接以查看详细信息时, 用于显示详细信息的页。

注:

如果您没有创建详细信息页, 则不为默认布局中的内容项显示用于显示详细信息的链接。

- **列表视图**: 选择部分布局以排列项。此列表可以包括定制布局。提供了以下布局:
 - **水平**: 在水平线上依次排列各项。
 - **垂直**: 在垂直线上依次排列各项。如果未选择布局, 则这是默认布局。
 - **两列**: 在多个行中每行排列两项。
 - **三列**: 在多个行中每行排列三项。
 单击所选布局旁的向右键可以修改该布局的设置。
 - **空列表视图**: 选择应如何显示空列表。
 - **选项**: 选择“内容搜索”组件中指定的搜索查询中是否应存在自动刷新。
 - **间距**: 选择将如何显示项。
6. 使用“样式”选项卡可以使用预定义样式或者您自己的定制选项, 设置包含组件的框架格式。

表 10-1 查询运算符

运算符	示例	支持的数据类型	说明
eq	?q=name eq "John" ?q=type eq "DigitalAsset" ?q=type eq "Employee" and fields.DOB eq "1994/09/26T16:23:45. 208" ?q=type eq "Employee" and fields.DOB eq "1994/09/26T16:23:45. 208"	文本、引用、数字、十进制、布尔、日期时间	等于运算符 (eq) 匹配查询中提供的确切值。此运算符不适用于多值数据类型。与此运算符一起提供的值不区分大小写, 但标准字段除外。此运算符甚至会考虑值中的特殊字符。

表 10-1 (续) 查询运算符

运算符	示例	支持的数据类型	说明
co	<pre>?q=(type eq "Employee" AND name co "john alex") ?q=(type eq "Car" AND fields.features co "manual")</pre>	文本、引用、数字、十进制、日期时间、大型文本	包含运算符 (co) 匹配标准中提供的每个单词。单词通过按特殊字符拆分值构成。它提供至少包含其中一个单词的结果（在此示例中为 john 或/和 alex）。搜索时，此运算符不会考虑值中的特殊字符。此运算符无法对禁用词执行搜索。要了解有关禁用词的详细信息，请参阅 Apache Lucene 文档。 对于单值属性，此运算符适用于文本、大型文本；而对于多值属性，此运算符适用于文本、引用、数字、十进制、日期时间、大型文本。要了解可能的日期时间格式，请参阅下文支持的日期/日期时间格式（24 小时）表。与此运算符一起提供的值不区分大小写。
sw	<pre>?q=type eq "Employee" AND name sw "Joh" ?q=type eq "Employee" AND fields.city sw "Los"</pre>	文本	开头为运算符 (sw) 仅匹配字段条件中提供的初始字符值。此运算符不适用于多值数据类型。与此运算符一起提供的值不区分大小写。
ge	<pre>?q=(type eq "Employee" AND fields.age ge "40") ?q=type eq "DigitalAsset" AND updatedAt ge "20171026"</pre>	数字、十进制、日期时间	大于或等于运算符 (ge) 仅匹配数值和日期时间值。要了解可能的日期时间格式，请参阅下文支持的日期/日期时间格式（24 小时）表。此运算符不适用于多值数据类型。
le	<pre>?q=(type eq "Employee" AND fields.weight le "60.6")</pre>	数字、十进制、日期时间	小于或等于运算符 (le) 仅匹配数值和日期时间值。要了解可能的日期时间格式，请参阅下文支持的日期/日期时间格式（24 小时）表。此运算符不适用于多值数据类型。
gt	<pre>?q=(type eq "Employee" AND fields.age gt "20")</pre>	数字、十进制、日期时间	大于运算符 (gt) 仅匹配数值和日期时间值。要了解可能的日期时间格式，请参阅下文支持的日期/日期时间格式（24 小时）表。要了解可能的日期时间格式，请参阅下文支持的日期/日期时间格式（24 小时）表。此运算符不适用于多值数据类型。

表 10-1 (续) 查询运算符

运算符	示例	支持的数据类型	说明
lt	<pre>?q=(type eq "Employee" AND fields.age lt "20") ?q=type eq "Employee" AND createdAt lt "1994/09/26T16:23:45. 208"</pre>	数字、十进制、日期 时间	小于运算符 (lt) 仅匹配数值和日期时间值。要了解可能的日期时间格式，请参阅支持的日期/日期时间格式 (24 小时) 部分。此运算符不适用于多值数据类型。
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>	文本、大型文本	<p>短语查询或邻近搜索 (匹配) 运算符 (mt) 用于查找在相对于某个单词的特定距离内的单词。结果按最佳匹配排序。要搜索内容项时 (例如，对于标准 "petrol 20kmpl" 中提供的值，需要发现可能包含 "petrol fuel mileage runs 20KMPL in the speed way" 的实际内容时)，此运算符很有用。</p> <p>此外，匹配运算符还允许在给定值中使用通配符，并支持在单个值中使用单字符和多字符通配符搜索。? 用于表示单个字符通配符，* 用于表示多个字符。"John" 和 "Joan" 可以按针对单个字符的 "Jo?n" 和针对多个字符的 "Jo*" 进行搜索。</p> <p>此运算符适用于单值和多值数据类型。此运算符无法对禁用词执行搜索。要了解有关禁用词的详细信息，请参阅 Apache Lucene 文档。与此运算符一起提供的值不区分大小写。</p>
sm	<pre>?q=(type eq "Employee" And fields.city sm "Rome")</pre>	文本、大型文本	相似度查询运算符。此运算符用于搜索与指定标准相似的值 (也称为模糊搜索)，默认情况下，使用两个编辑内容的最大值来搜索结果。"Rome" 与 "Dome" 相似。此运算符适用于单值和多值数据类型。与此运算符一起提供的值不区分大小写。
AND	<pre>?q=(type eq "Employee" AND name eq "John" AND fields.age ge "40")</pre>	N/A	AND 运算符可以用于在多个查询条件之间放置 AND 条件。此运算符的优先级高于 OR。
OR	<pre>type eq "Employee" AND name eq "John" OR fields.age ge "40"</pre>	N/A	OR 运算符可以用于在多个查询条件之间放置 OR 条件。

表 10-1 (续) 查询运算符

运算符	示例	支持的数据类型	说明
()	?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)	N/A	括号, 用于对标准中的条件进行分组的封闭运算符。此运算符的优先级最高, 其次是 AND, 然后是 OR。

内容搜索

作为企业用户, 您可以使用内容搜索组件并指定搜索所返回的操作。

您可以插入定制搜索栏来更改或刷新页面上显示的内容, 或者选择其他操作, 例如, 打开搜索结果页或显示预警。

要向页面中添加内容搜索组件, 请执行以下操作:

1. 导航到您要编辑的页, 并确保  设置为编辑。
2. 将内容搜索组件添加到页。
3. 要编辑组件及其外观, 请单击其菜单图标 , 然后选择设置。
4. 使用“一般信息”选项卡可以修改占位符文本、间距、对齐方式及其他表示选项。
5. 使用“样式”选项卡可以设置框架的格式, 框架中所含组件具有预定义样式或具有您自己的定制字体、边框、背景色等选择。
6. 如果您希望使用搜索组件来刷新当前页上的数据, 请将内容列表组件添加到页。转到设置并选择内容类型和任何其他查询信息, 例如要显示的项数、是否对结果分页或者滚动时延迟加载 (在页滚动后加载内容)。现在, 您可以转到查看模式并尝试搜索。
7. 如果您希望使用搜索组件来刷新当前页上的数据, 并且页上有多个内容列表组件, 则在除了要显示结果的内容列表之外, 需要关闭所有其他内容列表的自动查询。转到“内容列表”设置, 取消选中自动刷新内容搜索组件中的搜索查询。
8. 如果希望在一页上有多个内容搜索组件 (每个搜索组件会带来特定的结果内容列表), 则必须在“内容列表”组件中取消选中所有自动刷新选项并使用“触发/操作”来将内容搜索组件关联到其相应内容列表。
9. 对于搜索组件, 使用“链接”选项卡将操作与组件关联。从“选择链接类型”下拉列表选择一个选项。选择触发器操作, 然后单击在搜索查询上以查看可用的触发器或创建新的触发器。查找您要搜索的内容列表, 展开它, 然后将搜索内容拖到操作列表中。在搜索下, 选择搜索字符串。您选择的内容列表将在执行此操作于列表中选中。
10. 您还可以使用内容搜索组件将查询传递到搜索结果页。在搜索组件的“链接”选项卡上, 选择搜索页以显示搜索结果页。您可以使用默认页, 也可以链接到您所创建的搜索结果页。从“页”下拉列表中选择页并为页选择显示操作。您可以将结果定制为特定的内容类型, 在新窗口中打开结果, 等等。

以下指针可帮助您设计高效的搜索结果页：



- 创建一页并将其指定为搜索页。默认情况下，该页标记为隐藏，不过您可以在页属性中更改设置。
- 编辑页并将内容列表添加到页。您可以编辑内容列表的设置，指定内容类型。或者可从搜索组件链接设置（上方）指定内容类型。如果您使用链接设置，这让您可以使用根据启动搜索所用的组件，显示来自不同内容类型结果的搜索结果页。
- 要编辑显示，请更改设置，例如选择延迟加载分页（因为页面很可能会专门用于搜索结果）。
- 在搜索结果页上，您还可以放入内容搜索组件。这会回显用于启动页的搜索字符串，让用户根据需要细化搜索。

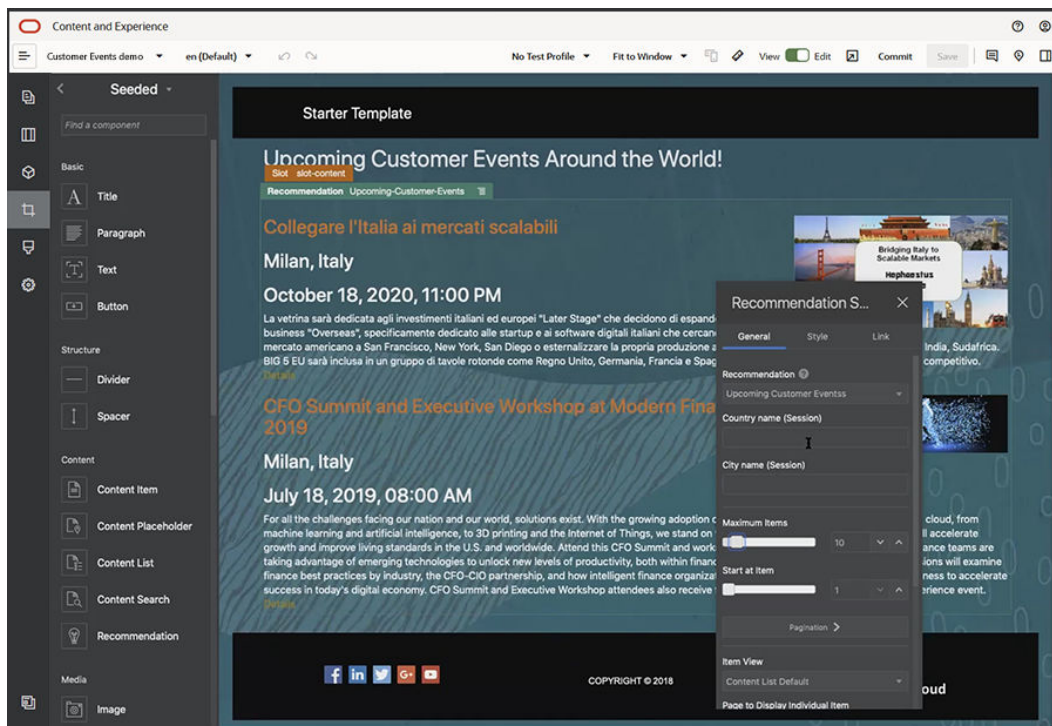
推荐

作为企业用户，您可以使用**推荐**组件，通过基于位置或关注领域显示资产来为网站访问者提供个性化体验。资料档案库内容提供者创建推荐时会定义一组规则来查找与受众属性（例如站点访问者的地理位置信息）相匹配的资产。例如，具有欧洲 IP 地址的站点访问者在站点首页上可能会首先看到欧洲的事件公告，而北美洲的站点访问者将看到美国和加拿大的事件。

与数字资产相同，推荐与资料档案库相关联，可以通过工作流进行编辑和移动，以供资料档案库的内容提供者审核和发布。不过，即使您没有内容提供者权限，也仍可以查看和测试推荐，以便先查看推荐的工作方式，然后在站点或无头 (headless) 体验中使用推荐。

要向页面添加推荐，请执行以下操作：

1. 单击侧面菜单中的站点，选择要向其添加推荐的站点，然后在右键单击菜单中选择打开，或单击操作栏中的 。
2. 将站点切换到编辑模式，并选择要使用的更新或创建新的更新。
3. 从侧面菜单中选择组件。
4. 打开已植入。
5. 单击并拖动“内容”部分下面的**推荐**以将其放到页面上。
6. 从“推荐”菜单 () 中选择设置。



7. 在推荐设置的一般信息选项卡中，选择要使用的推荐。页面将刷新并显示推荐的资产。
8. （可选）通过单击所选推荐旁边的箭头来输入此推荐所用受众属性的默认值。在完成值的添加之后，单击后退。
9. 更改所需的任何其他属性，例如，要使用的版本、可显示的最大项数或默认内容布局。完成后，关闭推荐设置并单击保存。

注：

如果已经创建了具有预定义受众属性值的测试概要信息，请从菜单栏中的测试概要信息菜单中选择概要信息，以预览推荐如何响应这些输入。测试概要信息值将覆盖组件设置中所配置的默认值。

10. 如果您的推荐在其规则中使用“当前日期（系统）”，而且您希望像对其他日期那样测试您的推荐，请从测试概要信息菜单中选择系统日期并选择要测试的日期。

其他组件

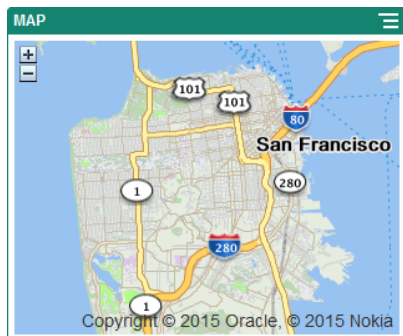
现在来了解一些组合不同内容类型的组件。

- [地图](#)
- [标题](#)
- [文章](#)
- [图像（具有文本）](#)
- [组件组](#)
- [在页上使用 Cobrowse](#)
- [Oracle Intelligent Advisor](#)



- Oracle Visual Builder

地图

添加地图到站点以允许用户通过互动方式浏览某个位置周围的区域。



要将地图添加到页，请执行以下操作：


1. 导航到您要编辑的页，并确保  设置为编辑。
2. 向页中添加组件。Oracle 地图组件显示默认位置。
3. 要更改地图的位置以及调整其显示属性，请单击菜单图标 ，然后选择设置。
4. 输入地图的开始位置。您可以使用地址、邮政编码或者逗号分隔的纬度和经度（例如，40.5,-57.6）



如果使用地址，请确保提供足够的信息以匹配单个位置。如果地址匹配了多个位置，则地图保持空白。

5. 指定地图的初始缩放级别。您可以选择允许用户使用鼠标、轨迹板或屏幕控件调整缩放级别。
6. 指定任意样式、对齐方式和显示选项：
 - 缩放：启用屏幕和设备选项以允许用户调整地图的缩放级别。
 - 平移：启用屏幕和设备选项以允许用户移动地图的焦点区域。
 - 显示标记：使用大头针图标标记开始位置。
 - 比例：包括一个显示当前缩放级别的比例的指示器。
 - 概览：包括一个在较大的环境中显示当前视图的嵌图。

标题


您可以添加标题，以便通过标题文本、背景图像和段落来增强站点中特定区域的吸引力。


1. 导航到您要编辑的页，并确保  设置为编辑。
2. 向页中添加组件。










3. 在组件的其他区域中单击以添加文本。您可以在主标题中添加内容，并在主标题下方添加段落。文本采用组件的默认样式的格式设置。按 Enter 可以添加其他文本行。
4. 如果要更改任何文本部分的默认格式设置，请选择要设置格式的文本，然后在格式设置工具栏中选择任意选项，例如字体、颜色或对齐方式。
5. 要删除使用这些选项应用的格式设置，请选择文本，然后单击 。
6. 要编辑组件及其外观，请单击其菜单图标 ，然后选择设置。
7. 选择组件可设置有关所显示内容的详细信息。

- 图像：

- 单击选择可在标题中使用图像。可从站点资料档案库中选择图像，从与您共享的文档文件夹中选择图像，或将图像上传到文档文件夹。找到并选择要使用的图像。如

果您未看到任何数字资产，请单击 ，然后将集合筛选器更改为全部。如果从文档文件夹中选择了图像，则可以链接到文件，而不必将它复制到站点。要链接到文件，请选择使用对原始文件的引用而不是将文件复制到站点。如果没有选择此选项，文件副本将随站点一起存储并可供从站点中引用。链接到原始文件可以避免复制内容。即使文件的权限发生更改或者限制查看，站点访问者仍可以通过链接来查看内容。单击确定。如果选择了数字资产，则可以选择特定重现。如果未选择重现，则将使用原始大小。如果要在发布站点时发布最新的资产版本，请选择使用最新的资产版本。

要编辑图像，请单击 ，并通过以下任一操作编辑图像：


- * 要剪裁图像，请单击  剪裁。在剪裁工具栏中选择预定义的图像纵横比之一，或根据需要在图像上拖动剪裁手柄。满意后，在剪裁工具栏中单击剪裁。
- * 要旋转或翻转图像，请单击  旋转。在旋转工具栏中，输入定制旋转角度，使用按钮向左或向右旋转图像，或者选择水平或垂直翻转图像。
- * 要将水印添加到图像，请单击  水印。将文本添加到图像，然后使用水印工具根据需要更改文本大小、样式、颜色和不透明度。
- * 要更改图像格式，请单击  选项，然后从格式下拉列表中选择新格式。
- * 要更改背景色，请单击  选项，然后从背景色下拉菜单中选择一个选项。
- * 如果编辑的是 .jpg 或 .webp（在 Google Chrome 浏览器中可用）格式的图像，您可以更改图像质量以创建较小的文件。请单击  选项，然后在质量框中输入新的百分比。
- * 要撤消或重做更改，请单击  或 。要删除所做的所有更改，请单击重置。
- * 要更改图像的缩放比例，请使用缩放控件 ()。
- 标题：输入要在工具提示中显示的文本。
- 替代文本：输入将显示用于可访问性用途的替代文本。
- 标题：输入显示在图像下方的标题。



- 对齐方式、宽度和间距：根据需要更改图像布局。
 - 标题：更改标题部分中的标题文本间距。
 - 段落：更改标题部分下方文本的间距。
8. 使用“一般信息”选项卡可以修改间距、对齐方式以及其他表示形式选项。
 9. 使用“样式”选项卡可以添加有关文本的格式设置并且定制背景色、字体和边框。










文章

您可以添加文章到站点，这可以将标题、段落和图像等组件组合到一个易于使用的组件中。

1. 导航到您要编辑的页，并确保  设置为编辑。
2. 向页中添加组件。
3. 在组件的其他区域中单击以添加文本。您可以向文章标题及其下方的子标题添加内容。然后，您可以在子标题下方添加文章的文本。所有文本采用组件的默认样式的格式设置。按 Enter 可以添加其他文本行。
4. 如果要更改任何文本部分的默认格式设置，请选择要设置格式的文本，然后在格式设置工具栏中选择任意选项，例如字体、颜色或对齐方式。
5. 要删除使用这些选项应用的格式设置，请选择文本，然后单击 。
6. 要编辑组件及其外观，请单击其菜单图标 ，然后选择设置。
7. 选择组件可设置有关所显示内容的详细信息。
 - 图像：
 - 单击选择可在文章中使用图像。可从站点资料档案库中选择图像，从与您共享的文档文件夹中选择图像，或将图像上传到文档文件夹。找到并选择要使用的图像。如果您未看到任何数字资产，请单击 ，然后将集合筛选器更改为全部。如果从文档文件夹中选择了图像，则可以链接到文件，而不必将它复制到站点。要链接到文件，请选择使用对原始文件的引用而不是将文件复制到站点。如果没有选择此选项，文件副本将随站点一起存储并可供从站点中引用。链接到原始文件可以避免复制内容。即使文件的权限发生更改或者限制查看，站点访问者仍可以通过链接来查看内容。单击确定。如果选择了数字资产，则可以选择特定重现。如果未选择重现，则将使用原始大小。如果要在发布站点时发布最新的资产版本，请选择使用最新的资产版本。





要编辑图像，请单击 ，并通过以下任一操作编辑图像：

- * 要剪裁图像，请单击  剪裁。在剪裁工具栏中选择预定义的图像纵横比之一，或根据需要在图像上拖动剪裁手柄。满意后，在剪裁工具栏中单击剪裁。
- * 要旋转或翻转图像，请单击  旋转。在旋转工具栏中，输入定制旋转角度，使用按钮向左或向右旋转图像，或者选择水平或垂直翻转图像。


- * 要将水印添加到图像，请单击  水印。将文本添加到图像，然后使用水印工具根据需要更改文本大小、样式、颜色和不透明度。
 - * 要更改图像格式，请单击  选项，然后从格式下拉列表中选择新格式。
 - * 要更改背景色，请单击  选项，然后从背景色下拉菜单中选择一个选项。
 - * 如果编辑的是 .jpg 或 .webp（在 Google Chrome 浏览器中可用）格式的图像，您可以更改图像质量以创建较小的文件。请单击  选项，然后在质量框中输入新的百分比。
 - * 要撤消或重做更改，请单击  或 。要删除所做的所有更改，请单击重置。
 - * 要更改图像的缩放比例，请使用缩放控件 (  )。
 - 标题：输入要在工具提示中显示的文本。
 - 替代文本：输入将显示用于可访问性用途的替代文本。
 - 标题：输入显示在图像下方的标题。
 - 对齐方式、宽度和间距：根据需要更改图像布局。
 - 标题：更改文章标题的间距。
 - 段落（子标题）和段落：更改文章标题下方文本的间距。
8. 使用“一般信息”选项卡可以修改组件的间距、对齐方式以及其他表示形式选项。
 9. 使用“样式”选项卡可以添加有关文本的格式设置并且定制背景色、字体和边框。










图像（具有文本）

您可以添加图像和文本组件，这可以将段落和图像等组件组合到一个易于使用的组件中。

1. 导航到您要编辑的页，并确保  设置为编辑。
2. 向页中添加组件。
3. 在组件的段落区域中单击以添加文本。所有文本采用组件的默认样式的格式设置。按 Enter 可以添加其他文本行。
4. 如果要更改任何文本部分的默认格式设置，请选择要设置格式的文本，然后在格式设置工具栏中选择任意选项，例如字体、颜色或对齐方式。
5. 要删除使用这些选项应用的格式设置，请选择文本，然后单击 。
6. 要编辑组件及其外观，请单击其菜单图标 ，然后选择设置。
7. 选择组件可设置有关所显示内容的详细信息。
 - 图像：
 - 单击选择可使用图像。可从站点资料档案库中选择图像，从与您共享的文档文件夹中选择图像，或将图像上传到文档文件夹。找到并选择要使用的图像。如果您未看到任何数字资产，请单击 ，然后将集合筛选器更改为全部。如果从文档文件夹

中选择了图像，则可以链接到文件，而不必将它复制到站点。要链接到文件，请选择使用对原始文件的引用而不是将文件复制到站点。如果没有选择此选项，文件副本将随站点一起存储并可供从站点中引用。链接到原始文件可以避免复制内容。即使文件的权限发生更改或者限制查看，站点访问者仍可以通过链接来查看内容。单击确定。如果选择了数字资产，则可以选择特定重现。如果未选择重现，则将使用原始大小。如果要在发布站点时发布最新的资产版本，请选择使用最新的资产版本。




要编辑图像，请单击 ，并通过以下任一操作编辑图像：

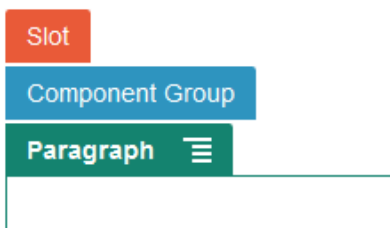
- * 要剪裁图像，请单击  剪裁。在剪裁工具栏中选择预定义的图像纵横比之一，或根据需要在图像上拖动剪裁手柄。满意后，在剪裁工具栏中单击剪裁。
 - * 要旋转或翻转图像，请单击  旋转。在旋转工具栏中，输入定制旋转角度，使用按钮向左或向右旋转图像，或者选择水平或垂直翻转图像。
 - * 要将水印添加到图像，请单击  水印。将文本添加到图像，然后使用水印工具根据需要更改文本大小、样式、颜色和透明度。
 - * 要更改图像格式，请单击  选项，然后从格式下拉列表中选择新格式。
 - * 要更改背景色，请单击  选项，然后从背景色下拉菜单中选择一个选项。
 - * 如果编辑的是 .jpg 或 .webp（在 Google Chrome 浏览器中可用）格式的图像，您可以更改图像质量以创建较小的文件。请单击  选项，然后在质量框中输入新的百分比。
 - * 要撤消或重做更改，请单击  或 。要删除所做的所有更改，请单击重置。
 - * 要更改图像的缩放比例，请使用缩放控件 ()。
 - 标题：输入要在工具提示中显示的文本。
 - 替代文本：输入将显示用于可访问性用途的替代文本。
 - 标题：输入显示在图像下方的标题。
 - 对齐方式、宽度和间距：根据需要更改图像布局。
 - 段落：更改文本的间距。
8. 使用“一般信息”选项卡可以修改组件的间距、对齐方式以及其他表示形式选项。
 9. 使用“样式”选项卡可以添加有关文本的格式设置并且定制背景色、字体和边框。

组件组


您可以结合使用一个或多个组件来创建组件组，并可以命名和重用该组。

保存组件组时，组件组将使用您提供的名称保存为定制组件，然后显示在编辑器的定制组件列表中。

1. 导航到您要编辑的页，并确保  设置为编辑。
2. 向页中添加组件组。使用  来标识组件组。
3. 将一个或多个组件拖放到定制组件的框架中。
4. 使用在内容框中定位和调整组件大小的方法，在组件组中定位和调整组件的大小。
5. 要编辑组件及其外观，请单击其菜单图标 ，然后选择设置。如果您单击组件名称而不是菜单图标，则可以查看并选择组件组（或内容框）的菜单图标：



使用组件组的“设置”选项卡可以指定组件组的位置、背景图像以及其他适用于整个组件组的设置。



6. 在您准备好保存更改到组件组时，单击组件组，单击其菜单图标 ，然后选择保存。
 - a. 在对话框中，输入组件组的名称。可以使用字母、数字、下划线 (_) 和连字符 (-)。如果输入空格，则使用连字符自动取代空格。
如果这是新的组件组，则无法使用现有定制组件的名称。如果您已将现有组件组添加到页中，修改了组件组，然后尝试保存更改，则您既可以提供名称来创建新的组件组，也可以选择覆盖现有组件组来使用您的更改更新现有组件组。
 - b. 单击保存。
该组件组将使用指定名称保存为定制组件。如果您是所有者或者其他人与您共享了组件，该组件会显示在组件管理器中自己的文件夹内，以及编辑器中定制组件的列表内。您可以按照共享任何定制组件的方法来共享组件组。

在页上使用 Cobrowse

Oracle Cobrowse Cloud Service 是一款协作工具，让您可以与其他人共享屏幕或者启动共同浏览会话。例如，您可能希望在订单表单上包括此功能，这样组织代表可以在客户下订单时查看客户的屏幕。

要使用此功能，必须先为站点启用功能。可以启用两种类型的启动程序脚本：一种使用定制按钮（开始点 2），一种使用默认“Cobrowse”按钮（开始点 1）。在为站点启用该功能和添加所需脚本时，您需要确定使用什么类型的启动程序。有关详细信息，请参见[启用 Cobrowse 集成](#)。



在为站点启用 Cobrowse 之后，可以配置任意页来允许共同浏览。

1. 导航到您要编辑的页，并确保  设置为编辑。
2. 单击 。在“Cobrowse”部分中选中复选框。

3. 单击关闭。

如果启用“开始点 1”脚本，在下次查看页时或者按下热键（如果配置）时，您将看到显示默认的 "Cobrowse" 按钮。

如果启用“开始点 2”脚本，您需要将定制按钮添加到页。

1. 请确保  设置为编辑。
2. 向页中添加组件。Cobrowse 启动程序组件在组件的“集成”部分中列出。
3. 要编辑组件及其外观，请单击其菜单图标 ，然后选择设置。您可以调整按钮的标签、外观、大小和对齐方式。使用“样式”选项卡可以添加有关文本的格式设置并且定制背景色、字体和边框。如需更多高级样式设置，请在当前站点模板的主题设计文件夹中，在 `design.json` 和 `design.css` 文件中编辑或添加样式类。样式类前缀为 `scs-cobrowse`。

在站点发布之后，站点的访问者可以使用 "Cobrowse" 按钮启动与组织代表的会话。访问者单击 "Cobrowse" 按钮，系统向其提供安全会话 ID。访问者通过电话将 ID 中继给组织中有权访问 Cobrowse 代理控制台的代表。代理使用控制台启动会话，该会话继续，直至访问者或代理终止会话。使用“开始点 2”设置时，启动程序组件具有固定 ID `cec-start-cobrowse`。在 Cobrowse 控制台中使用该 ID。

使用说明

使用 Cobrowse 即时模式 (ICB) 时，视频或嵌入式 iFrame 在页上不可见，除非使用相同的 Cobrowse 站点 ID 启用了 iFrame 内容。因此，在 ICB 模式下，一些 Oracle Content Management 组件在 Cobrowse 代理控制台中不呈现。使用 Cobrowse Advanced (ACB) 模式可呈现以下组件：

- 视频
- Youtube
- 文档管理器
- Facebook 点赞
- Twitter 关注
- Twitter 共享
- Facebook 推荐
- 对话

在即时模式下，使用 iFrame 获取内容的定制组件也不会呈现。

有关 Oracle Cobrowse Cloud Service 的详细信息，请参见《*Cobrowse Deployment and Use Guide*》中的 "[Cobrowse Overview](#)"。有关将 Cobrowse 用于安全站点或开发中站点的详细信息，请参见[启用 Cobrowse 集成](#)



Oracle Intelligent Advisor

Oracle Intelligent Advisor（以前称为 Oracle Policy Automation）用于实施在线互动式提问方案，例如故障排除反馈或服务资格评估。它通过在自然语言 Microsoft Word 和 Excel 文档中获取规则并按照这些规则构建名为“互动式提问”的互动式客户服务体验，从而跨渠道提供建议。

要使用 Intelligent Advisor 功能，必须先配置并启用该功能。服务管理员为您的服务启用该功能，包括添加用于访问正在使用的 Intelligent Advisor 中心的主机名、URL、用户名和密码。这两个服务之间的集成需要 SSO 登录，因此这两个服务必须位于同一个身份域中。请参见《*Integrating and Extending Oracle Content Management*》中的 "Integrate with Intelligent Advisor"。

在 Intelligent Advisor 端，必须在主机站点上创建并存储交互式提问。此外，还必须对 Oracle Content Management 服务进行授权，以供 Intelligent Advisor 主机使用。

配置并启用 Intelligent Advisor 后，您可以将 Intelligent Advisor 组件添加到您的站点中的页面。

1. 导航到您要编辑的页，并确保  设置为编辑。
2. 向页中添加组件。该组件显示在“组件”列表的“集成”部分中。
3. 要编辑组件及其外观，请单击其菜单图标 ，然后选择设置。您可以调整组件的标签、外观、大小和对齐方式。在“样式”选项卡中，可以使用与 Intelligent Advisor 主机中的交互式提问关联的默认样式。如需更多高级样式设置，请在当前站点模板的主题设计文件夹中，在 design.css 文件中编辑或添加样式类。样式类前缀为 scs-opainterview-。

发布站点后，站点访问者会看到在 Intelligent Advisor 组件中选择的交互式提问。有关 Intelligent Advisor 的更多详细信息，请参见 [Intelligent Advisor 文档库](#)。

Oracle Visual Builder

Oracle Visual Builder 是应用程序开发基础结构的托管环境。它提供了基于标准的开源解决方案，可用于开发、协作处理应用程序并在 Oracle Cloud 中部署应用程序。

初始步骤

要使用 Oracle Visual Builder，必须启用和配置该服务。服务管理员为您的服务启用了该功能，包括添加创建和存储应用程序时所在主机的主机名。请参见 *Integrating and Extending Oracle Content Management* 中的 "Integrate with Oracle Visual Builder"。集成这两个服务需要 SSO，因此这两个服务必须在同一个身份域中。

在 Oracle Visual Builder 中，必须先完成以下操作，才能将此功能与 Oracle Content Management 结合使用：

- 必须在 Oracle Visual Cloud Service 站点上启用跨源资源共享 (CORS)。
- 必须创建应用程序，使其可用于嵌入，并对其进行配置，使其可与 Oracle Content Management 一起使用。
- 必须创建 Web 应用程序并使其可以嵌入 iframe 中。必须在 Web 应用程序中导入和引用 Sites SDK。必须将名为 "id" 的页面 URL 参数添加到 Web 应用程序。

创建 Oracle Visual Builder 组件

启用集成并创建和准备好应用程序和 Web 应用程序后，必须为要添加到站点页的每个应用程序创建新组件。

1. 在 Oracle Visual Builder 中，获取发布的 Web 应用程序的 URL。单击包括 Web 应用程序的活动项目，然后单击 Web 应用程序。复制地址栏中的 URL。
2. 在 Oracle Content Management 中，依次单击开发人员和查看所有组件。此时将显示已注册的远程组件和布局。
3. 单击创建并选择创建可视构建器组件。

4. 输入组件的名称。您无法使用已由另一个组件或布局使用的名称。
可以使用字母、数字、下划线 (_) 和连字符 (-)。如果输入空格，将使用下划线自动取代空格。
不要将以下名称用于模板、主题、组件、站点或站点页：authsite、content、pages、scstemplate_*、_comps、_components、_compsdelivery、_idcservice、_sitescloud、_sitesclouddelivery、_themes、_themesdelivery。下面的名称虽然可用于站点页，但不要将其用于模板、主题、组件或站点：documents、sites。
5. (可选) 输入组件的说明。
6. 将 URL 复制到活动 Web 应用程序或活动经典应用程序。
7. 单击创建。创建了组件之后，该名称显示在组件的列表中。可以单击列表中的组件名称来浏览构成组件或布局的文件夹和文件。
8. 要选择分配给组件的默认图标之外的图标，请执行以下操作：
 - a. 从列表中选择组件。
 - b. 单击属性
 - c. 单击组件徽标选项卡。
 - d. 在徽标库中单击某个徽标，然后单击完成。

将组件添加到站点页

现在可以将应用程序的组件添加到您的其中一个站点页上。可以将该组件添加到公共页或安全页。

1. 导航到您要编辑的页，并确保  设置为编辑。
2. 向页中添加组件。该组件显示在“定制”部分中。
3. 要编辑组件及其外观，请单击其菜单图标 ，然后选择设置。您可以调整组件的标签、外观、大小和对齐方式。在“样式”选项卡中，可以使用与 Oracle Visual Builder 主机中的组件相关联的默认样式。如需更多高级样式设置，请在当前站点模板的主题设计文件夹中，在 design.css 文件中编辑或添加样式类。样式类前缀为 scs-component。

编辑站点时，可以在预览模式下查看组件。发布站点后，站点访问者可以看到，选定用于该组件的 Oracle Visual Builder 应用程序在页面的 iFrame 中运行。

第 III 部分

发布和管理站点

此部分详细介绍了如何使用、保护、改进和发布站点。其中包括以下章节：

- [管理站点](#)
- [发布站点](#)
- [保护站点](#)
- [使用多语言站点](#)
- [使用站点重定向或 URL 映射](#)
- [提高站点性能](#)

11

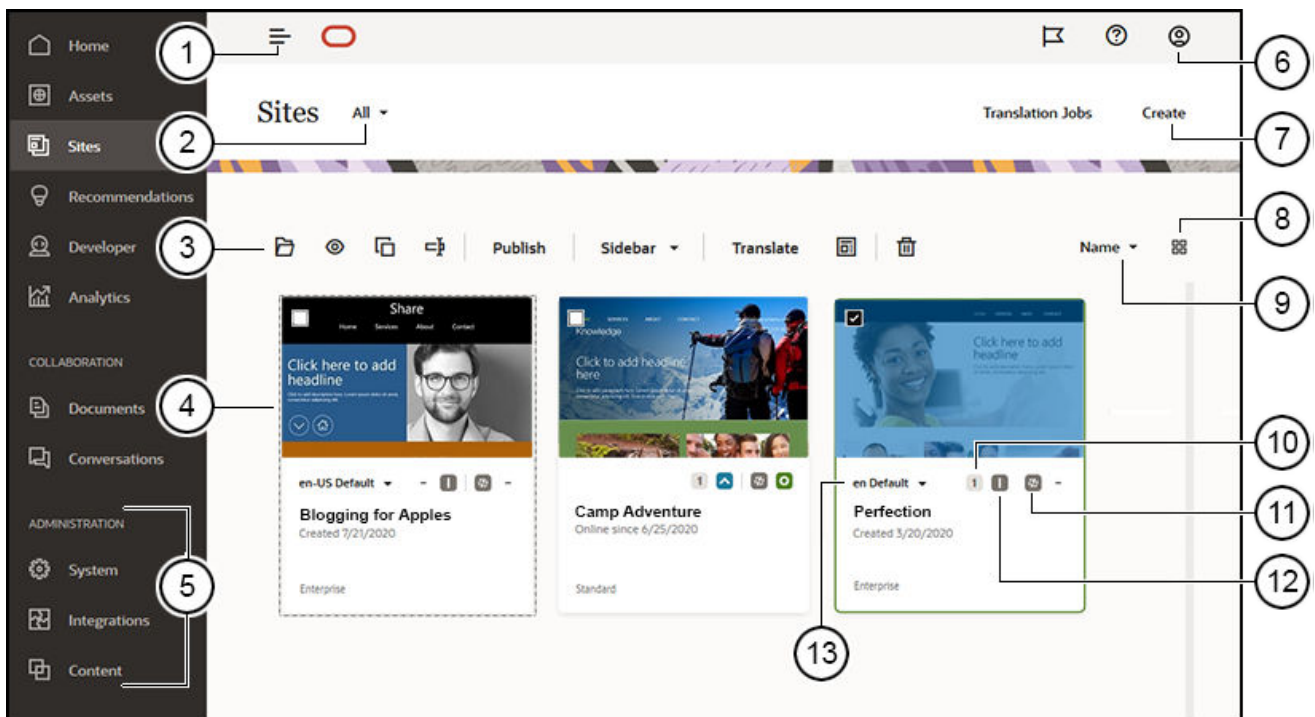
管理站点

要更改站点布局或内容，请在编辑器中创建并打开更新。要创建和管理站点自身和站点的属性，请使用站点管理器中的选项。

- [初步了解“站点”页](#)
- [管理站点和站点设置](#)
- [使站点联机或脱机](#)
- [更改站点说明、徽标或嵌入属性](#)
- [设置搜索引擎属性](#)
- [定制站点设置](#)
- [启用 Cobrowse 集成](#)
- [添加分析跟踪](#)

初步了解“站点”页

“站点”页是使用网站的入手点。















标注	说明
1	面板切换用于隐藏和显示导航菜单。
2	您可以使用筛选器菜单筛选站点列表以查看您可以访问的所有站点或其中一部分站点。如果启用了站点管理，您可以查看待处理的站点请求。
3	<p>在列表中选择站点后，操作栏将可用。使用操作栏上的选项可打开站点、查看站点、更改站点状态或重命名站点以及执行其他任务。</p> <p>显示的选项取决于您在选定站点上的角色。例如，如果您创建了站点，则您将具有该站点的管理者角色，可以执行列出的所有任务。但是，如果某人已与您共享站点并为您分配了查看者角色，则您可以查看站点属性，但无法更改任何内容。</p> <p>操作栏选项还取决于站点的当前状态。重命名和删除选项仅在站点脱机时可用。</p> <p>提示：要查找快捷方式？右键单击站点以打开上下文菜单并选择选项。</p>
4	站点列表显示您拥有或已与您共享的所有站点。列表中包括站点名称、更新数量、站点是否安全以及站点的状态（联机、脱机，或有待处理的更新）。
5	管理菜单在您以内容管理员身份登录时显示。通过此处的选项，您可以更改服务设置、设置集成并设置资料档案库以及其他与资产相关的功能。
6	用户菜单提供了用于设置首选项、提供反馈、访问帮助以及注销的选项。单击用户图片以显示菜单选项。
7	单击创建 可启动创建网站的流程。管理员必须先启用站点创建并提供一个或多个模板，然后您才能创建站点。如果您在站点页上未看到创建选项，或者在模板页上未看到模板，请与您的服务管理员联系。
8	单击视图图标可选择是在网格视图、列表视图还是表视图中显示站点。
9	使用排序选项可更改站点的显示顺序。
10	单击更新编号可编辑站点的现有更新或为站点创建新的更新。如果站点没有任何更新，您可以通过打开站点并将站点构建器更改为编辑模式来创建新的更新。
11	单击脱机、联机或重新发布 图标可更改站点的状态，或者发布已提交但未发布到活动站点的更改。
12	安全性图标显示站点是否需要登录。
13	使用语言列表可查看站点的关联本地化策略中指定的语言。要预览某个本地化版本，请选择相应语言，然后打开站点。


管理站点和站点设置

站点管理包括创建和编辑站点，以及管理站点中使用的内容。您可以使用站点构建器创建、复制和删除站点页和页内容。使用文件夹和文件管理器创建、复制、共享和删除整个站点。

在“站点”页上和站点构建器中使用菜单栏或右键单击菜单上的选项可以执行以下任务。

任务	说明
筛选站点视图	<p>要筛选站点列表，请在筛选器菜单中选择以下选项之一：</p> <ul style="list-style-type: none"> 全部 — 显示您可以访问的所有站点。如果您是站点管理员且启用了站点管理，则您可以访问您的环境中的所有站点。 由您所有 — 显示您拥有的所有站点。 与您共享 — 显示已与您共享的所有站点。 联机 — 显示处于联机（活动）状态的站点。 脱机 — 显示处于脱机状态的站点。 请求（仅在启用站点管理时可用） — 显示待处理的站点请求。如果您是站点管理员，您可在您的系统中看到所有站点请求，否则，您只能看到您提交的站点请求。 回收站 — 显示您拥有的已删除站点。如果您是站点管理员且启用了管理，则回收站还显示所有已删除站点。
创建站点	<p>要创建站点，请单击创建。在创建站点时，可以从模板开始。模板包含您在开始构建站点时需要的全部内容，包括站点代码框架、带有示例页和内容的默认站点、具有样式设置的主题以及图像等资源，甚至包括定制组件。请参见创建站点。</p> <p>您可以通过查看站点页上的筛选器菜单来了解是否启用了站点管理。如果您看到请求选项，则表示启用了站点管理。请参见了解站点管理。</p>
查看和管理站点请求	<p>要查看待处理的请求（仅在启用站点管理时可用），请在筛选器菜单中选择请求。如果您是站点管理员，您可在您的系统中看到所有站点请求，否则，您只能看到您提交的站点请求。请参见“管理站点请求”。</p>
编辑站点	<p>要编辑站点，请在右键单击菜单中选择打开，或单击操作栏中的 。请参见编辑站点。</p>
讨论站点或为站点添加批注	<p>在站点构建器中，您可以讨论站点并为站点添加批注。选择站点，然后在右键单击菜单中选择打开，或单击操作栏中的 。要讨论站点，请单击  以打开对话面板。要为站点添加批注，请单击 。请参见初步了解“站点构建器”页。</p>
查看站点	<p>要查看站点上线（联机）时的外观，请选择站点，然后在右键单击菜单中选择查看，或单击操作栏中的 。</p>
发布站点	<p>要发布站点，请选择站点，然后在右键单击菜单或操作栏中选择发布。请参见发布站点更改。</p>
重新发布	<p>如果您之前发布了站点，但关联的项目或策略发生了更改（例如，更新了关联的本地化策略以包含更多语言），那么您可以重新发布该站点以刷新这些更改。选择站点，然后在右键单击菜单或操作栏中选择重新发布。</p>
取消发布	<p>如果您之前发布了站点，但现在您想从 Oracle Cloud 中的托管位置删除站点文件，请选择站点，然后在右键单击菜单或操作栏中选择取消发布。站点必须处于脱机状态才能取消发布。</p>
使站点联机或脱机	<p>站点必须已发布，然后才能联机。要使站点联机或脱机，请选择站点，然后在右键单击菜单中选择联机或脱机。此外，还可以单击站点磁贴中的  使站点联机，或单击站点磁贴中的  使站点脱机。当站点处于联机状态时，用户可以通过分配的地址 (URL) 使用 Web 浏览器查看该站点。当站点处于脱机状态时，站点不可供公众查看；您只能在 Oracle Content Management 中查看该站点。请参见使站点联机或脱机。</p>

任务	说明
重命名站点	<p>要重命名站点，请右键单击站点，然后在右键单击菜单中选择重命名，或单击操作栏中的 。输入站点的名称，该名称应该与同一个服务器上的所有现有站点名称不同。</p> <p>站点联机时无法重命名。要使站点脱机，您必须是站点所有者或者具有该站点的管理者角色。</p>
复制站点	<p>您可以复制站点，以在构建站点时有个好的开头。原始站点上的所有内容，包括主题、所有未完成更新、页、页内容以及所有其他资产（例如图像）将复制到您提供的新名称下方的新站点中。您的新站点处于脱机状态并可供编辑。</p> <p>注：如果启用了站点管理，您可能需要获得站点批准，然后才能创建站点副本。</p> <p>如果您要复制使用来自多个资料档案库的内容的站点，您需要使用 OCE 工具包来完成。请参见使用 OCE 工具包进行开发和使用 cec 命令行实用程序。</p> <p>选择要复制的站点，然后在右键单击菜单中选择复制，或单击操作栏中的 。请参见复制站点。</p>
从站点创建模板	<p>如果您希望使用某个站点作为其他站点的起点，您可以从该站点创建模板。选择站点，然后在右键单击菜单中选择创建模板，或单击操作栏中的 。请参见从站点创建模板。</p>
	<div style="border: 1px solid #0070C0; padding: 10px; background-color: #E6F2FF;"> <p> 注：</p> <p>如果您要从使用来自多个资料档案库的内容的站点创建模板，您需要使用 OCE 工具包来完成。请参见使用 OCE 工具包进行开发和使用 cec 命令行实用程序。</p> </div>
删除或还原站点	<p>如果您有合适的权限，则可以删除站点及其内容。删除站点时，站点文件夹中的所有内容均放到回收站中，包括所有未完成的更新、站点页、页内容以及已添加到页中的资产（例如图像）。</p> <p>如果您创建了站点（您是站点所有者）或者其他人与您共享了站点并授予您内容提供者或管理者角色，则您可以删除或还原该站点。</p> <p>站点联机时无法删除。要使站点脱机，您必须是站点所有者或者具有该站点的管理者角色。</p> <p>此外，还无法删除已发布的站点。必须先取消发布内容。如果站点包含来自多个资料档案库的内容，您需要取消发布每个资料档案库中的关联内容。</p> <p>要删除站点，请选择站点，然后从右键单击菜单中选择删除，或单击操作栏中的 。系统会提示您将站点和所有站点更新移动到回收站。删除的站点会保留在回收站中，直至：</p> <ul style="list-style-type: none"> • 您还原该站点。 • 您永久地删除该站点。 • 达到您的回收站限额。 • 回收站会根据服务管理员设置的间隔自动清空。默认值为 90 天。 <p>要还原站点，请在“站点”菜单中选择回收站。从列表中选择站点，然后选择还原。</p>

任务	说明
向站点添加成员	<p>如果管理员已启用共享，您可以与其他 Oracle Content Management 用户共享您的站点，并允许这些用户根据您授予他们的权限在 Oracle Content Management 中查看、修改或管理站点。</p> <p>任何可以访问您的服务的用户将被视为成员。共享站点时，您向成员分配角色，定义他们在您的站点上可以执行和不可以执行的操作。成员只能根据您分配的角色来使用您的站点。</p> <p>选择要共享的站点，然后从右键单击菜单中选择成员，或单击操作栏上的子工具栏，然后选择成员以打开成员子工具栏。成员子工具栏打开后，单击添加成员。</p> <p>输入一个或多个用户名或电子邮件地址，并分配下列角色之一：</p> <ul style="list-style-type: none"> 查看者：查看者可以在编辑器中查看站点，但不能更改任何内容。 下载者：对于站点，下载者角色提供了与查看者角色相同的权限，用户可以从站点创建新模板。 内容提供者：与查看者相同，还可以编辑站点、删除站点页以及在站点脱机时删除站点。 管理者：与内容提供者相同，还可以添加用户和分配其角色、向联机站点发布更改以及将站点在联机和脱机之间切换。将自动为站点创建者（所有者）分配管理者角色。
	<div style="border: 1px solid #0070c0; padding: 10px; background-color: #e6f2ff;"> <p> 注：</p> <p>如果您要共享有权访问多个资料档案库的站点，只会共享默认资料档案库。任何其他资料档案库都需要单独共享。</p> </div>
更改站点属性（说明、标识等）	要更改站点说明、添加标识、允许嵌入站点、添加虚 URL 以及查看有关站点的有价值信息（包括站点 URL、站点所有者以及其他详细信息），请选择站点，然后从右键单击菜单中选择属性，或单击操作栏上的子工具栏，然后选择属性。请参见 更改站点说明、徽标或嵌入属性 。
翻译站点	如果关联的资料档案库允许，您可以将企业站点翻译为多种语言，方法是导出站点文件，翻译这些文件，然后导入翻译的文件。请参见 翻译站点 。
设置搜索引擎属性	在站点构建器中，您可以提供关键字和文本来帮助搜索引擎识别站点的内容。请参见 设置搜索引擎属性
定制站点设置（收藏图标、控制器文件等）	在站点构建器中，您可以指定用于不同浏览器和平台的站点图标，或者添加控制器文件来处理链接行为。请参见 定制站点设置
启用共同浏览	在站点构建器中，您可以启用协作工具 Oracle Cobrowse Cloud Service。您可以通过它与其他人共享屏幕或启动共同浏览会话。例如，您可能希望在订单表单上包括此功能，这样组织代表可以在客户下订单时查看客户的屏幕。请参见 启用 Cobrowse 集成
向站点添加分析跟踪	在站点构建器中，您可以向站点添加 JavaScript 跟踪代码的片段进行 Web Analytics 跟踪，以便更轻松地与外部分析提供方（例如 Google、Adobe 或 Oracle Infinity）集成。请参见 添加分析跟踪 。
保护活动站点	<p>发布站点并使其可供联机使用后，该站点可供任何人公开访问。但是，如果您是站点所有者或具有管理者角色，则可以限制活动站点仅可供注册用户或一部分用户访问。要更改站点的状态，您必须是站点所有者或者具有该站点的管理者角色。</p> <p>选择要保护的站点，然后从右键单击菜单中选择属性，或单击操作栏上的子工具栏，再选择属性，然后单击站点安全选项卡。请参见更改站点安全性。</p>
添加站点重定向	如果站点 URL 更改，则重定向将一个 URL（源）转发到另一个 URL（目标）。这有助于保留用户书签和搜索引擎评级。请参见 添加站点重定向 。

更改站点说明、徽标或嵌入属性

大部分站点属性在创建或更新站点时设置。查看属性时，您可以获取有关站点的有价值信息，包括站点 URL、站点所有者和其他详细信息。

要更改站点说明、站点标识、添加虚站点以及允许站点可嵌入，请执行以下操作：

1. 选择站点，然后从右键单击菜单中选择属性，或单击操作栏上的子工具栏，然后选择属性。
2. 要更改站点的可选说明，请单击站点属性选项卡，然后输入或更改说明。
3. 您无法直接更改站点 URL，不过您可以选择和复制站点 URL，然后将其粘贴到文档、演示文稿和电子邮件中，用于提供对站点的访问。

如果站点联机，则单击  可直接转到联机站点。

4. 要允许将站点用作嵌入式站点，切换可嵌入站点旁的是。
5. 要更改站点图像，请单击站点徽标选项卡，然后单击更改。查找并选择要使用的图像。图像必须为 .png、.jpeg 或 .jpg 文件，并且纵横比应为 4:3（矩形）。最佳大小为 300 x 225 像素，因为更小的图像可能会失真，而更大的图像可能会影响到性能。
6. 完成后单击关闭。

设置搜索引擎属性

您可以提供关键字和文本来帮助搜索引擎标识站点的内容。



可以在站点级别和页级别定义搜索引擎优化 (SEO) 设置。站点级别设置放大或覆盖单独页上的类似设置，如下表所述。

选项	站点级别	页级别
说明或页说明	提供站点本身未包含的有关站点的一般信息。站点说明包括在站点的每一页中 仅当未使用页设置为主页设置值的情况下，此说明还会用作主页的页级别说明。	提供页本身未包含的有关页的一般信息。页说明是站点中每一页包括的站点说明之外的补充。
关键字	标识应用到站点上所有页的术语或概念。 这些值累加（附加）到为单独页指定的关键字中。	标识应用到单独页的术语或概念。 页关键字可用于标识不在页的文本中显示或者在图像中显示的术语或概念。
页眉或页标题	添加脚本或标记以用于对您的站点进行分析或跟踪。站点页眉内容包括在站点的每一页中。	添加脚本或标记以用于对您的站点进行分析或跟踪。页标题内容是每一页上包括的站点标题内容之外的补充。
页脚或页脚注	添加脚本或标记以用于对您的站点进行分析或跟踪。站点页脚内容包括在站点的每一页中。	添加脚本或标记以用于对您的站点进行分析或跟踪。页脚注内容是每一页上包括的站点页脚内容之外的补充。

如果在站点级别选择了以下搜索排除选项，则设置应用到所有页并覆盖单独页上的设置。如果您未在站点级别选择该选项，则只有单独指定了该选项的那些页才使用该选项。

选项	站点级别	页级别
从搜索引擎中隐藏	如果选中，则将 NOINDEX 元标记添加到每个页，这样搜索引擎不会索引站点上任何页的内容。在这种情况下，站点及其所有页不会显示在 Web 搜索结果中。	如果选中，则将 NOINDEX 元标记添加到当前页，这样搜索引擎不会索引该页的内容。在这种情况下，单独页不会显示在 Web 搜索结果中。
在搜索引擎中隐藏页链接	如果选中，则将 NOFOLLOW 元标记添加到每个页，这样搜索引擎不会跟踪站点上任何页的链接（并随之索引目标）。	如果选中，则将 NOFOLLOW 元标记添加到当前页，这样搜索引擎不会跟踪该页的链接（并随之索引目标）。
在搜索引擎中隐藏页说明	如果选中，则将 NOSNIPPET 元标记添加到每个页，这样搜索引擎在搜索结果中的页后面不会包括说明（在上面指定）。	如果选中，则将 NOSNIPPET 元标记添加到当前页，这样搜索引擎在搜索结果中的该页后面不会包括页说明（在上面指定）。

要更改搜索引擎优化 (SEO) 设置，请执行以下操作：

1. 打开站点进行编辑。
2. 单击子工具栏中的 ，然后单击  **SEO**。
3. 为站点提供可选说明。站点说明包括在站点的每一页中。
仅当未使用页设置为主页设置值的情况下，此说明还会用作主页的页级别说明。
4. （可选）指定以逗号分隔的关键字以帮助搜索引擎标识站点的内容。
站点关键字标识应用到站点上所有页的术语或概念。这些值累加（附加）到使用页设置为单独页指定的关键字中。
5. （可选）添加页眉脚本或标记以用于对您的站点进行分析或跟踪。页眉内容包括在站点的每一页中。验证在页眉中使用的任何代码，确保代码正常工作并且不会给站点带来任何安全风险。
6. （可选）添加页脚脚本或标记以用于对您的站点进行分析或跟踪。页脚内容包括在站点的每一页中。验证在页脚中使用的任何代码，确保代码正常工作并且不会给站点带来任何安全风险。
7. （可选）选择一个或多个选项以排除相应信息，使其不显示在搜索结果中，如上表中所述。
8. 要保存当前更新中所有待处理更改，请单击保存。

启用预呈现服务以优化搜索引擎

您可以启用使用 Oracle Content Management 构建的站点，以预呈现静态 HTML 页，从而对搜索 Crawler 请求进行响应。

如果预呈现服务处于启用状态而且请求来源于搜索 Crawler，则会在高速缓存中搜索该页面。

- 如果在高速缓存中找到了该页面而且该页面存在的时间小于 15 天，则会为请求提供缓存页面。
- 如果该页面在高速缓存中的时间超过 15 天或者在高速缓存中找不到，则会从 Oracle Content Management 提取新版本，更新高速缓存，并提供新的预呈现页面。

- 如果在高速缓存中或 Oracle Content Management 中找不到该页面，则会返回错误页面。

公共站点或新站点上尚未缓存的失效页面一天呈现一次。如果某个页面已经缓存，则不会更新高速缓存。预呈现服务处于启用状态时，对于尚未更新和重新发布的站点，预呈现的高速缓存会每小时更新一次。如果您对站点页面进行更改而且想要测试这些更改对优化的影响，则可以在站点属性中手动更新预呈现的高速缓存。

1. 单击侧面导航中的站点。
2. 选择站点，然后从右键单击菜单中选择属性，或单击操作栏上的子工具栏，然后选择属性。
3. 单击属性子工具栏中的 **SEO**。
4. 此时列出上次刷新高速缓存的日期和时间。要更新高速缓存，请单击立即刷新。进度栏上会显示缓存状态。

 **注：**

如果在站点 SEO 设置中将某个页面设置为对于搜索引擎隐藏，则该页面将不预呈现或缓存。

要在 Oracle Content Management 中启用预渲染服务，请执行以下操作：

1. 单击侧面导航的“管理”区域中的系统。
2. 在“系统设置”菜单中选择站点的 **SEO**。
3. 单击已启用。
4. 定义所需的任何其他用户-代理。

表 11-1 Oracle Content Management 中硬编码的用户-代理

用户-代理	用户-代理	用户-代理	用户-代理
baiduspider	facebookexternalhit	twitterbot	rogerbot
linkedinbot	embedly	quora link preview	showyoubot
outbrain	pinterest/0.	developers.google.com/+web/snippet	slackbot
vkShare	W3C_Validator	redditbot	Applebot
WhatsApp	flipboard	tumblr	bitlybot
SkypeUriPreview	nuzzel	Discordbot	Google Page Speed
Qwantify	pinterestbot		

表 11-2 config.cfg 文件的 PrerenderUserAgents 属性中的其他预配置用户-代理

用户-代理	用户-代理	用户-代理	用户-代理
AddSearchBot	AdIdxBot	AdsBot-Google	AdsBot-Google-Mobile-Apps
AppEngine-Google	Baidu-YunGuanCe	Bingbot	BingPreview
DuckDuckBot	DuckDuckGo-Favicons-Bot	endeca webcrawler	Exabot

表 11-2 (续) config.cfg 文件的 PrerenderUserAgents 属性中的其他预配置用户-代理

用户-代理	用户-代理	用户-代理	用户-代理
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

设置“静态站点传送”选项

如果公司使用编译的站点，您可以控制静态站点的缓存时间长度，以及对于站点编译所支持的自适应移动布局，移动用户-代理调用的内容。

- [在发布时启用自动编译](#)
- [覆盖编译站点的默认高速缓存控制标头](#)
- [指定移动用户-代理以支持编译的自适应布局](#)

在发布时启用自动编译

Oracle Content Management 中内置的编译服务可帮助您轻松地编译站点，而无需任何其他配置。您的服务管理员也可以手动设置编译服务进行测试，以验证站点编译或使用定制库。无论将编译服务用于什么用途，您都可以选择在发布或重新发布站点时编译站点。

1. 在登录到 Oracle Content Management 之后，单击侧面导航菜单中的站点。
2. 选择要修改的模板，然后单击属性。
3. 单击属性对话框中的静态传送。
4. 在启用自动编译部分，启用在发布后编译站点。
5. 完成之后，单击完成。

发布或重新发布启用了自动编译的站点时，将在站点页的站点磁贴上跟踪和显示发布状态。发布完成后，将在站点属性对话框的静态传送部分中跟踪编译过程。编译完成后，站点属性静态传送部分将会列出上次编译的日期和时间，并提供一个用于下载编译日志的链接。

覆盖编译站点的默认高速缓存控制标头

默认情况下，编译的站点在用户的浏览器中缓存 300 秒（5 分钟）。服务管理员可以更改此默认值，但是站点开发人员可以覆盖站点属性中特定站点的默认值。

1. 在登录到 Oracle Content Management 之后，单击侧面导航菜单中的站点。
2. 选择要修改的模板，然后单击属性。
3. 单击属性对话框中的静态传送。
4. 在缓存响应标头部分中，输入 `Cache-control: max-age=`，然后输入您希望在用户浏览器中缓存页的秒数。例如，`Cache-control: max-age=600` 将页缓存 10 分钟。
5. 完成之后，单击完成。

如果您的实例使用 Akamai，要保留现有的 Akamai 设置，应将缓存响应标头部分留空。要覆盖现有的 Akamai 设置，应输入 `Edge-Control: !no-store,max-age=1800,downstream-ttl=1800`，其中黑体项是默认设置（以秒为单位）。

- `!no-store` 指示此设置应覆盖属性中相应的 Akamai 缓存配置。
- `max-age` 确定 Akamai 应缓存此页的时间长度。默认值为 1800 秒 (30 分钟)。在这段时间内，Akamai 将满足对此页的请求，而不从 Oracle Content Management 请求此页。
- `downstream-ttl` 指示 Akamai 将 "Cache-Control: max-age" 标头及其响应发送到客户端浏览器，并指示这些浏览器将此页缓存所分配的时间长度。默认值为 1800 秒 (30 分钟)。

指定移动用户-代理以支持编译的自适应布局

在对站点进行编译时，可以创建移动页以支持自适应布局。您可以指定用户-代理，这将导致服务器向站点提供移动页而不是标准编译页。与浏览器发送的用户-代理标头匹配时，在此处输入的值将被视为不区分大小写的子字符串。请注意，对于移动设备上的浏览器，用户-代理字符串中通常使用关键字 `Mobile`。

1. 在登录到 Oracle Content Management 之后，单击侧面导航菜单中的站点。
2. 选择要修改的模板，然后单击属性。
3. 单击属性对话框中的静态传送。
4. 在移动用户-代理部分中，对于您要为其提供移动页的用户代理，输入用逗号分隔的用户-代理子字符串列表。如果子字符串的任何部分与浏览器的用户-代理字符串相匹配，则会提供移动页。

指定和配置虚名 URL

设置站点虚名 URL 允许用户访问指定域（而不是 Oracle Content Management 实例域）中的 Oracle Content Management 站点。例如，实例 URL 可能为 `https://myinstance.cec.ocp.oraclecloud.com/site/MyCustomerSite/`，但 `https://www.example.com` 等更友好的 URL 更容易记住，更有助于品牌推广且更便于使用。此外，您还可以根据需要，将 Oracle Content Management 站点托管在定制路径下，例如 `https://www.example.com/store/`。

要使用虚域，需要执行几个步骤。

- 使用内容传送网络 (Content Delivery Network, CDN)。
- 使用域名系统 (Domain Name System, DNS) 管理虚域，以将域规范名称 (Canonical Name, CNAME) 记录映射到 CDN。
- 在 CDN 上部署有效证书以保护虚域。
- 配置 Oracle Content Management 站点以接受虚域上的请求。
- 配置 CDN 以将请求正确路由到 Oracle Content Management。

使用内容传送网络

内容传送网络 (CDN) 是由分布在各地的服务器组成的平台，用于提高网站的性能和安全性。CDN 可以最大程度地缩短用户和服务器之间的距离，同时优化内容请求的处理性能。尽管 CDN 的主要目标是提升用户体验，但 CDN 还可以用于变更传输中的请求，使用户看到的内容清晰明了（即使后端处理并非如此）。

要支持在虚域上托管 Oracle Content Management 站点，需要将 CDN 配置为处理来自所配置虚域的所有请求，并将其正确路由回 Oracle Content Management，然后对请求做出变更，以便 Oracle Content Management 正确安全地处理请求。

使用域名系统 (DNS) 管理域

任何域均可用作 Oracle Content Management 站点的虚域。将用作虚域的域配置用于 Oracle Content Management 站点之前，您必须先控制该域。

由于 DNS 的限制，您可能无法在没有 `www` 或其他子域（例如 `store.example.com`）的情况下使用根域（例如 `example.com`）。咨询 DNS 和 CDN 提供商以确定是否可以使用根域。

由于 DNS 在域级别而不是路径级别起作用，因此，如果 Oracle Content Management 托管在域的某些路径下而其他服务器托管在另外的路径下，路由需要由 CDN 处理。DNS 只能用于分离域和子域级别的通信。

部署证书

需要由 CDN 创建和托管用于保护虚域的证书。证书可以保护单个域、多个域和子域，或带通配符的子域（例如 `*.example.com`）。对于虚域，可以接受任何组合。所有受保护的域都将显示在证书详细信息中，因此，如果不打算公开共享这些详细信息，则应使用单独的证书。

注：

创建和托管证书的过程通常与具体 CDN 相关，CDN 需要规定如何恰当地执行该过程。

为 Oracle Content Management 站点配置虚域

使用虚域时，要使 Oracle Content Management 站点正确加载，您必须先向 Oracle Content Management 指出哪个站点 URL 在使用虚域。您可以在 Oracle Content Management 的站点属性选项卡中执行此操作。

1. 在 Oracle Content Management 中，单击侧面导航中的站点。
2. 选择要使用虚域的站点，然后从右键菜单中选择属性，或者在操作栏中选择子工具栏。
3. 在虚域字段中，输入虚域并单击保存。

 注:

Oracle Content Management 可能需要一个小时或更长时间进行设置，之后才能接受域的请求。您可以在站点属性面板中随时检查进度。

配置内容传送网络以路由请求

在 Oracle Content Management 正确配置并可接受请求之后，针对虚域提出的请求将通过 DNS 路由到 CDN，CDN 必须将请求正确转发到 Oracle Content Management。例如，如果具有站点 URL `https://myinstance.cec.ocp.oraclecloud.com/site/MyCustomerSite/` 的 Oracle Content Management 站点配置了虚域 `https://www.example.com/store`，则必须将 CDN 配置为：

- 识别虚域：`https://www.example.com/store`
- 标识使用该虚域的源 Oracle Content Management 实例：`https://myinstance.cec.ocp.oraclecloud.com/`
- 附加特定站点的站点路径：`site/MyCustomerSite/`
- 并将完整站点 URL 发送到源 Oracle Content Management 实例：`https://myinstance.cec.ocp.oraclecloud.com/site/MyCustomerSite/`
- Oracle Content Management 接收请求并将响应发送到 CDN，CDN 满足用户浏览器的请求，以仅向用户显示虚域：`https://www.example.com/store`

CDN 配置步骤通常与具体 CDN 相关，因此请与您的 CDN 提供商协作，以正确配置源定义。源定义将告知 CDN 如何访问源 Oracle Content Management 实例以及站点路径（用于附加正确的站点路径）。

 注:

请求不能包含以下任何路径：

- `/documents*`
- `/content/published*`
- `/osn*`
- `/pxysvc*`
- `/site*`

针对这些路径的请求不应包括站点路径。这些请求应解析到源 Oracle Content Management 实例的根，才能正确处理。

启用 Cobrowse 集成

Cobrowse 功能是用于 Oracle Cobrowse Cloud Service 的一款协作工具。

服务管理员必须先将与 Cobrowse Cloud Service 的集成作为接受的集成添加。请参见《*Integrating and Extending Oracle Content Management*》中的 "Integrate with Oracle Cobrowse Cloud Service"。

启用 Oracle Cobrowse Cloud Service 集成后，可以为站点配置该功能，然后将该功能添加到特定站点页以供使用。

要在站点上启用共同浏览，请执行以下操作：

1. 打开站点进行编辑。
2. 单击子工具栏中的 ，然后单击 。
3. 在 "Cobrowse" 部分中，选择在此站点上启用 **Oracle Cobrowse**。
4. 输入站点的启动程序脚本。Oracle Cobrowse Cloud Service 管理员可以访问 Cobrowse 管理控制台以获取合适的启动程序 Javascript 代码片段。有两种不同类型的启动程序。
 - 开始点 1: "Cobrowse" 按钮自动添加到页上。
 - 开始点 2: 允许您定制添加到页上的按钮和界面。
5. 单击关闭。

为站点启用 Cobrowse 之后，您可以将其添加到页上，也可以定制如何在页上使用该功能。有关详细信息，请参见[在页上使用 Cobrowse](#)。

将 Cobrowse 用于安全站点和站点构建器测试

Cobrowse 有两种模式：Instant (ICB) 和 Advanced (ACB)。对于安全站点或预览仍在开发中的站点，在 ICB 模式下使用共同浏览需要额外的配置。此配置在 Cobrowse 管理控制台中完成。请参见《[Using Standalone Cobrowse](#)》中的 "Log in to the Agent Console"。

在“定制功能”字段中，添加功能，允许代理查看已发布安全站点上使用密码保护的资源：

```
function () {
return {
passwordProtectedPatterns: [
"<PROTOCOL>://<DOMAIN>/authsite/*?*\"",
"<PROTOCOL>://<DOMAIN>/documents/*?*\"",
"<PROTOCOL>://<DOMAIN>/content/*?*#*\""
]
}
}
```

要为已发布安全站点和查看/预览开发中的站点使用此功能，请另外添加以下代码：



```
function () {
return {
passwordProtectedPatterns: [
"<PROTOCOL>://<DOMAIN>/authsite/*?*\"",
"<PROTOCOL>://<DOMAIN>/documents/*?*\"",
"<PROTOCOL>://<DOMAIN>/content/*?*#*\"",
"<PROTOCOL>://<DOMAIN>/sites/*?*#*\"",
"<PROTOCOL>://<DOMAIN>/_themes/*?*\"",
"<PROTOCOL>://<DOMAIN>/_sitescloud/*?*\"",
"<PROTOCOL>://<DOMAIN>/_compdelivery/*?*\""
]
}
}
```

配置受保护资源是一个新 Cobrowse 功能。它使用与 Cobrowse 页屏蔽相同的通配符 URL 模式。有关详细信息，请参见《*Cobrowse Deployment and Use Guide*》中的 [Configure page masking](#)。

添加分析跟踪

在站点构建器中，您可以向站点或页面中添加 JavaScript 跟踪代码片段以用于 Web Analytics 跟踪，使其更容易与外部分析提供方（例如 Google、Adobe 或 Oracle Infinity）集成。

要向站点添加分析跟踪，请执行以下操作：

1. 打开站点进行编辑。
2. 单击子工具栏中的 ，然后单击  分析。
3. 单击开关，以启用分析跟踪。
4. 在 **JavaScript 跟踪片段** 框中，添加新片段或编辑现有片段。
管理员可能已为您的环境提供代码片段。如果是这样，将在框中显示该代码片段。您可以定制该脚本，也可以添加您自己的脚本。如果您编辑由管理员提供的片段，则会出现一则消息，指出脚本已修改。要删除定制内容，请单击还原为最新的租户脚本。

下面是 Google Analytics 跟踪片段的示例：

```
<!-- 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>
```

您必须保存并发布此更改，必要时，先使站点联机，然后为站点收集分析内容。

查看分析数据

您发布站点并使其联机后，可以在供应商的站点（例如 Google Analytics）上查看跟踪的分析数据。如果您使用了用于 Oracle Infinity 分析跟踪的片段，请转至 Oracle Infinity 主页并单击分析来查看数据并选择或创建报告。

12

发布站点

下面我们来了解如何使站点联机、使其脱机以及发布站点更改。


- [使站点联机或脱机](#)
- [发布站点更改](#)

使站点联机或脱机

当站点处于联机状态时，具有适当访问权限的用户可以通过分配的地址 (URL) 使用标准 Web 浏览器查看该站点。当站点处于脱机状态时，站点不可供公众查看。您只能在 Oracle Content Management 中查看站点。

在将站点联机之前，站点必须已发布。要发布站点，请参见[发布站点更改](#)。

右侧的状态图标显示站点是联机还是脱机：

- 如果站点之前从未发布，将显示破折号 (-)。
- 如果站点联机，将显示 。
- 如果站点脱机，将显示 。

有关哪些人可以访问联机或脱机站点，请参见[了解站点安全性](#)。

要更改站点的状态，您必须是站点所有者或具有管理者角色，或者，如果启用了站点管理，则站点管理员可以更改任何站点的状态，无论该站点是否已与之共享。

1. 在“站点”页上，从列表中选择站点。
2. 要使站点联机或脱机，请在右键单击菜单中选择[联机](#)或[脱机](#)。此外，还可以单击站点磁贴中的  使站点联机，或单击站点磁贴中的  使站点脱机。

系统将提示您确认选择。

在使站点联机时，将创建站点的完全呈现的 HTML 版本，并将它复制到 Oracle Cloud 中的托管位置。联机站点在站点名下显示其 URL。默认 URL 的格式为：

```
https://service_name.identity_domain.sites.oraclecloud.com/site_name
```


在您使站点脱机时，站点及其文件夹和文件将从 Oracle Cloud 的托管位置中删除。

发布站点更改

要发布更改，您必须是站点所有者或者具有管理者角色。

发布站点更改时，请注意以下几点：

- 发布对联机站点的更改时，提交的任何更改都将立即对可访问该站点的所有用户可见。

- 发布对站点所做更改时，可以发布所有定向到站点通道的资产，也可以仅发布站点页面上使用的资产（包括推荐）。例如，如果您发布所有定向到站点通道的资产，而且站点中包含的列表引用了不是直接包含在站点中的内容项，引用的资产也将发布。
 - 发布对多语言站点的更改时，只会发布在站点关联的本地化策略中定义的语言对应的翻译。
 - 如果之前发布了某些语言的翻译，但现在站点关联的本地化策略中不再包含这些语言的定义，则会从已发布的站点中删除这些语言的翻译内容。
 - 如果从多语言站点中删除了某页，则在发布站点时也会删除对应的翻译页。
1. 要发布站点更改，请选择站点，然后在右键单击菜单或操作栏中选择发布。
 2. 选择是发布站点以及所有定向到站点发布通道的资产，还是发布站点以及所有添加到站点页面的资产。
 3. Oracle Content Management 会验证所有站点字符串和资产是否通过了要求的审批并具有所需的翻译。如果站点关联的本地化策略所要求的翻译不存在，资产未标记为已翻译，或者要求审批的资产未标记为已批准，您将无法发布站点。展开项目可以查看更多详细信息。如果存在问题，请更正后重新尝试发布。如果所有项都有效，请单击发布。系统将提示您确认选择。
 4. 如果站点尚未联机 ()，您需使站点联机，以使用户可访问该站点。

13

保护站点

发布站点并使其可供联机使用后，您可能希望控制哪些人可以访问该站点。

当您保护站点时，您可以指定哪些用户组可以基于分配的角色访问您发布的（联机）站点。这些角色是服务范围的角色，由服务实例的管理员分配。

- [了解站点安全性](#)
- [更改站点安全性](#)

如果您是管理员，则还可以执行其他与站点安全性相关的操作，例如启用定制登录；允许共享站点和主题；限制站点、模板和组件的创建；启用管理；执行其他操作。请参见管理 *Oracle Content Management* 中的“配置站点和资产设置”。

了解站点安全性

您可以应用安全性，来控制谁可以查看已发布（联机）站点、谁可以查看站点上的安全内容并与之交互以及谁可以查看和编辑未发布（脱机）站点。

站点安全性

发布站点并使其可供联机使用后，您可能希望控制哪些人可以访问该站点。根据系统和站点管理员配置环境的情况，您可以使站点可供任何人公开访问、限制站点仅可供注册用户访问或者限制站点仅可供特定用户访问。

您必须是站点所有者或者具有管理者角色，才能更改站点安全性或任何其他设置。要更改登录要求，站点必须处于脱机状态。但是，要更改指定的用户或用户角色，站点可以处于联机状态。在您使站点脱机时，站点及其文件夹和文件将从 Oracle Cloud 的托管位置中删除。

如果启用了站点管理，则可用的安全选项可能受模板策略限制；如果禁用了站点管理，则可用的安全选项可能受租户策略限制。请参见[了解站点管理](#)。

当您保护站点时，您可以指定哪些用户组可以基于分配的角色访问您发布的（联机）站点。这些角色是服务范围的角色，由服务实例的管理员分配。

- **云用户**：已验证的用户使用用户名和密码来登录服务实例。这包括所有通过验证的用户，不论用户具有还是不具有访问者角色或用户角色都可以。
- **访问者**：仅具有此角色的用户才能访问站点。例如，可以为以下用户授予此角色：他们可以查看已发布的站点，但无权访问此 Oracle Content Management 实例中的文件夹和文件。

注：

这不包括具有用户角色的用户，除非这些用户是站点所有者，或者已明确与这些用户共享站点。

- **服务用户**：仅具有此角色的用户才能访问站点。例如，可以为以下用户授予此角色：他们可以查看已发布的站点，并且有权访问此 Oracle Content Management 实例中的文件夹和文件。

- **特定用户**：仅添加为站点成员的用户可以查看已发布的站点。

站点共享

使用站点共享，您可以指定能够访问您未发布 (脱机) 站点的各个用户，并允许他们根据您的权限查看、修改或管理站点。如果您是站点所有者，或者已与您共享该站点并且向您授予了管理者角色，则您可以共享站点。

注：

您分配给用户的任何共享角色会作为其安全角色的补充。例如，如果某个用户具有访问者角色，但您与该用户共享了站点并为其指定了内容提供者角色，则该用户可以修改脱机站点；而只具有访问者角色的其他用户则只能查看联机站点。

- **查看者**：查看者可以在编辑器中查看站点，但不能更改任何内容。
- **下载者**：对于站点，下载者具有与查看者角色相同的权限。
- **内容提供者**：与查看者相同，还可以编辑站点、删除站点页以及在站点脱机时删除站点。
- **管理者**：与内容提供者相同，还可以添加用户和分配其角色、向联机站点发布更改以及将站点在联机和脱机之间切换。将自动为站点创建者 (所有者) 分配管理者角色。

创建站点时，将使用该站点名称创建通道。为了与其他人共享通道，您必须共享该站点并至少向某人提供一个内容提供者角色，该人员才能使用该通道发布资产。要发布站点，用户必须具有管理者角色。

组件共享

一些组件提供共享资源 (例如文件夹、文件或对话) 的访问权限。组件共享既考虑站点安全性 (谁可以查看已发布站点)，又考虑资源共享 (谁可以查看和使用文件夹、文件和对话)。

例如，当我们向站点添加文档管理器组件时，站点的所有访问者都可以查看文件夹的内容，另外，根据访问者角色以及其他任何权限，他们或许能够添加、修改或删除文件夹中的内容。

一般考虑事项：

- 站点作者授予的文件夹访问权限不能超过他们自己拥有的访问权限。例如，如果作者具有文件夹的下载者访问权限，则他们不能向站点访问者授予内容提供者权限。
- 组件中设置的权限可以补充访问者的权限。例如，如果访问者具有文件夹的查看者权限 (或者没有权限)，文档管理器组件可以根据组件中的所选角色授予更高的权限。这些增强权限仅在组件本身内有效。
- 如果站点访问者拥有的权限超过为组件指定的权限，则个别权限会覆盖在组件上设置的权限。
- 对文件夹授予的权限适用于该文件夹中嵌套的文件夹和文件。

对于公共站点：

- 对话组件仅在安全站点上受支持。

- 默认情况下，文档管理器组件向所有访问者提供相关文件夹的下载者权限。您可以根据上面列出的指南更改角色，并且您可以通过组件本身的设置来限制向用户显示的选项。
- 文件夹列表和文件列表组件向所有用户授予下载者访问权限。无论是什么角色，用户都可以查看和下载文件。

安全站点 URL

在您使站点联机时，将创建站点的完全呈现的 HTML 版本，并将其复制到 Oracle Cloud 中的托管位置。联机站点在站点名下显示其 URL。

不安全站点的默认 URL 格式为：

```
https://service_name.identity_domain.sites.oraclecloud.com/site_name
```

安全站点的默认 URL 格式为


```
https://service_name.identity_domain.sites.oraclecloud.com/authsite/site_name
```

请注意 URL 中添加了 authsite。

更改站点安全性

发布站点并使其可供联机使用后，您可能希望控制哪些人可以访问该站点。根据系统和站点管理员配置环境的情况，您可以使站点可供任何人公开访问、限制站点仅可供注册用户访问或者限制站点仅可供特定用户访问。

您必须是站点所有者或者具有管理者角色，才能更改站点安全性或任何其他设置。要更改登录要求，站点必须处于脱机状态。但是，要更改指定的用户或用户角色，站点可以处于联机状态。在您使站点脱机时，站点及其文件夹和文件将从 Oracle Cloud 的托管位置中删除。要使站点脱机，请从“站点”页中选择站点，然后在右键单击菜单中选择脱机，或单击操作栏中的

。系统将提示您确认选择。

要更改站点安全性，请执行以下操作：


1. 在“站点”页上，选择站点，然后在右键单击菜单中选择属性，或在操作栏中单击。
2. 单击站点安全性选项卡。

站点安全性选项卡上的可用选项取决于系统管理员配置站点安全设置的情况，如果启用了站点管理，站点管理员在此站点所基于的模板中配置安全设置的情况。请参见[站点入门](#)和[了解站点管理](#)。

3. 如果要求在站点联机时只有已注册用户在登录后才能查看，请单击“需要登录”旁的是。要删除要求并使站点在联机时可供公众访问，请单击“需要登录”旁的否。
4. 选择哪些注册用户组可以访问联机站点。要选择单个组，请先取消选择云用户。
 - 云用户：仅已验证的用户才能访问站点。已验证的用户使用用户名和密码登录您的域。这包括具有 **Oracle Content Management Cloud** 访问者角色或 **Oracle Content Management Cloud** 用户角色的用户。
 - 访问者：仅具有此角色的用户才能访问站点。这包括具有 **Oracle Content Management Cloud** 用户角色的用户。
 - 服务用户：只有可以登录此 Oracle Content Management 实例的用户才能访问站点。

- **特定用户**：指定可以访问站点的个人。单击**添加成员**。在搜索字段中输入用户名或用户名的一部分。从显示的列表中选择用户，并重复此操作以添加更多用户。完成后，请单击**添加**。要删除用户，请单击用户名下方的菜单中的**删除**。
5. 单击**保存**可保存更改并关闭窗口。

站点显示已脱机，需要登录。

6. 要使站点联机，请在右键单击菜单中选择**联机**，或单击操作栏中的 。单击**确认**以继续，然后单击**确定**。

在使站点联机时，将创建站点的完全呈现的 HTML 版本，并将它复制到 Oracle Cloud 中的托管位置。联机站点在站点名下显示其 URL。

不安全站点的默认 URL 格式为：

```
https://service_name-identity_domain.cec.ocp.oraclecloud.com/site/  
site_name
```

安全站点的默认 URL 格式为

```
https://service_name-identity_domain.cec.ocp.oraclecloud.com/site/  
authsite/site_name
```

请注意 URL 中添加了 `authsite`。

您可以添加注销 URL 并将其实施为链接或按钮或显示在菜单中的页。请参见[段落](#)、[按钮](#)和[添加页](#)。

注销 URL 的格式为：

```
https://service_name-identity_domain.cec.ocp.oraclecloud.com/cloudgate/  
logout.html?postlogouturl=%2Fsite%2Fauthsite%2Fsite_name
```

 **注：**

如上所示，`postlogouturl` 需要采用编码格式。

14

使用多语言站点

了解站点翻译以获得定向的多语言体验。

- [多语言站点概览](#)
- [翻译站点](#)
- [管理站点翻译作业](#)

多语言站点概览

可以将站点翻译为本地化策略中指定的任何语言。要查看站点的关联本地化策略中指定的语言，请在站点磁贴中单击语言旁边的向下箭头。要预览某个本地化版本，请选择相应语言，然后打开站点。



注：

Oracle Content Management 入门版无法创建多语言站点。要利用完整的功能集并消除站点限制，请升级到 Oracle Content Management 高级版。

选择站点以进行翻译时，将创建站点文件的 .zip 文件。

注：

如果站点包含来自多个资料档案库的资产，只有来自默认资料档案库的资产包含在该 .zip 中。

该 .zip 文件包含以下文件和文件夹：

- assets 文件夹 — 仅当您选择翻译整个网站或只翻译目标资产时，才会存在此文件夹。
 - root 文件夹

- * 每个内容项对应的一个 `<contentItem_ID>.json` 文件 — 包括内容项中字段的名称、说明和可翻译字符串，以及有关不应编辑的内容项的其他信息。
- `job.json` — 描述翻译作业的文件。不要编辑此文件。
- `site` 文件夹 — 仅当您选择翻译整个网站或只翻译站点内容时，才会存在此文件夹。
 - `root` 文件夹
 - * 每个站点页对应的一个 `<page_ID>.json` 文件 — 包括页上组件的名称、标题、说明、关键字、页眉、页脚和可翻译字符串，以及有关不应编辑的页的其他信息。如果您有定制组件，它们可能已配置为使用可翻译字符串。请参见[为多语言站点开发可翻译组件](#)。
 - * `siteinfo.json` — 包含说明、关键字、页眉和页脚。
 - * `structure.json` — 包含导航和站点结构。
 - `job.json` — 描述翻译作业的文件。不要编辑此文件。

翻译站点

如果您在某个已翻译站点中的页面上有资产（直接显示在页中或在内容列表中），并且您已翻译这些资产，这些资产将以与站点相同的语言显示。您还可以将内容项与站点分开翻译，请参见[本地化内容项](#)。

翻译包含来自多个资料档案库的资产的站点时，只有来自默认资料档案库的资产包含在翻译作业中

注：

Oracle Content Management 入门版无法使用翻译。要利用完整的功能集并消除站点限制，请升级到 Oracle Content Management 高级版。

创建翻译作业

导出翻译包进行手动翻译或使用翻译连接器时，需要创建翻译作业。

1. 选择要翻译的站点，然后单击翻译。可能需要单击更多才能看到翻译选项。
2. 输入要导出的站点文件的信息 — 翻译作业，然后单击创建：
 - 输入翻译作业的名称。
 - 选择站点的翻译目标语言。
 - 选择是导出整个站点的站点内容（页、结构和站点信息）及目标资产、只导出站点内容，还是只导出目标资产。
 - 选择翻译连接器或者选择导出翻译包进行手动翻译。
 - 完成后，单击创建。

翻译作业完成后，翻译包将导入到 Oracle Content and Experience Oracle Content Management。

手动翻译源语言文件

如果要手动翻译内容，需要在翻译作业完成后下载站点文件的 .zip 文件。

1. 单击横幅中的翻译作业，选择翻译作业，然后单击下载。
2. 对于选为目标语言的每种语言，在 .zip 文件中创建一个与 root 文件夹同级的文件夹，例如 de、es 和 fr。可以翻译一部分选定的语言。例如，可以现在翻译为德语 (de)，以后再翻译为西班牙语 (es) 和法语 (fr)。
3. 将 root 文件夹中的所有 .json 文件复制到每个语言文件夹。
4. 将所有 .json 文件中的字符串翻译为适当语言。不要删除 .json 文件中的任何字符串并且不要重命名该文件。
5. 压缩 assets (如果您的翻译作业包含资产) 和 site (如果您的翻译作业包含站点内容) 文件夹，以及 job.json、root 和包含翻译文件的所有语言文件夹。


导入手动翻译的包

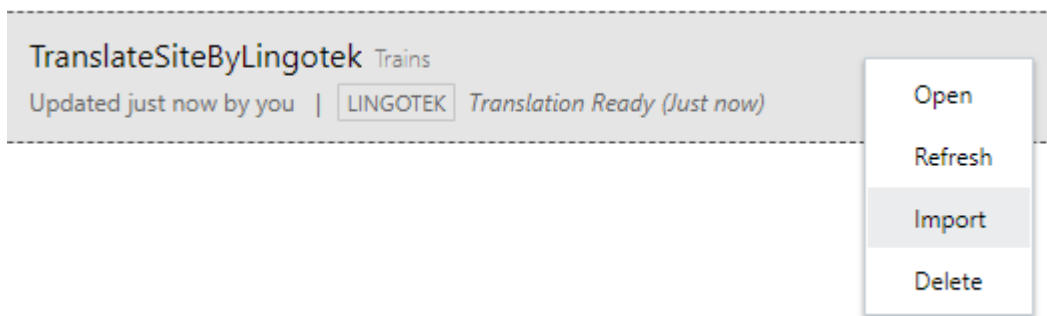
无论翻译是手动还是自动完成，都需要导入完成的翻译包。

1. 在翻译作业页上，单击导入。
2. 单击上载，选择已翻译站点文件的 .zip 文件，然后单击打开。
3. 上载完成后，单击确定
4. Oracle Content Management 验证作业中定义的翻译是否都位于 .zip 文件中。如果要查看哪些页和资产包含在翻译作业中，请单击对话框中的链接。
5. 准备好导入翻译时，单击导入。
导入的状态显示在横幅的上面。可以通过单击详细信息查看作业的详细信息。

从翻译连接器导入翻译的包

无论翻译是手动还是自动完成，都需要导入完成的翻译包。

1. 单击  并选择翻译作业。
2. 右键单击完成的翻译作业并选择导入。



3. Oracle Content Management 验证作业中定义的翻译。如果要查看哪些页和资产包含在翻译作业中，请单击对话框中的链接。
4. 准备好导入翻译时，单击导入。
导入的状态显示在横幅的上面。可以通过单击详细信息查看作业的详细信息。

导入翻译后，您需要发布站点并确保站点已联机，以使翻译在站点上可见。

如果在您导入翻译包后在翻译服务处对翻译进行了任何更改，可以单击刷新来更新翻译。





如果在翻译后更改站点，则需要翻译任何新字符串或编辑过的字符串。在站点构建器中编辑站点时，可编辑站点的默认语言版本。您进行的任何站点结构更改（例如，添加组件或重新排列页）将复制到站点的本地化版本中。之后，您可以创建新翻译作业来翻译更新的字符串。

管理站点翻译作业

选择站点以进行翻译时，将创建一个翻译作业。然后可以下载要进行翻译的文件，翻译这些文件，再导入已翻译的文件。

要创建翻译作业，请参见[翻译站点](#)。

“翻译作业”页列出所有翻译作业及其状态：

- （就绪）— 站点文件的 .zip 文件随时可以下载。
- （正在进行）— .zip 文件已下载。状态将保持正在进行，直到已成功导入所有目标语言的所有翻译。
- （完成）— 已成功导入此作业的所有目标语言的翻译。
- （失败）— 翻译作业失败。您应该已在横幅上方看到有关作业失败原因的失败消息。如果您需要再次查看该消息，您可以重新提交翻译作业。请更正问题，然后重新提交作业。

您可以执行以下操作：

- 要查看翻译作业的详细信息，请打开该作业。详细信息包括源语言 and 所有选定目标语言以及那些翻译的状态。
- 要下载站点文件的 .zip 文件，请选择作业，然后单击下载。
- 要删除某个作业，请选择该作业，然后单击删除。
- 要导入翻译，请依次单击导入和上载，选择已翻译站点文件的 .zip 文件，然后单击确定。
Oracle Content Management 验证作业中定义的所有翻译是否都位于 .zip 文件中。如果要查看哪些站点页和资产包含在翻译作业中，请单击对话框中的链接。准备好导入翻译时，单击导入。

用于翻译的区域设置

在提交要翻译的项时，通过代码标识目标语言，以便语言服务提供商知道要将该项翻译并返回的语言。例如，**fr** 表示法语，**de** 表示德语。

可以扩展这些代码以表示更多地区方言。例如，**de-LI** 是表示列支敦士登所用德语的代码，**de-LU** 是表示卢森堡所用德语的代码。但是，如果语言服务提供商不支持某个地区方言，则提供的代码将截断为两个字符的基本语言。例如，对于 **de-LI** 和 **de-LU**，代码将截断为 **de**。

但是，如果语言服务提供商支持一种地区方言，但并不支持所有地区方言，可能会发生替代。例如，**ms-BN** 是表示文莱所用马来语的代码，但是，如果语言服务提供商不支持该方言，则该方言可能会切换到语言服务提供商支持的方言（例如 **ms-MY**，这是表

示马来西亚所用马来语的代码)。如果语言服务提供商无法区分不同方言(例如,表示伯利兹所用英语的 **en-BZ** 和表示牙买加所用英语的 **en-JM**),则其将截断为基本语言(在本例中为表示英语的 **en**)。

用于翻译的定制区域设置

开发人员可以根据组织的需求创建定制区域设置。定制区域设置代码包括基本语言、任何地区方言代码(如果适用)、一个用来表示定制区域设置的 **x** 以及组织所需的任何其他用来标识定制的内容。例如,英语的定制区域设置代码可能为 **en-JM-x-custom**。

由于定制区域设置对组织是唯一的,因此在提交进行翻译时,定制区域设置代码会截断为基本语言和地区方言(如果语言服务提供商支持)。在以上示例中,**en-JM-x-custom** 将截断为 **en-JM**,这会消除代码中特定于定制内容的部分。或者,如果语言服务提供商不支持牙买加对应的地区方言代码(JM),则它可能会截断为仅保留基本语言 **en**。

为 URL 重定向设置区域设置别名


可以轻松地在运行时和运行时预览时用于站点 URL 中的区域设置来设置别名。如果组织使用的定制区域设置可能很长,因此增加了 URL 的复杂性,这尤其有用。

例如,您可能有一个定义为 **en-GB-x-cornish** 的定制区域设置,它在 URL 中看上去类似如下所示:

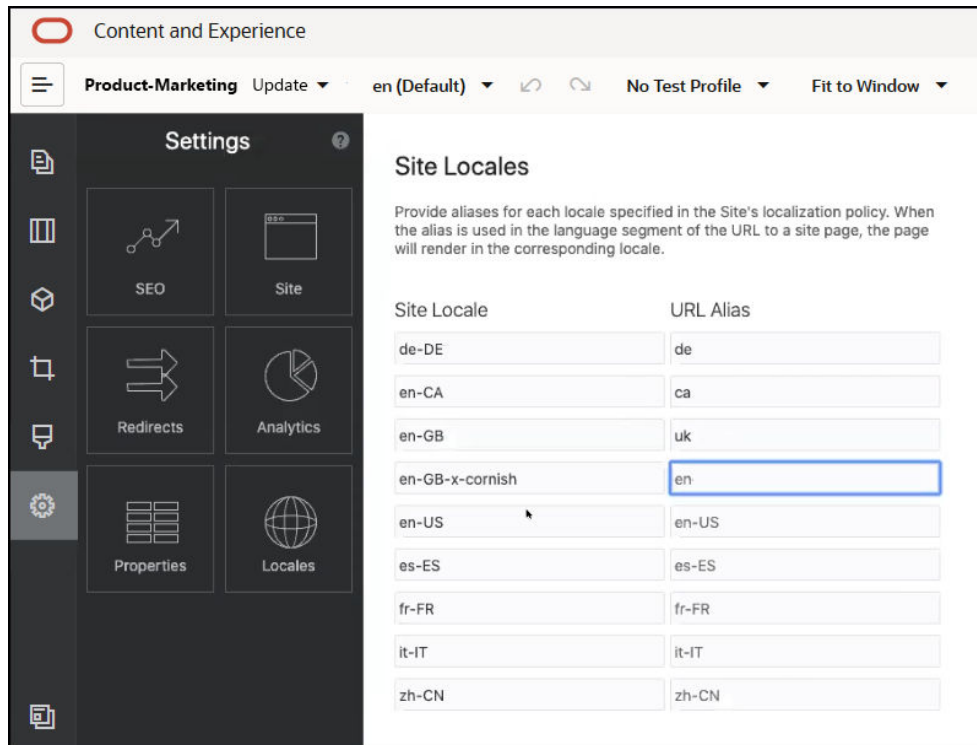
```
https://example.com/site/BlogSite/en-GB-x-cornish/home.html
```


添加别名后,可以重定向到较简单的 URL:

```
https://example.com/site/mysite/en/home.html
```

1. 打开站点进行编辑。
2. 单击子工具栏中的 , 然后单击区域设置。

将在对应的“URL 别名”字段旁边显示站点中使用的所有区域设置列表。



3. 在要为其设置别名的每个站点区域设置旁边输入别名，依次单击关闭和保存。
4. 要预览在运行时用于 URL 中的别名，请单击 .
5. 发布更新时，更改也将发布并投入使用。

使用站点重定向或 URL 映射

当您重新构建或移动网站时，可以将用户请求从旧 URL 重定向到当前 URL。为 URL 指定 30x 重定向可以在站点重新设计方案之间维护书签或已发布的链接。

当您从其他基础结构技术转到 Oracle Content Management 托管的站点时，搜索引擎中信誉评级较高的页可以转到不同 URL。借助重定向可以重新组织站点的 URL 结构并保留搜索引擎评级。

- [计划重定向](#)
- [添加站点重定向](#)
- [在 JSON 文件中指定重定向规则](#)
- [将重定向规则文件上传到站点](#)
- [映射站点 URL](#)

计划重定向

您可以为指定的 URL 指定发送 HTTP 30x 响应的重定向。如果请求不匹配指定的重定向之一，则将正常处理 URL，并以正常方式返回页。

您可以创建指定重定向的 JSON 文件，并将该文件上传到服务器。服务器将在处理传入的请求 URL 时使用该 JSON 文件。

可以使用以下两种重定向规则将传入 URL 重定向到新位置：

- [简单的字符串到字符串匹配](#)
- [简化的通配符匹配](#)

简单的字符串到字符串匹配

对于简单的字符串匹配和替换，您可以指定明确的 URL，然后通过将每个 URL 直接映射到目标 URL 来重定向相应的 URL。

下表显示了一些简单的字符串到字符串匹配示例。

源 URL	目标位置 URL
/index.html	/home.htm
/products/widget	/items/knickknack
/index?page=widgets	/items/widgets

字符串到字符串映射比较简单，很容易理解和测试。这些规则使用简单的字符串匹配快速进行评估，并映射查找结果。

但是，使用 URL 查询参数时的灵活性较低。它们需要完全匹配。额外的 URL 参数或不同顺序的参数可能会导致规则不匹配。

简化的通配符匹配

简化的通配符匹配允许一个规则匹配多个 URL，同时还限制获取结果所需的正则表达式回溯数量。

由于正则表达式的编写可能较复杂，并且构造不佳的正则表达式的求值时间可能无法确定 (ReDoS)，因此另一种类型的规则允许使用简化的匹配机制。它使用通配符 ("*") 来匹配传入 URL 中的零 (0) 个或多个字符，并使用带索引值的关键字 'wildcard' 将 URL 的传入部分复制到重定向的 URL。

下表显示了一些简化的通配符匹配示例。

源 URL	目标位置 URL
/old/*	/new/<\$wildcard(1)\$>
/dispatch.asp?page=* & facet=Lang*	/page<\$wildcard(1)\$>/<\$wildcard(2)\$>

简化的通配符匹配使得 URL 匹配功能比简单的字符串到字符串匹配强大，但却没有无限制的正则表达式处理。语法很简单，您可以对 URL 使用模式匹配以便能够处理大量具有一种模式的 URL。

由于它们是基于正则表达式的，因此通配符规则的求值速度会略低于简单字符串匹配。大量规则可能会对一般页传送性能造成负面影响。



添加站点重定向

如果站点 URL 更改，则重定向将一个 URL (源) 转发到另一个 URL (目标)。这有助于保留用户书签和搜索引擎评级。

可以使用两种类型的重定向：

- 永久重定向，使用 301 HTTP 服务响应代码
- 临时重定向，使用 302 HTTP 服务响应代码

要上载 `redirect.json` 文件，请执行以下操作：

1. 打开站点进行编辑。
2. 单击子工具栏中的 ，然后单击  重定向。
3. 单击选择要上载的文件，选择要使用的文件，然后单击确定。
4. 发布更新时，更改也将发布并投入使用。

在 JSON 文件中指定重定向规则

可在 JSON 文件中指定 URL 的重定向规则。

在 JSON 文件中使用以下格式指定 URL 的重定向规则。

```
{
  "redirectRules":
  [
    {
```

```

        "type": "string",
        "comment": "this rule is applied first",
        "expression": "/index.htm",
        "location": "/home.html"
    },
    {
        "type": "wildcard",
        "expression": "/items/*?page=*",
        "location": "/<$page$>?item=<$wildcard(1)$>",
        "code": 302
    }
]
}

```

JSON 文件中的外部包含结构是数组。该数组包含规则实例。

将按顺序首先评估 "string" 规则，然后是 "wildcard" 规则。一旦一个规则匹配，将放弃评估后面的规则，并生成相应的重定向。

每个规则都具有以下属性：

- "comment" 属性是可选字符串，对规则的评估没有影响。它包括注释或评注。
- "expression" 属性是必需字符串，与传入站点相对 URL 匹配。在通配符规则中，星号 (*) 标记匹配零个或多个字符。
- "location" 属性是必需字符串，指示重定向的位置或目标。重定向可以是完整或相对 URL。
- "code" 属性是可选整数，提供发出重定向时使用的 HTTP 响应代码。该值必须是以下整数之一：
 - 301：指示资源永久移动。省略 "code" 属性时，该值是默认值。
 - 302：指示资源临时移动。
- "type" 属性是可选字符串，指示重定向规则的类型。该值必须是以下字符串之一：
 - "string" 指定更快速的规则，其表达式与整个输入 URL 完全匹配。
 - "wildcard" 指定通配符规则，其可与许多 URL 匹配。这是省略该属性时的默认值。

位置标记

可以使用位置标记帮助生成重定向位置。以下位置标记中的每个标记都可以帮助指定重定向：

- <\$urlPath\$>：匹配 URL 的路径部分。
- <\$urlQueryString\$>：来自匹配 URL 的完整 URL 查询字符串。
- <\$urlQueryStringExcept(name1,name2)\$>：来自匹配 URL 的完整 URL 查询字符串，去掉命名的参数。
- <\$wildcard(N)\$>：匹配 URL 中匹配通配符的基于 1 的指数。（这类似于正则表达式中的 \1..\9。）
- <\$name\$>：命名的查询字符串参数的值。例如，如果输入中具有查询字符串 msmith: ? page=42，则可以在位置中使用 <\$page\$> 以将 '42' 放入该位置。

限制

以下限制应用于整个 redirects.json 文件以及它包含的规则：

- Oracle Content Management 接受的最大整体文件大小为 250 KB。
- redirects.json 文件中的最大规则数为 1,000。
- 规则的最大 "expression" 长度为 1,000 个字符。
- 规则的最大 "location" 长度为 2,000 个字符。
- 通配符规则表达式中的最大 '*' 标记数为 10。

字符串匹配示例

规则：

```
{
  "type": "string",
  "expression": "/old/page.jsp?id=material&type=glass",
  "location": "/new/<$id$>.htm"
}
```

以下 URL 将匹配该规则：

/old/page.jsp?id=material&type=glass

- 得到的位置将为： /new/material.htm
- 整个 URL 匹配，包括查询字符串。
- 虽然 <\$id\$> 用于位置，但是对于此示例，它并不是必需的，因为只有一个可能的查询字符串可以匹配。位置可能已写入为 /new/material.htm。

以下 URL 将不匹配该规则：

- /old/page.jsp
(规则的表达式提供必须匹配的查询字符串。)
- /old/page.jsp?id=material&type=glass&index=2
(候选 URL 中额外的 &index=2 不与规则表达式完全匹配。)
- /old/page.jsp?type=glass&id=material
(查询字符串参数的排序在 "string" 规则中必须匹配。)

通配符匹配示例

规则：

```
{
  "type": "wildcard",
  "expression": "/old/*/pages/*?id=*&item=sheet-*",
  "location": "/new/<$id$>/<$wildcard(4)$>.html"
}
```

以下 URL 将匹配该规则：

- /old/phones/android/pages/info.asp?id=XT1045&item=sheet-specs
– 得到的位置将为： /new/XT1045/specs.html

- URL 的路径部分匹配，因此还针对匹配的条件检查查询字符串。
- 此示例中的参数碰巧匹配规则表达式中的参数排序，但这不是必需的。
- /old/phones/android/pages/info.asp?item=sheet-specs&id=XT1045
 - 得到的位置将为： /new/XT1045/specs.html
 - URL 的路径部分在问号 (?) 之前匹配规则表达式，因此还检查参数是否存在匹配。
 - 虽然参数在规则表达式中按不同顺序列出，但参数是单个匹配的。
- /old/phones/android/pages/info.asp?id=XT1045&item=sheet-specs&unrelated=thing
 - 得到的位置将为： /new/XT1045/specs.html
 - URL 的路径部分匹配，因此还针对匹配的条件检查查询字符串。
 - 候选 URL 具有额外的 &unrelated=thing 参数，但是因为规则表达式中命名的查询参数匹配，所以该规则视为匹配。
 - unrelated 参数作为标记 <\$unrelated\$> 在位置中提供，并将具有值 thing，即使它对规则匹配没有帮助。

以下 URL 将不匹配：

- /old/pages/info.jsp
(URL 的路径部分与规则表达式的路径部分不匹配。)
- /old/phones/android/pages/info.asp
(URL 的路径部分与规则表达式的路径部分匹配，但是规则表达式中的查询参数不匹配。)
- /old/phones/android/pages/info.asp?id=cellular
(URL 的路径部分与规则表达式的路径部分匹配，但并非规则表达式中的所有查询参数都匹配。)

定义标记数组

您还可以在 `redirects.json` 文件中创建标记定义数组，以帮助配置支持多个虚名 URL 的重定向。这样，您可以根据传入 URL 的特征进行适当的重定向。

在 `redirects.json` 文件中使用以下格式定义用于重定向规则 URL 的标记。

```
{
  "tokenDefinitions":
  [
    {
      "token": "sitePrefix",
      "type": "hostmatch",
      "expression": "example.com"
      "value": ""
    },
    {
      "token": "sitePrefix",
      "type": "hostmatch",
      "expression": "*.com"
      "value": "/site/Starter-Site"
    }
  ],
}
```

```

    {
      "token": "gotoRedirect",
      "type": "pathmatch",
      "expression": "*oracle*"
      "value": "https://www.oracle.com"
      "flags": "caseinsensitive"
    },
  ]
}

```

tokenDefinitions 具有以下属性：

- "token"：要定义的标记的名称。
- "type"：以下项之一：
 - "hostmatch"，用于与传入 URL 的主机值匹配。
 - "pathmatch"，用于与传入 URL 的路径名值匹配。
 - "querymatch"，用于与传入 URL 的查询值匹配。
- "expression"：应该用于进行匹配的表达式。支持通配符。
- "value"：应该用于标记的值。
- "flags"：默认情况下，表达式匹配区分大小写，除非 flags 值设置为 caseinsensitive

计算标记的值时，将按顺序枚举 tokenDefinitions 数组。将使用第一个匹配定义。如果没有标记定义满足该标记，则将改用空字符串。出于方便和性能考虑，常用标记应该放置在 tokenDefinitions 列表的顶部。

tokenDefinitions 具有以下约束：

- 您最多可以创建 250 个标记定义。
- token 名称必须少于 100 个字符。
- expression 最多可以具有 10 个通配符。
- expression 必须少于 1000 个字符。
- value 必须少于 1000 个字符。

示例

例如，您可能具有以下 redirects.json 文件：

```

{
  "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"  
    },  
]  
}
```

在这种情况下，规则的 location 属性具有 `<$pathPrefix$>` 标记。pathPrefix 标记在 tokenDefinitions 部分中进行定义。如果传入 URL 与 "vanity.com" 匹配，则 pathPrefix 值将设置为 /fashion。这将用于 location 响应中，从而产生 /fashion/about/new-privacy-policy.html。

假设第一个虚域 URL 为 `http://example.com/legacy-privacy-policy.html`。这将与第一个且是唯一的重定向规则匹配。

此规则的已声明 location 为 `<$pathPrefix$>/about/new-privacy-policy.html`。在这种情况下，需要对 `<$pathPrefix$>` 标记求值。为此，将枚举 tokenDefinitions 数组以查找匹配项。

将考虑第一个标记定义。它的 token 是所需标记，因此将进一步对其求值。表达式 vanity.com 与传入 URL 的 example.com 不匹配，因此，此定义不满足要求，枚举将继续进行。

此时，没有更多的标记定义，因此空字符串用于 `<$pathPrefix$>` 标记的值。为此重定向返回的最终位置为 `/about/new-privacy-policy.html`。

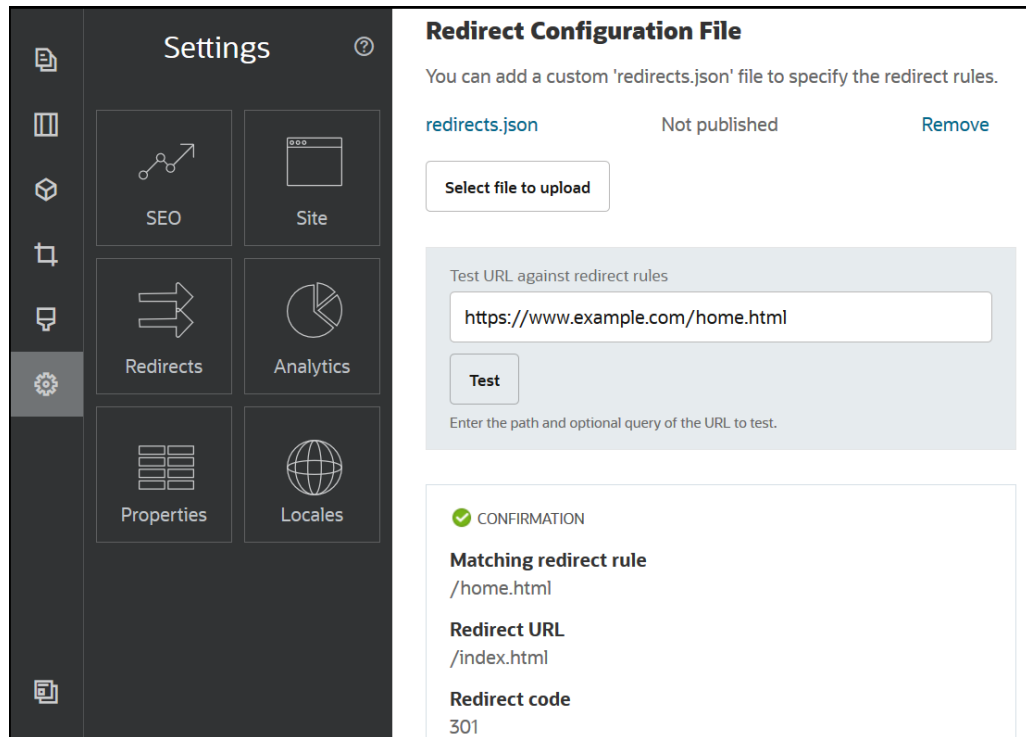
假设第二个虚域 URL 为 `http://vanity.com/legacy-privacy-policy.html`。与第一个 URL 一样，此规则的已声明 location 为 `<$pathPrefix$>/about/new-privacy-policy.html`。在这种情况下，需要对 `<$pathPrefix$>` 标记求值。为此，将枚举 tokenDefinitions 数组以查找匹配项。

将考虑第一个标记定义。与之前一样，它的 token 是所需标记，因此将进一步对其求值。表达式 vanity.com 与传入 URL 的 vanity.com 匹配，因此，此定义满足要求，并且值 /fashion 将用作标记的值。

因为已找到标记的匹配项，所以标记定义数组的枚举停止，并且最终位置计算为 `/fashion/about/new-privacy-policy.html`。

测试站点重定向



编辑站点时，您可以通过打开设置面板并单击重定向来测试站点重定向。输入要测试的 URL，然后单击测试。



将重定向规则文件上传到站点

可以将重定向规则上传到 Oracle Content Management 中的站点。

要将 `redirect.json` 文件上传到站点，请执行以下操作：

1. 打开站点进行编辑。
2. 单击子工具栏中的 ，然后单击 。
3. 单击选择要上传的文件，选择要使用的文件，然后单击确定。
4. 发布更新时，更改也将发布并投入使用。

映射站点 URL

使用 Oracle Content Management 创建和发布站点之后，您可以配置域名系统 (DNS)，这样就可以通过注册的域名来访问此站点，例如 `www.mysite.com`

域名系统 (DNS) 指定通过将您的域名映射到您的站点位置或者规范名称 (CNAME) 时，用户可以在什么位置找到您的 Web 页。

要映射域名，需要以下内容：

- Oracle Content Management 实例的 URL。通常为以下格式：
`service-tenant.documents.datacenter.oraclecloud.com`
- 由您的域名注册机构注册的域名。
例如 `www.example.com`。它还可以是子域，例如 `www.example.com/subdomain`。
- 具有内容传送网络 (CDN) 提供程序的账户 Oracle Content Management 将提供与 Akamai 的集成。要为您的实例配置 Akamai，请与 Oracle 技术支持联系。

如果您要使用自己的 CDN，而不是 Oracle Content Management 提供的 Akamai，请执行以下步骤。

不同域名系统提供商有不同的 Web 界面和不同的步骤来更新 CNAME 记录。以下步骤提供了您需要的信息以及所遵循的一般步骤。

要将站点 URL 映射到域名，请执行以下操作：

1. 向该域的内容传送网络提供商申请安全套接字层 (SSL) 证书。例如：`https://www.example.com`。
2. 配置内容传送网络，以便：
 - a. 内容传送网络接受该域的所有传入请求，并使用安全协议 (https) 进行转发。
 - b. 源指向来自 Oracle Content Management 的域：
`service-tenant.documents.datacenter.oraclecloud.com`
3. 更改 DNS 服务器区文件，以将域名映射到内容传送网络提供商提供的边缘服务器：
域 CNAME CDN 服务器
4. 等待更新传播。根据 DNS 服务，这可能需要 2 到 48 个小时才能在所有位置完成。
传播更改之后，您可以使用域名访问站点。例如：

`https://www.mysite.com/site_name`

默认情况下，Oracle Cloud REST API for Content Management 的端点在您使用为站点提供的标准 URL 时可用。例如，文件夹和文件列表组件使用 REST API 执行文件夹和文件操作。如果使用定制 URL，请确保对具有您的域名的端点拥有访问权限。例如：

`https://www.mysite.com/documents`

16

提高站点性能

可以通过利用浏览器高速缓存提高内容传送以及在浏览器中呈现的性能。首屏 (Above the fold, ATF) 呈现也可以改进网站呈现。

- [利用高速缓存提高性能](#)
- [首屏 \(Above the Fold, ATF\) 呈现](#)

利用高速缓存提高性能

交付内容项、数字资产和站点应充分利用访问者的浏览器高速缓存，从而提高浏览器中内容交付和呈现的性能。

站点、主题、内容项和数字资产将在访问者的浏览器高速缓存中缓存一段时间。更新站点、主题、内容项或数字资产后，URL 中的 cache-buster 键将更改，从而浏览器必须提取不同的 URL 并获得新项。

高速缓存键通过仅引用当前资源，帮助管理浏览器高速缓存的使用。虽然高速缓存键包含在 URL 中，但它是一个逻辑元素，而不是像通常那样是物理位置（文件夹）。高速缓存键中的更改不指向其他物理位置来查找资源；它只是通知服务器提取当前版本的资源。

资源可以是静态的，例如 CSS、JS 和图像文件，也可以是动态的，例如页数据、站点数据和内容项数据。有五种类别的资源用于构建网站：

- 产品资源 — 作为产品的一部分的资源，只要该产品发布新版本或打补丁时，就会进行更新。
- 站点资源 — 作为站点的一部分的资源，例如 `structure.json`、页数据和图像。发布站点时会更新这些资源。下文中介绍了控制器。
- 主题资源 — 作为主题的一部分的资源，例如布局、CSS 和图像。发布主题时会更新这些资源。
- 组件资源 — 作为定制组件的一部分的资源。这些资源包括 HTML、JS 和 CSS，以及构成组件的图像文件。发布组件时会更新这些资源。如果一个组件发生了更改并重新发布，则高速缓存键将针对所有组件发生更改，因为它是针对所有组件的单一键。
- CaaS 资源 — 为内容项和数字项提供服务的资源。发布或重新发布内容项或者更改集合目标时会更新这些资源。

以下主题介绍了 Oracle Content Management 运行时和站点构建器的高速缓存：

- [运行时高速缓存](#)
- [站点构建器高速缓存](#)

运行时高速缓存

对于运行时，Oracle Content Management Cache-Control 标头设置为 15 天。对于所有资源，都会向 URL 添加高速缓存键。

只要 URL 相同，浏览器都会从其本地高速缓存提供资源（如果有）。资源更新后，会在 URL 中更新高速缓存键，这可强制浏览器向服务器发出新的请求并更新本地高速缓存。

控制器（包含高速缓存键）也会缓存 1 分钟。因此，在长达 1 分钟的时间内会看不到任何更新的高速缓存键。

在运行时，服务器会返回 controller.html 及产品、站点、主题、组件和 CaaS 资源的最新高速缓存键。包含键的脚本将添加到 controller.html；例如：

```
<script type="text/javascript">
  var SCSCacheKeys = {
    product: '123',
    site: '456',
    theme: '789',
    component: '012',
    caas: '345'

  };
</script>
```

controller.js 使用这些键构造 URL，例如下表中的 URL。

资源类型	示例
产品资源	<i>/sitePrefix/productCacheKey/_sitesclouddelivery/...</i> <i>/mySite/_cache_947d/_sitesclouddelivery/</i>
主题资源	<i>/sitePrefix/themeCacheKey/_themesdelivery/themeName/...</i>
组件资源	<i>/sitePrefix/compCacheKey/_compdelivery/compName/...</i>
站点资源	<i>/sitePrefix/siteCacheKey/content/...</i> <i>/sitePrefix/siteCacheKey/structure.json</i> <i>/sitePrefix/siteCacheKey/pages/100.json</i>
CaaS 资源	<i>RegularCaaSUrl?cacheKey=caasCacheKey</i>

像这样将高速缓存键插入 URL 中后，Oracle Content Management 可以通过实际更改 URL 以使浏览器认为这实际上是新资源来强制浏览器加载更新的资源。

 注:

对于安全站点，仅缓存产品、主题和组件资源，而不缓存站点和 CaaS 内容。

站点构建器高速缓存

在站点构建器中，静态资源将缓存 15 天。

使用站点构建器时，将对产品、主题和组件资源进行缓存。（不对站点和 CaaS 资源进行缓存。）启动或刷新站点构建器时，将重新生成主题和组件高速缓存键。

如果对主题或组件进行更改并希望该更改显示在站点构建器中，则需要刷新站点构建器 (F5)。

首屏 (Above the Fold, ATF) 呈现

在采用 ATF 呈现的情况下，网站外观的加载速度高于不采用时的正常加载速度。目标是先呈现页上可见的所有部分，然后在用户向下滚动之前，呈现页上最初不可见的剩余部分。

内容框可以具有“首屏”标志，即在选项卡上显示图标。

要以此新方式呈现内容框，必须为其标记 `scs-atf`，如下所示：

```
<div class="scs-slot scs-atf" id="headline"></div>
```

组件在完成呈现时需要通知呈现器。现成的组件默认执行此操作。定制组件可以进行额外调用，并需要执行以下操作：

1. 向呈现器通知它要求呈现器等到自己完成呈现。
2. 在完成呈现时通知呈现器。

对于第 1 项，在定制组件的 `appinfo.json` 文件中，加入以下属性：

```
"initialData": {  
  . . .  
  "customRenderComplete": true,  
  . . .  
}
```

对于第 2 项，在组件的 `render.js` 文件中，通过调用以下方法来确保在完成时告知呈现器：

```
SitesSDK.setProperty('renderComplete', true);
```

如果 ATF 内容框中的一些组件未及时传回其已完成，则呈现器将等待 2 秒，然后再继续处理页的其余部分。如果您认为这个时间不够长，您可以通过在页模板中声明以下全局变量来延长该时间。

```
var SCSAtfPassTimeout = 3000;
```

 注:

时间以毫米为单位, 因此此示例是将超时时间设置为 3 秒。

API 可提供 ATF 过程的诊断数据。您可以在调试控制台中调用以下方法, 也可以从页面访问它 (如果需要) :

```
SCSRenderAPI.getRenderMetrics();
```

例如:

```
{currentTime: 16243.400000000001, renderStartTime: 264.36,
atfPassEndTime: 306.535, mainPassStartTime: 316.475, mainPassEndTime:
331.38500000000005, ...}
```

1. atfComponentCount:13
2. atfPassEndTime:306.535
3. completionCount:23
4. completionRecords:Array(23)
 1. 0:{atf: true, componentId: "a7afdd33-3fbb-4329-bc1b-6be60056a995", time: 280.065}
 2. 1:{atf: true, componentId: "edfcfcb4-b0d3-422f-aa59-5c925bbbbee", time: 283.54}
 3. 2:{atf: true, componentId: "c1c3aec8-e52f-406c-8c29-ab69c05877ed", time: 283.56000000000006}
 4. 3:{atf: true, componentId: "b3a31dc6-62a1-44d9-9c80-bdb2c5bedaaa", time: 284.13000000000005}
 5. 4:{atf: true, componentId: "c05aa1a2-c11c-4ef5-9051-4799c5bee24a", time: 284.15500000000003}
 6. 5:{atf: true, componentId: "bafd4047-06ec-4739-9b23-9db74f573f30", time: 294.665}
 7. 6:{atf: true, componentId: "e7d49528-0357-4b45-801e-b3a2716a086c", time: 297.995}
 8. 7:{atf: true, componentId: "a5f33674-4022-4138-8cc5-fef00c02a557", time: 299.78000000000003}
 9. 8:{atf: true, componentId: "ccfedc98-1dbd-440e-b867-5e683cea2ec5", time: 301.19500000000005}
 10. 9:{atf: true, componentId: "d691bc44-fed9-474a-9806-2191f46a5e2e", time: 302.46}


```
11. 10:{atf: true, componentId: "cf613054-05d8-40dd-83a0-718760d7bc73",
time: 303.79}

12. 11:{atf: true, componentId: "b4a6ef98-ffc8-48c7-987c-63346ee97bcc",
time: 305.115}

13. 12:{atf: true, componentId: "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: 320.45500000000004}

16. 15:{atf: false, componentId: "a48b5abb-49b2-4456-90bd-a3de998150c8",
time: 320.48}

17. 16:{atf: false, componentId: "a9650e6d-7e7e-42a2-b758-58f2aeab18a2",
time: 322.61500000000007}

18. 17:{atf: false, componentId: "aca9836a-f955-4aa7-8db2-fd3cf1189dea",
time: 324.23500000000007}

19. 18:{atf: false, componentId: "e3d7941c-fbc7-4da9-963b-e3810b6467d4",
time: 325.85}

20. 19:{atf: false, componentId: "eecde809-da54-4066-9326-73f9d9c35fe4",
time: 327.315}

21. 20:{atf: false, componentId: "e8f4fb16-4e15-4570-b7de-304e99e449a7",
time: 328.74}

22. 21:{atf: false, componentId: "a7baa06e-7f30-42c7-94f4-e171ab2edcd6",
time: 330.09000000000003}

23. 22:{atf: false, componentId: "fd603b96-2beb-4e87-a54f-12d0e264cd0a",
time: 331.38500000000005}

24. length:23

25. __proto__:Array(0)

5. componentCount:23

6. currentTime:16243.400000000001

7. mainPassEndTime:331.38500000000005

8. mainPassStartTime:316.475

9. renderStartTime:264.36

10. __proto__:Object
```

第 IV 部分

开发站点

此部分详细介绍了如何开始站点开发，可帮助您了解和开发模板、主题、样式、组件和其他有用项。其中包括以下章节：

- [定制设计和样式](#)
- [了解背景用法](#)
- [设置触发器和操作](#)
- [开发模板](#)
- [开发主题](#)
- [开发布局](#)
- [开发组件](#)

17

定制设计和样式

Oracle Content Management 的每个主题都必须具有设计，用于指定页的外观以及该主题的默认站点中所用组件的样式设置。

- [关于设计](#)
- [设计文件](#)
- [定制对话列表样式](#)
- [定制文件夹列表和文件列表样式](#)
- [定制社交栏图标](#)
- [为 Oracle Intelligent Advisor 配置交互式提问样式扩展](#)

关于设计

Oracle Content Management 的每个主题都必须具有设计，用于指定页的外观以及该主题中所用组件的样式设置。

用户创建站点时，他们必须选择模板，该模板默认情况下包括主题。您可以为站点创建主题或选择使用不同的主题。

创建主题以及设计页布局时，必须指定可用样式，这些样式会显示在将可用于用户的每个组件类型的“设置”面板中（段落、标题、图像等）。设计内可以有多个样式，以便为不同组件指定默认设置。可以针对文本字体和字号、图像框架等进行选择，然后将其保存为命名的设计。设计中的项会在呈现时影响页布局的外观。

请参见《通过 *Oracle Content Management* 构建站点》中的“使用站点页”。

设计文件

两个文件视为主题的默认设计文件：`design.json` 和 `design.css`。

- `design.json` 指定组件的样式
- `design.css` 提供类值的定义（例如颜色和字体）

这些文件位于主题结构中的 `/designs/default/` 目录中。

design.json 文件

`design.json` 文件具有以下结构：

```
{
  "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": []
    }
  }
}
```

每个 "styles":[] 条目都可以包含特定组件的样式列表。例如，标题组件提供以下默认样式：

```
"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"
  }
}]
```

 注:

如果为组件创建定制样式并将其映射到 `design.json` 中的样式，则无需像随系统提供的那些样式那样使用完整名称。只需指定要使用的字符串。例如，名称不使用 `"COMP_STYLE_BOX"`，仅使用 `"Box"`。这意味着仅名称 `"Box"`（而不是 `"COMP_STYLE_BOX"`）将显示在该组件“设置”面板的样式选项卡的列表中。

名称值映射到要在用户界面中显示的实际词语，例如：

```
"COMP_STYLE_FLAT": "Flat",  
"COMP_STYLE_HIGHLIGHT": "Highlight",  
"COMP_STYLE_DIVIDER": "Divider",
```

design.css 文件

`design.css` 文件提供类值的定义。这里有几个示例。

```
.scs-title-default-style {  
  color: #333333;  
  display: block;  
  font-family: "Helvetica Neue", "Helvetica", "Arial", sans-serif;  
  font-size: 24px;  
  font-weight: normal; }
```

```
.scs-title-style-2 {  
  background-color: #DEF300;  
  color: #333333;  
  font-family: adobe-clean, sans-serif;  
  padding-top: 2em;  
  padding-bottom: 2em; }
```

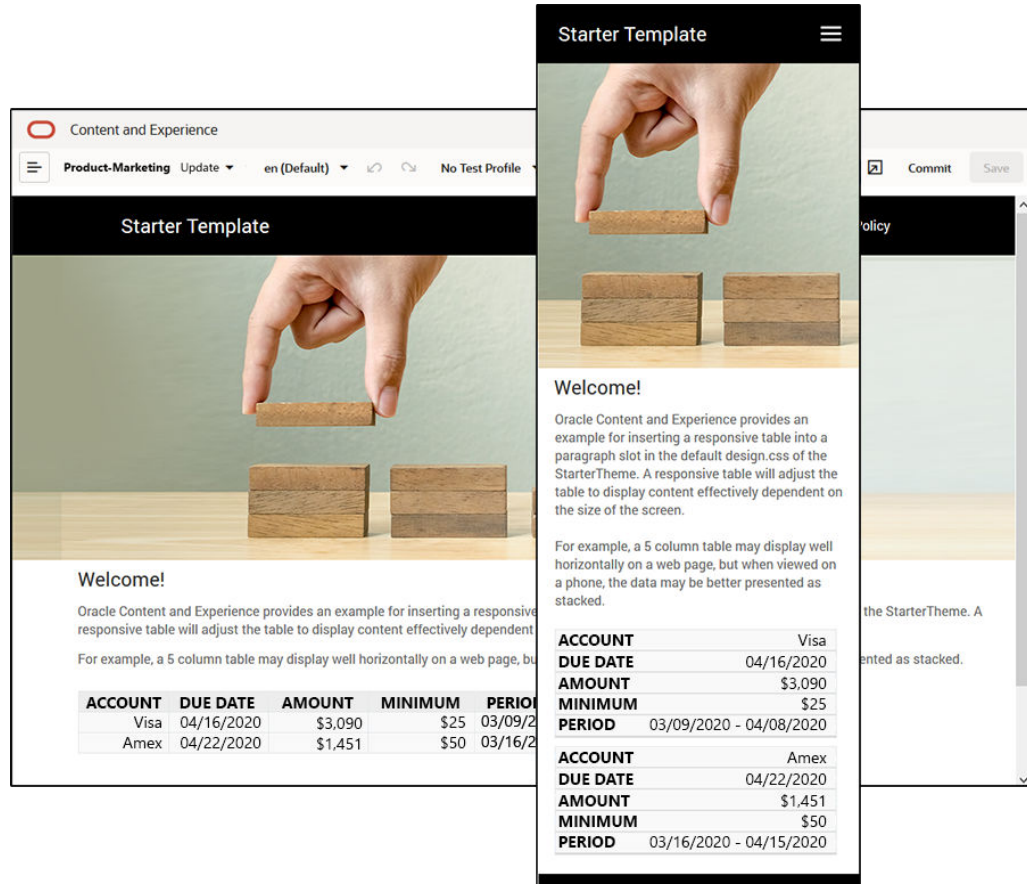
```
.scs-button-default-style .scs-button-button:hover {  
  background: #f7f8f9;  
  border: 1px solid #c4ced7;  
  color: #0572ce;  
  box-shadow: inset 0 1px 0 #f7f8f9;  
  text-shadow: 0 1px 0 #f7f8f9; }
```

```
.scs-button-default-style .scs-button-button:active {  
  background: #0572ce;  
  border: 1px solid #0572ce;  
  color: #fff;  
  box-shadow: inset 0 1px 0 #0572ce;  
  text-shadow: 0 1px 0 #0572ce; }
```

自适应表设计

Oracle Content Management 为段落组件内的自适应表提供了一个示例 CSS，该 CSS 允许堆叠移动设备上显示的行数据。

自适应表将根据屏幕大小调整表以有效地显示内容。例如，一个 5 列的表在 Web 页上可以很好地沿水平方向显示，但是当在手机上查看时，以堆叠方式呈现数据可能会更好。请注意，自适应表需要有一个标题行才能正确工作。



请注意，在所生成的如下 HTML 中，为其中的值与列标题文本相匹配的每个表单元格添加了一个 data-label 属性。

```
<thead>
  <tr>
    <th scope="col">ACCOUNT</th>
    <th scope="col">DUE DATE</th>
    <th scope="col">AMOUNT</th>
    <th scope="col">MINIMUM</th>
    <th scope="col">PERIOD</th>
  </tr>
</thead>

<tbody>
  <tr>
    <td data-label="ACCOUNT">Visa</td>
    <td data-label="DUE DATE">04/16/2020</td>
```

```
<td data-label="AMOUNT">$3,090</td>
<td data-label="PERIOD">03/09/2020 - 04/08/2020</td>
</tr>
```

一旦每个单元格中都有该属性，TD 就会在您应用下面的 CSS 规则时相互堆叠。


```
.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;
}
```

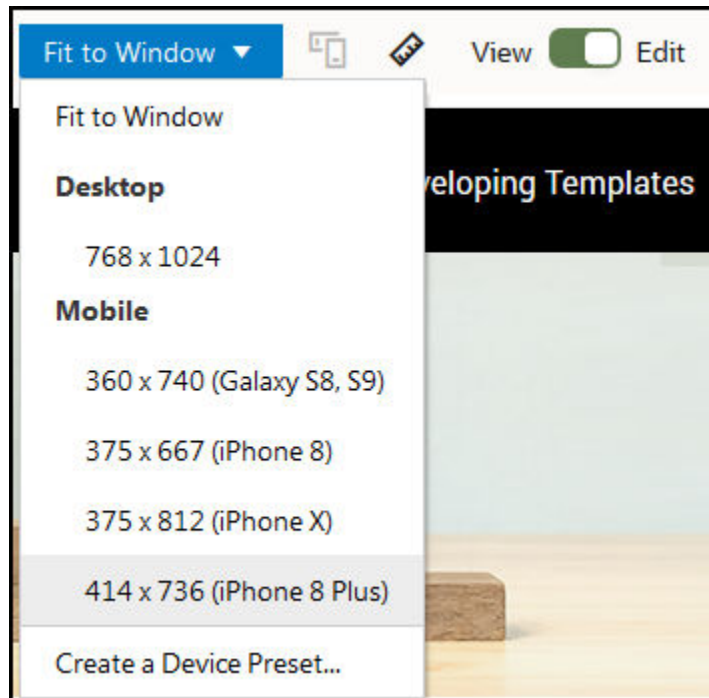
请注意，下面的媒体规则会查询屏幕大小，而且仅在屏幕大小小于 767 像素时才生效。

```
@media screen and (max-width: 767px) {
```

有关使用 css 允许所插入的表进行响应的代码示例位于所提供的 StarterTheme 的默认 design.css 中。如果从 StarterTheme 构建站点，则插入到段落内容框中的表将在默认情况下进行响应。要插入表，请执行以下操作：

1. 在 Oracle Content Management 中，打开一个站点并切换到编辑模式。
2. 创建新更新或者选择要修改的现有更新。
3. 将新的段落组件拖到页面上并单击要在其中插入表的位置，或者单击现有段落中要在其中插入表的位置。
4. 在多信息文本工具栏上，单击  并设置表属性。确保选择第一行作为标题，然后调整表宽，使其在预期要使用的最小屏幕上有效地工作。例如，如果您预期在手机上查看该站点，则默认宽度 767 像素将有可能太宽，无法在手机屏幕上完全显示，即使堆叠放置也是如此。您希望将表宽设置得小些（如 300 像素）或者将宽度设置为 100%。

完成后，切换回到查看模式并选择屏幕宽度小于 767 像素的查看选项，以预览结果。您必须在查看模式下，因为表在编辑模式无法以响应方式工作。



如果您在基于其他主题构建站点时希望使用自适应表，则需要将 StarterTheme design.css 中的代码复制到正在使用的主题的设计的 design.css 文件。

1. 要复制 StarterTheme design.css 中的代码，请单击 Oracle Content Management 侧面导航中的开发人员。
2. 单击查看所有主题。
3. 选择 **StarterTheme** 并单击打开。
4. 单击 designs 以打开该文件夹，然后单击 **default**。
5. 从 StarterTheme 中选择 design.css 文件并单击下载。
6. 在文本编辑器中打开该文件，找到该文件中以 *An example CSS of how to render a table responsively* 注释开头的部分。
7. 选择从当前注释到下一个注释之前的代码，并复制它。

```
/**
 * 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. 重复上述步骤以下载您要修改的模板的 `design.css` 文件，打开模板，然后将复制的代码粘贴到该文件中。
9. 保存所做的更改并将修改后的 `design.css` 文件作为新修订版上载到您正在修改的主题。

定制对话列表样式

可以通过在 `design.css` 文件中添加选择器来定制对话列表组件的样式。

可使用以下 CSS 选择器定制对话列表组件的样式。

选择器名称	说明
<code>.scs-convo-list-cust .scs-convo-list-container</code>	组件的最外层 DIV
<code>.scs-convo-list-cust .scs-convo-list-title</code>	列表中选择的对话的标题
<code>.scs-convo-list-cust .scs-convo-list-line-separator</code>	列表标题与列表之间的分隔符
<code>.scs-convo-list-cust .scs-convo-list-convo-title</code>	列表中的对话的标题

选择器名称	说明
<code>.scs-convo-list-cust .scs-convo-list-convo-line-separator</code>	每个对话之间的分隔符
<code>.scs-convo-list-cust .scs-convo-list-active</code>	列表中选择的对话的标题
<code>.scs-convo-list-cust .scs-convo-list-convo-posts</code>	对话的帖子数
<code>.scs-convo-list-cust .scs-convo-list-convo-unread</code>	对话的未读消息数
<code>.scs-convo-list-cust .scs-convo-list-convo-updated</code>	对话的上次更新日期
<code>.scs-convo-list-cust .scs-convo-list-no-convo-msg</code>	列表为空时的消息
<code>.scs-convo-list-cust .scs-convo-list-no-auth-msg</code>	在无用户授权的情况下在公共站点中呈现对话列表时的消息

请参见《通过 Oracle Content Management 构建站点》中的“使用样式和格式设置”。

示例

此示例说明如何使用定制 CSS 更改对话列表的字体类型、样式和颜色。



以下代码显示了用于创建该示例的定制 CSS：

```
.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;
    }
}

```

定制文件夹列表和文件列表样式

可以通过在 `design.css` 文件中添加选择器来定制文件夹列表和文件列表组件的样式。

可以使用文件夹列表组件列出 Oracle Content Management 账户的指定文件夹中的文件夹。文件夹列表可以自动与页上的文件列表组件和文档管理器通信，以显示在文件夹列表中选择文件夹内的文件。

您可以使用文件列表组件提供 Oracle Content Management 账户中指定文件夹内文件的视图。文件列表可以自动与页上的文件夹列表组件通信，以显示在文件夹列表中选择文件夹内的文件。

文件夹列表 CSS 选择器

可使用以下 CSS 选择器定制文件夹列表组件的样式。

选择器名称	说明
<code>.scs-folder-list-cust .scs-folder-list-container</code>	组件的最外层 DIV
<code>.scs-folder-list-cust .scs-folder-list-folder-title</code>	文件夹名称
<code>.scs-folder-list-cust .scs-folder-list-line-separator</code>	文件夹名称与子文件夹列表之间的分隔符
<code>.scs-folder-list-cust .scs-folder-list-sub-folder-title</code>	子文件夹名称
<code>.scs-folder-list-cust .scs-folder-list-sub-folder-title-active</code>	选择的子文件夹名称

选择器名称	说明
<code>.scs-folder-list-cust .scs-folder-list-no-folder-msg</code>	没有子文件夹可显示时的消息

文件列表 CSS 选择器

可使用以下 CSS 选择器定制文件列表组件的样式。

选择器名称	说明
<code>.scs-file-list-cust .scs-file-list-container</code>	组件的最外层 DIV
<code>.scs-file-list-cust .scs-file-list-folder-title</code>	文件夹名称
<code>.scs-file-list-cust .scs-file-list-line-separator</code>	文件夹名称与文件列表之间的分隔符
<code>.scs-file-list-cust .scs-file-list-row</code>	包含文件相关信息的行
<code>.scs-file-list-cust .scs-file-list-left-col</code>	位于组件左侧部分的文件缩略图
<code>.scs-file-list-cust .scs-file-list-mid-col</code>	组件的中间部分，包含文件的名称、说明、上次修改和大小。
<code>.scs-file-list-cust .scs-file-list-file-title</code>	位于应用程序中间部分的文件名
<code>.scs-file-list-cust .scs-file-list-file-desc</code>	位于应用程序中间部分的文件说明
<code>.scs-file-list-cust .scs-file-list-file-lastModified</code>	文件的上次修改
<code>.scs-file-list-cust .scs-file-list-file-size</code>	文件大小，与上次修改之间有个垂直分隔符
<code>.scs-file-list-cust .scs-file-list-file-size-no-sep</code>	文件大小，不带垂直分隔符（不显示上次修改）
<code>.scs-file-list-cust .scs-file-list-right-col</code>	应用程序的右侧部分
<code>.scs-file-list-cust .scs-file-list-file-download-icon</code>	位于应用程序右侧部分的下载图标
<code>.scs-file-list-cust .scs-file-list-no-file-msg</code>	没有文件可显示时的消息

请参见《通过 *Oracle Content Management* 构建站点》中的“文件列表”和“文件夹列表”。

定制社交栏图标

您可以创建要在主题的默认站点的社交栏中使用的定制社交图标。

显示在站点的社交栏中的社交图标由站点的主题设计确定。如果更改站点的主题，则社交图标会随主题一起更改。Oracle Content Management 主题附带了适用于 Facebook、Twitter、LinkedIn、Google+ 和 YouTube 的常用社交图标。

您可以通过编辑 `design.json` 和 `design.css` 文件向社交图标组件添加定制社交图标。

design.json 文件

在 design.json 文件中，您可以使用名称和类结构指定新图标，如以下示例代码所示：

```
"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 文件

在 design.css 文件中，您可以使用名称和 url 指定内容添加新图标，如以下示例代码所示：

```
.scs-facebook-icon {
  background-image: url("facebook.png"); }
.scs-twitter-icon {
  background-image: url("twitter.png"); }
.scs-linkedin-icon {
  background-image: url("linkedin.png"); }
```

为 Oracle Intelligent Advisor 配置交互式提问样式扩展

可以设置 Oracle Intelligent Advisor (OIA)（以前称为 Oracle Policy Automation）交互式提问的样式，使其外观与您公司站点的外观统一。

可以在 design.css 文件中配置以下 CSS 类选择器。所有这些类选择器都有前缀 "scs-opainterview-"。

每个类定义指定组件的所有样式设置。类具有完全控制，且不使用现有 OIA 样式。

选择器	应用到	说明
interview	interview	交互式提问区域包含整个交互式提问内容，包括页眉、页脚和导航区域。
interviewContent	交互式提问内容	交互式提问内容区域包含屏幕标题和控件，但不包含页眉、页脚和导航区域。
screenTitleBlock	屏幕标题块	屏幕标题块包含屏幕标题所在区域以及相应行中的任何其他窗口部件，例如屏幕下拉列表和/或“下一个”按钮和“返回”按钮。
screenTitle	屏幕标题	屏幕标题区域就是包含屏幕标题的区域。
nextButton	“下一个”按钮	“下一个”按钮。
backButton	“返回”按钮	“返回”按钮。
restartButton	“重新启动”按钮	“重新启动”按钮。
exitButton	“退出”按钮	“退出”按钮。
header	header	页眉区域。
footer	footer	页脚区域
question	问题文本	问题文本的样式设置。
control	控件的容器	包含控件的元素的样式设置。
label	标签控件	标签控件的样式设置。
controlError	所有控件	错误文本容器的样式设置。
controlErrorText	所有控件	错误文本 span 的样式设置。
textInput	单行文本框、密码和掩码内容	文本输入控件的样式设置。
textAreaInput	多行文本框	多行文本输入控件的样式设置。
calendarInput	日历	日历输入控件的样式设置。支持 iconColor 字段（允许更改日历图标的颜色）和 keepIcon 字段（指示是否应显示日历图标）。
dropDownInput	下拉列表	下拉列表输入控件的样式设置。
filterDropDownInput	筛选的下拉列表	筛选的下拉列表输入控件的样式设置。支持 iconColor 字段（允许更改下拉箭头的颜色）。
listInput	固定列表	固定列表输入控件的样式设置。
radioInput	单选按钮	单选按钮输入控件的样式设置。可以使用 borderColor 和 fillColor 选项为其设置样式。iconType 属性可以更改所用图标的类型。当前，仅支持 'tick' 和 'fill' 这两个替代选项。
checkboxInput	复选框	复选框输入控件的样式设置。可以使用 borderColor 和 fillColor 选项为其设置样式。iconType 属性可以更改所用图标的类型。当前，仅支持 'square' 和 'fill' 这两个替代选项。
autoCompleteInput	定制搜索	使用 customSearch 扩展时，autocomplete 字段的样式设置。
captchaInput	CAPTCHA 输入字段	用户输入 CAPTCHA 的输入字段的样式设置。
signatureInput	签名控件	签名控件的样式设置。支持额外的 inkColor 字段（允许更改签名的笔墨）。
explanationHeader	解释控件	解释控件的顶层可展开标题的样式设置。

选择器	应用到	说明
explanationText	解释控件	展开的解释控件文本的样式设置。
signatureClearButton	签名控件	签名控件上的清除按钮的样式设置。
uploadAddButton	上载控件	上载控件上的添加按钮的样式设置。
entityRemoveButton	实体收集控件	实体收集控件。

采用 `design.css` 中定义的样式扩展的示例

```
.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;
}
```

了解背景用法

您可以为站点页以及页上的各个内容框指定背景色和图像。页、内容框和组件层的背景依次叠加。例如，如果为内容框指定背景，它将分层放在为页指定的背景上面。

- [关于背景和主题](#)
- [如何实施背景](#)
- [设置存储位置](#)

关于背景和主题

背景功能主要用于站点页和内容框。为页和内容框设置的背景得由用户（而不是开发人员）在主题级别进行配置。

可以在主题内容框中配置背景效果，但这些效果可能会覆盖编辑站点时为页设置的任何背景效果。

背景功能不会变更主题或组成主题文件。页的背景设置无法变更主题，并且站点中基于同一布局的其他页不会继承源页的任何背景设置。通过主题添加或更改背景需要更新相应主题。

用户可以在站点的每个页上配置不同背景，也可以在使用同一布局的不同页上配置不同背景。这无法在主题中实现。尽管可以使用主题为页指定背景，但用户的背景设置可能会覆盖此指定。（背景设置仅对特定页应用覆盖；它不以任何方式修改主题本身。）

应注意避免出现主题覆盖用户在站点页和内容框中设置的背景样式。在下面几种情况下可能会发生这种情况：

- 在 `<body>` 和内容框上使用基于元素的 "style" 属性时，主题可能会覆盖站点中的页背景设置。
- 在主题级联样式表 (cascading style sheet, CSS) 文件中将主题背景样式标记为 " !important" 时，主题可能会覆盖站点中的页背景设置。

请参见《通过 Oracle Content Management 构建站点》中的“更改背景或主题”。

如何实施背景

为实施页和内容框背景功能，Oracle Content Management 在页的 `<head>` 中动态创建 CSS 样式表。

对于页背景设置，样式的选择器是基于标记的选择器 (body)。对于内容框背景设置和基于 ID 的选择器，将使用内容框 ID。

例如，设置页的背景颜色可能会在页的 `<head>` 中生成以下 CSS 标记：

```
body
{
    background-color: #fa7c9d;
}
```

类似地，在内容框上设置背景图像可能会在页的 <head> 中生成以下 CSS 标记：

```
#PageFooter
{
    background-image: url("footer_image.png");
}
```

此实施意味着直接在主体标记或内容框元素的 "style" 属性中指定的样式可覆盖在 <head> 代码中的样式表中配置的设置。

❗ 重要：

主题开发人员应注意不要用基于元素的样式覆盖背景设置。

请参见《通过 Oracle Content Management 构建站点》中的“更改背景或主题”。

设置存储位置

呈现页时，背景设置将动态写入页的 <head> 代码中的 "style" 标记。

背景设置将持久保留在页模型文件中（例如，<pageid>.json）。特别是，页背景设置存储在 properties.styles 部分中，内容框背景设置存储在 slots[<slot_id>].styles 部分中。

背景设置存储在页 JSON 文件中，特别是在 "styles" 内，如以下典型示例中所示。

```
{
  "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>\",
    \"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

设置触发器和操作

可以配置组件（包括内嵌框架中呈现的组件）之间的通信，以便组件中的触发器对其他组件调用操作。

- [关于触发器和操作](#)
- [设置触发器](#)
- [设置操作](#)

关于触发器和操作

可以配置组件（包括内嵌框架中呈现的组件）之间的通信，以便组件中的触发器对其他组件调用操作。

触发器是 Oracle Content Management 组件间通信的一部分。任何组件都可以引发任意数量的触发器。组件可以为触发器提供有效负载，有效负载将传递给引发触发器时执行的任何操作。您可以选择将为每个触发器执行的操作。构建为一起工作的组件可以自动引发触发器，以对其他组件执行操作，而无需用户交互。

基本流程包括：

1. 注册触发器
2. 引发触发器
3. 注册操作
4. 执行操作以验证设置

例如，您可以使用按钮组件来执行一个或多个操作，例如，显示或隐藏页组件以及显示消息。一个组件中可以有一个业务办公地点列表，单击该列表中的某个地点时，将在另一个组件中显示有关该地点的详细信息。

对于定制的组件，触发器和操作是组件注册数据的一部分，而不是组件实施的一部分。在注册数据中，存在包含组件支持的触发器和操作列表的 "triggers": [], 和 "actions": [], 条目。实际语法与本地组件和远程组件的语法相同，只是位置和检索方式不同。

请参见《通过 Oracle Content Management 构建站点》中的“使用触发器和操作”。

设置触发器

组件可以包括将执行其他组件中的操作的触发器。您必须注册要由组件引发的触发器。

组件可以为触发器提供有效负载，有效负载将传递给引发触发器时执行的任何操作。您可以选择将为每个触发器执行的操作。构建为一起工作的组件可以自动引发触发器来执行操作，而无需用户交互。

注册触发器

对于定制组件，将作为该组件的注册数据的一部分来注册触发器。要添加触发器，请使用组件支持的每个触发器更新 "triggers" 属性数组。还必须指定触发器支持的有效负载，以便可以创建用户界面来允许用户将有效负载内的值映射到操作支持的属性。

1. 编辑 `appinfo.json` 文件并检查 "triggers": [], 条目：

```
"triggers": [{
  "triggerName": "helloWorldWhoAreYou",
  "triggerDescription": "Show Who I Am",
  "triggerPayload": [{
    "name": "whoAreYou",
    "displayName": "Who I Am"
  }]
}],
```

2. 将文件同步到站点服务器。

在此示例触发器条目中，您定义了 `triggerName` ("helloWorldWhoAreYou")。名称值必须唯一。然后，您为触发器提供了说明 ("Show Who I Am")，用户界面对话框使用它来显示触发器。最后，您为触发器定义了单一值有效负载；用户将能够选择此有效负载中的条目并将它们映射到操作中的字段。

注册触发器后，当您转至组件的“设置”面板中的链接选项卡时，应该能够看到并选择该触发器。

引发触发器

组件可以在任何点引发触发器。通常，触发器由用户交互引发，例如单击按钮或选择表中的行。组件可以基于任何标准引发触发器，例如，当数据由于 REST API 调用而更改时。引发触发器时，您可以执行任意数量的操作。

下面是有关如何引发触发器的示例：

1. 编辑 `render.js` 文件，并在将调用 Sites SDK 以引发触发器的 `viewModel` 对象中添加 JavaScript 函数。

```
self.raiseTrigger = function (triggerName) {
  SitesSDK.publish(SitesSDK.MESSAGE_TYPES.TRIGGER_ACTIONS, {
    'triggerName': 'helloWorldWhoAreYou',
    'triggerPayload': { "whoAreYou": "This is " + self.whoAreYou()
+ "!" }
  });
};
```

2. 在用户界面中添加条目来调用函数以引发触发器 (`-edit template.html`) 并在 `</div>` 前添加按钮。

```
<button data-bind="click raiseTrigger">Who Am I?</button>
```

3. 将 `render.js` 文件同步或上载到 Oracle Content Management 实例服务器。

在 `viewModel` 对象中，您创建了 JavaScript 函数，单击按钮时将调用该函数。此函数调用 Sites SDK 以告知它触发为此触发器 "helloWorldWhoAreYou" 定义的所有操作。

它还传递具有单个字段 "whoAreYou" 的 triggerPayload。"helloWorldWhoAreYou" 和 "whoAreYou" 这些值与您在前面步骤中注册触发器时输入的那些值匹配。

 **注：**

执行操作时没有预定义的顺序。虽然每个操作将按其列出的顺序进行调用，但是在调用下一个操作之前不需要等待该操作完成。如果操作进行异步调用，则执行下一个操作之前该操作可能未完成。

设置操作

您可以设置组件来利用操作注册，以便可以将其放在页上，用于在您的组件中执行操作。

注册操作

引发了触发器时会对组件调用操作。组件可以注册任意数量的操作，并且还可以定义操作支持的有效负载。用户选择操作时，他们可以填充要传递到相应操作的有效负载。

与注册触发器一样，可以在主题的 `appinfo.json` 注册数据中注册组件支持的操作。

下面是有关如何注册操作的示例：

1. 编辑组件的 `appinfo.json` 文件并更新 `"actions": []`，条目。

```
"actions": [{
  "actionName": "helloWorldChangeWhoIAm",
  "actionDescription": "Change Who I Am",
  "actionPayload": [{
    "name": "whoAreYou",
    "description": "Who are you?",
    "type": {
      "ojComponent": {
        "component": "ojInputText"
      }
    },
    "value": ""
  }]
}]
```

2. 注册后，操作将在操作对话框中可见，当您在组件的“设置”面板上的链接选项卡上单击触发器时，将调用该操作。

执行操作

注册了操作后，您便能够在组件中执行操作的组件放在页上。要使组件执行操作，它必须监听 `EXECUTE_ACTION` 消息。此消息还包括传递到操作的有效负载，您必须从其提取预期值。

例如，要监听 `EXECUTE_ACTION` 消息，请编辑 `render.js` 文件并使用以下条目更新 `ViewModel` 对象：

```
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);
        }
    };
};
```

这段代码创建一个执行操作的 JavaScript 函数，然后在发出 EXECUTE_ACTION 消息时使用 Sites SDK 调用该函数。

每当发出 EXECUTE_ACTION 消息时都会调用该操作，并且由组件决定仅处理设计为由其处理的操作。为此，您必须检查操作的名称以确保它是您可以处理的操作。

操作的有效负载是值数组。在本示例中，假定值是数组中的第一个条目。通常，您必须从数组中查找您关注的有效负载值。

 **注：**

由于操作监听程序是回调，因此您应使用 JavaScript 闭包或适当地绑定函数，以确保在执行函数时，您有权访问 ViewModel。

开发模板

模板包包含站点的开发版本、具有页布局的主题、样式和导航以及站点中使用的关联组件。Oracle Content Management 提供一系列模板，可用于创建站点以及用作创建定制模板起点。

- [关于模板](#)
- [基本模板结构](#)
- [创建模板](#)
- [导出模板](#)
- [导入模板](#)
- [使用入门级模板](#)
- [基于 Bootstrap 或网站设计模板创建站点模板](#)
- [通过开发人员云服务开发模板](#)

关于模板

模板包含用户开始创建网站时所需的所有部分，包括带有示例页和内容的站点，带有样式、导航、资产（例如图像）的主题，以及关联的组件。

Oracle Content Management 提供了各种用于创建站点的模板。这些模板通常由管理员在初始化服务时安装。请参见管理 *Oracle Content Management* 中的“配置站点设置”。

无论何时创建新站点，都必须选择模板。模板将主题与站点和组件相结合来推动功能或解决方案，例如合作伙伴门户或营销活动。

有关现成模板的列表，请参见[了解模板](#)。

虽然开发人员和用户可以创建新模板以及修改和替换现有模板，但是开发人员的一个主要任务是设计新模板。此过程基本包括以下步骤。

1. 通过复制现有模板（例如 JET 入门级模板）来创建新模板。此操作还将获得与该模板关联的主题。

例如：

```
cec create-template My_JET_Template -f JETStarterTemplate
```

2. 以 .zip 文件形式将模板导出到开发环境。
3. 打开模板程序包中的文件并进行更改。
4. 以 .zip 文件形式创建修订的模板程序包。
5. 使用 Oracle Content Management 界面将新模板导入到您的实例。
6. 共享模板，以便其他人可以使用它。

还可以使用 Oracle Content Management 界面通过添加和修改页布局和资产来修改模板，扩展站点结构，将组件添加到站点页，以及添加要显示在使用此模板中主题的任何站点中的种子内容。

另请参见[管理模板](#)。

基本模板结构

模板的基本结构包括站点（具有资产、布局、页和内容）、关联的主题以及任何定制组件。

创建站点时，必须选择模板以提供站点结构和初始内容、包含设计和布局规范的主题以及任何定制组件。

模板在特定结构中进行组织，如以下示例中所示，其中显示了基本文件夹和文件。

```
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
```

组件和主题文件夹及文件在本指南的其他位置进行介绍。请参见[关于开发组件和基本主题结构](#)。

注释:

- 通常，主题在模板之间进行共享，除非您使用 JET 入门级模板或入门级模板，其使用主题的副本。请参见[使用入门级模板](#)。
- 主题不再包含站点。站点文件夹和文件位于 `/template` 文件夹中。

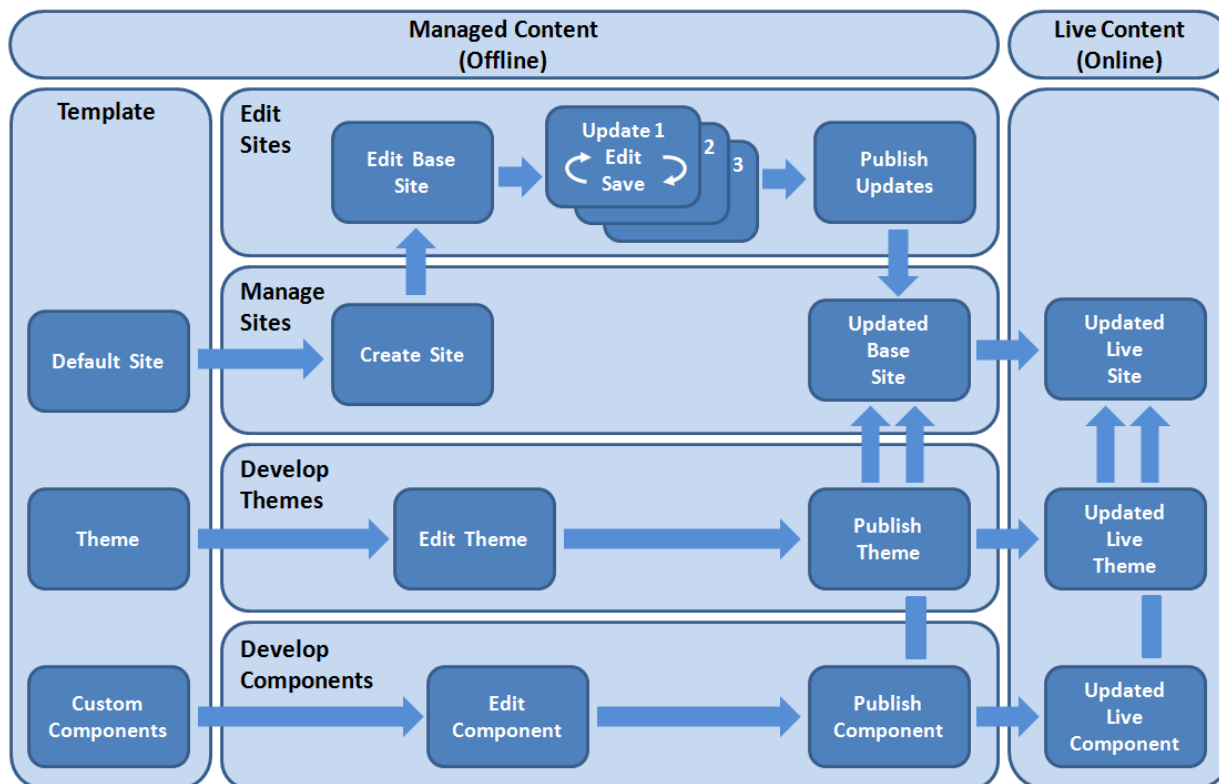
`template_name/template` 文件夹包含站点的文件夹和文件。

- `assets`: 包含用户界面中的模板详细信息页中显示的图像。
- `content`: 包含站点中使用的托管内容。
- `layouts`: 此时不使用。
- `pages`: 包含所有页 JSON 文件及数据。使用格式 `nnn.json`，其中 `nnn` 是页 ID。
- `variants`: 包含站点的所有更新的详细信息。
- `_folder.json`: 包含模板的元数据，例如站点作者、站点名称、项 GUID、简要和详细站点说明。
- `_folder_icon.png`: 表示用户界面中的站点。
- `componentsused.json`: (已废弃。) 记录站点内使用的定制组件 (如果有)。维护此项仅用于向后兼容。
- `controller.html`: 包含在浏览器中显示站点的键代码。如果要对此文件进行更改，Oracle 建议通过界面中的站点设置执行此操作。可以脱机修改该文件。请参见[定制控制器文件](#)。
- `siteinfo.json`: 标识站点名称和关联主题的名称，以及站点的其他元数据。不要修改此文件。
- `structure.json`: 为页 (父和子页) 定义站点的层次。为站点设置导航时，可以使用呈现 API 来提取树结构。请参见[站点导航](#)和[呈现 API 参考](#)。

创建模板

如果您希望使用某个站点作为其他站点的起点，您可以从该站点创建模板。此外，您还可以复制现有模板并对副本进行更改来创建新模板。

在您从现有站点创建模板时，新模板使用该站点的副本作为默认站点。模板引用站点使用的主题以及站点页中使用的任意定制组件。主题和定制组件不复制到模板，而是通过与站点相同的方式来引用。模板反映了在创建该模板时所使用的站点。以后对创建模板时所用站点进行的更改不会体现在存储于模板中的站点上。



如果通过复制现有模板并重命名副本来创建新模板，可对副本进行更改。请注意，复制模板时，不会复制该模板的共享信息。

不要将以下名称用于模板、主题、组件、站点或站点页：authsite、content、pages、scstemplate_*、_comps、_components、_compsdelivery、_idcservice、_sitescloud、_sitesclouddelivery、_themes、_themesdelivery。下面的名称虽然可用于站点页，但不要将其用于模板、主题、组件或站点：documents、sites。

如果您要创建自己的定制模板（包含站点和主题），最好使用 Oracle Content Management 提供的 JET 入门级模板或入门级模板，该模板包含站点和关联主题的基本元素。入门级模板包含写入站点页的信息和说明，以帮助您在定制模板中规划布局以及设计站点和主题。

请参见[管理模板](#)和[使用入门级模板](#)。

导出模板

您可以导出模板以脱机修改，然后将其作为新模板导入或者替换现有模板。也可以导出模板，以将其移动到其他 Oracle Content Management 实例并在该处导入。


导出模板时，您实际上是将模板作为单个 .zip 文件复制到 Oracle Content Management 中的文件夹内。您可以直接从文件夹下载模板程序包，将其解压缩并处理

单个文件。完成模板文件的处理之后，创建包含模板程序包的 .zip 文件，将其导入到 Oracle Content Management 以覆盖原始模板或者创建新模板。

 **注：**

导出模板时，不包括模板的共享信息。

要导出模板，请执行以下操作：

1. 在 Oracle Content Management 侧面导航中，单击开发人员。
此时将显示开发人员页。
2. 单击查看所有模板。
此时将显示现有模板的列表。
3. 选择模板，然后在右键单击菜单中选择导出，或单击操作栏中的 。
4. 导航到文件夹，或者单击创建，提供名称和可选的说明，然后单击创建来创建新文件夹。
要打开文件夹，请单击文件夹图标或文件夹名称。
5. 单击关联文件夹的复选框，然后单击确定来选择文件夹。
使用模板名称和 .zip 扩展名，在所选文件夹中创建模板程序包文件。

导入模板

您可以导出模板以脱机修改，然后将其作为新模板导入或者替换现有模板。也可以导出模板，以将其移动到其它 Oracle Content Management 实例并在该处导入。

导出模板时，您实际上是将模板作为单个 .zip 文件复制到 Oracle Content Management 中的文件夹内。您可以直接从文件夹下载模板程序包，将其解压缩并处理单个文件。完成模板文件的处理之后，创建包含模板程序包的 .zip 文件，将其导入到 Oracle Content Management 并覆盖原始模板或者创建新模板。

要导入模板程序包，请执行以下操作：

1. 在 Oracle Content Management 侧面导航中，单击开发人员
此时将显示开发人员页。
2. 单击查看所有模板。
此时将显示现有模板的列表。
3. 单击创建并选择导入模板程序包。
4. 上传模板程序包之后，导航到包含模板程序包的文件夹。要打开文件夹，请单击文件夹图标或文件夹名称。

如果您尚未上传模板程序包，请执行以下操作：

- a. 导航到您要将模板程序包上传到的文件夹，或者单击新建，提供名称和可选的说明，然后单击创建以创建文件夹。
- b. 单击上载。
- c. 找到并选择模板程序包，然后单击打开。

进度栏显示文件名和上载状态。

5. 单击文件名旁边的复选框，然后单击确定来选择模板程序包。

如果导入模板的内容与任何现有模板、主题或定制组件没有冲突，系统将创建新 Oracle Content Management 文件夹，其关联的主题以及任意定制组件。

6. 如果模板、主题或定制组件的名称或 ID 已经存在，系统将提示您解决冲突。

根据冲突的性质，系统提供相应的选项，您可以创建新模板、主题或定制组件；或者在某些情况下，您可以使用导入的版本覆盖现有模板、主题或定制组件。

使用入门级模板

可使用 Oracle Content Management 提供的入门级模板副本创建包含主题和定制组件的新站点。

入门级模板收集了在一个程序包中构建定制网站所需的所有内容：默认站点、布局、导航、示例内容、主题、关联的内容项等。与随 Oracle Content Management 提供的其他模板不同，入门级模板为您提供了易于使用的基本框架，可用于创建新站点（包括新主题）。



注：

模板之间通常会共享主题，但入门级模板使用主题副本。

如果管理员在安装和配置服务时启用模板，则 Oracle Content Management 中会提供入门级模板 StarterTemplate 和 JETStarterTemplate 以及其他模板。Oracle JavaScript 扩展工具包 (JavaScript Extension Toolkit, JET) 入门级模板包含了模板的最新 JET 样式设置，加入了一些页内容（作为 JET 组件），并提供了用于构建基于 JET 的模板和站点的入门级组件。

基本流程

下面是使用入门级模板的基本流程：

1. 创建新站点并选择入门级模板。在创建新站点时会创建新的入门级主题。



注：

应仔细选择站点名称。您为站点提供的名称会复制为新主题的名称，发布站点后，用户将可以看到主题。主题的名称在创建后无法更改。

入门级站点和主题包含开始开发站点和主题所需的一组文件夹和文件。

2. 将新主题同步到桌面。您可以在桌面上工作来使用布局和静态资产扩展和定制主题。由于这是主题副本，并且它使用特定于模板的名称，您可以对该主题进行更改，而不会影响源主题。
3. 在站点构建器中打开站点。您可以使用站点构建器向站点结构中添加内容、将组件和交互添加到页以及添加将成为新模板的默认站点一部分的内容。您可以重用或删除入门级模板中提供的站点内容。

4. 站点和主题准备就绪后，将更改与 Oracle Content Management 实例同步，然后从选择的站点创建新模板。
5. 共享模板，以便其他人可以使用它。

使用入门级模板创建模板

要使用入门级模板，请执行以下操作：

1. 在 Oracle Content Management 侧面导航中，单击站点。
此时将显示现有站点的列表。
2. 单击创建。
3. 在“创建站点”对话框中，选择要用作站点基础的入门级模板。
新站点将使用随入门级模板提供的主题，该主题将重命名以匹配新站点名称。
4. 在对话框中，输入站点的名称。站点 URL 中将使用此名称。可以使用字母、数字、下划线 (_) 和连字符 (-)。如果输入空格，将使用下划线自动取代空格。

不要将以下名称用于模板、主题、组件、站点或站点页：authsite、content、pages、scstemplate_*、_comps、_components、_compsdelivery、_idcservice、_sitescloud、_sitesclouddelivery、_themes、_themesdelivery。下面的名称虽然可用于站点页，但不要将其用于模板、主题、组件或站点：documents、sites。

注：

Oracle Content Management 站点 URL 的路径区分大小写。查询或片段字符串中的大小写由开发人员在其定制代码中管理。

```
https://[host[:port]][/]path[?query][#fragment]
      \_____/      \_____/
      Location      Data
```

5. (可选) 输入站点的说明。
6. 准备好之后，单击创建。

进度栏显示新站点名称和创建状态。在创建站点时，名称将显示在站点列表中。其初始状态为脱机。

要在列表中快速查找您新建的站点，请按照上次更新时间对列表排序。您刚刚创建的站点将显示在列表顶部。

注：

对于您创建的站点，系统会自动为您分配管理者角色。

7. 使用桌面应用程序将主题同步到桌面。
现在，您应该可以看到主题的文件层次和文件。

8. 入门级主题包含一组最低限度的文件夹和文件，例如：


```
theme
  assets
    css
      main.css
    js
      topnav.js
  designs
    default
      design.css
      design.json
      facebook.png
      googleplus.png
      linkedin.png
      twitter.png
      youtube.png
      x-close.png
  layouts
    index.html
    resonancepages
      404.html
    _folder.json
    _folder_icon.jpg
    components.json
    viewports.json
```

x-close.png 文件包含 Cookie 许可弹出窗口的默认关闭图标。

/layouts 文件夹包含入门级页布局文件 (index.html)，其中包含以下内容：

- 一组 HTML 标记，用于允许该文件用作页布局。
- 单个内容框，包含植入文本和说明，例如，如何将主题同步到桌面、如何添加新页布局、如何将组件添加到页布局以及如何使用新页布局构建站点层次。
- 一个简单的 JavaScript 导航文件，提供了有关如何使用呈现器 API JavaScript 函数和对象的示例。遍历站点层次并生成所需的 HTML 标记以允许在站点中导航时需要呈现的 API。

另请参见[基本主题结构](#)。

9. 完成主题修改后，将主题文件夹和文件同步到 Oracle Content Management 实例。要查看对主题所做更改在站点中的外观和行为，请在站点构建器中打开该站点。这可能是个迭代过程。
10. 要查看或修改站点，请选择站点，然后在右键单击菜单中单击打开，或单击操作栏中的 。
11. 将编辑器模式切换为编辑，以便可以对站点进行更改。您可以修改现有页，并使用入门级主题中提供的页布局添加新页。

 注：

如果您熟悉页布局结构和用法，您可以从入门级模板中提供的部分中删除不需要的部分，并将需要的部分上的布局切换到新布局之一。

12. 完成对站点的更改后，保存站点，然后单击发布以将更新合并到基本站点。
13. 选择站点并基于该站点创建模板。这将为新模板拉入资产和主题。
14. 与您希望其能够使用该模板的成员共享该模板。

基于 Bootstrap 或网站设计模板创建站点模板

Oracle Content Management 的开放式体系结构意味着您可以使用在其他编码框架（例如 Foundation 或 Bootstrap）中完成的工作内容。只需少量更改，就可以将 Bootstrap 模板转换为主题，并使其成为 Oracle Content Management 模板的一部分。

基本流程

下面概括列出了各步骤，后面各部分详细介绍了这些步骤：

1. [先决条件](#)
2. [创建站点](#)
3. [同步主题文件夹](#)
4. [设置基本主题](#)
5. [更新站点页](#)
6. [更新导航](#)
7. [更新站点布局](#)
8. [发布站点](#)
9. [创建新模板](#)

先决条件

- 设置并运行 Oracle Content Management 桌面应用程序，以便将文件夹和文件同步到本地计算机。
- 将 Bootstrap 模板主题文件夹、文件和内容下载到本地计算机，并已将其准备就绪，可供使用。

创建站点

基于 Oracle Content Management 入门级模板创建站点：

1. 在侧面导航中，单击站点。
此时将显示现有站点的列表。
2. 单击创建。
3. 在“创建站点”对话框中，选择要用作站点基础的 **JET** 入门级模板或入门级模板。
4. 在对话框中，输入站点的名称。站点 URL 中将使用此名称。可以使用字母、数字、下划线 (_) 和连字符 (-)。如果输入空格，则使用连字符自动取代空格。

不要将以下名称用于模板、主题、组件、站点或站点页：authsite、content、pages、scstemplate_*、_comps、_components、_compsdelivery、_idcservice、_sitescloud、_sitesclouddelivery、_themes、_themesdelivery。下面的名称虽然可用于站点页，但不要将其用于模板、主题、组件或站点：documents、sites。

 **注：**

Oracle Content Management 站点 URL 的路径区分大小写。查询或片段字符串中的大小写由开发人员在其定制代码中管理。

```

https://[host[:port]][/]path[?query][#fragment]
      \_____/      \_____/
      Location      Data
    
```

5. (可选) 输入站点的说明。
6. 准备好之后，单击创建。

进度栏显示新站点名称和创建状态。在创建站点时，名称将显示在站点列表中。其初始状态为脱机。

要在列表中快速查找您新建的站点，请按照上次更新时间对列表排序。您刚刚创建的站点将显示在列表顶部。

同步主题文件夹

当您基于入门级模板创建站点时，系统会创建入门级模板主题的副本并使用站点名称及后接的主题名称为其命名。例如，My_New_Site 的主题为 My_New_SiteTheme。

可使用桌面应用程序将站点的主题文件夹和文件同步到本地计算机。请参见《通过 Oracle Content Management 协作处理文档》中的“同步入门”。

现在，您应该可以在您的本地桌面上看到模板主题文件夹层次和文件。下面是一个示例：

```

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
```

设置基本主题

1. 将这些 Bootstrap 文件复制到您本地桌面上的已同步主题文件夹中以覆盖现有文件。
 - html 文件放在 `theme_name/layouts` 文件夹中
 - css 文件放在 `theme_name/assets/css` 文件夹中
 - js 文件放在 `theme_name/assets/js` 文件夹中
 - image 文件放在 `theme_name/assets/images` 文件夹中，可以将这些文件分组在包含用于背景、页脚和人员等的图像的子文件夹中
2. 修改 layout 文件夹中的 html 文件以更新相对路径并添加所需元素。对于典型的 Bootstrap 主题，`/assets` 文件夹可能存在很多相对路径，因此，您必须修改它们以指向主题文件夹。

修复 css、js 和 images 文件夹的路径以使用：

```
_scs_theme_root_/assets/css/  
_scs_theme_root_/assets/js/  
_scs_theme_root_/assets/images/
```

注：

完成此步骤后，`_scs_theme_root` 部分将自动调整为正在使用主题的环境。

3. 每个 Oracle Content Management 布局都有三个要求：
 - a. 在 html 文件的 `<head>` 标记中包含以下标记：
 - `<!--$SCS_RENDER_INFO-->`
 - `<!--$SCS_SITE_HEADER-->`
 - `<!--$SCS_PAGE_HEADER-->`
 - b. 在布局文件的末尾包含呈现器脚本（在 `<body>` 标记内部）。在站点构建器和运行时环境中，以下两个路径都会自动调整。

```
<script data-main="/_sitescloud/renderer/renderer.js" src="/  
_sitescloud/renderer/require.js"></script>
```

- c. 在 `renderer.js` 文件的 `include` 后面包含以下标记：
 - `<!--$SCS_PAGE_FOOTER-->`
 - `<!--$SCS_SITE_FOOTER-->`


验证修改后的主题文件是否已与 Oracle Content Management 中的站点同步。

更新站点页

在站点构建器中打开站点时，最初会看到默认站点中的页。您可以删除不需要的页，并将要保留的页上的布局切换到新布局之一。




1. 在侧面导航中，单击站点。

此时将显示现有站点的列表。

2. 选择站点，然后在右键单击菜单中选择打开，或单击操作栏中的 。
3. 输入更新的名称及可选说明，然后单击创建。

对于更新名称，您可以使用字母、数字、下划线 (_) 和连字符 (-)。如果输入空格，则使用连字符自动取代空格。

如果已具有站点的更新，请从列表中选择此更新，然后单击 。

4. 站点构建器将以预览模式打开。要进行更改或者使用子工具栏中的导航选项，请确保“编辑”开关  设置为编辑。
5. 要编辑特定页，请使用子工具栏中的站点树或使用站点自己的导航来选择该页。
6. 要删除不需要的页，请选择该页，然后单击 。
7. 要添加新页，请单击添加页。可以通过拖放页来在站点树中重新定位页。
8. 要更改与页关联的布局，请在站点树中选择该页，然后单击  以显示页设置。
转到页布局字段，然后从菜单中选择其他布局。页布局的数量和类型取决于与您的站点关联的主题。
9. 单击保存以将更改保存到当前更新。可以继续使用当前更新，也可以根据需要创建新的更新。

更新导航

在站点预览中查看导航时，它与当前层次不匹配，这是因为布局中的硬编码导航来自 Bootstrap 主题。

可更新导航以使用从站点层次动态生成的代码替换 Bootstrap 主题中的硬编码代码。

编辑主题文件的本地已同步副本。

1. 从布局中删除硬编码导航代码。下面是必须从典型 Bootstrap 主题中删除的硬编码导航示例。这是典型的 header 部分，包含 logo、‘Hamburger’ 菜单上（在页太窄时使用的）‘Toggle Navigation’ 部分（响应部分）和其他页的硬编码页导航。

```
<header id="header" class="header navbar-fixed-top">
  <div class="container">
    <h1 class="logo">
      <a href="index.html"><span class="text">Velocity</span></a>
    </h1><!--//logo-->
    <nav class="main-nav navbar-right" role="navigation">
      <div class="navbar-header">
        <button class="navbar-toggle" type="button"
          data-toggle="collapse" data-target="#navbar-collapse">
          <span class="sr-only">Toggle navigation</span>
        </button>
      </div>
    </nav>
  </div>
</header>
```

```

span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
    </button><!--//nav-toggle-->
</div><!--//navbar-header-->
<div id="navbar-collapse" class="navbar-collapse
collapse">
        <ul class="nav navbar-nav">
            <li class="active nav-item"><a
href="index.html">Home</a></li>
            <li class="nav-item"><a
href="features.html">Features</a></li>
            <li class="nav-item"><a
href="pricing.html">Pricing</a></li>
            <li class="nav-item dropdown">
                <a class="dropdown-toggle" data-
toggle="dropdown" data-hover="dropdown" data-delay="0" data-close-
others="false" href="#">Pages <i class="fa fa-angle-down"></i></a>
                <ul class="dropdown-menu">
                    <li><a href="download.html">Download
Apps</a></li>
                    <li><a href="blog.html">Blog</a></li>
                    <li><a href="blog-single.html">Blog
Single</a></li>
                    <li><a href="blog-category.html">Blog
Category</a></li>
                    <li><a href="blog-archive.html">Blog
Archive</a></li>
                    <li><a href="about.html">About Us</a></li>
                    <li><a
href="contact.html">Contact</a></li>
                </ul>
            </li><!--//dropdown-->
            <li class="nav-item"><a href="login.html">Log
in</a></li>
            <li class="nav-item nav-item-cta last"><a
class="btn btn-cta btn-cta-secondary" href="signup.html">Sign Up
Free</a></li>
        </ul><!--//nav-->
    </div><!--//navabr-collapse-->
</nav><!--//main-nav-->
</div><!--//container-->
</header><!--//header-->

```

2. 编写 JavaScript 代码以遍历站点结构信息并生成导航代码，然后在布局中包含该 JavaScript 代码，例如：

```

<script type="text/javascript" src="_scs_theme_root_/assets/js/
navbar.js"></script>

```

3. 修改 topnav.js 文件的精确输出以匹配您的特定 CSS 中所需的标记。

更新站点布局

此时，站点功能正常，但布局中没有可编辑的区域（内容框）。

1. 在布局中找到或添加 DIV 元素并将其指定为内容框。

内容框是布局中 class 属性值为 "scs-slot" 的 DIV 元素。每个内容框都必须具有唯一的 id 属性。例如：

```
<div id="slot-content1" class="scs-slot scs-responsive"></div>
```

要使内容框根据浏览器视窗的大小自动调整，请包含 class 属性 "scs-responsive"。

2. 根据需要重复此步骤以在该布局或其他布局中创建其他内容框。

发布站点

完成并保存对站点的所有更改后，必须发布该站点。

发布站点会获取当前更新中的所有更改并将其合并到基本站点中，从而使更新中的内容进入新的基本站点。

创建新模板

现在，您有了一个具有关联主题的功能正常的站点，因此可以将其做成模板以与其他人共享，以便他们可以基于此模板创建站点。

1. 在 Oracle Content Management 侧面导航中，单击开发人员。
2. 单击查看所有模板。
3. 单击创建，然后选择从现有站点。
4. 选择使用入门级模板和 Bootstrap 模板创建的新站点。
5. 输入新模板的名称，然后单击创建。
6. 要打包模板以用于 Oracle Content Management 实例，请选择模板，然后选择导出菜单选项以创建可以下载的 .zip 文件。

通过开发人员云服务开发模板

可以使用开发人员云服务为 Oracle Content Management 开发模板。

按照以下步骤在开发人员云服务中开发模板，在本地对其进行测试，然后将其导出到 Oracle Content Management 中：

1. 在本地计算机上设置 OCE 工具包。
2. 登录 Oracle Content Management 的开发人员云服务控制台。
3. 在开发人员云服务中创建项目。
4. 将 OCE 工具包添加到新的 Git 资料档案库中的项目代码。
5. 在开发人员云服务中创建模板。

您可以创建新模板以进行开发、[在开发人员云服务中复制现有模板](#)或从 Oracle Content Management [导入模板](#)。

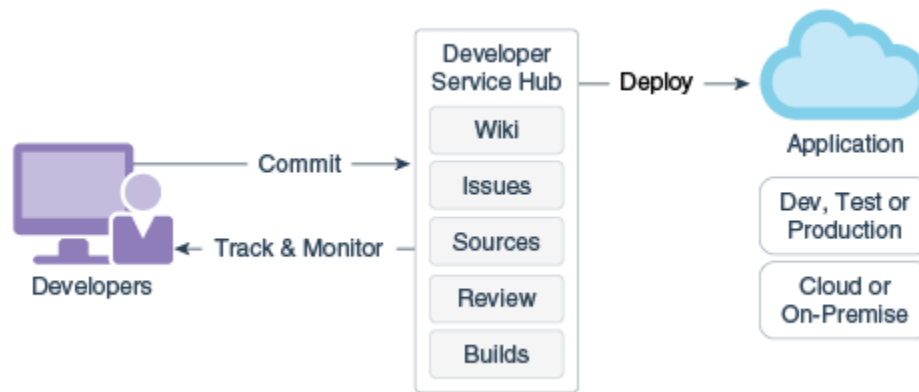
6. 在本地测试工具中测试模板。
7. 合并更改。
8. 将模板从开发人员云服务导出到 Oracle Content Management 中。

登录 Oracle Content Management 的开发人员云服务控制台

开始在开发人员云服务控制台上为 Oracle Content Management 开发定制组件。

作为 Oracle Cloud 服务的管理员，您可以使用我的服务管理来创建和管理您的云服务。如果您是 Oracle Content Management 的服务实例管理员以及标准开发人员服务的实例管理员，则可以设置它们并开始使用它们：

1. 使用为您的账户提供的信息登录 Oracle Cloud。
2. 登录“我的服务管理”，以创建和管理您的 Oracle Content Management 实例和标准开发人员服务。



3. 根据要求验证 Oracle Developer Cloud Service 电子邮件。
4. 使用您的服务的订阅详细信息设置 Oracle Content Management 实例，然后转到实例的 Oracle Content Management URL。
5. 转到标准开发人员服务的 URL。
6. 登录 Oracle Developer Cloud Service 账户。

访问开发人员云服务 URL 并登录控制台。

在开发人员云服务中创建项目

您可以使用“内容体验云”项目模板在开发人员云服务中创建项目，也可以创建具有空 Git 资料档案库的项目，然后从 Oracle Content Management 实例导入 OCE 工具包。

- 使用 Oracle Content Management 模板创建开发人员云服务项目
- 通过从 Oracle Content Management 下载 OCE 工具包在开发人员云服务中创建项目
- 将 OCE 工具包添加到新的 Git 资料档案库中的项目代码

在开发人员云服务中创建模板

您可以使用 cec 命令行实用程序基于可用的源模板创建 Oracle Content Management 模板。

可使用 `cec create-template` 命令基于可用的源模板之一创建模板。在命令行上键入 `cec create-template -h` 将提供可用的源模板。

下面是模板创建示例：

```
cec create-template CafeSupremoLite_yourname -f CafeSupremoLite
```

Windows： 此命令创建符号链接，以便在您创建模板时，在外部 HTML WYSIWYG 编辑器中呈现主题（例如 `_scs_theme_root_`）。要在 Windows 中创建符号链接，通常需要使用管理权限运行该命令行实用程序。如果不使用 WYSIWYG 编辑器编辑主题，则不需要使用管理权限运行，并且可以忽略符号链接创建错误。

上述示例创建 `CafeSupremoLite_yournameTheme` 模板，并在 `cec-components/src/main/` 处提供源代码。下表显示了在开发人员云服务中创建站点模板后源代码的位置。

模板源代码	主题	模板的组件
<code>cec-components/src/main/templates</code>	<code>cec-components/src/main/themes</code>	<code>cec-components/src/main/components</code>

本地测试工具也显示组件，并且支持按模板和类型筛选它们。

您可以使用任何文本或代码编辑器编辑主题和组件文件。请参见[使用本地测试工具进行测试](#)。在编辑主题或组件后可刷新浏览器以查看更改。

❗ 重要：

模板、主题和组件的源代码位于 `src/main/`。不应修改 `src/main` 以外的任何文件，因为 Oracle Content Management 本地服务器正常运行需要它们。

在开发人员云服务中复制模板

可以在开发人员云服务中复制现有 Oracle Content Management 模板。

要从 `src/main/templates` 复制现有模板之一，请使用 `cec copy-template` 命令。如果模板包含来自其他资料档案库的资产，可以选择提供资料档案库映射，否则不会复制这些资产。

以下示例将 `Temp1` 模板复制到名为 `Temp2` 的新模板：

```
cec copy-template Temp1 -n Temp2
```

将模板导入到开发人员云服务中

可以将 Oracle Content Management 中的模板导入到开发人员云服务中以便进行进一步开发。

如果您有从 Oracle Content Management 服务器创建的模板 zip 文件，您可以将该文件导入到开发人员云服务中以便进行进一步开发，例如，编辑主题或组件。请使用以下命令：

```
cec import-template <location of the template zip file>
```

指定 Oracle Content Management 中包含 zip 文件的文件夹。

请参见[关于模板](#)和[导出模板](#)。

合并更改

在本地计算机上创建组件、模板或内容布局或者编辑源代码后，需要将新的和更改的组件与模板合并到项目的 Git 资料档案库中。

要将更改合并到 Git 资料档案库中，请在终端窗口中按顺序输入以下命令。

```
cd cec-components git pull
git add .
git status
git commit -a -m "Your comments" git pull
git push
```

从开发人员云服务导出模板

可以从开发人员云服务导出模板 zip 文件，然后使用该文件在 Oracle Content Management 中创建站点。

模板开发完成后，可以运行以下命令导出模板。该命令响应会告诉您模板的 zip 文件在 Oracle Content Management 中的创建位置。

```
cec export-template CafeSupremoLite_yourname
```

请参见[关于模板](#)和[导入模板](#)。

21

开发主题

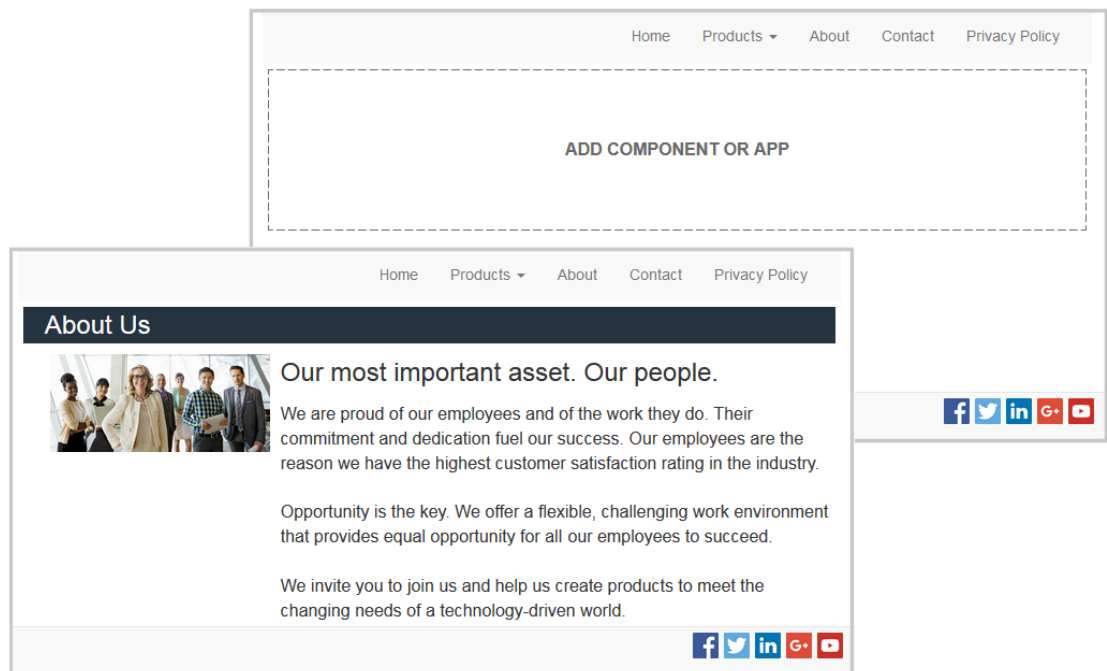
主题定义站点的一般外观（即整体样式），包括颜色方案、字号、字体类型以及页背景。主题为站点中的页提供一致的视觉效果。您可以创建独特主题和主题的变体，同时指定设计和示例内容，之后可以使用它们创建站点以推广您的品牌和愿景。

- [关于主题](#)
- [基本主题结构](#)
- [站点导航](#)
- [创建主题](#)
- [将组件与主题关联](#)
- [站点呈现 API](#)

关于主题

主题定义站点的一般外观，包括内容、外观和行为。主题为站点中的页提供一致的视觉效果。

设计新主题意味着指定布局、样式、示例内容、导航以及用作新站点起点的所有基本信息。主题设计者设置站点外观和行为的预期。设计主题时应考虑其使用方式；例如，大多数或所有用户都预期使用移动设备访问站点。如果您的用户希望创建许多相似站点，则设计定制主题非常有用。可以使用页布局针对可在主题间共享的通用模式设计主题。



主题包含用于设计站点内容、外观和行为的页布局。您可以更改设计和设置，添加内容来创建站点，用于推销您的风格、品牌和愿景。

主题包括：

- 背景图像的资产或作为页布局的一部分的其他内容（图像、JavaScript 文件等）
- 站点的样式设置 (CSS)
- 多个页布局（HTML 文件）
- 用于构造站点导航的代码（JavaScript 文件）
- 可用于组件的基本样式列表（在 `design.css` 和 `design.json` 文件中指定）

主题还包括种子数据，该数据用于填充从其中一个页模板创建的新页。例如，用户为产品部分创建新页并选择名为 `new_product.html` 的页布局。如果主题包含名为 `new_product-pageseed.json` 的文件，则第一次创建新页时，将使用页种子文件的内容填充该页。与示例内容一样，此种子数据可以修改并且只是为您提供构建页的起点。

您可以创建使用组件子集的主题，这些组件要使用该主题。用户为他们的站点选择该主题时，他们将仅看到为该主题指定的组件。请参见[将组件与主题关联](#)。

每个网站使用一个主题。从模板创建站点时，您从模板继承主题。您可以在任何时候更改一个站点的主题。Oracle Content Management 提供了多种带有主题的模板，可供您在入门时使用。

如果站点使用新的未发布主题，则在您首次使站点联机时，主题会自动随站点一起发布。如果您对主题进行了更改并希望更新联机站点以显示更改，则必须明确发布主题。只有主题所有者或具有管理者权限的用户可以明确发布主题。

注：

如果您发布对某个主题的更改，则所有使用该主题的联机站点将会反映这些更改。例如，如果您更改了在主题中指定的默认字体并发布主题，则使用该主题的所有站点都将使用新的默认字体。

另请参见“[管理主题](#)”。

基本主题结构

主题的基本结构包括 Oracle Content Management 中存储的文件夹中指定的设计、导航和样式。主题是站点的模板的一部分。

用户选择模板来创建新站点时，将自动加载关联的主题数据。如果使用 JET 入门级模板或入门级模板，将自动复制主题，而不是引用主题。作为开发人员，如果您使用入门级模板，则需要您自己的主题副本。

主题在特定文件夹和文件结构中进行组织，如以下示例中所示，其中显示了基本文件夹和文件：

```
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
```

某些文件夹包含特定类型的信息，包括以下文件夹：

- `assets`：JavaScript、级联样式表 (Cascading style sheet, CSS)、图像以及布局引用的其他支持文件。
- `designs`：`design.css` 和 `design.json` 文件，用于指定组件的样式选项。
- `layouts`：页模板的 HTML 文件，用于显示站点页。

 **注：**

所有 HTML 文件在文件开头都必须具有 DOCTYPE 元素，如下所示：`<!DOCTYPE html>`

- `publish`：主题已用于发布的站点中后，将列出此目录，其中包含文件的副本。如果您使用桌面应用程序同步了主题，则此目录可见，但是如果您作为模板的一部分导出了主题，则不包括此目录。
- `responsepages`：用于处理错误 (404) 的特殊页。

 **注：**

如果站点中的页标记为错误页，将忽略来自主题的 404 错误消息，改为使用指定的错误页。

- `viewport.json`：指定主题的视窗设置。
- `_folder.json`：指定主题的名称和 GUID。例如：

```
{
  "themeName": "MarketingCampaignTheme",
  "itemGUID": "TB79D65F699B022AC4E11F4D4EE870070A1ADD86BBBB"
}
```

第一次导入主题或者复制主题时，Oracle Content Management 创建 GUID。主题名称在创建主题时由主题开发人员分配。

- `components.json`：（已废弃。）记录主题内使用的定制组件。维护此项仅用于向后兼容。

创建新主题时，您将使用两个主要文件。以下文件为组件设置样式：

- design.css
- design.json

最好将导航信息放入一个 JavaScript 文件；例如，名为 `nav.js` 的文件。主题的 `/assets/js/` 文件夹是此类文件的合适位置。

站点导航

站点的层次存储在与相应站点关联的 `structure.json` 文件中。层次会加载到内存中，并作为 `SCS.structureMap` 对象在页上下文中提供。

站点构建器读取 `structure.json` 文件以在站点构建器中绘制站点树。`structure.json` 文件将包含站点页的代码。例如：

```
"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
}
```

站点页中需要有导航 JavaScript 代码，才能读取相应结构并提取站点的导航链接。随 Oracle Content Management 提供的模板包括说明此操作工作方式的示例导航 JavaScript 文件。

随 Oracle Content Management 提供的部分主题中使用的 `topnav.js` 文件就是一个示例，说明了如何将 `SCS.structureMap` 对象与呈现 API 调用（例如 `SCSRenderAPI.getPageLinkData`）一起使用来遍历站点结构，并提取在页中呈现导航菜单所需的 HTML 标记。下面是示例 `topnav.js` 文件中的代码：

```
function renderNode(id, navBar)
{
    if (id >= 0)
    {
        var navNode = SCS.structureMap[id];
        if( navNode &&
            (
                ( typeof navNode.hideInNavigation != "boolean" ) ||
                ( navNode.hideInNavigation === false )
            ) )
        {
            var navItem = document.createElement("li");
            var navLink = document.createElement("a");
            var navText = document.createTextNode(navNode.name);

            var linkData = SCSRenderAPI.getPageLinkData(navNode.id) || {};
            if( linkData.href ) {
                navLink.href = linkData.href;
            }
            if( linkData.target ) {
                navLink.target = linkData.target;
            }

            navLink.appendChild(navText);
            navItem.appendChild(navLink);

            if (navNode.children.length > 0)
            {
                var navSub = document.createElement("ul");

                for (var c = 0; c < navNode.children.length; c++)
                {
                    renderNode(navNode.children[c], navSub);
                }

                navItem.appendChild(navSub);
            }
            navBar.appendChild(navItem);
        }
    }
}

function renderNav()
{
    var topnav = document.getElementById("topnav");           // expected to be
    an empty <div>
```

```
    if (topnav)
    {
        var navBar = document.createElement("ul");

        renderNode(SCS.navigationRoot, navBar);

        topnav.appendChild(navBar);
    }
}

// Must wait for all our script to be ready...
if (document.addEventListener)
{
    document.addEventListener('scsrenderstart', renderNav, false);
}
else if (document.attachEvent)
{
    document.documentElement.scsrenderstart = 0;
    document.documentElement.attachEvent("onpropertychange",
        function(event)
        {
            if (event && (event.propertyName == "scsrenderstart"))
            {
                renderNav();
            }
        }
    );
}
```

可以使用呈现 API 调用生成将在站点编辑和预览模式下以及在发布的联机站点中有效的导航链接。请参见[呈现 API 参考](#)。

建议将导航信息放在一个 JavaScript 文件（例如 `topnav.js`）中。JavaScript 文件通常存储在主题的 `/assets/js/` 文件夹中，在随 Oracle Content Management 提供的示例主题中可以看到该文件。

创建主题

您可以复制现有主题并对副本进行更改来创建新主题。还可以在新主题中导入并使用 Bootstrap 内容。

注：


无论是创建新主题还是对现有主题进行更新，始终要生成主题的副本并处理该副本。通过示例站点或实际站点的副本测试该副本，以确保它正常运行。请注意，对现有主题进行的更改会在修订的主题发布后立即在使用该主题的所有站点上实施。

复制主题

1. 在主页上，单击开发人员。
此时将显示开发人员页。

2. 单击查看所有主题。

此时将显示现有主题的列表。您可以通过单击“视图”图标并选择列表中的某个选项来控制主题的显示方式。

3. 选择主题，然后在右键单击菜单中选择复制，或单击操作栏中的 。

将复制该主题的所有文件夹和文件，包括任何示例页和内容。

 注：

复制主题时，不复制该主题的共享信息。

4. 输入所复制主题的名称。您无法使用已由另一个主题使用的名称。

可以在名称中使用字母、数字、下划线 (_) 和连字符 (-)。如果输入空格，则使用连字符自动取代空格。

不要将以下名称用于模板、主题、组件、站点或站点页：authsite、content、pages、scstemplate_*、_comps、_components、_compsdelivery、_idcservice、_sitescloud、_sitesclouddelivery、_themes、_themesdelivery。下面的名称虽然可用于站点页，但不要将其用于模板、主题、组件或站点：documents、sites。

5. (可选) 输入主题的说明。

6. 单击复制。

进度栏中显示新主题名称和复制状态。复制了主题之后，该名称显示在主题的列表中。可以单击主题列表中的主题名称来浏览构成主题的文件夹和文件。

7. 使用 Oracle Content Management 桌面应用程序将主题文件夹和文件同步到您的本地系统。这样，您可以浏览本地文件夹并直接处理文件。对主题进行的更改将自动同步。您可以使用喜爱的 HTML、代码或文本编辑工具进行更改。

设置是否可以在站点中使用定制样式

主题管理员可以指定主题随附的样式或者定制样式。

主题 `components.json` 文件中的设置指定是否可以在站点构建器中定制样式。这是为了控制站点内容提供者在构建站点时保持在站点的样式内，不使用其他字体和颜色等。

要不允许在站点中使用定制样式，请执行以下操作：

1. 隐藏组件的设置面板中的定制选项。

在主题的 `components.json` 文件中，添加以下对象以及其他组件定义：

```
[
  {
    "showStyleClassOnly": "true"
  }
]
```

2. 定制 CKEditor 的工具栏组和按钮以删除定制样式：

在主题的 `components.json` 文件中，可以指定 CKEditor 的 `toolbarGroups` 和 `removeButtons` 属性，以为 `Title` 和 `Paragraph` 组件定制组和组内的按钮。

例如，添加以下对象以及其他组件定义，以从 Styles 组中删除样式、字体和颜色：

```
[
  {
    "name": "",
    "list": [
      {
        "type": "scs-title",
        "id": "scs-title",
        "config": {
          "toolbarGroups": [
            {
              "name": "basicstyles",
              "groups": ["basicstyles"]
            },
            {
              "name": "styles",
              "groups": ["styles"]
            },
            {
              "name": "colors",
              "groups": ["colors"]
            },
            "/",
            {
              "name": "undo",
              "groups": ["undo"]
            },
            {
              "name": "links",
              "groups": ["links"]
            },
            {
              "name": "paragraph",
              "groups": ["list", "indent"]
            },
            {
              "name": "align",
              "groups": ["align"]
            },
            {
              "name": "cleanup",
              "groups": ["cleanup"]
            }
          ],
          "removeButtons":
            "Styles,Subscript,Superscript,Strike,Anchor,Blockquote,Link,Unlink,Font,TextColor,BGColor"
        }
      },
      {
        "type": "scs-paragraph",
        "id": "scs-paragraph",
        "config": {
```

```

        "toolbarGroups": [
            {
                "name": "basicstyles",
                "groups": ["basicstyles"]
            },
            {
                "name": "styles",
                "groups": ["styles"]
            },
            {
                "name": "colors",
                "groups": ["colors"]
            },
            "/",
            {
                "name": "undo",
                "groups": ["undo"]
            },
            {
                "name": "links",
                "groups": ["links"]
            },
            {
                "name": "paragraph",
                "groups": ["list", "indent"]
            },
            {
                "name": "align",
                "groups": ["align"]
            },
            {
                "name": "insert",
                "groups": ["image", "table"]
            },
            {
                "name": "cleanup",
                "groups": ["cleanup"]
            }
        ],
        "removeButtons":
        "Styles,Subscript,Superscript,Strike,Anchor,Blockquote,Link,Unlink,Font,TextColor,BGColor"
    }
}
]
]

```

3. 防止将样式粘贴到 CKEditor 中。

粘贴遵循配置的样式和按钮。在前面的示例中，Font 被从粘贴的数据中删除。

不允许定制样式具有以下效果：

- 在所有内置组件中，样式选项卡隐藏 (o) 定制选项。

- 在 CKEditor 中，用于设置样式的工具栏按钮已隐藏，字体系列、字体颜色等的指定已覆盖。
- 此外，在 CKEditor 中，不能对粘贴的文本设置样式。允许使用粗体或斜体。

隐藏主题的组件和部分布局

您可以通过编辑主题的 `component.json` 文件来隐藏组件和部分布局，以便防止站点开发人员使用它们。您可能为了提升一致的外观而这样做。请参见[隐藏主题的组件和部分布局](#)。

通过站点验证主题

完成主题编辑后，您需要检查所有项是否按计划与站点协同工作。

1. 确保您复制的文件夹和文件与 Oracle Content Management 桌面应用程序已同步并且所有修改都已保存。
2. 在站点构建器中打开站点（可以是测试或现有脱机站点）并将该站点切换为使用新主题。
3. 通过添加页和使用设置、查看图像、检查导航以及您在新主题中更改的其他任何项，测试具有该主题的站点。
4. 检查是否正确呈现所有内容，包括所有文件和链接。
5. 发布主题，以便站点实施主题更改，然后在运行时环境中进行检查。

请参见[发布主题](#)。

使用 Bootstrap 主题

由于 Oracle Content Management 主题与 Bootstrap 主题之间的相似性，所以可以转换现有 Bootstrap 主题页和内容以用于 Oracle Content Management 主题。

Bootstrap 是用于创建网站和 Web 应用程序的免费且开源的工具集合。它包含用于界面组件和 JavaScript 扩展的基于 HTML 和 CSS 的设计模板。

您将遵循用于创建主题的相同说明，但是您处理代码（使用您选择的任何编辑器）以从 Bootstrap 主题导入和编辑所需的任何页或内容。

您不可能将整个 Bootstrap 主题都用于其所有页和内容。通常，您将选择该主题的某些部分用于 Oracle Content Management 主题，可能仅几个页模板以及仅其中的某些部分。例如，Bootstrap 主题包含模板内的多个块，包括页眉、导航块、正文块和页脚块。由于 Bootstrap 主题作为可从文件系统直接查看的文件的集合来提供（无 Web 服务器），它们在每页内包含许多重复内容（它们不得不这样，因为未包含页组合引擎）。在 Oracle Content Management 主题中使用此信息时，您需要获取这些部分并将其添加到可重用的 Oracle Content Management 页模板，然后使用 Oracle Content Management 从这些模板动态组合多个页。


导航部分即是将信息添加到可重用模板的一个示例。典型 Bootstrap 主题中的导航将在所有页上复制，但是当您将其移至 Oracle Content Management 主题中时，必须使用 JavaScript 代码动态经过站点的层次并生成导航结构。然后，您只需在所有页上包括该脚本，它们都将获得导航，并且在添加页或从站点删除页时，该导航将进行调整。

请参见[基于 Bootstrap 或网站设计模板创建站点模板](#)。

隐藏主题的组件和部分布局

默认情况下，所有组件和部分布局都可供站点开发人员使用。有时，您可能希望隐藏某个组件或部分布局，以便在使用特定布局时该组件或部分布局不可用。例如，如果您希望提升不使用任何按钮组件或滑块部分布局的一致外观，则可以通过编辑主题的 `components.json` 文件来在主题中隐藏这些组件或布局。

要隐藏主题中的组件和部分布局，请执行以下操作：

1. 在 Oracle Content Management 中，单击左侧导航菜单中的开发人员。
2. 在“开发人员”页上，单击查看所有主题。
3. 在“主题”页上，选择要修改的主题，然后在右键单击菜单中单击打开，或单击操作栏中的 。
4. 选择 `components.json` 文件并单击下载。
5. 在文本编辑器中打开 `components.json` 文件。
6. 添加用于指定组件类型和 ID 的对象，将属性 `hidden` 设置为等于 `true`。例如，下面是用来隐藏按钮组件的示例条目：

```
"type": "scs-button",  
"id": "scs-button",  
"hidden": true
```

同样，下面是用来隐藏滑块部分布局的示例条目：

```
"type": "scs-sectionlayout",  
"id": "scs-sl-slider",  
"hidden": true
```

`components.json` 文件中用来隐藏按钮组件和滑块部分布局的完整条目如下所示：

```
[  
  {  
    "name": "",  
    "list": [  
      {  
        "type": "scs-button",  
        "id": "scs-button",  
        "hidden": true  
      },  
      {  
        "type": "scs-sectionlayout",  
        "id": "scs-sl-slider",  
        "hidden": true  
      }  
    ]  
  },  
  {  
    "name": "Starter",  
    "list": [  

```

```
{
  "type": "component",
  "id": "StarterComponent",
  "themed": true
},
{
  "type": "component",
  "id": "StarterFooter",
  "themed": true
}
]
}
```


7. 进行了所需更改后，将 components.json 文件作为新版本上传到主题。

隐藏主题中组件的对齐方式、宽度或间距选项

默认情况下，大多数组件允许构建站点的人员在组件的设置对话框中指定对齐方式、宽度和间距选项。但是，作为开发人员，您可能希望对定制组件隐藏这些选项，从而防止内容提供者使用的内容呈现方式与站点主题不一致。例如，如果您希望提升外观的一致性，将按钮居中并保持将间距设置为 30 像素，则可以通过编辑主题的 components.json 文件在设置中隐藏用于更改它们的选项。

有时您可能还希望覆盖定制组件的初始默认值，这也可以通过编辑 components.json 文件来完成。对初始值的所有更改只应用于在编辑的 components.json 文件已作为新版本上传后添加到页面的组件。

要隐藏主题中组件的对齐方式、宽度或间距的设置选项，请执行以下操作：

1. 在 Oracle Content Management 中，单击左侧导航菜单中的开发人员。
2. 在“开发人员”页上，单击查看所有主题。
3. 在“主题”页上，选择要修改的主题，然后在右键单击菜单中单击打开，或单击操作栏中的 。
4. 选择 components.json 文件并单击下载。
5. 在文本编辑器中打开 components.json 文件。
6. 添加用于指定组件类型和 ID 的对象，将要隐藏的属性设置为等于 true。例如，下面是用于隐藏设置对话框中按钮组件的对齐方式、宽度和间距属性的条目：

```
"type": "scs-button",
"id": "scs-button",
"hideAlignmentAndWidth": true,
"hideSpacing": true,
```

如果要覆盖按钮组件的默认对齐方式和间距属性，下面是如何指定初始值的示例：

```
"initialData": {
  "alignment": "center",
```

```
"marginTop": 30,  
"marginBottom": 30
```

components.json 文件中用于指定默认值以及隐藏主题中所用按钮组件的设置对话框中的对齐方式、宽度和间距选项的完整条目如下所示：

```
[  
  {  
    "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. 进行了所需更改后，将 components.json 文件作为新版本上载到主题。

将组件与主题关联

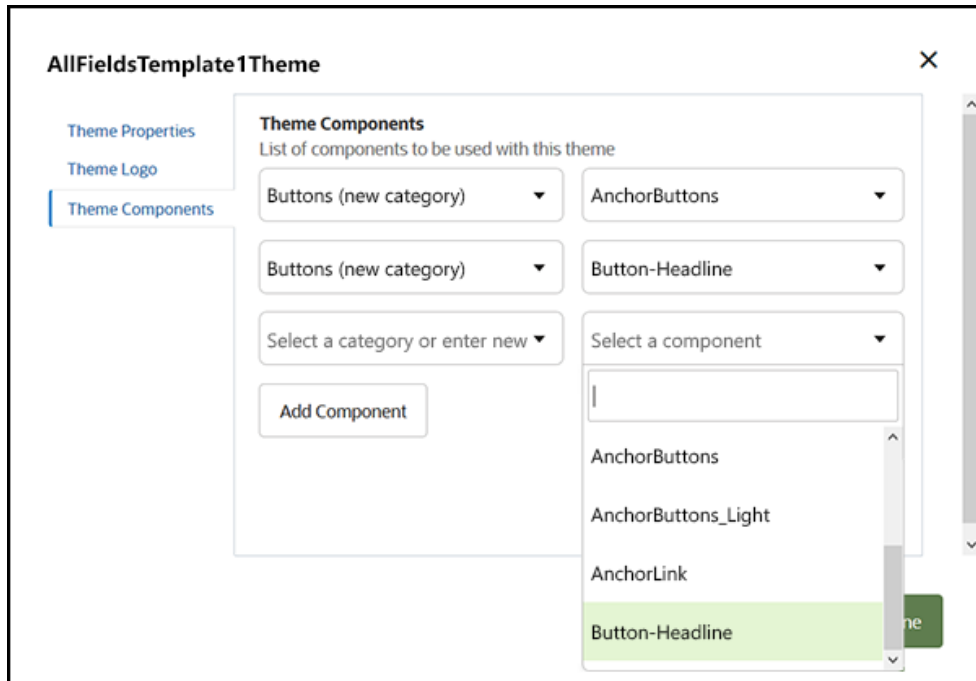
可以将组件、部分布局和组件组与主题关联，以将特定组件子集与该主题一起使用。

作为开发人员，您可以为某个模板创建一个主题和多个组件，以便市场营销人员使用此模板来创建用于促销您组织的产品的站点。将组件与主题关联后，当用户在站点构建器中选择主题组件时，它便可用。在主题组件选项卡上，用户仅会看到已与主题关联的组件。将组件与主题关联可确保此组件将与站点模板一起导出，即使它未在站点上使用也是如此。

要将组件与主题关联，请执行以下操作：

1. 在 Oracle Content Management 中，单击左侧导航菜单中的开发人员。

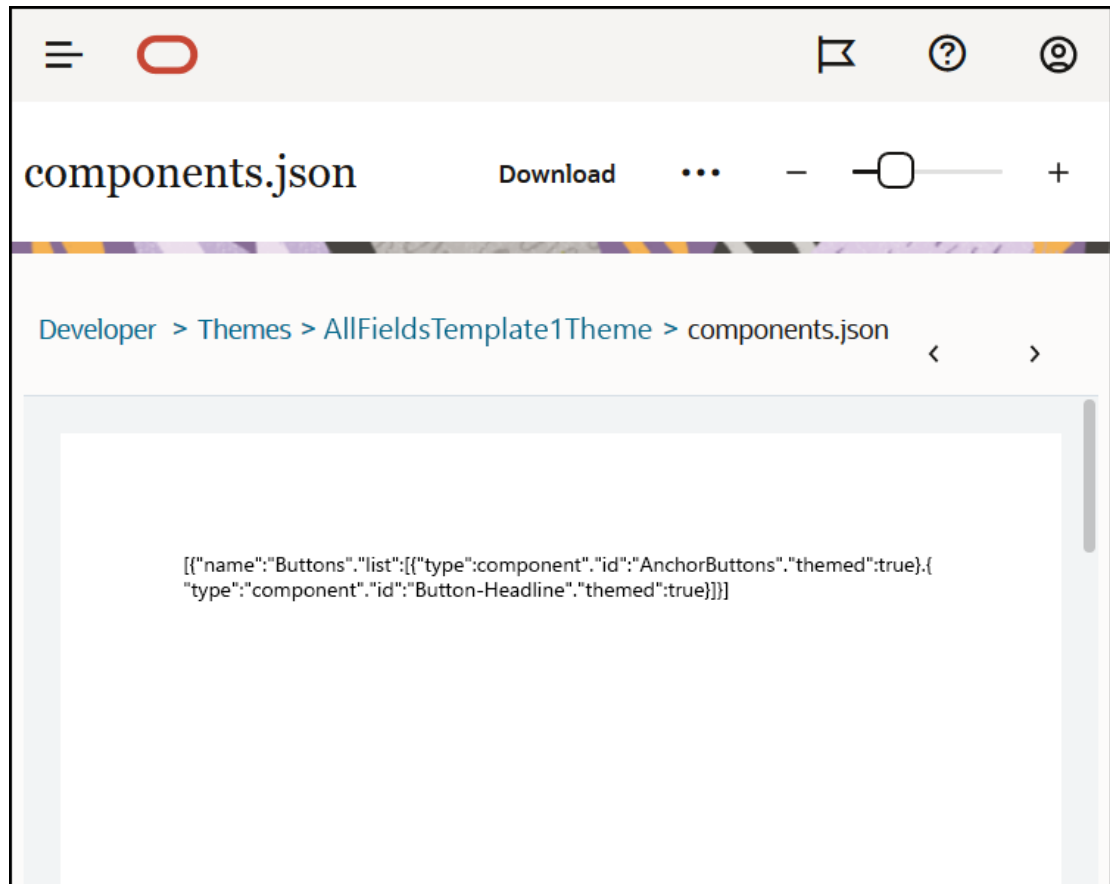
2. 在开发人员页上，单击查看所有主题。
3. 在“主题”页上，选择主题，然后在右键单击菜单中单击属性，或在操作栏中单击。
4. 单击主题组件以打开主题组件选项卡。
5. 选择一个或多个要与主题关联的组件，并对组件进行分类：
 - a. 从选择组件下拉列表中，选择定制组件。
 - b. 对于选择的第一个组件，输入类别名称以便为组件创建类别。
 - c. 对于选择的每个其他组件，选择现有类别或创建新类别。
 - d. 单击添加组件。



6. 单击保存以将选择的组件与主题关联。
7. 在主题页上，选择主题，然后单击“属性”以重新打开主题组件选项卡。
8. 验证选择的组件是否已保存。

要从主题删除关联的组件，请单击主题组件选项卡上组件名称旁边的 **X**，然后在删除组件对话框中单击是。

可以在模板对应的详细信息页上查看与模板使用的主题关联的组件列表。对与主题关联的组件的引用存储在相应主题对应的 `components.json` 文件中。



配置主题关联后，可以将模板部署到 Oracle Content Management 并与市场营销团队共享此模板。市场营销人员为其站点选择该主题后，他们在站点构建器中的主题组件选项卡上将只看到您为该主题指定的组件。定制和全部组件列表也包含关联的主题化组件。

在 OCE 工具包中将组件与主题关联。

要在 OCE 工具包中将组件与主题关联，可以使用以下 `cec` 命令。组件将显示在站点构建器中的主题组件选项卡上，以及定制和全部组件列表中。

```
cec add-component-to-theme <component>
```

以下 `cec` 命令删除组件与主题的关联。

```
cec remove-component-from-theme <component>
```

有关 OCE 工具包的信息，请参见使用 OCE 工具包进行开发。

站点呈现 API

适用于 Oracle Content Management 的[站点呈现 API](#) (SCSRenderAPI) 是一个存在于所有 Oracle Content Management Web 页上的窗口全局对象。它主要负责呈现页面的内容框和组件，并为主题布局上存在的 JavaScript 代码提供接口。如果您以开发人员身份处理主题或组

件，则可能会发现这些事件和功能。运行时 SCSRenderAPI 呈现视图和预览显示模式，而设计时 SCSRenderAPI 呈现导航、编辑和批注显示模式。

在运行时，[站点呈现 API](#) 有三个一般用途：

- 使用组件和内容填充页面上的内容框。
- 满足由页面布局上的 JavaScript 代码发出的信息请求。
- 在页面呈现过程的生命周期内引发事件。

在设计时，[站点呈现 API](#) 有四个一般用途：

- 使用组件和内容填充页面上的内容框。
- 满足由页面布局上的 JavaScript 代码发出的信息请求。
- 在页面呈现和编辑过程的生命周期内引发事件。
- 与站点构建器交互以允许对页面进行编辑和批注。

通过在主题布局上放置以下脚本标记来将[站点呈现 API](#) 加载到 Oracle Content Management Web 页中：

```
<script data-main="/_sitescloud/renderer/renderer.js" src="/_sitescloud/  
renderer/require.js"></script>
```

这通常放在布局底部。（请注意，此标记中的 URL 将自动针对相应的环境（设计时或运行时）进行调整。）SCSRenderAPI 对象以异步方式加载；定制 JavaScript 代码可以通过处理 `scsrenderstart` 事件来监听站点呈现 API 的可用性。

22

开发布局

布局定义内容在页上的排列方式，以及用于为 Oracle Content Management 站点中使用的页生成 HTML。

- [关于布局](#)
- [搜索引擎优化 \(Search Engine Optimization, SEO\)](#)
- [了解 components.json 文件和格式](#)
- [在站点构建器中定制工具栏组](#)
- [限制内容框中的组件](#)
- [将布局内容设为可编辑](#)
- [创建部分布局](#)
- [创建支持延迟加载的部分布局](#)
- [使用 API 开发定制部分布局](#)
- [开发内容布局](#)

关于布局

布局定义内容如何在站点页上排列。不同布局可以包含不同数量的命名内容框。内容框是一个横跨页的区域，其中可以容纳一种或多种类型的内容。

每个主题有多个页布局。请参见[关于主题](#)。

将页添加到站点之后，您选择用于该页的布局。每个布局在页上都有一个称为内容框的区域，在其中可以拖放内容。这些内容框中的内容完全由您决定。这可以是标题、文本和分隔线乃至多媒体、图库和社交媒体的任何内容。

布局包含有效 HTML 结构以及 Oracle Content Management 呈现器理解的特殊标记。布局必须以 DOCTYPE 语句开头以便将浏览器配置为在符合标准的模式下呈现页；例如：<!DOCTYPE html>。特定组件需要此语句来获得最佳结果。

此示例代码显示最小布局：

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <meta http-equiv="X-UA-Compatible" content="IE=edge">
5
6     <script src="/_themes/ [--$SCS_THEME_NAME--]/assets/js/
topnav.js"></script>
7     <link rel="stylesheet" type="text/css"
8         href="/_themes/ [--$SCS_THEME_NAME--]/assets/css/main.css">
9     <link rel="stylesheet" type="text/css"
10        href="/_themes/ [--$SCS_THEME_NAME--]/designs/
[!--$SCS_DESIGN_NAME--]/design.css">
11
```

```

12      <!--$SCS_RENDER_INFO-->
13      <!--$SCS_SITE_HEADER-->
13      <!--$SCS_PAGE_HEADER-->
14 </head>
15 <body>
16     <div id="topNavigation"></div>
17     <div id="mainContentSlot" class="scs-slot scs-responsive"></
div>
18
19     <script data-main="/_sitescloud/renderer/renderer.js"
20           src="/_sitescloud/renderer/require.js"></script>
21     <!--$SCS_SITE_FOOTER-->
22 </body>
23 </html>

```

在浏览器中呈现页时将扩展各种标记。

- [!--\$SCS_THEME_NAME--]

此项扩展为当前为站点选择的主题的名称。使用此标记允许复制主题，因为使用此标记的 URL 将引用当前主题。
- [!--\$SCS_DESIGN_NAME--]

此项扩展为当前为站点选择的设计的名称。这样允许布局由主题内的多个设计使用。
- [!--\$SCS_RENDER_INFO--]

此项扩展为用于保留页的页层次和组件呈现信息的脚本标记。此项应放置在布局的 <head> 部分中。
- [!--\$SCS_SITE_HEADER--]

此项扩展为在搜索引擎优化 (Search Engine Optimization, SEO) 属性的“页眉”字段中指定的站点页眉值。可以在此处输入要放在所有页上的站点范围标记。请参见[设置搜索引擎属性](#)。
- [!--\$SCS_PAGE_HEADER--]

此项扩展为在站点构建器中“页设置”属性中的“页标题”字段中找到的页标题值。可以在此处输入要为此页放置的特定于页的标记。
- [!--\$SCS_SITE_FOOTER--]

此项扩展为在搜索引擎优化 (Search Engine Optimization, SEO) 属性的“页脚”字段中找到的站点页脚值。可以在此处输入要放在所有页上的站点范围标记。请参见[设置搜索引擎属性](#)。

注：

标记还可以使用 <!--\$ 前缀和 --> 后缀作为分隔符代替 [!--\$ 和 --]。

站点联机时，链接中的以下标记将替换为了解其使用的上下文的实际值。这样，链接便可以在编辑站点时起作用，并且在联机时在已发布的站点中起作用。

- /_sitescloud/ 替换为 /_sitesclouddelivery/
- /_themes/ 替换为 /_themesdelivery/

内容框是布局中 class 属性值为 "scs-slot" 的 DIV 元素。用户可以在内容框中添加组件以填充站点内容。通过分配 "scs-slot" 类属性，可以将多个 DIV 元素指定为内容框。每个内容框都必须具有唯一的 id 属性。

 **注：**

内容框不能嵌套，但是可以通过使用 CSS 覆盖它们来给出建议。如果您要执行此操作，请使用组件组或部分布局。请参见[创建部分布局](#)。

浏览器视窗更改分辨率时，额外具有类属性值 "scs-responsive" 的内容框将刷新其内容。这样，内容框可以针对大的桌面显示或小的移动设备相应地进行呈现。

最后的 <script> 标记（示例中的行 19）加载 Oracle Content Management 呈现代码。此代码负责在页上绘制组件，并且它还允许定制代码访问呈现器 API。如果没有此 <script> 标记，则站点构建器无法更改基于布局的页。

搜索引擎优化 (Search Engine Optimization, SEO)

您可以提供关键字来帮助搜索引擎识别站点的内容。

SEO 设置

搜索引擎优化 (Search engine optimization, SEO) 设置在站点级别和页级别定义。SEO 文本将滚入到所有现成模板的页脚中。

请参见[设置搜索引擎属性](#)。

站点访问者的 Cookie

站点访问者将 Cookie 用于 SEO，每个浏览器和每个站点一个 Cookie，以用于计费用途。每个 Cookie 需要每小时更新一次，或在 24 小时后更新。

每个站点必须包含向访问者通知 Cookie 的弹出窗口。



Footer

Text entered above will be displayed in the Cookie consent message.

Oracle Content and Experience uses cookies for the purpose of identifying unique user visitors. To comply with EU Cookie Regulations your site should display the Cookie consent message and include a separate page explaining use of cookies on the site to visitors. For more information about use of cookies in Oracle Content and Experience, see your Service Agreement.

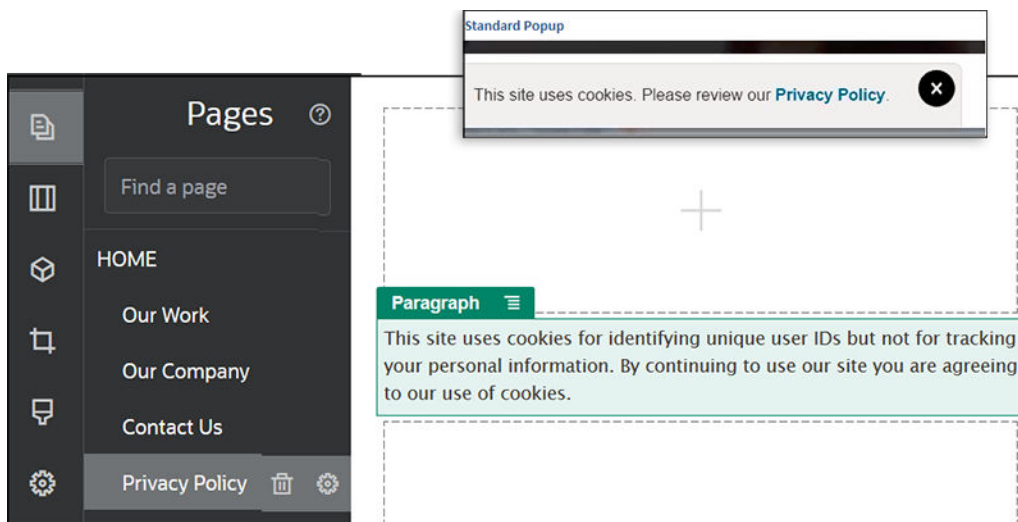
弹出窗口应包含以下信息：

- SEO 页眉和页脚文本
- 页脚中包含特定 ID 的 Div，将在此处选取并链接文本

```
<div class="scs-cookie-consent" style="display: none;">This site uses cookies. Please review our
<a href="[/!-$SCS_PAGE--]500[/!-$SCS_PAGE--]">Privacy Policy</a>. <a href="#" alt="close"
class="scs-cookie-consent-close"></a></div>
```

“隐私政策”页

文本将链接到“隐私政策”页，其中包含忠告文本。



搜索引擎优化的内部 HTML

在站点构建器中保存组件数据时，可以保存在运行时呈现组件时会生成的内部 HTML。此内部 HTML 存储在页数据中，以便在呈现页时，数据可以取代组件插入页中。在页重现的最早期阶段可能会发生这种情况，这可允许搜索引擎有更多的机会成功搜索页内容。

了解 components.json 文件和格式

主题必须具有 components.json 文件，该文件位于 /ThemeName/ components.json 中，用于指定主题中使用的组件。

此文件必须包含有效 JSON，该文件必须至少包含空 JSON 数组 []。

components.json 文件语法列出所有本地组件并且完全支持组件的分类。（远程组件在组件目录中注册。）

无论将什么组件添加到主题级别 components.json 文件（包括无组件），Oracle Content Management 都会填充用户可用的默认组件集。此默认集在源代码中定义。以下列表显示了内嵌框架中呈现的组件和（植入）组件。此外，在服务级别注册且设置为可用于您实例中用户的任何远程组件都将可用于站点构建器。

以下本地组件随 Oracle Content Management 提供。

名称	类型	ID
标题	scs-title	scs-title
段落	scs-paragraph	scs-paragraph
图像	scs-image	scs-image
图库	scs-gallery	scs-gallery
图库网格	scs-gallerygrid	scs-gallerygrid
文档	scs-document	scs-document
按钮	scs-button	scs-button
地图	scs-map	scs-map
分隔线	scs-divider	scs-divider
分隔符	scs-spacer	scs-spacer
YouTube	scs-youtube	scs-youtube
社交栏	scs-socialbar	scs-socialbar
视频	scs-video	scs-video
文章 (定制组件)	scs-component	scs-comp-article
标题 (定制组件)	scs-component	scs-comp-headline
图像和文本 (定制组件)	scs-component	scs-comp-image-text

内嵌框架中呈现的下列组件随 Oracle Content Management 提供。它们不包括注册的远程组件。

名称	类型	ID
对话	scs-app	对话
文档管理器	scs-app	文档管理器
文件夹列表	scs-app	文件夹列表
文件列表	scs-app	文件列表
Facebook 点赞	scs-app	Facebook 点赞
Facebook 推荐	scs-app	Facebook 推荐
Twitter 关注	scs-app	Twitter 关注
Twitter 共享	scs-app	Twitter 共享

一般格式

components.json 文件的一般格式如下：

- 在每个组件内指定组件的属性。顶级 "components" 或 "apps" 属性已废弃。
- 每个组件都具有 "type" 属性。组件只能具有特定值（所有值都列在默认组件的表中）。
- 每个组件都具有 "id" 属性，其必须唯一。此属性用于区分具有相同 "type" 的组件。以前，应用程序具有 "appName" 属性。虽然 "id" 不可用时 "appName" 仍起作用，但是 "appName" 属性已废弃。

- 每个组件都具有 "name" 属性，这是用户界面中的显示名称。如果未指定回退值，对于组件，该值是相应默认组件的名称，对于远程组件，该值是 ID。

下面是 components.json 文件的示例：

```
[
  {
    "name": "COMP_CONFIG_TEXT_CATEGORY_NAME",
    "list": [
      {
        "type": "scs-title",
        "id": "my-headline",
        "name": "My Headline",
        ...
      },
      {
        ...
      },
      ...
    ],
  },
  {
    "name": "My own category name",
    "list": [ ... ]
  }
]
```

一般结构是类别对象的 JSON 数组。每个类别对象都具有 "name" 属性和 "list" 属性。"name" 属性可以是映射到本地化字符串的键。如果这些默认类别不够，您可以提供自己的类别名称，该名称将不进行本地化。下表列出了可用默认类别和相应键。

键	类别名称（英语）
COMP_CONFIG_CONTENT_CATEGORY_NAME	Content
COMP_CONFIG_CUSTOM_CATEGORY_NAME	Custom
COMP_CONFIG_MEDIA_CATEGORY_NAME	Media
COMP_CONFIG_SOCIAL_CATEGORY_NAME	Social
COMP_CONFIG_TEXT_CATEGORY_NAME	Text

每个类别对象中的 "list" 属性都包含组件对象的数组。每个组件或对象都必须具有 "type" 和 "id" 属性。其他属性是可选的。

- "type" 属性必须等于在默认组件中找到的一种类型。如果 "type" 尚未存在，将不显示组件。
- "id" 属性在组件间必须唯一。如果发现 "id" 已存在，将不显示组件。
- "name" 属性是用户界面中组件的显示名称。此项将替换应用程序先前的 "appName" 属性（现在是远程组件）。
- 所有其他属性都与先前发行版中相同。

将新组件添加到 components.json

不允许修改默认组件。但是，您可以基于现有默认组件创建新组件。例如，可以基于用于设置一些默认文本的 "scs-title" 组件来创建新组件。添加新组件所需的最低要求是指定 "type" 和 "id" 属性。

- "type" 必须等于在默认组件中找到的一种类型。如果 "type" 尚未存在，将不显示组件。
- "id" 在组件间必须唯一。如果发现 "id" 已存在，将不显示组件。

下面是用于添加新标题组件的代码示例。此组件将与默认标题组件一起显示。

```
[
  {
    "name": "COMP_CONFIG_TEXT_CATEGORY_NAME",
    "list": [
      {
        "type": "scs-title",
        "id": "my-headline"
      }
    ]
  }
]
```

下面是用于添加具有显示名称和默认文本的新标题组件的代码示例。

```
[
  {
    "name": "COMP_CONFIG_TEXT_CATEGORY_NAME",
    "list": [
      {
        "type": "scs-title",
        "id": "my-headline",
        "name": "My Headline",
        "initialData": {
          "userText": "This is a second title component"
        }
      }
    ]
  }
]
```

请注意，标题组件采用默认标题组件的所有属性作为基础，并在其上应用主题级别的修改来创建新组件。

向后兼容性

仍可以读取采用以前格式的 components.json 文件。

- 具有顶级 "components" 或 "apps" 属性的文件。
- 如果文件包含 "apps" 属性，仍将加载此属性下的用户定义的远程组件。
- 如果文件包含顶级 "apps" 属性，则假定下面列出的任何远程组件都具有类型 "scs-app"。

- 如果 "appName" 属性存在，则将 "id" 设置为 "appName" 值。显示名称将与 "name" (如果指定) 相同，或者回退到 "id" 值。

在站点构建器中定制工具栏组

对于定制组件，可以定制显示在站点构建器工具栏中的现成工具栏组的副本。

您可以基于现成组件创建自己的版本，使其具有受限制的集合并在定制标记上可用。任何定制都不影响现成组件。

您可以基于现成组件创建自己的版本，使其具有受限制的集合并在定制标记上可用，但它不会影响任何现成组件。

工具栏组定义单击标题或段落组件以对其进行编辑时在工具栏上看到的内容。您可以删除和重新排序标题或段落组件支持的内容，但不允许使用其他插件。

要在站点构建器中定制工具栏组，使用的语法需要与 CKEditor 用于其 toolbarGroups 配置的语法相同。请参见 <https://ckeditor.com/latest/samples/toolbarconfigurator/index.html#basic>。

标题和段落组件支持以下组：

- "basicstyles" — 限定为粗体/斜体/下划线 <https://ckeditor.com/cke4/addon/basicstyles>
- "styles" — 字体样式 https://docs.ckeditor.com/ckeditor4/latest/guide/dev_styles.html
- "colors" — 文本和背景颜色 https://docs.ckeditor.com/ckeditor4/latest/guide/dev_colorbutton.html
- "undo" — 对 CKEditor 中的当前实例执行撤消/重做
- "links" - 用于链接对话框的定制插件
- "paragraph" — 带项目符号/编号的列表和缩进支持
 - "list"
 - "indent"
- "align" — 左对齐/右对齐/居中对齐
- "cleanup" — 删除选定文本的样式

段落组件还支持图像和表插入插件：

- "insert"
 - "image"
 - "table"

此外，还可以使用行分隔符条目：

- "/"

 注:

如果您在工具栏组配置中设置了任何其他值，在创建站点构建器工具栏之前将删除该值。您无法提供 "extraPlugins"。只支持 "name"/"groups" 配置。将忽略所有 "items" 条目。

例如，如果您要阻止用户定义字体、颜色、样式或大小，则可以按如下所示更新工具栏配置。对于 "id" 值，您需要指定不同于现成值的定制值。

```
[{
  "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;"
    }
  }
}]
```

验证

使用上述代码替换 `components.json` 文件，然后编辑站点（如果已在编辑，则刷新浏览器）。此时，在编辑标题组件时，它不再显示所选内容的字体样式或颜色。段落组件将继续显示这些内容，而可用字号列表限于 16、24 和 48。

默认工具栏组

标题和段落的默认工具栏组如下：

- 标题

```
[{
  "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"]
}
]

```

限制内容框中的组件

对于任何布局内容框，您都可以对允许在相应内容框中使用的组件指定特定限制。

如果在内容框中限制组件，则当用户拖动不允许使用的组件时，会看到一条警告消息，并且无法在相应内容框中添加或移动组件。

要配置此限制，请编辑主题中的布局（例如，布局文件 `themes\theme_name\layouts\oneslot.htm`），然后向内容框 `div` 添加定制数据属性。

下面是定制属性的格式。与以前版本的主要区别在于用户必须仅指定组件的 `id`。以前，要限制组件，用户必须使用详细的 `<type> <id>` 语法（仍支持）。

```

data-allowed-items='["<id>:", "<type>", "<type>:<id>", ...]'
data-disallowed-items='["<id>:"<type>", "<type>:<id>", ...]

```

注：

`data-allowed-items` 和 `data-disallowed-items` 的值可以使用双引号 (") 或单引号 (')。在以下示例中，`data-allowed-items` 在 JSON 数组两边使用了单引号，而 `data-disallowed-items` 在 JSON 数组两边使用了双引号：

```

<div id="slot101"
  class="scs-slot"
  data-allowed-items='["scs-app", "scs-title"]'
  data-disallowed-items=["'File List', 'scs-map']">
</div>

```

下表列出了随 Oracle Content Management 提供的组件及各自的 ID。为了防止出现任何命名冲突，请勿为任何定制（本地或远程）组件 ID 添加前缀 `scs-`，也不要使用此表中列出的任何类型或 ID。

名称	类型	ID
文档管理器	scs-app	文档管理器
Facebook 点赞	scs-app	Facebook 点赞
Facebook 推荐	scs-app	Facebook 推荐
文件列表	scs-app	文件列表
文件夹列表	scs-app	文件夹列表
Twitter 关注	scs-app	Twitter 关注
Twitter 共享	scs-app	Twitter 共享
按钮	scs-button	scs-button
文章（定制组件）	scs-component	scs-comp-article
标题（定制组件）	scs-component	scs-comp-headline
图像和文本（定制组件）	scs-component	scs-comp-image-text
组件组	scs-componentgroup	scs-componentgroup
内容搜索	scs-contentsearch	scs-contentsearch
内容列表	scs-contentlist	scs-contentlist
内容占位符	scs-component	scs-contentplaceholder
内容项	scs-component	scs-contentitem
分隔线	scs-divider	scs-divider
文档	scs-document	scs-document
图库	scs-gallery	scs-gallery
图库网格	scs-gallerygrid	scs-gallerygrid
图像	scs-image	scs-image
地图	scs-map	scs-map
段落	scs-paragraph	scs-paragraph
社交栏	scs-socialbar	scs-socialbar
分隔符	scs-spacer	scs-spacer
标题	scs-title	scs-title
YouTube	scs-youtube	scs-youtube

用户可以创建本地或远程组件。此示例中提供的名称 (`My_Local_Component`) 是可用于指定要在内容框中限制的此组件的 ID。

Create Local Component ✕

Provide a name for your component

My_Local_Component

Use only letters, numbers, hyphens, and underscores in component names.

Provide a description for your component (optional)

Add an optional description for this component

Component Type

Default

Cancel Create

将布局内容设为可编辑

您可以在布局中配置特定文本或图像内容，以将其设为可由处理基于该布局的页的用户对其进行编辑。

可以在任何主题（包括 Bootstrap 主题）中使用此功能。修改会自动采用原始页的样式。

可以在布局中向以下 HTML 标记 (tag) 添加简单的标记 (markup)：<p>、<h1> 到 <h6>、<div> 和 。

文本和图像格式设置选项包括：

- 文本：允许用户指定粗体、斜体、下划线和链接。
- 图像：允许用户指定指向图像文件的链接。更改标题的属性（用户将光标悬停在图像上时看到的内容），以及添加用于可访问性的替代文本。

要修改文本或图像元素以便用户可以对其进行编辑，有两个条件：

1. 将 `scs-editable` 添加到 `class`。
2. 添加唯一的 `id` 属性。

修改了布局后，该功能可用于基于该布局的所有页（即使是新页也是如此）。

如果复制并粘贴页，则修改将会复制到新页。

修改标题标记

下面是有关如何在布局中修改标题标记以便用户可以对其进行编辑的示例。

1. 将布局文件同步到本地桌面，或编辑 `<h1>` 标题对应的 HTML 源文件。
2. 在标题对应的代码行中，将 `scs-editable` 添加到 `class`，并添加属性 `id="test-heading"`，以使其允许为其附加编辑器。例如：

```
<h1 class="brand-heading scs-editable" id="test-heading">Sample  
Heading Value</h1>
```

3. 保存文件。
4. 与 Oracle Content Management 同步并重新加载浏览器。
当用户将光标悬停在标题上时，标题周围应显示黑色边框，这表示内容可以编辑。当用户单击标题时，边框将变为绿色，这表示用户现在可以编辑标题内容。
5. 如果还未发布主题，请发布。如果已发布主题，则当您刷新浏览器时，将会显示更改。

发布主题后，站点用户可以单击标题并在站点构建器中对其进行编辑。

系统会为 `scs-editable` 元素创建伪组件，以便您可以在站点构建器中对其进行更改，并将其与页数据一起存储。运行时，在呈现页之前，控制器会将 `scs-editable` 标记替换为您在站点构建器中设置的值。

修改图像标记

在布局中修改图像标记以使用户可以对其进行编辑的过程与文本相似。

1. 将 `scs-editable` 添加到 `class`。
2. 添加唯一的图像 `id`。

用户可以单击图像，然后更改属性以使用其他图像。

下面是用户可以编辑的图像的示例代码：

```

```

创建部分布局

创建部分布局，以在站点页上的内容框内排列内容。

企业用户可以根据您作为开发人员提供的部分布局来排列站点上的内容项。可以根据默认布局创建新的部分布局。

您可以导出部分布局以脱机修改，然后将其作为新部分布局导入或者替换现有部分布局。单独导出各个部分布局，或者在模板程序包中导出（模板程序包中包含定制组件和布局）。

提供以下现成的部分布局：


- 水平
- 两列
- 三列
- 垂直
- 选项卡式

- 滑块

可以在站点构建器中立即使用这些布局，而无需创建任何项。

这些部分布局的文件具有注释，包含有关部分布局文件的结构的更多详细信息。要查看注释，可以基于现成部分布局创建新的部分布局，然后导出新布局进行编辑，如以下过程中所述。

要创建部分布局，请执行以下操作：

1. 在主页上，单击开发人员。
2. 单击查看所有组件。
3. 从右侧的创建下拉菜单中，选择创建部分布局。
4. 在创建部分布局对话框中，为部分布局组件提供名称和说明。
5. 要导出部分布局进行编辑，请选择部分布局，然后在右键单击菜单中单击导出，或单击操作栏中的 。
 - a. 导航到现有文件夹，或单击创建以创建新文件夹并提供名称，还可以提供说明。
 - b. 选中文件夹旁边的复选框，然后单击确定。
 - c. 单击文件夹的图标或名称以打开它。

使用部分布局名称和 .zip 扩展名在选定文件夹中创建布局程序包文件。将文件下载到您的开发环境以编辑文件。

[使用 API 开发定制部分布局](#) 中包括有关部分布局 API 的信息。

6. 将修改的文件作为新部分布局导入或者替换现有部分布局。
 - a. 在主页上，单击内容选项卡，然后单击文档。
 - b. 将修改的部分布局上载到文件夹，该布局位于扩展名为 .zip 文件中，其中包括您导出的相同文件夹和文件名。
 - c. 在开发人员页上，单击查看所有组件。
 - d. 从创建菜单，选择导入组件。
 - e. 选中包含所修改部分布局的已上载 zip 文件旁边的复选框，然后单击确定。

您修改的部分布局将导入到所选的文件夹。

您也可以导出部分布局，以将其复制或移动到其他 Oracle Content Management 实例并在该处导入。

创建支持延迟加载的部分布局

内容列表可以调用部分布局以及其他组件（如果查询这些组件）。

呈现内容列表时，您可以选择部分布局来呈现返回的所有内容项。这样，您可以为内容项创建各种不同的布局，例如表、滑块或八列布局。这些定制部分布局还可以参与更高级的分页功能。

内容列表支持以下分页：

- 分页
- 滚动时加载
- 单击时加载

对于标准分页功能，部分布局不需要执行任何操作。用户单击下一页时，将用下一组项重新呈现该布局。但是，对于滚动时加载和单击时加载，将不重新呈现部分布局，而是将其他组件添加到部分布局。这主要用于无限滚动模型，其中您加载前 n 项，然后随着用户向下滚动页，您将提取并呈现下一组项。为了支持滚动时加载和单击时加载，定制部分布局需要进行以下相关操作

1. `render.js`：实施 `addComponent()` API。将对添加到部分布局的每个新组件调用此项。

```
// dynamic API for adding additional components through "load more"
when used in a Content List
addComponent: function (parentObj, component) {
    // create the component div and add it to the parent
    object
    $
    (parentObj).append(this.createComponentDiv(component));
}
```

2. `appInfo.json`：包括以下内容，以使内容列表知道部分布局支持 `addComponent()` api。

```
"contentListData": {
  "addComponent": true
},
```

更新 `appInfo.json` 后，当用户在设置面板中选择此部分布局并转至分页屏幕时，他们将看到单击时加载和滚动时加载选项。

使用 API 开发定制部分布局

可以在 Oracle Content Management 中使用部分布局 API（包括呈现 API 和编辑 API）开发定制部分布局。

有关要查看的入门级文件，请参见[创建部分布局](#)。部分布局的入门级文件包括有关部分布局文件结构的注释与详细信息。

呈现 API

从 `render.js` 模块加载的呈现 API 用于站点构建器中以及运行时。

呈现 API	说明	输入参数	返回结果
(构造器)	初始化部分布局呈现模块。	包含以下属性的 JavaScript 对象： <ul style="list-style-type: none"> sectionlayoutData (对象)：在页模型中找到的部分布局数据。 componentId (字符串)：部分布局的 componentId 值，通常为 GUID。 renderMode (字符串, 可选)：呈现操作的呈现模式。 customSettingsData (对象)：在 sectionLayoutData 中找到的 customSettingsData 副本。 	初始化部分布局呈现 API。
render	将适合部分布局的 DOM 元素发到页，包括子组件的容器 DIV。	container (元素)：应在其中呈现部分布局的标记的 DOM 元素。	在此方法返回后，将呈现子组件。 可以通过查找子 div[id] 元素来识别子组件。
addComponent	与内容列表组件结合使用来动态将子组件添加到部分布局。 此函数是可选的。	container (元素)：应在其中呈现新组件的 DOM 元素。 componentId (字符串)：要添加到部分布局的新组件的 ID。	在此方法返回后，将呈现其 ID 匹配 componentId 输入的元素。

编辑 API

如果与部分布局关联的 appinfo.json 文件中 hasEditHandlers 属性设置为 true，则将加载 edit.js 模块。

编辑 API 用于站点构建器中。

此模块中除构造器以外的所有函数都是可选的。

编辑 API	说明	输入参数	返回结果
(构造器)	初始化部分布局编辑模块。	包含以下属性的 JavaScript 对象： <ul style="list-style-type: none"> componentId (字符串)：部分布局的 componentId 值，通常为 GUID。 	初始化部分布局编辑 API。

编辑 API	说明	输入参数	返回结果
getCapabilities	返回描述部分布局的编辑功能的对象。	<p>描述部分布局的编辑功能的 JavaScript 对象。</p> <p>输入后，将向该函数提供默认功能。该函数可以根据需要修改功能对象。</p> <p>功能对象可以包含以下功能：</p> <ul style="list-style-type: none"> • title (字符串)：要向用户显示的部分布局标题。 • settingsTitle (字符串)：要在“设置”面板对话框中显示的标题。 • hasSettings (布尔值)：指示部分布局是否支持“设置”面板。 • allowMove (布尔值)：指示部分布局是否允许移动子项。 • allowDelete (布尔值)：指示部分布局是否允许删除子项。 • isHidden (布尔值)：指示当前是否根据用户选项隐藏了部分布局。 • dropTarget (布尔值)：指示部分布局是否是拖放操作的目标。 • customMenuOptions (数组)：要添加到部分布局上下文菜单的定制菜单选项。每个菜单选项对象都具有以下属性： <ul style="list-style-type: none"> – label (字符串)：菜单项的显示文本。 – action (函数)：单击菜单项时要调用的函数。 – disabled (布尔值)：指示菜单项应以禁用状态显示。 – icon (字符串)：要与标签一起显示在菜单项中的 URL。（此属性保留供将来使用。） – checkmark (布尔值)：指示复选标记应与标签一起显示在菜单项中。 – subMenuItems (数组)：要显示在子菜单中的菜单选项。 	(对象)：部分布局的功能。
getCaptionContent	返回部分布局显示名称，此名称将显示在 UI 元素中。	无。	(字符串)：部分布局的显示名称。
filterCapabilities	<p>允许部分布局在菜单显示给用户之前修改功能对象。</p> <p>可以使用此 API 调整或删除菜单选项。（另请参见 getCapabilities。）</p>	描述部分布局的编辑功能的 JavaScript 对象。输入后，将向该函数提供默认功能。	(对象) 部分布局的功能。

编辑 API	说明	输入参数	返回结果
onDragOver	在执行拖放操作期间调用，以指示是否可以将拖动项放置在部分布局上。	eventObject (事件对象)：用于保存有关拖动事件的信息的事件对象。 dataTransfer (DataTransfer 对象)：用于保存有关要在部分布局上拖动的项的信息的 DataTransfer 对象。	(布尔值) 指示部分布局是否可以接受拖动项的值。 如果部分布局可以接受拖动项，则返回 <i>true</i> ，否则返回 <i>false</i> 。
onDrop	在执行拖放操作的放置部分期间调用，以指示是否应将拖动项放置在部分布局中。	eventObject (事件对象)：用于保存有关放置事件的信息的事件对象。 dataTransfer (DataTransfer 对象)：用于保存有关要放置在部分布局上的项信息的 DataTransfer 对象。	(布尔值) 指示部分布局是否已处理放置操作的值。 返回 <i>true</i> 表示绕过默认逻辑。
onAddComponent	向部分布局通知拖放操作已在部分布局中添加项。	eventObject (事件对象)：用于保存有关拖动事件的信息的事件对象。 dataTransfer (DataTransfer 对象)：用于保存有关要放置在部分布局上的项信息的 DataTransfer 对象。 componentId (字符串)：新添加项的 componentId 值。	部分布局通知。
onMoveComponent	向部分布局通知拖放操作已在部分布局中移动项。	eventObject (事件对象)：用于保存有关拖动事件的信息的事件对象。 dataTransfer (DataTransfer 对象)：用于保存有关要放置在部分布局上的项信息的 DataTransfer 对象。 componentId (字符串)：已移动项的 componentId 值。	部分布局通知。
getSettingsData	允许部分布局在显示“设置”面板之前更改设置数据。	settingsData (对象)：为部分布局计算的默认设置数据	(对象) 部分布局的设置数据。
updateSettings	允许部分布局在关闭了“设置”面板之后更改其设置。 在页模型中存储设置之前调用此 API。	parameters (对象)：从“设置”面板返回的原始参数对象。 sectionLayoutData (对象)：将存储的部分布局数据。将从 parameters (对象) 生成默认数据，并将其传递到此参数中的函数。	(对象) 要存储在页模型中的部分布局数据。
dispose	允许编辑模块释放内存、分离事件以及撤消分配与编辑处理程序关联的资源。 部分布局需要彻底重绘 (例如撤消/重做操作) 时调用此 API。	无。	启用重绘部分布局。

开发内容布局

内容布局通过站点页中使用的内容列表或内容占位符组件来帮助用户查看内容项中的数据。您可以为一种内容类型创建多个内容布局，以创建不同视图或表示内容项的不同部分。

例如，根据 Blog-Post 内容的使用方式和使用位置，Blog-Post 内容类型可能需要不同的内容布局。站点的主页可能会显示 Blog-Post 项列表，但在主页上单击了博客帖子时，[详细信息页](#)可能会显示有关相应博客帖子的详细信息。

主页包含配置为列出 Blog-Post 内容类型的项的内容列表，并使用 Blog-Post-Summary 内容布局作为项视图。

“详细信息”页在内容占位符中使用 Blog-Post-Header 内容布局来显示页眉图像和标题。两列部分布局包含两个内容占位符，宽度各占 70% 和 30%，并使用 Blog-Post-Content 和 Blog-Post-Author 内容布局。如您所见，使用了四个不同的内容布局来可视化同一内容类型。

可以采用以下两种方式中的任何一种来创建内容布局：

- 在 Oracle Content Management 中，选择开发人员 > 查看所有组件 > 创建 > 创建内容布局。
- 在 OCE 工具包项目中，使用 `cec create-contentlayout` 命令。

Oracle Content Management 会按内容类型创建默认内容布局。要修改默认内容布局，可以编辑以下文件：

- `assets/layout.html`
编辑此文件以更改 HTML 视图。
- `assets/design.css`
编辑此文件为内容布局设置样式。
- `assets/render.js`
编辑此文件以更改 `layout.html` 中使用的数据，或向内容布局添加动态行为。

内容布局组件呈现来自 Oracle Content Management 服务器的内容项。大多数资产存储在 Oracle Content Management 服务器中。有时，您可能希望在内容布局本身中使用本地可用的静态资产，例如，用于样式设置用途的背景图像。例如，在下面的内容布局中，可以在 `render.js` 中生成 `images/background.jpg` 的绝对 URL，并在 `layout.html` 中使用它。

最简单的绝对 URL 生成方法是使用内置的 `requirejs` 文件。定义 `'require'` 作为相关项，并使用 `require.toURL()` 生成 URL，如以下代码所示：

```
define([
  'require',
  'jquery',
  'mustache',
  'text!./layout.html',
  'css!./design.css'
], function (require, $, Mustache, templateHtml, css) {
  'use strict';

  var imageURL = req.toUrl('./images/background.jpg');
```

以下主题介绍了如何开发内容布局：

- [通过 Oracle Content Management 创建内容布局](#)
- [将布局视图传递到内容布局](#)
- [通过 API 生成“站点详细信息”页 URL](#)

- [通过开发人员云服务在本地开发内容布局](#)
- [扩展内容列表查询中的宏](#)
- [开发强健的内容布局](#)
- [创建示例博客模板](#)
- [将内容布局映射添加到模板](#)
- [使用本地测试工具测试内容布局](#)
- [将具有内容布局的模板导入到 Oracle Content Management 中](#)

相关主题

有关如何管理内容布局的信息，请参见[管理定制组件和布局](#)。

有关在站点中使用数字资产和其他内容项的信息，请参见《通过 *Oracle Content Management* 管理资产》中的“使用资产”和“管理数字资产”。

通过 Oracle Content Management 创建内容布局

创建内容布局，以便内容项显示在站点页上时对该项中的字段进行布局。将内容项添加到页时，它将使用所选的内容布局。

企业用户可以根据您作为开发人员提供的内容类型和布局来创建和使用内容项。可以根据默认布局创建新的内容布局。与内容类型关联的多个内容布局使站点设计者可以在不同的上下文中显示内容项，而无需更改内容。

如果在内容列表组件中使用内容布局，则内容布局为每个内容项重复一次。然后按部分布局排列内容布局。

您可以导出内容布局以脱机修改，然后将其导入以替换现有内容布局。

要创建内容布局，请执行以下操作：

1. 在 Oracle Content Management 主页上，单击侧面导航中的开发人员。
此时将显示开发人员页。
2. 单击查看所有组件。
3. 从右侧的创建下拉菜单中，选择创建内容布局。
4. 在创建内容布局对话框中，选择将使用布局的内容类型，选择要显示的字段，以及启用添加对于站点中所用定制设置的支持（如果您希望站点创建者在向页添加内容项和内容列表组件时能够添加定制设置）。

Create Content Layout ✕

Choose a Content Type

Create a content layout to display items of this type

Choose a content type ▼

Choose Fields to Display

Overview ▼

Add support for custom settings when used in Sites

Provide a name for your content layout

Use only letters, numbers, hyphens, and underscores in content layout names.

Add a name for this content layout

Provide a description for your content layout (optional)


Add an optional description for this content layout

Cancel Create

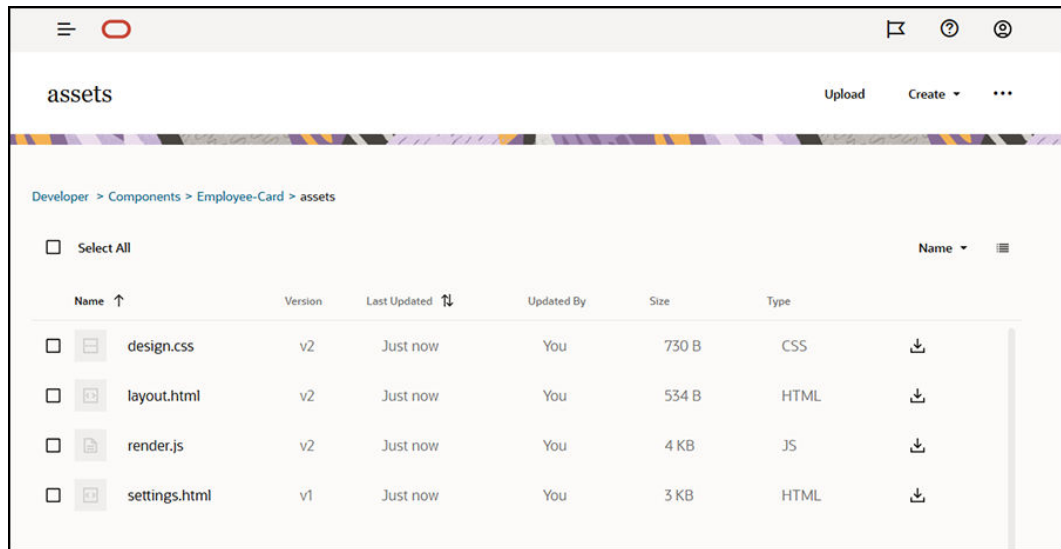
5. 提供内容布局组件的名称和说明，然后单击创建。
该内容布局将添加到您的组件。

 **注：**

仅字母数字字符、连字符和下划线可用于内容布局标题。

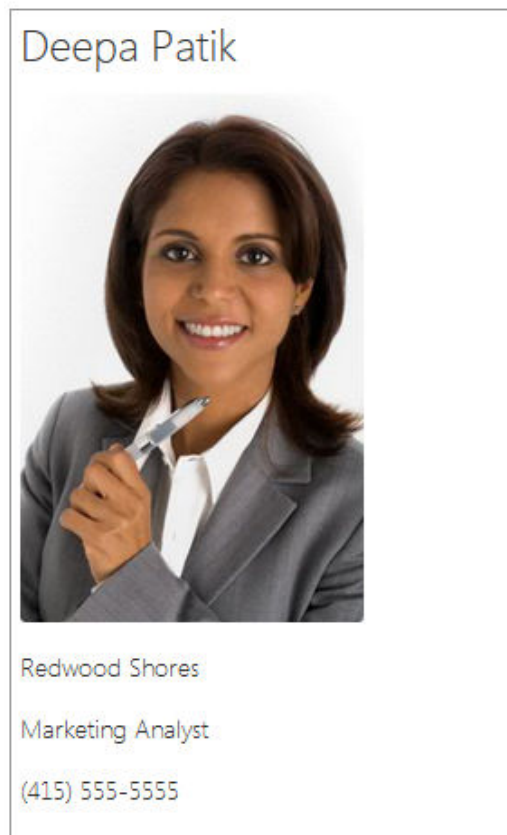
6. 要导出内容布局进行编辑，请选择内容布局，然后在右键单击菜单中单击导出，或单击操作栏中的 。
 - a. 导航到现有文件夹，或依次选择创建和文件夹以创建新文件夹并提供名称，还可以提供说明。
 - b. 选中文件夹旁边的复选框，然后单击确定。
 - c. 单击文件夹的图标或名称以打开它。

使用内容布局名称和 `.zip` 扩展名在选定文件夹中创建布局程序包文件。将文件下载到您的开发环境以编辑文件。

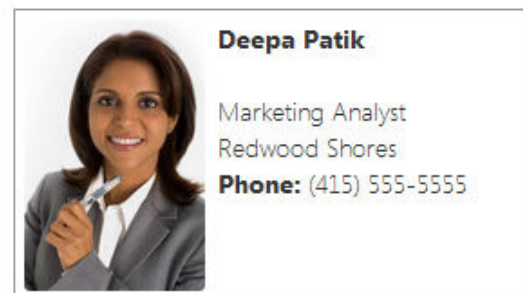


这些文件控制使用 Employee-Card 内容布局的内容项中字段的布局。如果启用了添加对于站点中所用定制设置的支持，则还将创建一个名为 **settings.html** 的附加文件，用于提供单个内容项的默认呈现，以便可以显示它。

Default Layout



Card Layout



7. 编辑 `design.css`、`layout.html` 和 `render.js` 文件以获得所需的内容布局。

例如，以下文件指定 Employee-Card 内容布局：

a. 编辑 design.css 文件：

```
.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. 编辑 layout.html 文件：

```
{{#data.employee_profile_pictureURL}}
<div class="scs-tile-layout-img-container">
  
```

```

</div>
{{/data.employee_profile_pictureURL}}
<ul class="scs-tile-layout">
  <li>
    <p><b>{{name}}</b></p>
  </li>
  <li>&nbsp;</li>
  <li>
    <p>{{data.employee_job_title}}</p>
    <p>{{data.employee_location}}</p>
    <p><b>Phone: </b>{{data.employee_phone}}</p>
  </li>
  {{#scsData.detailPageLink}}
  <li>
    <a href="{{scsData.detailPageLink}}" title="Go to detail
page"><span class="detail-page">Profile</span></a>
  </li>
  {{/scsData.detailPageLink}}
</ul>

```

c. 编辑 render.js 文件:

```

/* globals define */

define([
  'jquery',
  'mustache',
  'text!./layout.html',
  'css!./design.css'
], function($, Mustache, templateHtml, css) {
  'use strict';

  function ContentLayout(params) {
    this.contentItemData = params.contentItemData || {};
    this.scsData = params.scsData;
    this.contentClient = params.contentClient ||
(params.scsData ? params.scsData.contentClient : null);
    // retrieve the custom settings
    this.customSettings = this.scsData.customSettingsData
    || {};
  }

  ContentLayout.prototype = {

    render: function(parentObj) {
      var template,
          content = $.extend({}, this.contentItemData),
          contentClient = this.contentClient,
          contentType,
          secureContent = false;

      if (this.scsData) {
        content = $.extend(content, { 'scsData':
this.scsData });
        contentType = content.scsData.showPublishedContent

```

```
=== true ? 'published' : 'draft';
    secureContent = content.scsData.secureContent;
  }

  if (contentClient) {
    var params = {
      'itemGUID': typeof
content.data.employee_profile_picture === 'string' ?
      content.data.employee_profile_picture :
content.data.employee_profile_picture.id,
      'contentType': contentType,
      'secureContent': secureContent
    };
    content.data.employee_profile_pictureURL =
contentClient.getRenditionURL(params);
  }

  console.log(content);

  try {
    // Mustache
    template = Mustache.render(templateHtml,
content);

    if (template) {
      $(parentObj).append(template);
    }
  } catch (e) {
    console.error(e.stack);
  }
};

return ContentLayout;
});
```

有关编辑 `render.js` 和其他文件的信息，请参见[开发组件](#)。

8. 导入修改的文件以替换现有内容布局。
 - a. 在主页上，单击文档。
 - b. 将修改的内容布局上载到文件夹，该布局位于扩展名为 **.zip** 文件中，其中包括您导出的相同文件夹和文件名。

如果要将其作为新内容布局导入，则需要更改 `_folder.json` 中该内容布局的 GUID。
 - c. 在主页上，单击开发人员。

此时将显示开发人员页。
 - d. 单击查看所有组件。
 - e. 从创建菜单，选择导入组件。
 - f. 选中包含所修改组件的已上载 zip 文件旁边的复选框，然后单击确定。

您修改的内容布局将导入到组件。

您也可以导出内容布局，以将其复制或移动到其他 Oracle Content Management 实例并在该处导入。

将布局视图传递到内容布局

开发内容布局时，如果内容布局正用于站点，您可以获取基础组件的属性。

对于此特定用例，您可以获取为内容列表中的内容类型选择的内容布局视图。然后，您可以根据类别变更组件的呈现方式。如果无法访问此属性，您将需要创建实际上相同的两个内容布局。

`contentLayoutCategory` 属性可用于为内容项和内容列表组件呈现的内容布局。可以通过 Sites SDK 访问此属性，如下所示。

```
scsData.SitesSDK.getProperty('contentLayoutCategory', function  
(layoutCategory)  
    { console.log(layoutCategory);});
```

请参见 Oracle Content Management SDK。

仅当从 Oracle Content Management 站点中的内容项或内容列表呈现内容布局时，此属性才可用。从第三方应用程序呈现内容布局时，此属性不可用。

通过 API 生成“站点详细信息”页 URL

如果您要从自己的查询呈现某个内容布局中的内容项列表，您可以创建指向内容项的详细信息页的链接。您可以使用 Sites SDK `SCSRenderAPI.getPageLinkData` API 生成站点详细信息页 URL。

`detailPageId` 属性是内容项或内容列表中选择的详细信息页的 ID。如果未设置值，则返回的值是 `SiteStructureMap` 中设置了 `isDetailPage` 属性的第一个页。

要访问此属性，可以使用 Sites SDK。仅当将内容布局用于 Oracle Content Management 站点时，此属性才可用。只能通过 `scsData` 访问它，在创建内容布局时，它作为其中一个参数进行传递。例如：

```
scsData.SitesSDK.getProperty('detailPageId', function (detailPageId)  
{ console.log(detailPageId);});
```

有了 `detailPageId` 后，就可以使用它构造指向详细信息页的链接。

`SCSRenderAPI` 具有函数 `getPageLinkData()`，此函数接受 `pageId` 及其他选项，并通过选项传递来构造页的所需 URL。此函数的签名如下：

```
SCSRenderAPI.getPageLinkData(pageId,  
    options);
```

它包含以下参数：

- **pageId**：与 Sites SDK `detailPageId` 属性返回的 `detailPageId` 相同。
- **options**：
 - `contentType`

- contentId
- contentName

返回值是具有以下属性的对象：

- hideInNavigation
- href
- href

以下示例集合了上述所有内容：

```
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);
});
```

输出为："/sites/{site}/{detailPageName}/{contentType}/{contentId}/{contentSlug}"

如果 pageId 不是详细信息页，则不会将内容值添加到 URL。

请参见 Oracle Content Management SDK。

通过开发人员云服务在本地开发内容布局

可以通过 OCE 工具包使用开发人员云服务为 Oracle Content Management 在本地创建、编辑、配置和测试内容布局。

执行以下步骤为使用开发人员云服务开发内容布局做准备：

1. 在本地计算机上设置 [OCE 工具包](#)。
2. 登录 [Oracle Content Management 的开发人员云服务控制台](#)
3. 在开发人员云服务中创建项目。
4. 将 [OCE 工具包](#) 添加到新的 [Git 资料档案库](#) 中的项目代码。
5. [创建内容布局](#)

以下主题介绍了如何使用 OCE 工具包通过开发人员云服务在本地开发内容布局：

- [通过开发人员云服务创建内容布局](#)
- [定义 RequireJS 模块](#)
- [配置构造器函数参数](#)
- [呈现内容布局](#)
- [编辑 Mustache 模板中的内容布局](#)
- [添加动态 DOM 操纵](#)

- 在 `design.css` 文件中定义样式
- 获取引用项
- 获取媒体 URL
- 引发触发器
- 导航到包含搜索查询的搜索页
- 扩展宏和呈现多信息文本
- 指向“详细信息”页的链接

使用开发人员云服务模板开发内容布局后，在项目的 Git 资料档案库中[合并更改](#)。

通过开发人员云服务创建内容布局

可以使用 `cec create-contentlayout` 命令在开发人员云服务项目中按内容类型创建内容布局。

要在开发人员云服务项目中创建内容布局，请执行以下操作：

1. 在终端窗口中，转到 `cec-components` 目录。
2. 输入 `cec create-contentlayout` 以查看该命令的选项和示例：

```
Usage: cec create-contentlayout <name>
```

```
Creates a content layout based on a content type from a local template or from CEC server.
```

```
By default, an "overview" content layout is created. Optionally specify -s <style> to create in a different style.
```

```
Valid values for <style> are:
```

```
  detail
  overview
```

```
Options:
```

```
  --contenttype, -c <contenttype> Content layout is based on [required]
```

```
  --template, -t <template> Content type is from
```

```
  --server, -r flag to indicate the content type is from server
```

```
  --style, -s <style> Content layout style
```

```
  --addcustomsettings, -a Add support for custom settings when used in
```

```
Sites
```

```
  --help, -h Show help
```

```
[boolean]
```

```
Examples:
```

```
  cec create-contentlayout Blog-Post-Overview-Layout -c Blog-Post -t BlogTemplate
```

```
  cec create-contentlayout Blog-Post-Detail-Layout -c Blog-Post -t BlogTemplate -s detail
```

```
  cec create-contentlayout Blog-Post-Overview-Layout -c Blog-Post -t BlogTemplate -a
```

```
  cec create-contentlayout Blog-Post-Overview-Layout -c Blog-Post -r
```

```
cec create-contentlayout Blog-Post-Overview-Layout -c Blog-Post -r -s detail
```

3. 输入以下命令以查看服务器上可用的内容类型：

```
cec list-server-content-types
```

4. 如 "Usage" 中所示，可以按 Oracle Content Management 服务器中的内容类型创建内容布局，也可以按 `cec-components/src/main/templates` 下的模板中的内容类型创建内容布局。例如，以下命令按服务器中的类型创建内容布局：

```
cec create-contentlayout Blog-Post-Overview-Layout -c Blog-Post -r
```

可以编辑以下文件来修改内容布局：

- `assets/layout.html`
此文件指定 HTML 视图。请参见[编辑 Mustache 模板中的内容布局](#)。
- `assets/design.css`
此文件指定内容布局的样式。请参见在 `design.css` 文件中定义样式。
- `assets/render.js`
此文件指定 `layout.html` 中使用的数据，您可以在此文件中向内容布局添加动态行为。请参见[定义 RequireJS 模块](#)。

如果您创建了一个允许使用定制设置的布局（在创建布局时使用选项 `-a`），则还将创建一个名为 `settings.html` 的附加文件，用于提供单个内容项的默认呈现，以便可以显示它。

定义 RequireJS 模块

可以在 `render.js` 文件中定义 RequireJS 模块。站点会加载相关项，例如 JQuery、Mustache、RequireJS 文本插件和 RequireJS CSS 插件。

```
define([
    'jquery',
    'mustache',
    'text!./layout.html',
    'css!./design.css'
], function ($, Mustache, templateHtml, css) {
```

可以使用 Mustache 模板系统呈现布局。

内容布局的 `assets/render.js` 文件具有以下属性：

- 它应为 RequireJS 模块
- 它应返回 JavaScript 构造器函数。站点通过传递参数对象来调用构造器函数。该参数对象包含呈现布局所需的内容项数据和 API。
- 此构造器函数应包含用于处理内容布局呈现的 `render(parentObj)` 方法。它应将内容布局 DOM 对象附加到 `parentObj` 对象（此对象传递到 `render()` method）。
- RequireJS 模块可以使用相关项，包括 JQuery、Mustache、RequireJS 文本插件和 RequireJS CSS 插件。这些相关项将由站点加载。您也可以使用其他库。

配置构造器函数参数

站点创建构造器函数的新实例时，它会传递一个包含 `contentItemData`、`scsData` 和 `contentClient` 的参数来帮助进行内容布局开发。

下面是构造器函数的示例代码：

```
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;
```

该构造器函数参数包括以下对象：

- **params.contentItemData**：包含内容项，其中包括其名称、说明、ID 和数据。例如，可以使用 `params.contentItemData.data['blogpost_title']` 访问内容项中的字段 `'blogpost_title'`。
- **params.scsData**：从站点内调用构造器时，此对象传入信息。对于在第三方应用程序中呈现的内容布局，不存在此对象。此对象包含 Sites SDK 对象、用于引发触发器的方法 `contentTriggerFunction` 以及详细信息页链接。
- **params.contentClient**：这是从 Content SDK 创建的 `contentClient` 对象，用于调用内容布局。因此为其配置了适用于内容服务器的参数。如果您需要对内容服务器进行额外的调用，可以使用此 `contentClient` 对象，而不是创建自己的对象。此对象包含适用于内容的客户端 API。提供了一些 API 以用于查询、搜索以及获取内容项及其内容类型。此外，还提供了其他的帮助程序 API，例如，用于扩展多信息文本中所用宏的 `expandMacros()`。

呈现内容布局

`ContentLayout` 的 `render(params)` 方法基于模板呈现内容布局。默认情况下，呈现内容布局时使用 `Mustache` 模板，但您可以使用所需的任何模板技术。

`ContentLayout` 的 `render(params)` 方法可以使用以下代码来提供数据并呈现模板：

```
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);
  }
}
```

可以向从 `params.contentItemData` 创建的 `content` 对象添加所需数据。Oracle 建议将 `params.scsData` 中的属性合并到此对象中，以便模板也可以利用它们。应将呈现的模板附加到向 `render()` 方法传递的父对象。

编辑 Mustache 模板中的内容布局

可以编辑 `assets/layout.html` 文件中的默认 Mustache 模板，其中包含默认内容布局。

用于呈现博客标题的简单模板如下：

```
<h1>
  {{blogTitle}}
</h1>
```

添加动态 DOM 操纵

调用 `Mustache.render()` 并将模板附加到父对象后，可以将动态 DOM 操纵添加到 `render.js`。

例如，您可以连接监听程序，动态更改样式：

```
// Dynamic DOM Manipulation can be done here
$('h1').click(function (event) {
  alert('Title is : ' + $(this).text());
});
```

在 design.css 文件中定义样式

在 `design.css` 文件中，可以定义在内容布局模板中使用的任何样式。

`design.css` 文件通过使用 RequireJS CSS 插件加载到模块定义中。

获取引用项

您可以获取某种内容类型的引用项，该内容类型包含引用其他内容类型的引用数据字段。

例如，Blog-Post 内容类型中的 Author 字段是对 Author 内容类型的引用。在 Blog-Post 的内容布局中，`contentClient.getItems()` 将 Author 引用项的详细信息与当前 Blog-Post 项关联。

```
var author_id = this.contentItemData.data['starter-blog-post_author'].id;
var ids = [author_id];
// Get the author reference item
contentClient.getItems({
    'ids': ids
}).then(function (items) {
    // Use the item
}, function (error) {
    // Handle error
});
```

获取媒体 URL

您可以使用 `contentClient.getRenditionURL()` 获取数字资产（例如图像）的默认重现。

```
blogHeaderImage = contentClient.getRenditionURL({
    'id': data['blog-post_header_image'].id
}),
```

如果需要其他重现（例如 thumbnail），则可以使用 `contentClient.getItems()` 获取数字资产，然后参阅 `item.data.renditions.default` 和 `item.data.renditions.thumbnail`。

引发触发器

您可以使用 `scsData.contentTriggerFunction(payload)` 从内容布局引发触发器。

下面是在单击作者姓名时引发触发器的 Author 内容布局示例：

```
{{#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}}
```

有效负载是针对当前所选作者的搜索查询，页上的其他内容列表项可以监听它。

导航到包含搜索查询的搜索页

常见用例是在单击内容布局中的链接时导航到包含动态搜索查询的搜索页。

例如，假定您希望在单击内容布局中的 "More articles from this author"（此作者的更多文章）链接时导航到名为 "Authors"（作者）的搜索页，并传递搜索有效负载。以下代码可实现此目的。请注意，在站点页中运行时，全局对象 `SCS` 和 `SCSRenderAPI` 可用于内容布局。

```
$('.more-from-author').click($.proxy(function () {
    var childrenPages = SCS.structureMap[SCS.navigationRoot].children;
```

```

if (!childrenPages) return; // No pages

// Find the Authors page
for (var i = 0; i < childrenPages.length; i++) {
  var page = SCS.structureMap[childrenPages[i]];
  if (page.name === 'Authors') {
    var linkData = SCSRenderAPI.getPageLinkData(page.id);
    if (linkData && linkData.href) {
      var href = linkData.href,
          searchPayload = content.author_id + '*',
          contentType = "Starter-Blog-Post";
      // if both the page URL and the search query exists,
      navigate to the page passing in the query
      if (href && searchPayload) {
        var queryStart = href.indexOf('?') === -1 ? '?' :
        '&';

        // add in the contentType and search parameters
        // contentType isn't a required URL parameter
        // Payload contains search string only. No
        parameter name.
        href += queryStart + (contentType ? 'contentType='
+ contentType + '&' : '') + 'q=' + searchPayload;

        // navigate to the search results page
        window.location = href;
      }
    }
  }
}, this));

```

如果您希望在同一页中多次使用同一内容布局，最好在 CSS 选择器而不是类选择器中使用唯一 ID，例如 `$('.more-from-author').click(...)`。

例如：

```

template.html
  <div id="{{navigateId}}">...</div>

render.js
  content.navigateId = this.scsData.id + 'detailTrigger';
  $('##' + navigateId).click(...)

```

扩展宏和呈现多信息文本

内容项中的多信息文本可以嵌入数字图像。

为了在内容布局中正确呈现此多信息文本，多信息文本字段使用 `contentClient.expandMacros()` API。这解决了在多信息文本中对数字资产的所有引用。

```
data["starter-blog-post_content"] =  
  contentClient.expandMacros(data["starter-blog-post_content"]);
```

如果使用 Mustache 进行呈现，则应使用 `{{{ }}}` 呈现多信息文本值，因为多信息文本包含 HTML。在变量两边使用 `{{{ }}}` 后，Mustache 不会转义 HTML。

指向“详细信息”页的链接

“详细信息”页链接通过 `scsData.detailPageLink` 提供。

例如，如果您希望在单击博客标题时导航到“详细信息”页以显示博客详细信息，则可以按如下所示使用“详细信息”页链接：

```
<a href="{{scsData.detailPageLink}}">  
  <h2 class="post-title">  
    {{blogTitle}}  
  </h2>  
</a>
```

扩展内容列表查询中的宏

在内容列表查询中，可以为运行时计算的属性定义值，以显示最近更新的内容。

站点中的组件的大多数属性是静态的。用户可为组件的属性之一选择或输入固定字符串或值，并且无论何时或在何处运行页，该字符串或值都不会更改。但是，您可以为运行时计算的属性定义值。这可用于显示内容查询中最近更新的内容。用户可以输入“在过去 3 天”之类的日期。

您可以将 Mustache JS 扩展插入到多个属性。这些字符串中引用的值是从运行时执行的模型派生的。有一个现成的模型可处理为内容 REST API 调用格式化的日期。您可以使用其他值来扩展此模型以满足任何用户要求。

下面是可以为属性输入的字符串示例：

```
Content List component:  
  Additional Query String property:  
    updatedDate gt "{{#content.date}}today - 3 days{{/  
content.date}}"
```

将在运行时对表示日期的此 Mustache 条目求值，以便返回值根据运行时间更改（即，扩展为 `updatedDate gt "2220181002060000000"`）。这样，用户可以构建任何复杂的日期字符串，而不必输入预定义值。

支持的组件属性

以下属性支持 Mustache JS 模板语法：

- 内容列表
 - 其他查询字符串
 - 例如： `updatedDate gt "{{#content.date}}today - 3 days{{/content.date}}"`

- 标题/段落/文本
 - 通过 CKEditor 输入的多信息文本
 - 例如: "Content REST API format for date: {{#content.date}}now{{{/content.date}}}"

注:

在没有适用于 Mustache 模板的定制模型的情况下，标题/段落/文本中的扩展没什么用。但是，要验证您在其他查询字符串中输入的内容时，它很有用，因为当您在编辑和查看之间切换时会对其进行求值，并且它会立即可见。

支持的组件语法

现成支持 `content.date` 对象。此对象接受两个主要参数: `today` 和 `now`。

`today` 值采用当前浏览器时间，将其转换为今晚午夜，然后将该值转换为 UTC 时间。

- `{{#content.date}}today{{{/content.date}}` 扩展为今晚午夜的浏览器值（转换为 UTC 值并格式化为内容 REST API 日期格式）。例如:

```
2220181008065959999
```

- 然后可以采用以下方式扩充它:

```
today +/- [day | week | month | year]
```

- 此外，当您对其使用加法或减法时，`today` 值的行为会有所不同。如果使用减法，则它将使用凌晨时间。如果使用加法，则它将使用午夜时间。例如:
 - `{{#content.date}}today - 1 day{{{/content.date}}` 扩展为昨天的开始时间。
 - `{{#content.date}}today + 2 days{{{/content.date}}` 扩展为明天午夜后的那天。

`now` 值采用当前浏览器时间，并将其转换为 UTC 时间，而不进行任何调整。

- `{{#content.date}}now{{{/content.date}}` 扩展为当前浏览器时间（转换为 UTC 值并格式化为内容 REST API 日期格式）。
- 此外，还可以采用 `hour` 扩充 `now`。即:

```
now +/- [hour | day | week | month | year]
```

- 例如:
 - `{{#content.date}}now + 2 hours{{{/content.date}}` — 从现在起的两个小时（转换为 UTC 值并格式化为内容 REST API 日期格式）
 - `{{#content.date}}now - 1 day{{{/content.date}}` — 昨天的当前浏览器时间（转换为 UTC 值并格式化为内容 REST API 日期格式）

使用支持的组件语法

要在其他查询字符串中使用宏扩展，假如您要返回过去 3 周的所有内容，则输入以下内容：

```
updatedAt gt "{{#code.date}}today - 3 weeks{/code.date}}"
```

将仅返回日期，因此，要处理内容 REST API 调用，则在构造查询字符串时添加引号，就像在输入静态值时一样。

MustacheJS

有关语法，请参阅 Mustache JS 模板页，网址为 mustache.github.io/mustache.5.html。

已对扩展字符串时运行的 Mustache 实例进行了一项更改。Mustache 提供使用 `{{ }}` 的 text 扩展和使用 `{{{ }}` 的 html 扩展。这两者之间的差异是 text 扩展对字符串进行 HTML 编码；即，如果值扩展为 `a < b`，则结果将为 `a < b`。这不是您在构造 URL 的字符串时所需的。您可以告诉用户使用 HTML 扩展，但这就是一项额外工作，并且会产生更多问题，例如，解释为什么他们需要使用 `{{{ }}`。

为了避免这种情况，对 Mustache 进行了设置以便在使用 `{{ }}` 时它不会转义值。这意味着 `{{ }}` 和 `{{{ }}` 的行为相同。此外，这还将对任何结果进行编码作为练习留给用户（如果需要）。

OOTB Mustache 模型

Mustache 要求将模型应用于模板才能进行扩展。在前面的示例中，已现成定义了 `{{#content.date}}`，但开发人员可以添加新属性，例如 `{{person}}`。如果用户在 Mustache 模板中输入的值不在模型中，则结果将为空字符串。因此，对于 `Hello {{person}}`，除非开发人员将 `person` 添加到模型中，否则它将扩展为仅 `Hello`。

使用的模型对象是名为 `SCSMacros` 的全局对象。开发人员可以将任何其他条目添加到此对象中。在对模板进行求值时，该对象将传递到 Mustache。

现成的模型对象当前仅支持 `content.date` 对象：

```
{
  content: {
    date: <lambda implementation>
  }
}
```

定制 Mustache 模型

开发人员可以根据自己要求增强支持的对象。因此，他们可以引入 `lastTwoDays` 对象并将扩展简化为仅 `{{lastTwoDays}}`。

要扩展模型以支持某些内容（例如，前面示例中的 `Hello {{person}}`），需要将 `person` 对象添加到 `SCSMacros`。需要在运行页之前进行此操作。可以通过将脚本标记添加到页布局开头，在页布局中完成此操作。例如：

```
<script type="text/javascript">
window.SCSMacros = window.SCSMacros || {}; // define/get the SCSMacros
object
```

```
window.SCSMacros.person = "World";  
</script>
```

完成此更改后，Hello {{person}} 模板将扩展为：Hello World。

如果要将值传递给对象（例如，Hello {{#person}}personId{{/person}}），则需要实施 mustache lambda，并在实施中包装和扩展该值。

例如：

```
<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>
```

完成此更改后，Hello {{#person}}111{{/person}} 模板将扩展为 Hello Small，
Hello {{#person}}222{{/person}} 模板将扩展为 Hello Big。

注：

Mustache 扩展是同步执行的。如果您需要检索异步值，则需要尝试在尝试执行 Mustache 扩展之前在模型中解析这些值，尽管可以进行定制实施，但目前不支持此操作。

开发强健的内容布局

对于从内容 REST 调用获取的三种类型的响应数据，内容布局必须强健：

- 内容项，带 expand=all 参数时，包含的引用会扩展，并将会有大文本字段。
- 内容项，不带 expand=all 时，包含的引用不会扩展，但将会有大文本字段。
- 内容查询包含的引用不会扩展或没有大文本字段。

为了提高性能，自 Oracle Content Management 19.2.3 发行版起，检索资产数据的内容 REST 调用不再包含 expand=all 参数。expand 参数用于告知 Oracle Content Management 下钻并检索所有引用项以及响应中的当前项。需要更新依赖于通过 expand 参数检索的数据的定制内容布局，才能处理检索的数据不包含引用的字段值的情况。

呈现内容项

内容布局用于呈现内容项。它们接收内容项数据，将其呈现到 HTML 中，以及将其插入到页中。

尽管可以采用任何 JavaScript 技术来实施内容项，但默认情况下，内容布局利用 Mustache 模板来呈现内容项。要使 Mustache 模板呈现出来，它需要特定格式的数据。内容布局 `render.js` 文件需要确保它传递到模板的模型匹配该格式。

下面是在多个用例中呈现内容布局的情况：

- 在 Oracle Content Management 资产管理 UI 中使用时，数据可以处于“已编辑”状态，以便用户在保存更改之前预览更改。
- 在 Oracle Content Management 站点中的内容列表或内容项中使用时，使用有关运行它的站点的其他信息扩充数据。
- 通过 Content SDK 的 `contentClient.renderLayout()` 调用使用时，在此情况下 Content SDK 用户将其所需的任何数据直接传递到内容布局。

出于性能考虑，通常会在创建单个查询（可以返回所有所需数据）或多个查询（以便通过快速初始查询尽快呈现外框，然后通过后续查询填充各区域）之间进行权衡。您选择的模型取决于您的数据和用例。

此外，传递到内容布局的数据的格式可能会因数据检索方式而不同。例如，如果您将内容 REST 与扩展参数结合使用，则可能还会返回对其他内容项的字段引用（单个或成组）。如果未包括所引用的内容项，您将需要进行额外的 REST 调用。

要处理所有情况，内容布局开发人员应尝试灵活处理收到的数据的格式。此外，必要时，开发人员可以提取其他数据并将数据强制转换为呈现模板所需的格式。

标准化内容布局的数据结构

内容布局开发人员需要对内容布局接收的数据结构进行标准化。

如果所有数据都存在，则内容布局可以简单地呈现组件。如果数据不是全部存在，则内容布局可能需要进行额外查询。在所有情况下，内容布局都不应假定某种数据格式，而是将数据强制转换为将呈现的格式。

您需要确保具有预期的所有数据。如果数据不存在，则需要进行额外查询。数据中可能会缺少以下字段：

- 用于引用字段的 "fields" 条目
- 大文本字段

因为内容布局是为特定内容类型设计的，所以内容布局的开发人员知道所需字段的列表。对于这些字段中的每个字段，都需要提取数据，以便内容布局可以呈现。您有两个选项：提取缺少的数据，然后使用完整数据进行呈现；或者立即呈现，然后提取缺少的数据来填充空白。

选项 1：提取缺少的数据，然后使用完整数据进行呈现

创建 Promise 来检索所需的数据，然后在所有 Promise 返回时继续呈现。

例如，我们具有以下内容类型和相应字段：

- `starter-blog-author`
 - 字段

- * starter-blog-author_name — 文本字段
- * starter-blog-author_bio — 文本字段
- starter-blog-post
 - 字段
 - * starter-blog-post_title — 文本字段
 - * starter-blog-post_content — 大文本字段
 - * starter-blog-post_author — 对 starter-blog-author 项的引用

内容布局具有以下模板来呈现这些预期的字段值：

```

{{#fields}}
<div class="blog_container">
  <div class="blog-post-title">{{starter-blog-post_title}}</div>
  {{#starter-blog-post_author.fields}}
  <div class="blog-author-container">
    <div class="blog-author-details">
      <div class="blog-author-name">{{starter-blog-author_name}}</
div>
      <div class="blog-author-bio">{{starter-blog-author_bio}}</
div>
      <span class="more-from-author">More articles from this
author</span>
    </div>
  </div>
  {{/starter-blog-post_author.fields}}
  <div class="blog-post-content">{{starter-blog-post_content}}</div>
</div>
{{/fields}}

```

可以使用以下查询的数据调用内容布局：

- 具有 "expand" 的项查询 — 提供所有数据
 - /content/published/api/v1.1/items/{id}?expand=fields.starter-blog-post_author&channelToken=8dd714be0096ffaf0f7eb08f4ce5630f
 - 这是成功填充模板中的所有值所需的数据格式。如果使用其他查询中的任何一个，则需要额外的工作来提取数据并将其转换为此格式。
 - "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": "..."
  }
}
          
```

- 无 "expand" 的项查询 — 缺少引用的项字段 **"starter-blog-post_author.fields"**:
 - /content/published/api/v1.1/items/{id}?channelToken=8dd714be0096ffaf0f7eb08f4ce5630f
 - "fields": {
 - "starter-blog-post_title": "...",
 - "starter-blog-post_summary": "...",
 - "starter-blog-post_content": "...",
 - "starter-blog-post_author": {
 - "id": "CORE386C8733274240D0AB477C62271C2A02",
 - "type": "Starter-Blog-Author"
- SCIM 查询 — 缺少大文本字段 **"starter-blog-post_content"**, 缺少引用的项字段 **"starter-blog-post_author.fields"**:
 - /content/published/api/v1.1/items?q=(type eq "Starter-Blog-Post")&fields=ALL&channelToken=8dd714be0096ffaf0f7eb08f4ce5630f
 - "fields": {
 - "starter-blog-post_title": "...",
 - "starter-blog-post_summary": "...",
 - "starter-blog-post_author": {
 - "id": "CORE386C8733274240D0AB477C62271C2A02",
 - "type": "Starter-Blog-Author"

为了能够使用这些查询中的任何一个一致地进行呈现, 内部布局中的 `render.js` 需要确保扩展所有引用的字段并且存在大文本字段。

如果不是这样, 则需要重新查询这些项, 修复数据, 然后使用完整数据进行呈现。

`render()` 函数示例:

```
render: function (parentObj) {
  var self = this,
      template,
      contentClient = self.contentClient,
      content = self.contentItemData;

  var getRefItems = function (contentClient, ids) {
    // Calling getItem() with no "ids" returns all items.
    // If no items are requested, just return a resolved Promise.
    if (ids.length === 0) {
      return Promise.resolve({});
    } else {
      return contentClient.getItems({
        "ids": ids
      });
    }
  };

  var fetchIDs = [], // list of items to fetch
```

```
        referredFields = ['starter-blog-post_author'], // names of
reference fields
        largeTextFields = ['starter-blog-post_content'], // large text
fields in this asset
        fieldsData = content.fields;
        // See if we need to fetch any referenced fields
        referredFields.forEach(function (fieldName) {
            if(fieldsData[fieldName] && fieldsData[fieldName].fields) {
                // got data already, nothing else to do
            } else {
                // fetch this item
                fetchIDs.push(fieldsData[fieldName].id);
            }
        });

        // See if we need to fetch any large text fields
        for(var i = 0; i < largeTextFields.length; i++) {
            if(!fieldsData[largeTextFields[i]]) {
                // need to fetch this content item directly to get all the
large text fields
                fetchIDs.push(content.id);
                break;
            }
        }
        // now we have the IDs of all the content items we need to fetch,
get them all before continuing
        getRefItems(contentClient, fetchIDs).then(function (referenceData) {
            var items = referenceData && referenceData.items || [];

            // add the data back in
            items.forEach(function (referencedItem){
                // check if it's the current item
                if(referencedItem.id === content.id) {
                    // copy across the large text fields
                    largeTextFields.forEach(function (fieldName) {
                        fieldsData[fieldName] =
referencedItem.fields[fieldName];
                    });
                } else{
                    // check for any referenced fields
                    for (var i = 0; i < referredFields.length; i++) {
                        if(referencedItem.id ===
fieldsData[referredFields[i]].id){
                            // copy across the fields values
                            fieldsData[referredFields[i]].fields =
referencedItem.fields;
                            break;
                        }
                    }
                }
            });

            // now data is fixed up, we can continue as before
            try{
                // Mustache
```

```
        template = Mustache.render(templateHtml, content);

        if(template) {
            $(parentObj).append(template);
        }

    } catch (e) {
        console.error(e.stack);
    }
    });
}
```

选项 2：立即呈现，然后提取缺少的数据来填充空白

通过分离出可能不存在的项并在第二轮中呈现它们，可以提高性能。这将需要两个 Mustache 模板，第一个进行初始呈现，留下“洞”，然后在数据完整时通过第二次呈现来填充这些洞。

这需要设置 Mustache 模板以支持多轮次：针对“洞”具有单独的模板或者具有模型返回模板宏，而不是实际值。在任一情况下，您都需要“隐藏”这些洞，直到检索数据，然后填充这些洞并使用适当的 UI 动画显示它们，以避免页“跳跃”过多。

创建示例博客模板

BlogTemplate 示例演示了内容布局功能。

可以在开发人员云服务中创建模板，检查模板中的内容布局以及测试搜索功能：

1. 创建类型为 BlogTemplate 的模板：

```
cec create-template MyBlogTemplate -f BlogTemplate
```

将通过复制 BlogTemplate 创建新模板 MyBlogTemplate。

2. 打开 <http://localhost:8085/>，然后依次单击模板和 **MyBlogTemplate**。
3. 您将看到博客帖子列表。单击其中一个。您将转到“详细信息”页，该页使用三个不同的内容布局来呈现。
4. 单击 **More articles from this author**（此作者的更多文章）。您将转到搜索页。
5. 单击作者姓名。该内容布局将引发触发器，在右侧显示文章。
6. 再次转至主页并测试搜索。

将内容布局映射添加到模板

创建内容布局后，您可以通过单击内容布局映射将其添加到开发人员云服务项目中的本地模板。

可使用 `cec add-contentlayout-mapping` 命令为模板添加内容布局映射。需要提供内容布局基于的内容类型 (-c) 以及映射适用的模板 (-t)。例如：

```
cec add-contentlayout-mapping Blog-Post-Detail-Layout -c Blog-Post -t
BlogTemplate
```

默认内容布局映射是桌面的默认样式。您可以指定 `-s <layoutstyle>` 选项以按名称指定其他布局样式，例如 `Overview` 或 `Details`。

```
cec add-contentlayout-mapping Blog-Post-Detail-Layout -c Blog-Post -t
BlogTemplate -s Details
```

此外，您还可以使用 `-m` 选项为移动设备设置映射：

```
cec add-contentlayout-mapping Blog-Post-Detail-Layout -c Blog-Post -t
BlogTemplate -m
```

使用本地测试工具测试内容布局

将内容类型和内容布局映射添加到模板后，可以在本地测试工具中测试内容布局。

请参见[使用本地测试工具进行测试](#)。

使用本地测试工具进行测试

在将定制组件、模板和内容布局导入到 Oracle Content Management 之前，先在本地测试工具中运行它们。

要启动本地测试工具，请执行以下操作：

1. 在终端窗口中输入 `cd cec`。
2. 输入 `cec develop &` 或 `cec develop --server <server-name> &`
3. 打开浏览器并访问 `http://localhost:8085`，以查看本地测试工具中运行的组件、模板和内容布局。
4. 您可以在以下目录中查找组件、模板、主题等：
 - `cec/src/main/components`
 - `cec/src/main/templates`
 - `cec/src/main/themes`

将具有内容布局的模板导入到 Oracle Content Management 中

开发并测试内容布局后，可以从开发人员云服务项目导出包含内容布局的模板，然后将该模板导入到 Oracle Content Management 中。

请参见[导出模板](#)。

23

开发组件

可以开发用于 Oracle Content Management 站点的组件。

- [关于组件](#)
- [关于开发组件](#)
- [创建组件](#)
- [通过开发人员云服务开发定制组件](#)
- [为多语言站点开发可翻译组件](#)
- [构建包含“设置”面板的 H1 组件](#)
- [使用 HTML 模板创建简单的组件](#)
- [比较本地组件与远程组件](#)
- [呈现组件设置](#)
- [本地组件实施](#)
- [组件的样式类](#)
- [如何为内置组件设置样式](#)
- [设置组件属性](#)
- [在内嵌框架中呈现的组件](#)
- [关于在内嵌框架中呈现的组件的实例 ID 和结构](#)
- [远程组件的安全性](#)
- [注册远程组件](#)
- [删除组件](#)
- [Sites SDK](#)

关于组件

组件是可以添加到站点中页上的特定类型的内容。Oracle Content Management 中的组件包括段落、标题、图像和分隔线等项。

Oracle Content Management 支持以下类型的组件：

- 本地组件：文件存储在 Oracle Content Management 中
- 远程组件：文件存储在远程服务器上

本地组件可以设置为直接在页中呈现，也可以设置为在页中的内嵌框架中呈现。远程组件始终在内嵌框架中呈现。

Oracle Content Management 提供了一组包含各自模板（每个模板还包含主题和站点）的默认组件。您可以创建新组件，也可以在您的组件中使用这些默认组件。向站点添加了组件后，您

可以编辑组件的属性设置以满足您的要求，方法是指定页内容、字体和字号、图像框架和位置以及其他样式。您可以编辑的设置取决于组件类型。

这些组件随 Oracle Content Management 提供。

名称	类型	ID
标题	scs-title	scs-title
段落	scs-paragraph	scs-paragraph
文本	scs-text	scs-text
图像	scs-image	scs-image
图库	scs-gallery	scs-gallery
图库网格	scs-gallerygrid	scs-gallerygrid
文档	scs-document	scs-document
按钮	scs-button	scs-button
地图	scs-map	scs-map
分隔线	scs-divider	scs-divider
分隔符	scs-spacer	scs-spacer
YouTube	scs-youtube	scs-youtube
社交栏	scs-socialbar	scs-socialbar
文章 (定制组件)	scs-component	scs-comp-article
标题 (定制组件)	scs-component	scs-comp-headline
文件夹列表	scs-app	文件夹列表
文件列表	scs-app	文件列表
文档管理器	scs-app	文档管理器
Facebook 点赞	scs-app	Facebook 点赞
Twitter 关注	scs-app	Twitter 关注
Twitter 共享	scs-app	Twitter 共享
Facebook 推荐	scs-app	Facebook 推荐
图像和文本 (定制组件)	scs-component	scs-comp-image-text

请参见《通过 Oracle Content Management 构建站点》中的“排列页面内容”。

关于开发组件

通过开发自己的定制组件，您可以使用任何选定的页技术，开发可嵌入到站点页中的复合元素。实际上，您可以通过这种方式扩展随 Oracle Content Management 提供的组件列表。

所有已注册组件的条目都存储在组件目录中，此目录是 Oracle Content Management 中的一个文件夹，包含所有已注册组件的条目。

不要将以下名称用于模板、主题、组件、站点或站点页：authsite、content、pages、scs-template_*、_comps、_components、_compsdelivery、_idcservice、_sitescloud、_sitesclouddelivery、_themes、_themesdelivery。下面的名称虽然可用于站点页，但不要将其用于模板、主题、组件或站点：documents、sites。

组件类型

组件目录支持以下类型的组件：

- 本地组件
- 在内嵌框架中呈现的本地组件
- 远程组件

组件类型作为组件文件夹的扩展属性 "xScsAppType" 存储。有效值如下。

类型	说明
本地组件	所有相关项均应为本地相关项。
在内嵌框架中呈现的本地组件	与 Oracle Content Management 相同的域提供组件。
远程组件	从远程位置提供组件。

组件文件结构

根据实施方式，在组件目录中创建时，每种类型的组件都有一组不同的文件。

本地组件：

```
/Components/component-name
  appinfo.json
  _folder_icon.jpg
  assets
    settings.html
  render.js
```

使用内嵌框架的本地组件：

```
/Components/component-name
  appinfo.json
  _folder_icon.jpg
  assets
    settings.html
  render.js
  js
    sites.min.js
    knockout.min.js
    jquery.min.js
```

远程组件：

```
/Components/component-name
  appinfo.json
  _folder_icon.jpg
  keys.json
```

文件夹元数据

组件注册数据存储存储在文件夹元数据中。以下属性用于唯一标识组件及其类型。

属性	说明
应用程序名称	指定组件名称的文件夹的名称。
应用程序说明	文件夹的说明。
应用程序 GUID	每个组件都与一个 GUID 关联，并作为扩展属性 xScsItemGUID 存储。GUID 是在创建组件时由服务器生成的。
应用程序类型	此属性指定组件类型。它作为扩展属性 xScsApType 存储。

 **注：**

已废弃作为扩展属性 xScsAppIconUrl 存储的属性 imageUrl。

appinfo.json 文件

每种类型的组件的 appinfo.json 注册文件仅包含文件夹元数据中不可用的数据。在文件夹元数据中定义的组件属性不会复制到 appinfo.json 文件。

本地组件：

```
{
  "settingsData": {
    "settingsHeight": 80,
    "settingsRenderOption": "dialog",
    "settingsWidth": 300,
    "componentLayouts": [],
    "triggers": [],
    "actions": []
  },
  "initialData": {
    "customSettingsData": []
  }
}
```

在内嵌框架中呈现的本地组件：

```
{
  "endpoints": {
    "settings": {
      "height": "300",
      "width": "400"
    }
  }
  "initialData": {
    "customSettingsData": {}
  }
}
```

远程组件：

```
{
  "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": {}
}
}
```

本地组件没有注册的 "url" 值。它们使用特定文件，这些文件的内容可以编辑，但不能更改位置或名称。

- 本地组件使用文件 `assets/render.js` 和 `assets/settings.html`。
- 在内嵌框架中呈现的本地组件使用文件 `assets/render.html` 和 `assets/settings.html`。
- 远程组件使用指定的任何 "url" 值。

创建组件

可以创建用于 Oracle Content Management 的定制组件。

虽然 Oracle Content Management 提供了许多预定义的组件以用于构建网站，但作为开发人员，您可以创建定制组件来访问相同特性和功能。这些组件类型包括本地组件、内嵌框架中呈现的本地组件以及内嵌框架中呈现的远程组件。

本地组件

创建本地组件时，系统将向您提供具有全部功能的示例组件，您可以此为基础来创建自己的组件。选择默认值以创建 Knockout 版本的组件，或者选择模板以创建基于 Mustache 的版本。请参见《通过 Oracle Content Management 构建站点》中的“创建本地组件或布局”。

采用内嵌框架的本地组件

创建采用内嵌框架的本地组件时，系统将向您提供具有全部功能的示例组件，您可以此为基础来创建自己的组件。您可以按照创建本地组件的说明进行操作，还可以选择沙盒选项以创建内嵌框架版本的组件（存储在本地）。请参见[在内嵌框架中呈现的组件](#)。

远程组件

当创建使用内嵌框架的远程组件时，请选择沙盒选项。

将创建的文件复制到远程服务器，然后注册远程组件。先测试该组件，然后再将其包含在已发布站点中。

请参见[在内嵌框架中呈现的组件](#)和[注册远程组件](#)。

通过开发人员云服务开发定制组件

您可以通过 Oracle Developer Cloud Service 为 Oracle Content Management 开发模板、主题和定制组件。

包含 Oracle Content Management 的开发人员云服务集成提供了一个模板，其中包含可用于开发模板和组件的工具。此外，它还提供了示例单元测试以供开始使用。该集成包含一个 Git 资料档案库和多个工具（可帮助开发模板和组件），以及一个可用于快速迭代开发模板、主题和定制组件的本地测试工具。

借助开发人员云服务可以执行以下任务：

- 设置本地开发环境以使用 Oracle Content Management 实例在本地开发和测试模板、主题和组件
- 基于示例或入门级文件创建模板和组件，在测试工具中运行它们，浏览它们，以及在开发人员云服务环境中开发模板、主题和组件
- 将从 Oracle Content Management 创建的模板或组件导入到开发人员云服务环境以进行源管理和进一步开发
- 从开发人员云服务环境导出要导入到 Oracle Content Management 中的模板或组件，以在网站中使用
- 复制现有组件
- 编写单元测试
- 优化组件
- 将组件部署到 Oracle Content Management

以下主题介绍了如何使用开发人员云服务的集成为 Oracle Content Management 进行开发：

- [为 Oracle Content Management 开发定制组件](#)
- [优化组件（缩小）以提高性能](#)
- [运行连续集成作业](#)

为 Oracle Content Management 开发定制组件

可使用开发人员云服务和本地计算机为 Oracle Content Management 开发定制组件。

以下主题介绍了通过开发人员云服务开发和测试 Oracle Content Management 组件的步骤：

1. [在本地计算机上设置 OCE 工具包。](#)
2. [登录 Oracle Content Management 的开发人员云服务控制台。](#)
3. [在开发人员云服务中创建项目。](#)
4. [将 OCE 工具包添加到新的 Git 资料档案库中的项目代码。](#)
5. [开发定制组件](#)
6. [使用本地测试工具进行测试](#)
7. [编写并运行单元测试](#)

开发定制组件

可使用 `cec` 命令行实用程序在本地创建新组件和开发组件，然后将组件导出到 Oracle Content Management。

创建组件

要在本地创建和开发定制组件，请使用以下命令：

```
cec create-component <component-name> -f <source>
```

为 `source` 选择以下值之一：

- 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

示例：

```
cec create-component MyLocalComponent1 -f local
```

将在 `cec-components/src/main/components` 下的 Git 资料档案库中创建组件。

`src/main/components` 目录中将植入 `Sample-To-Do` 组件。您创建的所有组件都将存放此目录中。

复制组件

可以使用 `cec copy-component` 命令在开发人员云服务中复制组件：

```
cec copy-component <source> [<destination>]
```

此命令将名为 `<source>` 的现有组件复制到 `<destination>`。

将组件导出或部署到 Oracle Content Management

在本地服务器中开发并测试了组件后，可以使用以下命令导出该组件。这将创建组件 zip 文件。可以手动将此组件 zip 导入到 Oracle Content Management 中。

```
cec export-component <component name>
```

或者，可以使用以下命令从开发人员云服务将组件直接部署到 Oracle Content Management。

```
cec deploy <component name>
```

cec deployAll 命令将所有组件部署在 `src/main/components` 中。

将组件导入到开发人员云服务中

如果您有从 Oracle Content Management 服务器创建的组件 ZIP 文件，您可以将该文件导入到开发人员云服务中以便进行进一步开发。请使用以下命令：

```
cec import-component <location of the component zip file>
```

❗ 重要：

组件的源代码位于 `src/main/components`。不应修改 `src/main/components` 以外的任何文件，因为 Oracle Content Management 本地服务器正常运行需要它们。

编写并运行单元测试

针对您的定制 Oracle Content Management 组件，先使用示例单元测试来编写单元测试，然后运行它们。

开始使用示例单元测试

适用于 Oracle Content Management 的开发人员云服务示例包括以下可帮助您编写单元测试的文件：

- **src/test/unit**：包含适用于 `Sample-To-Do` 组件的单元测试。它用作为组件 JavaScript 代码编写单元测试的示例，其中包括 RequireJS 模块。使用了适用于 JavaScript 单元测试的 Mocha 和 Chai 框架。
- **index.html**：运行单元测试。此文件加载 Mocha、Chai 和主测试文件 `test-main.js`。
- **test-main.js**：加载单元测试模块 `Sample-To-Do-Test`，并运行 Mocha 单元测试。
- **Sample-To-Do-Test.js**：
 - 定义实际测试。此文件使用 RequireJS 加载组件源代码 `components/Sample-To-Do/assets/render.js`。
 - 测试 `add()`、`delete()`、`title()` 和 `placeholder()` 方法。

为组件编写单元测试

要为定制组件编写每个单元测试，请执行以下步骤：

1. 编写类似于 `Sample-To-Do-Test.js` 的单元测试。
2. 将编写的测试加载到 `test-main.js` 中。

在本地运行单元测试

要在本地运行单元测试，请执行以下步骤：

1. 在本地克隆 Git 资料档案库。
2. 在本地计算机上的终端窗口中，输入 `npm install`。
3. 输入 `npm start`。
4. 输入 `npm test` 或在浏览器中打开 `http://localhost:8085/unit/`。

优化组件（缩小）以提高性能

通过开发人员云服务缩小 JavaScript 代码，可提高组件性能并减小下载大小。

优化组件

针对 `Sample-Text-With-Image` 组件提供了一个优化组件源代码示例（可选）。可以使用 RequireJS 优化程序缩小和组合源代码。缩小组件会删除 JavaScript 代码中不需要的空格，从而提高性能并减小下载大小。优化涉及以下操作：

- 缩小 JavaScript。
- 压缩 CSS。
- 将 JavaScript、HTML 和 CSS 组合到单个文件 `render.js` 中。

此优化可减小下载大小并提高性能。

为组件启用优化

在 `Sample-Text-With-Image` 组件中，优化会缩小 `render.js` 中标记为相关项的所有文件（`template.html`、`data-defaults.js` 和 `design.css`）并将其组合到 `render.js` 中作为单个文件。已支持优化的其他组件如下：

- `Sample-Folder-List`
- `Sample-File-List`
- `Sample-Documents-Manager`
- `Sample-Process-Start-Form`
- `Sample-Process-Task-List`
- `Sample-Process-Task-Details`

要为本地组件启用优化，请执行以下步骤（必需）：

1. 将 `gulpfile.js` 从 `Sample-Text-With-Image` 组件复制到您的组件。
2. 在 `gulpfile.js` 中修复您的 `render.js` 相关项对应的模块名称。请参阅 `gulpfile.js` 中的注释。

 注:

如果您有现有组件，且 `assets` 下有 `css` 文件夹，请将该 `css` 文件夹重命名为 `styles` 以避免在优化期间出错。

运行连续集成作业

您可以运行连续集成作业以使站点保持最新。

请参阅开发人员云服务中的“CEC 定制组件开发入门”Wiki。可以为连续集成配置构建作业。

为多语言站点开发可翻译组件

定制组件的开发人员可以指定定制组件内的哪些字符串应参与多语言 (MLS) 站点的页翻译。

为此，在将数据保存到 `customSettingsData` 对象时，您需要在顶级引入 `nls` 属性。

例如：

```
SitesSDK.setProperty('customSettingsData', {  
  'nls': {  
    linkText: 'More...'  
  }  
});
```

创建翻译作业后，Oracle Content Management 将检查 `customSettingsData` 对象的顶级属性并导出页上每个定制组件实例的整个 `nls` 对象。翻译员将翻译这些值，然后可将其重新导入站点。

导入站点翻译后，将在翻译的区域设置的 `customSettingsData` 对象中返回 `nls` 对象的正确版本。

例如，如果您将站点翻译为法语，然后在法语区域设置中呈现页，则传递到定制组件的 `customSettingsData` 对象中的 `nls` 对象值将更新为：

```
{  
  'nls': {  
    'linkText': 'Plus...'  
  }  
},
```

`customSettingsData` 中的 `nls` 对象的格式应限制为名称/值对。这样有助于翻译，并确保在翻译的区域设置中呈现页时翻译的值可以正确应用于站点中的基本值。

构建包含“设置”面板的 H1 组件

您可以创建具有简单 HTML 模板和 CSS 的最小 Oracle Content Management 组件。此 H1 组件具有简单的“设置”面板，并且在 `design.json` 中具有主题对应的条目，以允许

其他 Oracle Content Management 用户在编辑器中使用组件时从三个内置样式中进行选取。

当您创建新组件时，将会获得一组现成可用的植入文件。这些植入文件涵盖了产品中某个组件的大多数功能。您可以更改植入代码以创建自己的组件，这仅需要一小部分的植入代码即可实现最终结果。

可以采用五个步骤构建包含“设置”面板的 H1 组件：

1. [创建新的本地组件](#)
2. [构建基本 H1 组件](#)
3. [为组件添加 CSS](#)
4. [添加“设置”面板以更改标题文本](#)
5. [更新主题以便其他人选取 H1 组件样式](#)

创建新的本地组件

可以通过 Oracle Content Management 创建一个立即就可以放到页面上的本地组件。这是创建任何新组件的起点。

要创建新的本地组件，请执行以下操作：

1. 单击侧面导航中的开发人员，然后单击查看所有组件。
2. 选择创建 > 创建本地组件。
3. 输入名称（例如 H1_Component）和（可选）说明。
4. 单击创建以创建新组件。

现在，您已成功创建组件，在组件页上的组件列表中以及您创建的任何站点的添加 > 定制组件选项板中，应该可看到此组件。可使用以下步骤验证组件创建情况（检查点 1）：

1. 使用任何植入模板创建新站点，例如，使用 StarterTemplate 模板创建名为 ComponentTest 的站点。
2. 选择编辑选项，然后为站点创建更新以在编辑器中打开它。
3. 在创建的站点中编辑页。
4. 单击左侧栏中的添加按钮 (+)，然后选择定制以显示定制组件列表。
5. 从定制组件选项板中选择 **H1_Component** 并将其放在页上。

现在应该会看到创建的本地组件的默认呈现效果。

6. 选择组件的上下文菜单。
7. 从下拉菜单中选择设置。

您可以更改设置以了解植入组件呈现的变化方式。

您可以修改植入文件以创建新的定制组件。

构建基本 H1 组件

您可以删除植入文件中的大部分内容来创建 H1 组件。它显示您在创建 viewModel 时植入的标题文本。以后可以为该组件提供设置和样式。

要查看本地组件的结构，请执行以下操作：

1. 使用 Oracle Content Management 桌面同步应用程序，找到组件并将其与文件系统同步。

- 在最新版本的桌面同步应用程序中，选择开始同步或选择要同步的文件夹选项。
 - 如果您没有桌面同步应用程序，您可以在 Oracle Content Management 的组件选项卡上选择组件，然后下钻以查看文件。
2. 如果列出组件下的文件，则可以看到以下文件：
- assets 文件夹中的组件文件：
 - render.js
 - settings.html
 - appinfo.json：包含组件说明的 JSON 文件。
请参见[关于开发组件](#)。
 - folder_icon.jpg：显示在组件目录中的图标。

要构建 H1 组件，请执行以下操作：

1. 打开 appinfo.json 文件，并将其内容替换为以下行：

```
{
  "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. 在您常用的文本编辑器中打开 assets 文件夹中的 render.js 文件。
3. 将 render.js 的内容更改为以下行：

```
/* globals define */
define(['knockout', 'jquery', 'text!./render.html'],
function(ko, $, template) {
  'use strict';
  // -----
  // Define a Knockout ViewModel for your template
  // -----
  var SampleComponentViewModel = function(args) {
    var SitesSDK = args.SitesSDK;
    // create the observables -- this allows updated settings to
    automatically update the HTML on the page
    this.headingText = ko.observable();
```

```
//
// Handle property changes from the Settings panel
//
this.updateCustomSettingsData = $.proxy(function(customData) {
    this.headingText(customData && customData.headingText);
}, this);
this.updateSettings = function(settings) {
    if (settings.property === 'customSettingsData') {
        this.updateCustomSettingsData(settings.value);
    }
};
// Register your updateSettings listener to receive SETTINGS_UPDATED
events

SitesSDK.subscribe(SitesSDK.MESSAGE_TYPES.SETTINGS_UPDATED, $.proxy(this.u
pdateSettings, this));

//
// Get the initial settings data for the component and apply it
//
SitesSDK.getProperty('customSettingsData',
this.updateCustomSettingsData);
};

// -----
// Create a knockout based component implementation
// -----
var SampleComponentImpl = function(args) {
    // Initialize the custom component
    this.init(args);
};
// initialize all the values within the component from the given
argument values
SampleComponentImpl.prototype.init = function(args) {
    this.createViewModel(args);
    this.createTemplate(args);
    this.setupCallbacks();
};
// create the viewModel from the initial values
SampleComponentImpl.prototype.createViewModel = function(args) {
    // create the viewModel
    this.viewModel = new SampleComponentViewModel(args);
};
// create the template based on the initial values
SampleComponentImpl.prototype.createTemplate = function(args) {
    // create a unique ID for the div to add, this will be passed to the
callback
    this.contentId = args.id + '_content_' + args.viewMode;
    // create a hidden custom component template that can be added to the
DOM
    this.template = '<div id="' + this.contentId + '">' +
        template +
        '</div>';
```

```
};  
//  
// SDK Callbacks  
// setup the callbacks expected by the SDK API  
//  
SampleComponentImpl.prototype.setupCallbacks = function() {  
  //  
  // callback - render: add the component into the page  
  //  
  this.render = $.proxy(function(container) {  
    var $container = $(container);  
    // add the custom component template to the DOM  
    $container.append(this.template);  
    // apply the bindings  
    ko.applyBindings(this.viewModel, $('#' + this.contentId)[0]);  
  }, this);  
  //  
  // callback - dispose: cleanup after component when it is  
  // removed from the page  
  //  
  this.dispose = $.proxy(function() {  
    // nothing required for this sample since knockout disposal  
    // will automatically clean up the node  
  }, this);  
};  
// -----  
// Create the factory object for your component  
// -----  
var sampleComponentFactory = {  
  createComponent: function(args, callback) {  
    // return a new instance of the component  
    return callback(new SampleComponentImpl(args));  
  }  
};  
return sampleComponentFactory;  
});
```

4. 在 `assets` 文件夹中，创建新文件 `render.html` 以作为组件的简单 HTML 模板。
5. 在 `render.html` 文件中使用以下内容：

```
<h1 data-bind="text: headingText()">  
</h1>
```

组件 `assets` 文件夹现在包含三个文件。

- `render.html`
- `render.js`
- `settings.html`

将新的 H1 组件添加到页中（检查点 2）。

为组件添加 CSS

可以添加 CSS，以为您的组件提供默认样式。

要添加 CSS，请执行以下操作：

1. 将 `design.css` 文件添加到组件的 `assets` 文件夹，包含以下内容：

```
.h1-component-default-style .scs-component-content {
  font-family: "Helvetica Neue", "Helvetica", "Arial", sans-serif;
  font-size: 24px;
  color:red;
  font-weight: normal; }
```

2. 添加到 `appinfo.json` 以声明将用于设置组件样式的样式类前缀。如果添加 `h1-component` 的 `styleClassName`，则将组件放到页上时，默认样式将为 `h1-component-default-style`。 `appinfo.json` 的新内容如下：

```
{
  "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. 通过将第 2 行 (`define(['knockout', 'jquery', 'text!./render.html'], function(ko, $, template) {})` 替换为以下行，更新 `render.js` 来加载 CSS 文件。

```
define(['knockout', 'jquery', 'text!./render.html', 'css!./design.css'],
function(ko, $, template, css) {
```

注：

由于更改了 `appinfo.json` 文件，因此必须刷新浏览器并将组件重新添加到页以获取更改。

验证您的组件现在是否会从 `appinfo.json` 文件获取其默认数据。

添加“设置”面板以更改标题文本

更新 `settings.html` 文件来提供可用于设置 H1 组件文本的设置面板。

要添加设置面板以更改标题文本，请执行以下操作：

1. 更新 `settings.html` 文件以具有以下内容：

```
<!DOCTYPE html>
<html lang="en">

<head>
  <!-- only allow embedding of this iFrame in SCS -->
  <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
  <title>H1 Component</title>
  <!-- include sample apps styling -->
  <link href="/_sitescloud/renderer/app/sdk/css/app-styles.css"
rel="stylesheet">
  <!-- include supporting files -->
  <script type="text/javascript" src="/_sitescloud/renderer/app/
apps/js/knockout.min.js"></script>
  <script type="text/javascript" src="/_sitescloud/renderer/app/
apps/js/jquery.min.js"></script>
  <!-- include the Sites SDK -->
  <script type="text/javascript" src="/_sitescloud/
renderer/app/sdk/js/sites.min.js"></script>
</head>

<body data-bind="visible: true" style="display:none; margin:0px;
padding:0px;background:transparent;background-image:none;">
  <div class="scs-component-settings">
    <div>
      <!-- Heading Text -->
      <label id="headingTextLabel" for="headingText"
class="settings-heading" data-bind="text: 'Heading Text'"></label>
      <input id="headingText" data-bind="value: headingText"
placeholder="Heading" class="settings-text-box">
    </div>
  </div>
  <script type="text/javascript">
// define the viewModel object
var SettingsViewModel = function() {
  var self = this;

  // create the observables for passing data
  self.headingText = ko.observable();

  // create rest of viewModel
  self.saveData = false;

  // Get custom settings
  SitesSDK.getProperty('customSettingsData', function(data) {
    // update observable
```

```
self.headingText(data.headingText);

// now that viewModel is initialized and can start saving data
self.saveData = true;
});

// save whenever any updates occur, but don't save if data hasn't
been loaded into the form yet
self.save = ko.computed(function() {
    var saveconfig = {
        'headingText': self.headingText()
    };
    // save data in page
    if (self.saveData) {
        SitesSDK.setProperty('customSettingsData', saveconfig);
    }
}, self);
});

// apply the bindings
ko.applyBindings(new SettingsViewModel());
</script>
</body>
```

2. 在站点构建器中选择您的组件，然后单击设置。

设置面板很小，设置为嵌入在组件设置面板的顶部。查找标题为“Heading Text”的部分。验证您是否可以更改标题文本（检查点 4）：

您更改设置面板中的标题文本时，组件将更新以显示新文本。

更新主题以便其他人选取 H1 组件样式

您可以在主题中为组件注册样式，以便其他人可以从“设置”面板或样式选项卡在您为组件提供的样式之间切换。

要更新主题以便其他用户选取组件样式，请执行以下操作：

1. 向组件 `design.css` 文件再添加一些样式。在每个样式前面添加组件的已注册 `styleClassName`（在 `appinfo.json` 中定义）作为前缀。对于此组件，前缀为 `h1-component`。

已添加了另外两个样式：`h1-component-gothic-style` 和 `h1-component-courier-style`。

`design.css` 的新内容现在如下所示：

```
.h1-component-default-style .scs-component-content {
font-family: "Helvetica Neue", "Helvetica", "Arial", sans-serif;
font-size: 24px;
color:red;
font-weight: normal; }
.h1-component-gothic-style .scs-component-content {
font-family: "Century Gothic","CenturyGothic","AppleGothic",sans-serif;
font-size: 32px;
font-weight: bold; }
```

```
.h1-component-courier-style .scs-component-content {
  font-family: "Courier";
  font-size: 32px;
  font-weight: bold; }
```

2. 在主题的 `design.json` 文件中注册样式。可以在站点当前使用的主题中查找此文件。下钻到 `designs` 文件夹中的主题文件，再下钻到 `defaults` 文件夹，然后向 `design.json` 添加有关组件的部分。

下文的粗体文本是要添加的内容示例：

```

    "news-article": {
      "styles": [{
        "name": "News Article 1",
        "class": "news-article-
default-style"
      },
      {
        "name": "News Article 2",
        "class": "news-article-
style-1"
      }
    ]
  },
  "h1-component": {
    "styles": [{
      "name": "Plain",
      "class": "h1-component-default-style"
    }, {
      "name": "Courier",
      "class": "h1-component-courier-style"
    }, { "name": "Gothic",
      "class": "h1-component-gothic-style"
    }
  ]
}
},
"componentIcons": {
  "scs-socialbar": {
    "icons": [
```

`design.json` 片段中添加的 `name` ("Plain", "Courier", "Gothic") 将显示在样式选项卡上组件的“设置”面板中，如下所示。选择后，它们将在组件上分别应用对应的样式 ("`h1-component-default-style`", "`h1-component-courier-style`", "`h1-component-gothic-style`")。

使用 HTML 模板创建简单的组件

您可以创建这样一个 Oracle Content Management 组件：它将 HTML 和 CSS 与 Mustache JS 模板结合使用，以根据您在“设置”面板上输入的数据来呈现其内容。

您提供的“设置”面板可以允许编辑将在 HTML 模板中呈现的数据。

此过程的示例使用 Mustache JS 模板。

可以使用 HTML 模板创建简单的组件，过程分两步：

1. [创建 HTML 组件](#)
2. [将用户提供的数据添加到 HTML 组件](#)

创建 HTML 组件

可以在 Oracle Content Management 组件目录中创建新的本地组件，然后通过在其 assets 文件夹中添加和更改文件来构建 HTML 组件。

要创建和构建 HTML 组件，请执行以下操作：

1. 创建组件并将其命名为 HTML Component。
请参见[创建新的本地组件](#)。
2. 将 `mustache.min.js` 文件添加到 HTML 组件的 `assets` 文件夹。
3. 在 `assets` 文件夹中创建名为 `render.html` 的新文件，其包含组件正文的 HTML。对于此示例，使用以下内容创建 `render.html` 文件：

```
<ul class="wrapper">
  <li class="box">
    <h1 class="title">One</h1>
    <p class="text"> Lorem ipsum dolor sit amet, consectetur adipiscing
elit, sed do eiusmod tempor incididunt ut labore et dolore magna
aliqua.</p>
  </li>
  <li class="box">
    <h1 class="title">Two</h1>
    <p class="text"> Lorem ipsum dolor sit amet, consectetur adipiscing
elit, sed do eiusmod tempor incididunt ut labore et dolore magna
aliqua.</p>
  </li>
  <li class="box">
    <h1 class="title">Three</h1>
    <p class="text">Lorem ipsum dolor sit amet, consectetur adipiscing
elit, sed do eiusmod tempor incididunt ut labore et dolore magna
aliqua.</p>
  </li>
  <li class="box">
    <h1 class="title">Four</h1>
    <p class="text">Lorem ipsum dolor sit amet, consectetur adipiscing
elit, sed do eiusmod tempor incididunt ut labore et dolore magna
aliqua.</p>
  </li>
</ul>
```

4. 在 `assets` 文件夹中创建名为 `design.css` 的新文件，其是组件的 CSS。对于此示例，添加以下行作为 `design.css` 文件的内容：

```
.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. 打开 `assets` 文件夹中的 `render.js` 文件，并将内容更改为以下内容。无论您在前面步骤中使用了什么 HTML 和 CSS，以下 `render.js` 文件都会为您将 HTML 和 CSS 呈现到页中：

```
/* globals define */
define(['jquery', './mustache.min', 'text!./render.html', 'css!./
design.css'], function($, Mustache, template, css) {
  'use strict';

  // -----
  // Create a Mustache-based component implementation
  // -----
  var SampleComponentImpl = function(args) {
    this.SitesSDK = args.SitesSDK;

    // Initialize the custom component
    this.createTemplate(args);
    this.setupCallbacks();
  };
  // create the template based on the initial values
  SampleComponentImpl.prototype.createTemplate = function(args) {
    // create a unique ID for the div to add, this will be passed
to the callback
    this.contentId = args.id + '_content_' + args.viewMode;
    // create a hidden custom component template that can be added
to the DOM
    this.template = '<div id="' + this.contentid + '">' +
      template +
      '</div>';
```

```
};
SampleComponentImpl.prototype.updateSettings = function(settings) {
  if (settings.property === 'customSettingsData') {
    this.update(settings.value);
  }
};
SampleComponentImpl.prototype.update = function(data) {
  this.data = data;
  this.container.html(Mustache.to_html(this.template, this.data));
};
//
// SDK Callbacks
// setup the callbacks expected by the SDK API
//
SampleComponentImpl.prototype.setupCallbacks = function() {
  //
  // callback - render: add the component into the page
  //
  this.render = $.proxy(function(container) {
    this.container = $(container);

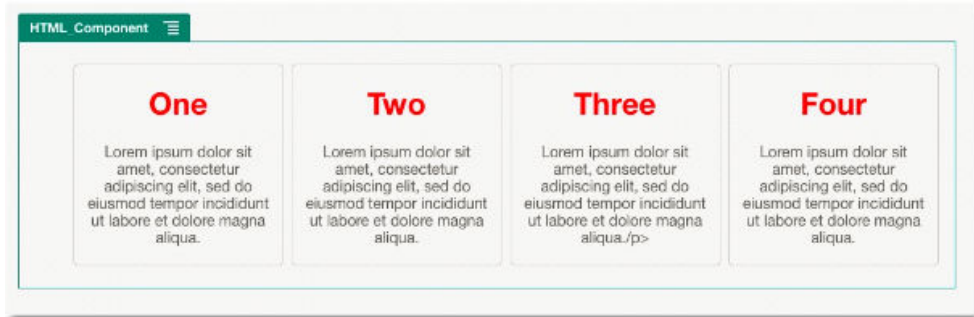
    this.SitesSDK.getProperty('customSettingsData', $.proxy(this.update,
    this));
  }, this);
  //
  // callback - SETTINGS_UPDATED: retrieve new custom data and re-render
  the component
  //
  this.SitesSDK.subscribe(this.SitesSDK.MESSAGE_TYPES.SETTINGS_UPDATED, $.pr
  oxy(this.updateSettings, this));
  //
  // callback - dispose: cleanup after component when it is removed
  from the page
  //
  this.dispose = $.proxy(function() {
    // nothing required
  }, this);
};
// -----
// Create the factory object for your component
// -----
var sampleComponentFactory = {
  createComponent: function(args, callback) {
    // return a new instance of the component
    return callback(new SampleComponentImpl(args));
  }
};
return sampleComponentFactory;
});
```

要验证是否已构建了 HTML 组件（检查点 1），请执行以下操作：

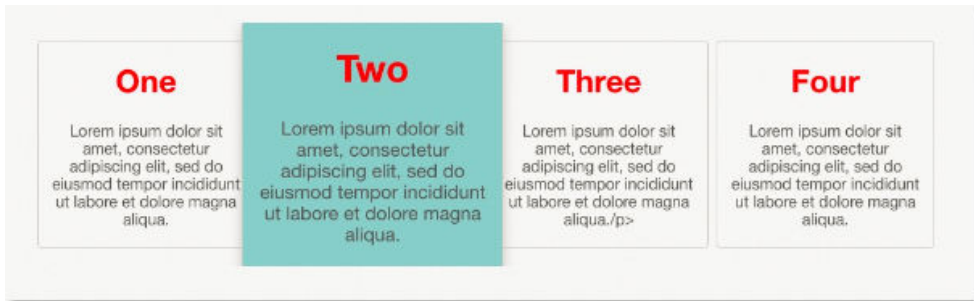
1. 检查组件中的 `assets` 文件夹是否具有以下五个文件。
 - `design.css`

- mustache.min.js
 - render.html
 - render.js
 - settings.html
2. 将新的 HTML 组件添加到您的测试站点中的页。在编辑器中，在编辑和预览模式下应看到页上的组件如下所示。

编辑模式



预览模式



将用户提供的的数据添加到 HTML 组件

您可以添加定制设置面板，以针对在 HTML 组件中定义的所有数据元素提供字段。然后，用户就可以在定制设置面板中为页上的组件实例填写值。

要将用户提供的的数据添加到 HTML 组件，请执行以下操作：

1. 在 `render.html` 文件中更新 HTML 模板以包含所需的任何数据元素。以下示例使用 Mustache JS 模板语言，因此，必须使用 `{{` 和 `}}` 语法添加数据元素，如下所示：

```
<ul class="wrapper">
  <li class="box">
    <h1 class="title">{{title1}}</h1>
    <p class="text">{{text1}}</p>
  </li>
  <li class="box">
    <h1 class="title">{{title2}}</h1>
    <p class="text">{{text2}}</p>
  </li>
</ul>
```

```

</li>
<li class="box">
  <h1 class="title">{{title3}}</h1>
  <p class="text">{{text3}}</p>
</li>
<li class="box">
  <h1 class="title">{{title4}}</h1>
  <p class="text">{{text4}}</p>
</li>
</ul>

```

2. 通过更改 HTML 组件中 `appinfo.json` 文件的内容，提供以下数据字段的默认值：

```

{
  "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. 添加定制设置面板，用于在 HTML 模板 `render.html` 文件中查找数据值以创建设置，供用户在编辑器中更改。将 `settings.html` 文件的内容更改为以下行：

```

<!DOCTYPE html>
<html lang="en">
<head>
  <!-- only allow embedding of this iFrame in SCS -->
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

  <title>H1 Mustache Component</title>

```

```
<!-- include sample apps styling -->
<link href="/_sitescloud/renderer/app/sdk/css/app-styles.css"
rel="stylesheet">

    <!-- include supporting files -->
    <script type="text/javascript" src="/_sitescloud/renderer/app/
apps/js/knockout.min.js"></script>
    <script type="text/javascript" src="/_sitescloud/renderer/app/
apps/js/jquery.min.js"></script>

    <!-- include the Sites SDK -->
    <script type="text/javascript" src="/_sitescloud/
renderer/app/sdk/js/sites.min.js"></script>
</head>
<body data-bind="visible: true" style="display:none; margin:0px;
padding:0px;background:transparent;background-
image:none;">
    <!-- ko if: initialized() -->
    <div class="scs-component-settings">
        <div>
            <!-- Width -->
            <label id="headingTextLabel"
for="headingText" class="settings-heading" data-bind="text:
'Heading Text'"></label>
            <input id="headingText" data-bind="value:
headingText" placeholder="Heading" class="settings-
text-box">
        </div>
    </div>
    <div data-bind="setSettingsHeight: true"></div>
    <!-- /ko -->
    <!-- ko ifnot: initialized() -->
    <div data-bind="text: 'waiting for initialization to
complete'"></div>
    <!-- /ko -->
    <script type="text/javascript">

        // set the iFrame height when we've fully rendered
        ko.bindingHandlers.scsCompComponentImpl = {
            init: function (element, valueAccessor,
allBindings, viewModel, bindingContext) {
                var body = document.body,
                    html =
document.documentElement;

                SitesSDK.setHeight(Math.max(
                    body.scrollHeight,
                    body.offsetHeight,
                    html.clientHeight,
                    html.scrollHeight,
                    html.offsetHeight));
            }
        };

        // define the viewModel object
```

```
var SettingsViewModel = function () {
    var self = this;

    // create the observables for passing data
    self.headingText = ko.observable('Heading 1');

    // create rest of viewModel
    self.initialized = ko.observable(false);
    self.saveData = false;

    // Get custom settings
    SitesSDK.getProperty('customSettingsData',

function (data) {

    //update observable
    self.headingText(data.headingText);

    // note that viewModel is initialized
    and can start saving data

    self.initialized(true);
    self.saveData = true;

});

    // save whenever any updates occur
    self.save = ko.computed(function () {
        var saveconfig = {
            'headingText':

self.headingText()

        };

        // save data in page
        if (self.saveData) {

SitesSDK.setProperty('customSettingsData', saveconfig);
        }
    }, self);

};

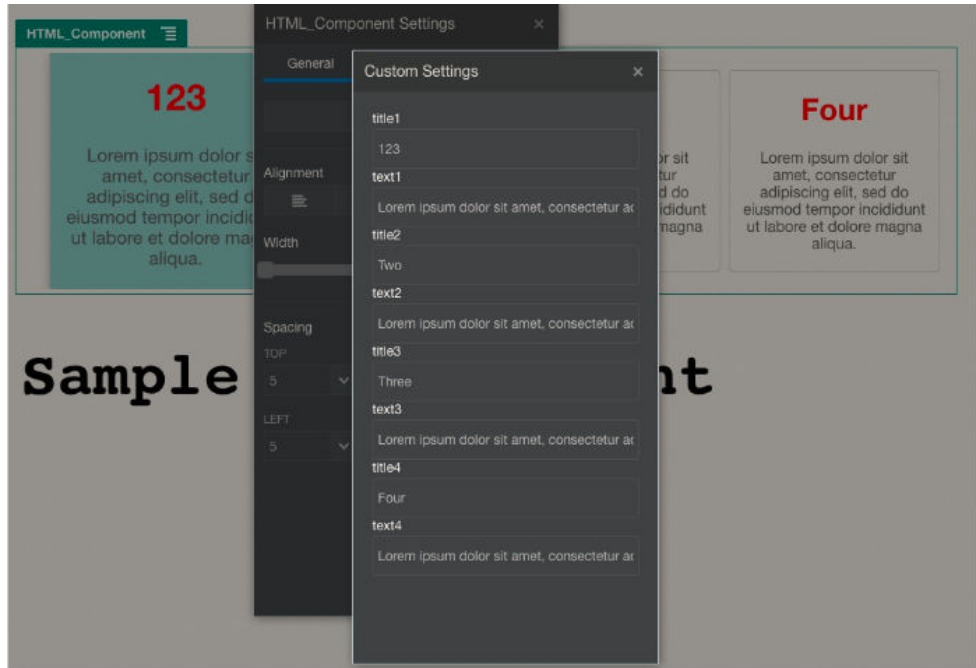
    // apply the bindings
    ko.applyBindings(new SettingsViewModel());
</script>
</body>
```

 **注:**

由于更改了 `appinfo.json` 文件，因此必须刷新浏览器并将组件重新添加到页以获取更改。

要验证组件现在是否会从 `appinfo.json` 文件获取其默认数据，以及您是否可以更改定制设置面板中的值（检查点 2），请执行以下操作：

1. 在更改 `appinfo.json` 文件后，记住要刷新浏览器并将组件重新添加到页以获取更改。
2. 打开 HTML 组件中的定制设置面板，然后输入数据。



3. 验证 Oracle Content Management 组件是否将 HTML 和 CSS 与 Mustache JS 模板结合使用，以根据您在定制设置面板上输入的数据来呈现其内容。

比较本地组件与远程组件

本地组件和远程组件的实施方式不同。

下表列出了组件实施方式的差异。它可能有助于您确定是要使用本地组件还是要使用在内嵌框架中呈现的远程组件。

本地组件实施	远程组件实施
将以内容为中心的复杂用户界面集成到 Oracle Content Management 中	将应用程序逻辑集成到 Oracle Content Management 中
执行 JavaScript 以在页中呈现内容	使用 <iframe> HTML 标记在页中呈现内容
使用任何 JavaScript 技术堆栈	可以使用任何技术，而不是仅使用 JavaScript
利用 Oracle Content Management JavaScript 堆栈，并可以重用 Oracle Content Management 组件	不与 Oracle Content Management JavaScript 堆栈集成
从 Oracle Content Management 服务器中存储的 /assets 文档添加资产相关项	所有资产访问都需要 HTTPS URL 和相应的证书
与模板一起由 Oracle Content Management 托管，并独立于主题发布	对于 URL 端点，需要托管的中间层服务器
由于代码是在页中执行，因此，如果组件代码中断，则可能会中断页	页独立于内嵌框架内容呈现，因此页不会因组件失败而中断
如果组件呈现较慢，可能会降低页呈现速度	页独立于内嵌框架内容加载，因此页会尽可能地加载，然后再加载任何内嵌框架的内容

 注：

虽然本地组件和远程组件的呈现方式不同，但站点构建器中的“设置”面板实施是相同的。这两种类型的组件的“设置”面板都使用内嵌框架呈现，并且此面板使用相同的 JavaScript SDK 以允许跨边界和跨域通信。

呈现组件设置

要呈现组件设置，可以使用组件设置 URL 和组件设置呈现选项。

组件设置 URL

组件设置 URL 使用内嵌框架进行呈现，通过参数进行调用，以便允许放到某页上的实际组件的特定设置。设置 URL 具有以下格式：

```
{Component Settings URL}?instance=<app-  
instance>&width=<width>&currCompId=<id of the app associated with the  
settings panel>&locale=<locale>
```

名称	类型	说明
Component Settings URL	URL	组件的组件设置 URL
width	数字	设置内嵌框架的宽度（以像素为单位）
currCompId	字符串	通过“设置”面板编辑的组件的当前组件 ID
locale	字符串	主机站点的当前区域设置（站点构建器）。格式为 <language>_<dialect>。示例： En_us。

本地组件设置呈现选项

有三个选项用于根据内嵌框架的大小和复杂性在本地组件的“设置”面板中呈现内嵌框架。在 `settingsRenderOption` 属性中指定每个选项。

 注：

这些选项仅可用于本地组件。远程组件“设置”面板始终在对话框中呈现。

- inline
 - 仅当有一些小属性供用户输入时，才使用此选项。
 - 内嵌框架将插入到一般信息选项卡，替代用于导航到“设置”面板的常用按钮。
 - 提供最集成的解决方案，需要最少的用户单击量，但空间有限。
- panel
 - 如果属性列表较长，但它们仍可以在标准“设置”面板的 300 像素内合理地显示，请使用此选项。

- 内嵌框架将滑动到视图中，将显示“返回”按钮以将用户返回“一般信息”选项卡。
- 提供集成的解决方案，您可以在其中与页上的内嵌框架交互。
- dialog
 - 用于处理需要复杂用户界面的更常规设置布局的默认模式。
 - 在页上的模式对话框中显示内嵌框架。

本地组件实施

组件实例由组件工厂对象创建。

组件注册文件中的组件工厂对象条目为 `initData.componentFactory`。这是对 JavaScript 文件的引用，RequireJS 使用该文件加载组件工厂。它必须返回实施 `componentFactory` 接口的 JavaScript 对象

组件工厂对象必须实施 API `customComponentFactory.createComponent(args)`，其创建组件的每个实例。

`args` 包含以下设置：

- `SitesSDK`：Oracle Content Management Sites SDK。
- `id`：添加到页的组件的唯一 ID (GUID)。
- `viewMode`：页正在呈现的当前模式。正在编辑页时，它是“编辑”。预览页时，它是“导航”。在运行时（即发布站点时），该值未定义。可以根据应为每种模式公开的功能，提供不同的实施。例如，页在“编辑”模式下运行时，链接不应处于活动状态。

注：

不需要为您的组件使用 JQuery 或 Knockout，但是如果您要利用 Oracle Content Management 功能（例如嵌套组件），则必须使用 Oracle Content Management 提供的 Knockout 版本。此版本的 Knockout 具有扩展的组件注册和处理程序，否则它们将对您不可用。

对于组件自身，在组件实例化时将传递 SDK，以便组件可以与页生命周期通信。页生命周期功能必须由组件实施，并由 Oracle Content Management 调用以在页上呈现组件。

提供了必需和可选 API 来实施组件。

必需 API

`customComponent.render(container)`：要求组件将自身插入到提供的 DOM 容器元素。

- `container`：定制组件 HTML 的 DOM 容器元素。

可选 API

`customComponent.dispose()`：在从页中删除组件时调用。为组件提供机会来删除不再需要的任何资源。

组件的样式类

您可以创建定义的样式列表，以使用户可以将其应用于您的组件。

为您的组件准备预定义的样式列表采用的模型与 Oracle Content Management 通过主题的设计文件为内容和体验提供的组件定义样式类相同。您要在组件的 `appinfo.json` 文件中为定制样式类命名。

在 `design.css` 和 `design.json` 文件中定义其他样式。`json` 文件提供从用户界面中显示的名称到实际基础 `css` 类名称的映射，`css` 文件提供每个样式类的详细信息。

`design.json` 文件包含组件的以下结构：

```
{
  "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": []
    }
  }
}
```

```

    }
  }
}

```

每个 "styles":[] 条目都可以包含特定组件的样式列表。"name" 可以是对内置本地化字符串的引用，也可以是要使用的指定值。例如，标题组件提供以下默认样式：

```

{
  "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"
  }
]
}

```

名称值映射到要在用户界面中显示的实际词语，例如：

```

"COMP_STYLE_FLAT": "Flat",
"COMP_STYLE_HIGHLIGHT": "Highlight",
"COMP_STYLE_DIVIDER": "Divider",

```

css 文件提供类值的定义：

```

.scs-title-default-style {
  color: #333333;
  display: block;
  font-family: "Helvetica Neue", "Helvetica", "Arial", sans-serif;
  font-size: 24px;
  font-weight: normal;
}

```

例如，在主题的 design.json 文件中，可以根据您在 components.json 文件中定义的 initialData.componentId 值为您的组件添加条目：

```

"componentId": "news-article"

```

design.json 文件中的对应条目如下：

```

"componentStyles": {
  "news-article": {
    "styles": [{
      "name": "News Article 1",
      "class": "news-article-default-style"
    },
    {

```

```

        "name": "News Article 2",
        "class": "news-article-style-1"
    }
  ],
},
}]

```

design.css 文件中的对应条目如下：

```

.news-article-default-style .scs-image {...}
.news-article-style-1 .scs-image {...}

```

如何为内置组件设置样式

通过覆盖和扩展内置样式，在 Oracle Content Management 中创建您自己的外观来设置内置组件的样式。

内置组件从两个位置获取它们的可视样式：

- comp.css，指定每个组件的基本外观的内置 CSS 文件
- design.css，属于站点正在使用的主题一部分的 CSS 文件

在 design.css 文件中，您可以覆盖和扩展内置 comp.css 样式来创建您自己的外观。在主题中，design.css 文件位于 designs/default 目录中。

以下主题介绍了 comp.css 文件中对所有内置组件通用的类并概述了如何定义主题：

- [组件样式设置基本信息](#)
- [特定于组件的样式设置](#)
- [设置组件属性](#)

组件样式设置基本信息

所有内置站点组件都具有相似的 CSS 类结构。

每个组件的最外层 <div> 元素都应用以下三个 CSS 类：

```
scs-component scs-type design-style
```

type 是组件类型（例如 image、gallery 或 divider）。design-style 是为组件选择的样式类，这在主题的文件中定义。

主题样式基本信息

主题的 design.json 文件列出可以应用于每种类型的组件（例如 button 或 image）的所有样式（frame、shadow、highlighted 等）。每个样式都有显示名称和类名称。显示名称显示在设置面板的样式选项卡上。类名称是指主题的 design.css 文件中的 CSS 选择器。例如，button 组件对应的条目如下：

```

"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"
    }
  ]
},

```

内置组件的名称已翻译，因此，您会看到一个用于从资源包获取样式名称的键。如果向页添加 `button` 组件，然后在设置 > 样式面板中选择简单样式，则 `design.json` 文件会将显示名称简单（键 `COMP_STYLE_SIMPLE`）与类名称 `scs-button-style-3` 关联。将使用以下类呈现 `button`：

```
scs-component scs-button scs-button-style-3
```

如果没有为给定组件选择样式，则将使用默认样式 `scs-type-default-style`。在上述示例中，将使用以下类呈现按钮：

```
scs-component scs-button scs-button-default-style
```

scs-component-content 样式

对于每个内置组件，在前面提到的 `scs-component <div>` 中，都有包含 CSS 类 `scs-component-content` 的 `content <div>`。换言之：

```
scs-component scs-type design-style
scs-component-content
```

在 `design.css` 文件中，通常使用 `scs-component-content` 类为组件周围的“框”设置样式（例如，应用边框或阴影）。

值得注意的是，在内置 `comp.css` 文件中，定义常见的 `scs-component-content` 类时使用了 `position:relative` 和 `display:inline-block` 以及其他 CSS 属性。

虽然 `scs-component-content` 可用于为每个组件周围的“框”设置样式，但要为组件完整设置样式，需要特定于组件的类。请参见[特定于组件的样式设置](#)。

特定于组件的样式设置

您可以对图像、按钮、文档、段落、标题、地图及其他组件应用特定样式。

图像组件

图像组件的 `scs-component-content` 类下具有以下 CSS 类结构：

```
scs-image-container
scs-image-link
```

```
scs-image-image  
scs-image-caption
```

`scs-image-image` 类应用于 `` 标记本身。`scs-image-caption` 类用于为标题设置样式（如果存在标题）。

`scs-image-link` 类仅在图像附加链接时存在。它和 `scs-image-container` 类通常都不需要定制样式设置。

默认情况下，图像标题以在图像底部伸展的半透明重叠方式呈现。



```
.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;  
}
```

要将标题放置在图像顶部并更改颜色，请在 `design.json` 文件中为图像组件添加额外的样式，然后在 `design.css` 文件中为其定义 CSS。



```
.scs-image-style-17 .scs-image-caption {  
  position: absolute;  
  top: 0px;  
  height: 35px;  
  font-weight: bold;  
  background-color: rgba(122, 213, 256, 0.54);  
  color: #515151;  
}
```

按钮组件

按钮组件具有以下类结构：

```
scs-button-button  
  scs-button-text
```

`scs-button-button` 类是可单击的 `<div>`，样式设置为看上去像按钮。`scs-button-text` 类用于为按钮中的文字设置样式。

例如，通过在 `design.json` 文件中为按钮组件添加额外的样式，然后在 `design.css` 文件中为其定义 CSS，对该组件的外观进行更改测试。




```
.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;
}
```

文档

文档组件具有以下类结构：

```
scs-document-container
  scs-document-cap
    scs-document-title
  scs-document-desc
```

`scs-document-container` 类用于包装文档查看器，通常不设置样式。

图库

图库组件具有单个用于包装基础 JSSOR 滑块组件的类：

```
scs-gallery-container
```

JSSOR 滑块使用多个类（您也可以为其设置样式）：

```
jssorb14 (navigator)
jssora02l (left arrow)
jssora02r (right arrow)
jssort07 (thumbnails)
```

图库网格

用于图库网格组件的类取决于在设置面板中选择的布局和剪裁：

```
scs-gallerygrid-container scs-gallerygrid-layout
  scs-gallerygrid-cell
    scs-image (multiple)
```

根据为图库网格选择的剪裁和布局设置，`layout` 的值将为 `stretch`、`crop`、`fit` 或 `flowing`。

`scs-gallerygrid-cell` 类仅适用于列布局。

社交栏

社交栏组件具有以下类结构：

```
scs-socialbar-container
  scs-socialbar-icon
```

`scs-socialbar-icon` 类应用于社交栏中的每个 `` 标记。

段落

段落组件仅有单个用于包装实际段落文本的类：

```
scs-paragraph-text
```

例如，要使在段落组件中提供的文本具有金属雕刻文字效果，请在 `design.json` 文件中添加额外的样式类，然后在 `design.css` 文件中为其定义 CSS。



```
.scs-paragraph-style-7 {  
  font-size: 24px;  
  font-family: Arial, Helvetica, sans-serif;  
  font-weight: 700;  
  padding: .3em;  
  color: #000000;  
  background: #666666;  
  text-shadow: 0px 1px 1px #ffffff;  
}
```

或者，如果您想要新奇效果，请使用类似于下一个示例的内容。



```
.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;
}
```

标题

标题组件也仅有单个用于包装实际文本的类：

```
scs-title-text
```

地图

地图组件具有单个用于包装地图重现的类：

```
scs-map-content
```

此类通常不设置样式。

定制本地组件

定制本地组件仅有单个用于包装实际组件的类：

```
scs-custom-component-wrapper
```

呈现您的定制本地组件的定制视图需要使用的 CSS 样式完全由您控制。本地组件以内嵌方式呈现，即，您可以直接应用在主题或 `design.css` 文件中定义的 CSS 样式。

定制远程组件

定制远程组件仅有单个用于包装其 `iframe` 的类：

```
scs-app-iframe-wrapper
```

除了应用定制远程组件中定义的 CSS 样式外，您还可以利用 Sites SDK 从主机站点提取 `design.css` 文件。

```
// 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);
    }
  }
});
```

因此，您可以使您的组件继承主机样式中的样式设置。

分隔线

尽管分隔线组件没有特定于组件的类，但可以为 `<hr>` 标记本身设置样式。

例如，您可以创建点线分隔线：

```
.design-style .scs-divider hr {  
border-top: 1px dotted #333333;  
}
```

视频、YouTube、分隔符

视频、YouTube 和分隔符组件没有特定于组件的类。


设置组件属性

可以配置组件属性以用于站点。

组件是 Web 页中的单独部件，其包括文本、标题、图像、按钮、分隔线、地图、图库、视频等。为主题创建设计时，还必须指定每种组件类型的默认设置。每个组件都有定义该组件外观和行为的设置，例如大小、对齐方式、间距、颜色和边框。设置根据组件不同而不同。还可以选择在组件可用于新主题中的站点后，用户是否可以更改组件属性。

作为示例，以下步骤说明如何配置段落组件的设置：

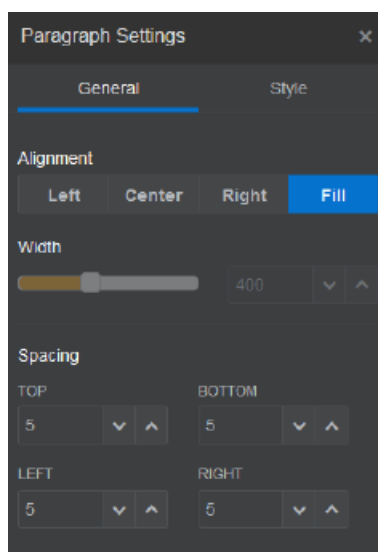
1. 在您的开发站点在编辑模式下打开的情况下，选择其中具有段落组件的页或者添加段落组件。


2. 单击段落组件，然后单击该组件角落的 ，然后选择设置。

您将看到显示设置面板，您可以在其中针对组件进行选择。

注：

设置选项特定于每种组件类型。如果使用本地或远程定制组件，您将看到定制设置链接。



- 例如，可以更改对齐方式、宽度、间距、字体、样式、颜色、大小等设置。组件外观将更改为新设置。
- 完成后，单击 。您的设置将应用于页。

在内嵌框架中呈现的组件

通过注册和添加来自外部服务器的组件（称为远程组件），以及还通过在创建本地组件时选择内嵌框架选项，可以在 Oracle Content Management 页中指定在内嵌框架中呈现的组件。这种类型的组件可以扩展站点的功能，例如添加社交组件或购物车结账组件。

对于远程组件，必须指定端点 URL。对于在内嵌框架中呈现的本地组件，从组件目录中组件的名称派生 URL。

- 端点 URL：窗口部件的内容从此 URL 提取并嵌入内嵌框架中。
- 设置 URL：在内嵌框架中呈现此 URL，以便在将组件放到页时配置该组件。

组件可以呈现静态或动态数据可视化，并显示用于扩展站点功能的表单或其他交互式用户界面。请参见[呈现组件设置](#)。

Oracle Content Management 提供的组件

随 Oracle Content Management 提供了一组在内嵌框架中呈现的组件。

名称	类型	ID
文件夹列表	scs-app	文件夹列表
文件列表	scs-app	文件列表
文档管理器	scs-app	文档管理器
Facebook 点赞	scs-app	Facebook 点赞
Twitter 关注	scs-app	Twitter 关注
Twitter 共享	scs-app	Twitter 共享
Facebook 推荐	scs-app	Facebook 推荐

组件注册

内嵌框架中呈现的远程组件可用于站点之前，必须先使用 HTTPS 注册端点 URL。此信息存储在组件目录中。端点必须允许 URL 显示在内嵌框架中；不要在页眉中设置 `x-Frame-Options="sameorigin"`。对于本地组件，因为文件存储在 Oracle Content Management 服务器上，所以将自动满足此端点条件。

内嵌框架中呈现的组件注册后，将生成新的 GUID 来表示该组件。如果此类组件在 Oracle Content Management 实例中注册多次，它将获得多个 GUID，因为它们表示该组件的每次注册。注册组件时，说明就是 Oracle Content Management 中存储的针对 GUID 的所有信息；组件将仍从其远程端点运行。请参见[注册远程组件](#)。

内嵌框架中呈现的组件注册后，还将生成实例 ID。此实例 ID 表示在特定 Oracle Content Management 租户中注册的组件。同一组件在同一 Oracle Content Management 租户内注册多次，或者在不同 Oracle Content Management 租户内注册，将具有不同的实例 ID。

远程组件设置持久性

内嵌框架中呈现的组件的设置 URL 在“设置”面板对话框中呈现时，将同时提供实例 ID 和组件 ID。这样，组件可以选择在其自己的服务器中自己持久保留任何设置，按实例 ID 和组件 ID 编制索引。或者，可以使用 Oracle Content Management Sites SDK，以允许针对组件 ID 在站点的页模型中最多存储 1.5 KB 的 JSON 数据。

对于内嵌框架中呈现的组件，使用 Sites SDK 持久保留设置具有两个优势：

- 组件可以轻松参与页版本、页更新和站点发布模型。
- 组件可以包含在浏览器中执行（与在后端系统中执行相对）的 HTML 端点。

请参见 Oracle Content Management SDK。

关于在内嵌框架中呈现的组件的实例 ID 和结构

组件实例 ID 是在站点中内嵌框架中呈现的组件的唯一标识符。

用户将内嵌框架中呈现的组件从组件目录拖放到站点页上时，将向 Oracle Sites Cloud Service 发出预配调用以生成新的唯一组件实例 ID。此 ID 保证是唯一的，并且在该服务上预配的所有此类组件实例都获取相同的实例 ID。

组件实例 ID 包含可用于保护内嵌框架中呈现的组件的设置和使用的其他信息，以便组件可以确定设置更新来自可信位置。

开发人员通过实例 ID 参数可以识别站点并验证调用方。对调用方验证是通过验证使用组件密钥生成的数字签名来进行的。该密钥是在组件注册过程中生成的。

组件实例包含由 ‘.’ 分隔符分隔的两个部分：数据和结构。

组件实例：数据

内嵌框架中呈现的组件实例的数据部分是 Base64 JSON 编码字符串。下面是 JSON 字符串结构：

```
{
  "instanceid": "BBDC7614F693B75110D811E6C0B77C935FAEC5112E5E",
  "permissions": "",
  "entitlements": "",
  "signdate": "1435426735293",
  "sitedomain": "service1-tenant4.localhost"
}
```

字段名称	说明
instanceid	Oracle Content Management 租户内嵌框架中呈现的组件的唯一标识符。
signdate	签名生成日期。
sitedomain	Oracle Content Management 实例的域名。
permissions	站点成员的权限集。在编辑模式下，其值为 "SITE_OWNER"；在其他情况下，它没有值。
entitlements	站点所有者购买的高级功能列表。

组件实例：签名

组件实例的数据部分先进行串行化，然后再由 APP_SECRET_KEY 签名。在注册组件时，必须生成此密钥并向开发人员显示。签名是通过使用密钥生成组件实例数据部分的散列（串行化 JSON 结构）计算得出的，如下所示：

```
$signature = HMAC (serialized JSON structure, APP_SECRET_KEY)
```

生成签名时使用的散列算法为 SHA256。令牌则是将串行化 JSON 结构和生成的签名组件连接的结果，如下所示：

```
$instance = {base64encoded serialized JSON structure}.  
{base64encoded $signature}
```

示例：

```
//base64 encoded serialized object           //signature  
eyJpbmN0YW5jZWlkIjoiQTRGOTE3REY5OTZEN0Q3ODBCMjUzODZFOTFEMDA3ODJGMjVBRjY2  
Rjc3OTIiLCJzaWduZGF0ZSI6IjE0NDU2MzcwNTk5MTciLCJzaXRlZG9tYWluIjoic2Vydmlj  
ZTEtdGVuYW50MS1cy5vcnFjbGUuY29tIiwicGVybnVlc2lvdnMiOiJTSVRFX09XTkVSIiw  
iZW50aXRzZWl1bnRzIjoiIn0=.5p3of7t110wuysF3zpm+YgICSHH8C/BHczdbVZx2VH8=
```

远程组件的安全性

Oracle Content Management 允许第三方开发人员将他们的定制组件集成到 Oracle Content Management 平台中，但是将这些组件存储在远程服务器上。

每个远程组件都必须具有在 Oracle Content Management 中注册的设置和呈现端点。除了端点，开发人员还需要提供对注册组件唯一的密钥。

Oracle Content Management 调用注册组件的端点以在站点页中实现内容。因为这些端点是对公共 Internet 公开的，所以开发人员应验证注册远程组件的端点是否是从 Oracle Content Management 调用的。为了验证调用方的真实性，将为 URL 的注册端点提供签名的令牌。通过使用注册过程中提供的远程组件的密钥来验证签名令牌中嵌入的数字签名，从而对调用方进行验证。

令牌的格式为：

```
{base64 encoded serialized JSON data}.{base64 encoded signature}
```

传递到注册的应用程序端点的令牌示例如下所示：

```
eyJpbmN0YW5jZWlkIjoiQTRGOTE3REY5OTZEN0Q3ODBCMjUzODZFOTFEMDA3ODJGMjVBRjY2  
Rjc3OTIiLCJzaWduZGF0ZSI6IjE0NDU2MzcwNTk5MTciLCJzaXRlZG9tYWluIjoic2Vydmlj  
ZTEtdGVuYW50MS1cy5vcnFjbGUuY29tIiwicGVybnVlc2lvdnMiOiJTSVRFX09XTkVSIiw  
iZW50aXRzZWl1bnRzIjoiIn0=.5p3of7t110wuysF3zpm+YgICSHH8C/BHczdbVZx2VH8=
```

令牌包含两个不同部分：数据和签名，由 '.' 分隔符分隔。

一般来说，在对组件的注册端点授予访问权限之前，开发人员应始终在编辑或预览模式下验证令牌。此外，在设置端点中验证调用方时，开发人员应始终注意查找令牌的

permissions 字段中的 `SITE_OWNER` 值。令牌的 **permissions** 字段仅在编辑模式下显示 `SITE_OWNER` 值。编辑会话过程中生成的令牌不会持久保留回页模型，将与运行时令牌切换，该运行时令牌在 **permissions** 字段中具有 `NULL` 值。

数据

实例的数据部分是 Base64 JSON 编码的字符串。下面是 JSON 字符串结构：

```
{
  "instanceid": "BBDC7614F693B75110D811E6C0B77C935FAEC5112E5E",
  "permissions": "",
  "entitlements": "",
  "signdate": "1435426735293",
  "sitedomain": "service1-tenant4.localhost"
}
```

字段名称	说明
instanceid	Oracle Content Management 租户的组件的唯一标识符。
signdate	签名生成日期。
sitedomain	Oracle Content Management 实例的域名。
permissions	站点成员的权限集。在编辑模式下，其值为 "SITE_OWNER"；在其他情况下，它没有值。
entitlements	站点所有者购买的高级功能列表。

签名

远程组件实例的数据部分先进行串行化，然后再由 `APP_SECRET_KEY` 签名。在注册组件时，必须生成此密钥并向开发人员显示。签名是通过使用密钥生成组件实例数据部分的散列（串行化 JSON 结构）计算得出的，如下所示：

```
$signature = HMAC (serialized JSON structure, APP_SECRET_KEY)
```

生成签名时使用的散列算法为 SHA256。令牌则是将串行化 JSON 结构和生成的签名组件连接的结果，如下所示：

```
$instance = {base64encoded serialized JSON structure}.
{base64encoded $signature}
```

注册远程组件

要在站点中使用远程组件，必须在 Oracle Content Management 中注册它。

您可以注册第三方远程组件以及您自己开发的远程组件。

要注册远程组件以在 Oracle Content Management 实例中使用，请执行以下操作：

1. 单击开发人员，然后单击查看所有组件。
2. 单击创建并选择注册远程组件。
3. 在“注册远程组件”窗口中，输入或选择下列信息：

- 名称：用户将看到的组件的名称。
- 说明：用户将看到的组件的说明。
- 组件 URL：在 iframe 中使用的端点，用于在页中呈现组件内容。必须为 HTTPS。
- 设置 URL：在 iframe 中使用的端点，用于呈现添加到页的远程组件的设置。必须为 HTTPS。
- 设置宽度：设置组件设置面板的默认宽度，以像素为单位。
- 设置高度：设置组件设置面板的默认高度，以像素为单位。
- 密钥：与远程组件关联的 192 位 AES 密钥，用于在预配组件时创建签名散列令牌。它用于加密，确保安全地读取和写入组件设置。

4. 单击注册。

创建了远程组件之后，该名称显示在组件的列表中。在组件列表中单击组件名称可以浏览用于注册组件的文件。

组件注册信息存储在由站点使用的目录中，该站点是在同一 Oracle Content Management 实例中创建的，但组件保持为远程服务。

对于组件所有者来说，组件图标会添加到站点构建器的“定制组件”面板中，使用您分配给组件的名称。您可以与其他用户共享组件，对方可在站点构建器的“定制组件”面板中看到该组件。

删除组件


如果您有合适的权限，则可以从组件管理器中删除组件，使其不再可供使用。删除组件时，组件文件夹及其关联的所有文件夹和文件将移到回收站中。

如果您创建了组件（您是组件所有者）或者其他人与您共享了组件并授予您管理者角色，则您可以从组件管理器中删除该组件。

注：

如果组件由任何站点或更新使用（包括回收站中的站点或更新），则无法删除。

要删除组件，请执行以下操作：

1. 在主页上，单击开发人员。
2. 单击查看所有组件。
此时将显示任意当前已注册的组件。
3. 选择组件名称，然后在右键单击菜单中选择删除，或单击操作栏中的 。
系统将提示您确认操作。
4. 要确认删除操作，请单击是。要停止删除操作，请单击否。
如果确认删除，则组件及其所有关联文件夹和文件将移动到回收站中。

已删除的组件文件夹会保留在回收站中，直至：

- 您还原该文件夹。
- 您永久地删除该文件夹。

- 达到您的回收站限额。
- 回收站会根据服务管理员设置的间隔自动清空。默认值为 90 天。

Sites SDK

为 Oracle Content Management 开发的组件在站点中呈现为组件，并且可以拖放在站点的特定页上的任何位置。

Sites SDK 处理组件与页之间的所有通信。

- `Sites.Settings.getProperty(propertyName, callbackFunction)`：提供用于检索定制组件实例的请求属性的回调。
- `Sites.Settings.setProperty(propertyName, propertyValue)`：根据定制组件实例存储请求属性。

对所有定制组件属性使用 `Sites.Settings.getProperty` 或 `Sites.Settings.setProperty`，然后使用 `SitesSDK.publish` 和 `SitesSDK.subscribe` 监听消息事件。

请参见 [Sites SDK 参考](#)。

24

定制控制器文件

在 Oracle Content Management 站点中，控制器文件用于显示站点中的每个页面。当浏览器发送需要 Web 页的请求时，服务器会返回控制器文件副本。对于请求的每个页面，会从服务器传送控制器文件同一副本。

- [关于控制器文件](#)
- [修改默认的 controller.html 文件](#)
- [关于 SCS 对象](#)
- [不应定制的控制文件部分](#)
- [使用标记来实现定制控制器文件可移植性](#)
- [定制控制器文件示例](#)

关于控制器文件

控制器文件是一种小型 HTML 页，用于动态为页面其余部分启动呈现序列。它是可以应用定制内容以影响站点每个页面行为的第一个位置。

控制器文件的主要任务是为控制器 JavaScript 提供并托管执行环境。然后由控制器 JavaScript 加载并显示页面。控制器文件中的定制内容能够覆盖和影响控制器 JavaScript 的操作。

默认的控制文件

首次创建 Oracle Content Management 站点时，一个默认的控制文件 controller.html 与站点关联。

```
<!DOCTYPE html>
<html>
<head>
<!-- The following meta tag is used for Internet Explorer browsers. It
indicates that the browser should use the latest rendering mode to display
the web page. -->
<meta http-equiv="X-UA-Compatible" content="IE=edge">

<!-- The following meta tag is used for browsers on mobile devices to set
the initial viewport scale to the full page. -->
<meta name="viewport" content="initial-scale=1">

<!-- The following script initially defines the SCS object. The SCS object
must be present, and this variable name is reserved for use by Oracle
Content Management.. -->
<script type="text/javascript">
var SCS = { sitePrefix: '/SampleSite/' };
</script>

<!-- The following script loads the full controller JavaScript, which is
```

```
used to display the web page. -->
<script src="/SampleSite/_sitesclouddelivery/renderer/controller.js"></script>
</head>

<!-- The body tag of the controller must have the id scsControllerBody.
This identifier is used by the Controller JavaScript. If JavaScript is
not enabled on the browser, the noscript tag content is displayed.-->
<body id="scsControllerBody"><noscript>This site requires JavaScript to
be enabled.</noscript>

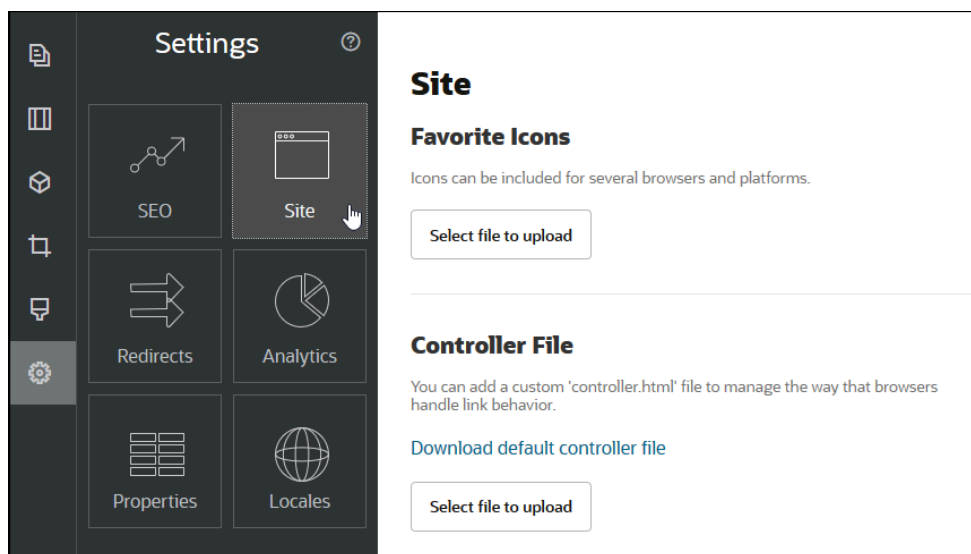
<!-- The following image tag displays an animated circle by default if
it takes too long for a page to display. The wait image must have the
id scsWaitImage. -->

</body></html>
```

修改默认的 controller.html 文件

下载站点的 controller.html 文件，并对其进行修改以影响每个站点页的行为。要下载站点的 controller.html 文件，请执行以下操作：

1. 在站点构建器中打开要修改的站点，并将其设置为编辑。
2. 选择现有更新或创建新更新。
3. 单击侧面导航菜单中的设置，然后单击站点。



4. 在“控制器文件”部分中，单击下载默认的控制文件。
5. 保存到本地驱动器，通过编辑来定制 controller.html 文件。
6. 完成编辑后，返回站点的“设置”页，然后在“控制器文件”部分中单击选择要上传的文件。
7. 导航到修改的控制文件并将其上传。

 注:

对控制器文件所做的定制内容仅用于联机站点。编辑站点时或在站点预览中不使用它们。要在为站点上载定制 controller.html 文件后使更改生效，必须提交站点更新并发布站点。

关于 SCS 对象

控制器 JavaScript 使用全局 SCS 对象中定义的变量和属性来呈现 Web 页。全局 JavaScript 对象必须在控制器文件中包含 controller.js 文件之前定义。SCS 对象中有两个主要属性可用于定制：

- [SCS.sitePrefix](#)
- [SCS.preInitRendering](#)
- [SCS.getDeviceInfo](#)

SCS.sitePrefix

SCS.sitePrefix 变量定义联机站点的路径前缀。通常，此项默认为 `site/<siteName>/`。

控制器 JavaScript 使用此值确定要显示的 Web 页。例如，如果浏览器请求 `/site/SampleSite/products/index.html` 处的页面，通过 sitePrefix `/site/SampleSite/`，JavaScript 可以计算出应显示站点中的 `products/index.html` 页面。请注意，sitePrefix 必须是以 `/` 字符开头和结尾的字符串值。

如果实施了代理或其他基础结构（例如 Akamai）和 URL 映射规则，则可通过修改此变量来定制站点前缀。例如，如果将 sitePrefix 设置为 `/intranet/ExampleSite/`，则可以通过在浏览器中使用路径名来传送网站，而不是使用默认的 `/site/SampleSite/` 路径前缀。

 注:

如果浏览器中的 URL 与站点前缀不匹配，则默认控制器 JavaScript 在内部使用 `/` 作为 sitePrefix。这样，可以使用没有定制内容的虚域提供站点。

SCS.preInitRendering

如果定义了 SCS.preInitRendering 函数，则先由控制器 JavaScript 代码调用此函数，之后再执行它的任何计算逻辑。此函数很有用，可在定制控制器文件中定义此函数以覆盖控制器 JavaScript 的基本操作。

如果定义 SCS.preInitRendering，它必须是函数。调用此函数时不使用任何参数，也不会有或处理任何返回值。

SCS.getDeviceInfo

通过使用 SCS.getDeviceInfo 函数，可以在控制器 JavaScript 中定制设备检测逻辑。此检测用于确定应传送页面的移动版本还是响应版本。

SCS.getDeviceInfo 函数不接收任何参数，它返回具有两个属性的 JavaScript 对象：

- **isMobile** — 布尔值属性，指示当前设备是移动客户端，例如智能手机。
- **isIOS** — 布尔值属性，指示当前设备正在运行基于 iOS 的操作系统。

如果定制控制器未覆盖 `SCS.getDeviceInfo`，则使用内置的默认实施。

不应定制的控制器文件部分

控制器文件的某些部分由 Oracle Content Management 服务器处理，以便使用 CDN 以及定义 URL 高速缓存段。在下面的示例中，以内嵌方式对这些部分加注了说明。

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="initial-scale=1">
<script type="text/javascript">
<!-- The global variable SCSCacheKeys will be inserted at this location
-->

var SCS = { sitePrefix: '/SampleSite/' };
<!-- A variable denoting the location of the CDN will be injected at
this location. -->
<!-- Additionally, the prefix "/site" will be inserted before the site
name segment. This is to support legacy controller files. -->

</script>
<script src="/SampleSite/_sitesclouddelivery/renderer/controller.js"></
script>
<!-- The src value will be updated to use the CDN if this syntax is
used. -->

</head>
<body id="scsControllerBody"><noscript>This site requires JavaScript to
be enabled.</noscript>
<!-- The id of the <body> should not be changed. -->


<!-- The id of the wait image tag should not be changed -->

</body></html>
```

使用标记来实现定制控制器文件可移植性

创建站点时，站点名称用于控制器文件中。这将控制器文件关联到指定站点，并造成无法在站点之间使用控制器文件。

如果在名称与控制器文件中所用名称不同的站点上使用该控制器文件，将不起作用。对于从具有定制控制器文件的站点模板创建的站点，也是如此。使用以下动态求值的标记，可以将控制器文件用于多个站点。

标记	说明
<code>[!-\$SCS_SITE_PREFIX-]</code>	此项求值结果为当前站点的站点前缀。值示例可能包括： <code>/site/MySite/</code> 或 <code>/site/authsite/MySecureSite/</code>
<code>[!-\$SCS_SITE_PATH-]</code>	此项求值结果为当前产品 CDN 位置。值没有尾随 '/' 字符。值示例可能包括： <code>/site/MySite/_cache_0000</code> 或 <code>/site/authsite/MySecureSite</code>
<code>[!-\$SCS_PRODUCT_PATH-]</code>	此项求值结果为当前产品 URL，包括高速缓存键（如果适用）。值没有尾随 '/' 字符。值示例可能包括： <code>https://www.example.com/cdn/cec/v21.1.2.23</code> 或 <code>https://www.example.com/cdn/cec/v21.1.3.18</code>

定制控制器文件示例

提供了以下定制控制器文件的示例：

- [更改站点前缀](#)
- [定制等待图](#)
- [定制收藏夹图标](#)
- [为非 JavaScript Crawler 定制 <noscript> 和 <meta> 标记](#)
- [预取 JavaScript 文件](#)
- [使用额外的标记来验证站点所有权](#)
- [增加设备检测](#)
- [使用标记来增强 controller.htm 可移植性](#)

更改站点前缀

以下示例控制器文件定义 `preInitRendering` 函数，以允许基于多个前缀传送站点。

注：

要利用此示例，必须配置 CDN 或其他代理以响应定义的前缀。此外，如果函数中定义的附加前缀都不匹配浏览器 URL，则将使用默认行为。

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="initial-scale=1">
<script type="text/javascript">
var SCS = { sitePrefix: '/SampleSite/' };

SCS.preInitRendering = function() {
    // List additional site prefixes here. All sitePrefix values MUST
    start and end with a '/' character.
    var additionalSitePrefixes = [
        '/corporate/intranet/SampleSite/',
        '/marketing/preflight/',
        '/qa/'
    ];

    // Determine if the actual browser URL matches one of the
    additional site prefixes
    var pageUrl = decodeURI(window.location.pathname);
    var i, prefix;
    for (i = 0; i < additionalSitePrefixes.length; i++) {
        prefix = additionalSitePrefixes[i];
        if (pageUrl.startsWith(prefix) || (pageUrl ===
prefix.slice(0, -1))) {
            // If we find a match, set the global site prefix
            variable
                SCS.sitePrefix = prefix;
                break;
            }
        }
    };
</script>
<script src="/SampleSite/_sitesclouddelivery/renderer/controller.js"></
script>
</head>
<body id="scsControllerBody"><noscript>This site requires JavaScript to
be enabled.</noscript>

</body></html>
```

定制等待图

以下示例控制器文件定制在呈现因网络延迟而延迟时显示的等待图。

注：

此示例使用图像的数据 URL，而不是在呈现期间发出单独的请求。

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="initial-scale=1">
<script type="text/javascript">
var SCS = { sitePrefix: '/SampleSite/' };
</script>
<script src="/SampleSite/_sitesclouddelivery/renderer/controller.js"></script>
</head>
<body id="scsControllerBody"><noscript>This site requires JavaScript to be enabled.</noscript>

</body></html>

```

定制收藏夹图标

以下示例控制器文件定义站点的定制收藏夹图标。当浏览器不从呈现的页面的文档对象模型动态加载收藏夹图标时，这很有用。

```

<!DOCTYPE html>
<html>
<head>
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="initial-scale=1">
<link rel="shortcut icon" href="/SampleSite/favicon.ico" />
<script type="text/javascript">

```

```
var SCS = { sitePrefix: '/SampleSite/' };
</script>
<script src="/SampleSite/_sitesclouddelivery/renderer/controller.js"></
script>
</head>
<body id="scsControllerBody"><noscript>This site requires JavaScript to be
enabled.</noscript>

</body></html>
```

为非 JavaScript Crawler 定制 <noscript> 和 <meta> 标记

以下示例控制器文件针对不处理 JavaScript 的 Crawler 定制 <noscript> 消息和 <meta> 标记。

注：

由于会为网站的每个页面提供控制器文件，因此同一元标记会出现在每个页面上。

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="initial-scale=1">
<meta name="description" content="A site with interesting content">
<script type="text/javascript">
var SCS = { sitePrefix: '/SampleSite/' };
</script>
<script src="/SampleSite/_sitesclouddelivery/renderer/controller.js"></
script>
</head>
<body id="scsControllerBody"><noscript>Please enable JavaScript to view this
site properly.</noscript>

</body></html>
```

预取 JavaScript 文件

一些浏览器允许声明当前或后续导航中所需的资源。控制器文件中的标记可以声明应预取或预加载的资源。

 注:

由于每个 OCE Web 页都使用 require.js 和 renderer.js，因此，它们适合应用预加载和预取技术。此外，当 href 列在 <link> 标记中的前面位置时，其值要由从 CDN 传送的 OCE 服务器修复。

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="initial-scale=1">
<link href="/SampleSite/_sitesclouddelivery/renderer/require.js"
rel="preload" as="script">
<link href="/SampleSite/_sitesclouddelivery/renderer/renderer.js"
rel="preload" as="script">
<script type="text/javascript">
var SCS = { sitePrefix: '/SampleSite/' };
</script>
<script src="/SampleSite/_sitesclouddelivery/renderer/controller.js"></
script>
</head>
<body id="scsControllerBody"><noscript>Please enable JavaScript to view
this site properly.</noscript>

</body></html>
```

使用额外的标记来验证站点所有权

第三方 Crawler 和搜索引擎可能需要额外的标记来验证站点所有权。可以根据需要将其注入到控制器文件中。

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="initial-scale=1">
<meta name="search-site-verification" content="your verification
string">
<script type="text/javascript">
var SCS = { sitePrefix: '/SampleSite/' };
</script>
<script src="/SampleSite/_sitesclouddelivery/renderer/controller.js"></
script>
</head>
<body id="scsControllerBody"><noscript>Please enable JavaScript to view
this site properly.</noscript>

</body></html>
```

增加设备检测

可以通过在控制器中定制设备检测代码来识别新移动设备或定制的设备用户代理标识符。

注:

- 控制器调用 `getDeviceInfo` 函数来确定是否在使用移动设备。此示例覆盖默认的 `getDeviceInfo` 调用。
- `isMobile` 属性确定呈现页面时使用的布局。
- `isIOS` 属性确定与地图组件一起使用的提供方。
- 内置的默认 `isMobile` 测试如下

```
/Mobi|iPhone|iPod|BlackBerry|IEMobile|Opera Mini/  
i.test(userAgent) && !/iPad/i.test(userAgent)
```

- 内置的默认 `isIOS` 测试如下

```
/iPad|iPhone|iPod/i.test(userAgent) && !window.MSStream
```

```
<!DOCTYPE html>  
<html>  
<head>  
<meta http-equiv="X-UA-Compatible" content="IE=edge">  
<meta name="viewport" content="initial-scale=1">  
<meta name="search-site-verification" content="your verification string">  
<script type="text/javascript">  
var SCS = { sitePrefix: '/SampleSite/' };  
  
SCS.getDeviceInfo = function() {  
  // Return an object with two Boolean properties, isMobile and isIOS.  
  var userAgent = navigator.userAgent;  
  return {  
    isMobile: /Mobi|iPhone/i.test(userAgent) && !/iPad/i.test(userAgent),  
    isIOS: /iPad|iPhone|iPod/i.test(userAgent)  
  };  
};  
</script>  
<script src="/SampleSite/_sitesclouddelivery/renderer/controller.js"></script>  
</head>  
<body id="scsControllerBody"><noscript>Please enable JavaScript to view this  
site properly.</noscript>  
  
</body></html>
```

使用标记来增强 controller.htm 可移植性

默认情况下，站点名称硬编码到 controller.html 文件中，因此很难重命名站点或在多个站点中重用相同的定制控制器。可以使用标记来缓解此问题。以下示例显示如何使用可移植性标记。

注：

示例中的 <link> 标记仅用于演示。基线 controller.html 不包括这些内容。

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="initial-scale=1">
<script type="text/javascript">
var SCS = { sitePrefix: '[!--$SCS_SITE_PREFIX--]' };
</script>
<script src="[!--$SCS_PRODUCT_PATH--]/_sitesclouddelivery/renderer/
controller.js"></script>
<link rel="shortcut icon" href="[!--$SCS_SITE_PATH--]/favicon.ico">
<link rel="preload" href="[!--$SCS_PRODUCT_PATH--]/_sitesclouddelivery/
renderer/require.js" as="script">
<link rel="preload" href="[!--$SCS_PRODUCT_PATH--]/_sitesclouddelivery/
renderer/renderer.js" as="script">
</head>
<body id="scsControllerBody"><noscript>Please enable JavaScript to view
this site properly.</noscript>

</body></html>
```

第 V 部分

使用其他工具进行站点开发

Oracle Content Management 提供了几种方式与在 Oracle Content Management 内部或 Oracle Content Management 外部使用其他第三方工具开发的应用程序和体验进行交互。

使用 OCE 工具包

Oracle Content Management OCE 工具包和 SDK 可帮助您开发定制应用程序，以使用 Oracle Content Management 资料档案库中管理的内容。您可以在 Oracle Content Management 中或使用第三方工具开发这些应用程序。

借助 OCE 工具包可以执行以下任务：

- 设置本地开发环境以使用 Oracle Content Management 实例在本地开发和测试组件、模板、主题和内容布局
- 从示例创建组件、站点模板和内容布局，在测试工具中运行它们，浏览它们以及在开发人员云服务环境中开发组件、模板、主题和内容布局
- 将在 Oracle Content Management 中创建的组件和站点模板导入到开发人员云服务项目和环境以进行源管理和进一步开发
- 从开发人员云服务环境导出组件、模板和内容布局，以在 Oracle Content Management 中使用
- 复制现有组件、模板或内容布局
- 编写单元测试
- 优化组件
- 将组件和模板部署到 Oracle Content Management

您可以从 <https://github.com/oracle/content-and-experience-toolkit#readme> 中获取更多信息。

体验业务流程

如果您使用 Oracle Content Management 以外的工具来创建体验，可将 Oracle Content Management 资料档案库连接到这些体验。这样内容创建者可以在工作时预览站点的更改，并在内容变化或发布时触发自动构建。该体验业务流程可使内容提供方和站点开发人员之间的工作流自动化，从而简化体验管理和发布。

25

使用 OCE 工具包进行开发

OCE 工具包可帮助您为 Oracle Content Management 开发站点模板、主题、定制组件和内容布局。

使用 OCE 工具包，您可以在自己的开发环境中工作，并且可以使用 Oracle Content Management 中的资产资料档案库、文件和文件夹。OCE 工具包提供了一些工具，可用于创建和开发定制组件和站点模板（包括主题和内容布局）。它包含用于快速迭代开发的本地测试工具和供开始使用的示例单元测试。

以下主题介绍了如何设置 OCE 工具包，并在本地计算机上或作为开发人员云服务项目使用该工具包进行开发：

- [在本地计算机上设置 OCE 工具包](#)
- [升级到 jQuery 3.5.x](#)
- [使用开发人员云服务为 Oracle Content Management 进行开发](#)
- [使用 OCE 工具包将更改从测试环境传播到生产环境](#)
- [从模板创建站点并为内容保持相同 GUID](#)
- [导入和导出分类](#)
- [导入和导出推荐](#)
- [使用 OCE 工具包开发定制字段编辑器](#)
- [将站点从一个服务器传输或更新到另一个服务器](#)
- [使用 OCE 工具包为站点页编制索引](#)
- [使用 OCE 工具包为多语言站点编制索引](#)
- [创建简化的组件以便于组件开发](#)
- [编译站点以提高站点页的运行时性能](#)
- [在 Oracle Content Management 服务器中创建新站点或资产翻译作业](#)
- [通过语言服务提供商翻译站点](#)

在本地计算机上设置 OCE 工具包

在本地计算机上，可以从 GitHub 下载 OCE 工具包。

要设置 OCE 工具包，请按照位于以下网址的说明操作：

<https://github.com/oracle/content-and-experience-toolkit/blob/master/README.md>

完成设置并准备使用 OCE 工具包：

1. [通过 npm 安装相关项](#)

2. 使用 `cec` 命令行实用程序
3. 使用本地测试工具进行测试

通过 npm 安装相关项

使用 npm (node package manager, 节点程序包管理器) 为您的项目安装站点相关项。

如果您使用代理访问 Internet, 请通过 `npm config` 命令为 npm 设置代理。请参见 <https://docs.npmjs.com/misc/config>。要为 bower 设置代理, 请参见 <https://bower.io/docs/config/>。



注:

确保在本地计算机上安装了 Node.js 8.0.0 或更高版本 (<https://nodejs.org/>)。

使用 cec 命令行实用程序

`cec` 跨平台命令行实用程序提供用于创建和管理模板及组件的命令。

使用 `cec` 命令行实用程序之前, 针对需要连接到 Oracle Content Management 服务器的命令, 在您的本地开发环境中创建源。

集成帮助提供了用于运行命令的信息以及示例。要查看 `cec` 集成帮助, 可以在命令行中键入 `cec` 命令:

- 在终端窗口中, 转到 `cec` 目录。
- 键入 `cec` 以列出 `cec` 命令。

```
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
                                           [alias: cfd]
  or folder hierarchy on OCM server.
  cec share-folder <name>                Shares folder with
                                           [alias: sfd]
  users and groups on OCM server.
  cec unshare-folder <name>              Deletes user or
                                           [alias: usfd]
  group access to a shared folder on OCM server.
  cec list-folder <path>                  Displays folder
                                           [alias: lfd]
  hierarchy on OCM server.
  cec download-folder <path>              Downloads folder
                                           [alias: dlfd]
  from OCM server.
  cec upload-folder <path>                Uploads folder to
  OCM server.                             [alias: ulfd]
```

<pre>cec delete-folder <path> server. cec download-file <file> from OCM server. cec upload-file <file> OCM server. cec delete-file <file> server.</pre>	<pre>Deletes folder on OCM [alias:] Downloads file <file> [alias: dlf] Uploads file <file> to [alias: ulf] Deletes file on OCM [alias:]</pre>
---	---

Components

<pre>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.</pre>	<pre>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]</pre>
---	--

Templates

<pre>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> <name> on the OCM server. cec share-template <name> users and groups on OCM server. cec unshare-template <name> access to a template on OCM server. cec update-template <action> local template. cec describe-template <name></pre>	<pre>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 [alias:] Shares template with [alias: stm] Deletes user or group [alias: ustm] Performs action on a [alias: ut] Describes the template</pre>
--	--

```

<name> package.                                     [alias: dst]
    cec create-template-report <name>             Generates an asset
usage report for the template <name> package.     [alias: cttr]

Themes
    cec add-component-to-theme <component>        Adds a component
to a theme.                                       [alias: actt]
    cec remove-component-from-theme <component>   Removes a
component from a theme.                           [alias:
rcft]
    cec control-theme <action>                   Performs action
<action> on theme on OCM server.                 [alias: ctt]
    cec share-theme <name>                       Shares theme with
users and groups on OCM server.                 [alias: sth]
    cec unshare-theme <name>                     Deletes user or
group access to a theme on OCM server.          [alias: usth]

Sites
    cec create-site <name>                        Creates Enterprise
Site <name>.                                       [alias: cs]
    cec copy-site <name>                          Copies Enterprise
Site <name>.                                       [alias: cps]
    cec update-site <name>                        Update Enterprise
Site <name>.                                       [alias: us]
    cec transfer-site <name>                      Transfers a site
from one OCM server to another.                  [alias: ts]
    cec transfer-site-content <name>             Creates scripts to
transfer site content from one OCM server to another. [alias: tsc]
    cec validate-site <name>                     Validates site
<name>.                                           [alias: vs]
    cec control-site <action>                    Performs action
<action> on site on OCM server.                 [alias: cts]
    cec share-site <name>                        Shares site with
users and groups on OCM server.                 [alias: ss]
    cec unshare-site <name>                      Deletes user or
group access to a site on OCM server.           [alias: uss]
    cec get-site-security <name>                 Gets site security
on OCM server.                                   [alias: gss]
    cec set-site-security <name>                 Sets site security
on OCM server.                                   [alias: sss]
    cec index-site <site>                        Index the page
content of site <site> on OCM server.           [alias: is]
    cec create-site-map <site>                   Creates a site map
for site <site> on OCM server.                  [alias: csm]
    cec create-rss-feed <site>                   Creates RSS feed
for site <site> on OCM server.                  [alias: crf]
    cec create-asset-report <site>               Generates an asset
usage report for site <site> on OCM server.     [alias: car]
    cec upload-static-site-files <path>          Uploads files to
render statically from a site on OCM server.    [alias: ulss]
    cec download-static-site-files <site>        Downloads the
static files from a site on OCM server.         [alias:
dlss]
    cec delete-static-site-files <site>          Deletes the static
files from a site on OCM server.               [alias: ]

```

cec refresh-prerender-cache <site> cache for a site on OCM server.	Refreshes pre-render [alias: rpc]
cec migrate-site <site> OCI IC server to EC server.	Migrates a site from [alias: ms]
Assets	
cec download-content OCM server.	Downloads content from [alias: dlc]
cec upload-content <name> to a repository on OCM server.	Uploads local content [alias: ulc]
cec control-content <action> <action> on channel items on OCM server.	Performs action [alias: ctct]
cec transfer-content <repository> transfer content from one OCM server to another.	Creates scripts to [alias: tc]
cec list-assets server.	Lists assets on OCM [alias: la]
cec create-digital-asset asset	Creates digital [alias: cda]
cec update-digital-asset <id> asset	Updates digital [alias: uda]
cec copy-assets <repository> another repository on OCM server.	Copies assets to [alias: ca]
cec create-asset-usage-report <assets> usage report for assets on OCM server.	Generates an asset [alias: caur]
Content	
cec create-repository <name> OCM server.	Creates a repository on [alias: cr]
cec control-repository <action> <action> on repositories on OCM server.	Performs action [alias: ctr]
cec share-repository <name> users and groups on OCM server.	Shares repository with [alias: sr]
cec unshare-repository <name> access to a repository on OCM server.	Deletes user or group [alias: usr]
cec create-channel <name> OCM server.	Creates a channel on [alias: cch]
cec share-channel <name> users and groups on OCM server.	Shares channel with [alias: sch]
cec unshare-channel <name> access to a channel on OCM server.	Deletes user or group [alias: usch]
cec create-localization-policy <name> policy on OCM server.	Creates a localization [alias: clp]
cec list-server-content-types from server.	Lists all content types [alias: lsct]
cec share-type <name> and groups on OCM server.	Shares type with users [alias: st]
cec unshare-type <name> access to a type on OCM server.	Deletes user or group [alias: ust]
cec download-type <name> OCM server.	Downloads types from [alias: dltp]
cec upload-type <name> server.	Uploads types to OCM [alias: ultp]
cec update-type <action> <action> on a type	Performs action [alias: utp]
cec download-recommendation <name>	Downloads a

```

recommendation from the OCM server.                                [alias:
dlr]
    cec upload-recommendation <name>                               Uploads a
recommendation to the OCM server.
[alias:  ulr]
    cec create-contentlayout <name>                                 Creates a content
layout based on a content type.                                     [alias:  ccl]
    cec add-contentlayout-mapping <contentlayout>                  Creates content
type and content layout mapping.                                   [alias:  aclm]
    cec remove-contentlayout-mapping <contentlayout>              Removes a content
layout mapping.                                                   [alias:  rclm]
    cec add-field-editor <name>                                     Adds a field
editor to a field in a content type.                               [alias:
afe]
    cec remove-field-editor <name>                                 Removes a field
editor from a field in a content type.                             [alias:  rfe]
    cec migrate-content <name>                                     Migrates content
from OCI IC server to EC server.                                   [alias:  mc]
    cec compile-content                                           Compiles the
content items generating HTML renditions.                          [alias:
cmpc]
    cec upload-compiled-content <path>                             Uploads the
compiled content to OCM server.                                    [alias:
ulcc]

Taxonomies
    cec download-taxonomy <name>                                   Downloads a
taxonomy from OCM server.                                         [alias:
dltx]
    cec upload-taxonomy <taxonomy>                                 Uploads a taxonomy
to OCM server.                                                    [alias:  ultx]
    cec control-taxonomy <action>                                  Performs action on
taxonomy on OCM server.                                           [alias:  cttx]

Translation
    cec list-translation-jobs                                       Lists translation
jobs.                                                               [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.
[alias:  rtj]
    cec ingest-translation-job <name>                               Gets translated
job <name> from translation connection and ingest.                 [alias:  itj]
    cec upload-translation-job <name>                               Uploads
translation job <name> to OCM server.
[alias:  utj]
    cec create-translation-connector <name>                        Creates

```

```
translation connector <name>.                                [alias:
ctc]
  cec start-translation-connector <name>                    Starts translation
connector <name>.                                           [alias: stc]
  cec register-translation-connector <name>                  Registers a translation
connector.                                                    [alias: rtc]
```

Groups

```
  cec create-group <name>                                     Creates an OCM group on
OCM server.                                                  [alias: cg]
  cec delete-group <name>                                     Deletes an OCM group on
OCM server.                                                  [alias: ]
  cec add-member-to-group <name>                             Adds users and groups
to an OCM group on OCM server.                               [alias: amtg]
  cec remove-member-from-group <name>                       Removes users and
groups from an OCM group on OCM server.                     [alias: rmfg]
```

Local Environment

```
  cec create-encryption-key <file>                          Create an encryption
key to encrypt/decrypt password for servers.                [alias: cek]
  cec register-server <name>                                 Registers a OCM
server.                                                       [alias: rs]
  cec set-oauth-token <token>                               Set OAuth token for a
registered server.                                           [alias: sot]
  cec list                                                    Lists local or server
resources.                                                    [alias: l]
  cec execute-get <endpoint>                                 Makes an HTTP GET
request to a REST API endpoint on OCM server                 [alias: exeg]
  cec install                                                 Creates source
tree.                                                         [alias: i]
  cec develop                                                 Starts a test
server.                                                       [alias: d]
  cec sync-server                                             Starts a sync
server.                                                       [alias: scs]
  cec webhook-server                                         Starts a webhook
server.                                                       [alias: whs]
```

Options:

```
--version, -v Show version number [boolean]
--help, -h Show Help
```

```
-----
cec create-folder
-----
```

```
Usage: cec create-folder <name>
```

Create a folder or folder hierarchy on OCM server. Specify the server with -s <server> or use the one specified in cec.properties file.

Options:

```
--help, -h Show Help
--server, -s <server> The registered OCM server
```

Examples:

```
cec create-folder Projects          Creates folder Projects under the Home
```

```
folder
  cec create-folder Projects/Blogs  Creates folder Projects under the
  Home folder and folder Blogs under Projects
  cec create-folder Projects -s UAT  Creates folder Projects under the
  Home folder on the registered server UAT
```

```
-----
cec share-folder
-----
```

Usage: cec share-folder <name>

Shares folder with users and groups on OCM server and assign a role. Specify the server with -s <server> or use the one specified in cec.properties file. The valid roles are

```
manager
contributor
downloader
viewer
```

Options:

```
--help, -h    Show Help
--users, -u    The comma separated list of user names
--groups, -g   The comma separated list of group names
--role, -r     The role [manager | contributor | downloader | viewer]
to assign to the users or groups [required]
--server, -s  <server> The registered OCM server
```

Examples:

```
cec share-folder Projects/Blogs -u user1,user2 -r manager
Share folder Projects/Blogs with user user1 and user2 and assign
Manager role to them
cec share-folder Projects/Blogs -u user1,user2 -g group1 -r manager
Share folder Projects/Blogs with user user1, user2 and group group1 and
assign Manager role to them
cec share-folder Projects/Blogs -g group1,group2 -r manager
Share folder Projects/Blogs with group group1 and group2 and assign
Manager role to them
cec share-folder Projects/Blogs -u user1,user2 -r manager -s UAT
Share folder Projects/Blogs with user user1 and user2 and assign
Manager role to them on the registered server UAT
```

```
-----
cec unshare-folder
-----
```

Usage: cec unshare-folder <name>

Deletes user or group access to a shared folder on OCM server. Specify the server with -s <server> or use the one specified in cec.properties file.

Options:

```
--help, -h    Show Help
--users, -u    The comma separated list of user names
```



```
--groups, -g The comma separated list of group names
--server, -s <server> The registered OCM server
```

Examples:

```
cec unshare-folder Projects/Blogs -u user1,user2
cec unshare-folder Projects/Blogs -g group1,group2
cec unshare-folder Projects/Blogs -u user1,user2 -g group1,group2
cec unshare-folder Projects/Blogs -u user1,user2 -s UAT
```

```
-----
cec list-folder
-----
```

Usage: cec list-folder <path>

Displays folder and all its content on OCM server. Specify the server with -s <server> or use the one specified in cec.properties file.

Options:

```
--help, -h Show Help
--server, -s The registered OCM server
```

Examples:

```
cec list-folder Releases/1
cec list-folder Releases/1 -s UAT
cec list-folder site:blog1
cec list-folder theme:blog1Theme
cec list-folder component:Comp1/assets
```

```
-----
cec download-folder
-----
```

Usage: cec download-folder <path>

Downloads folder and all its content from OCM server. Specify the server with -s <server> or use the one specified in cec.properties file. Optionally specify -f <folder> to save the folder on the local system.

Options:

```
--help, -h Show Help
--folder, -f <folder> Local folder to save the folder on OCM server
--server, -s <server> The registered OCM server
```

Examples:

```
cec download-folder Releases/1 Downloads folder
Releases/1 from OCM server and save to local folder src/documents/
cec download-folder / Downloads all
documents from OCM server and save to local folder src/documents/
cec download-folder Releases/1 -s UAT Downloads folder
Releases/1 from the registered server UAT and save to local folder src/
documents/
cec download-folder Releases/1 -f ~/Downloads Downloads folder
Releases/1 from OCM server and save to local folder ~/Download/
cec download-folder Releases/1 -f . Downloads folder
Releases/1 from OCM server and save to the current local folder
cec download-folder site:blog1 -f ~/Downloads/blog1Files Downloads all
```

```
files of site blog1 and save to local folder ~/Download/blog1Files
cec download-folder theme:blog1Theme Downloads
all files of theme blog1Theme and save to local folder src/documents/
blog1Theme/
cec download-folder component:Compl/assets Downloads
all files in folder assets of component Compl and save to local folder
src/documents/Compl/assets/
```

```
-----
cec upload-folder
-----
Usage: cec upload-folder <path>
```

Uploads folder and all its content to OCM server. Specify the server with `-s <server>` or use the one specified in `cec.properties` file. Optionally specify `-f <folder>` to set the parent folder on OCM server.

Options:

```
--help, -h Show Help
--folder, -f <folder> The parent folder on OCM server
--server, -s <server> The registered OCM server
```

Examples:

```
cec upload-folder ~/Downloads/docs
Uploads all content from ~/Downloads/docs to folder docs on the server
cec upload-folder ~/Downloads/docs/
Uploads all content from ~/Downloads/docs to the Home folder on the
server
cec upload-folder ~/Downloads/docs -f Mydoc
Uploads all content from ~/Downloads/docs to folder Mydoc/docs on the
server
cec upload-folder ~/Downloads/docs/ -f Mydoc
Uploads all content from ~/Downloads/docs to folder Mydoc on the server
cec upload-folder ~/Downloads/docs -s UAT
Uploads all content from ~/Downloads/docs to folder docs on the
registered server UAT
cec upload-folder ~/Downloads/docs/ -f site:blog1/settings/misc
Uploads all content from ~/Downloads/docs to folder settings/misc of
site blog1
cec upload-folder ~/Downloads/docs -f theme:blog1Theme
Uploads all content from ~/Downloads/docs to folder docs of theme
blog1Theme
cec upload-folder ~/Downloads/docs -f component:Compl
Uploads all content from ~/Downloads/docs to folder docs of component
Compl
```

```
-----
cec delete-folder
-----
Usage: cec delete-folder <path>
```

Deletes folder and all its content on OCM server. Specify the server with `-s <server>` or use the one specified in `cec.properties` file. Optionally specify `-p` to permanently delete the folder.

Options:
 --help, -h Show Help
 --server, -s <server> The registered OCM server
 --permanent, -p Delete the folder permanently

Examples:
 cec delete-folder Import/docs
 cec delete-folder Import/docs -s UAT
 cec delete-folder Import/docs -p
 cec delete-folder site:blog1/docs
 cec delete-folder theme:blog1Theme/docs
 cec delete-folder component:Comp1/docs

 cec download-file

Usage: cec download-file <file>

Downloads file <file> from OCM server. Specify the server with -s <server> or use the one specified in cec.properties file. Optionally specify -f <folder> to save the file on the local system.

Options:
 --help, -h Show Help
 --folder, -f <folder> Local folder to save the file
 --server, -s <server> The registered OCM server

Examples:
 cec download-file Releases/Projects.pdf Downloads
 the file from OCM server and save to local folder src/documents/
 cec download-file Releases/Projects.pdf -s UAT Downloads
 the file from the registered server UAT and save to local folder src/
 documents/
 cec download-file Releases/Projects.pdf -f ~/Downloads Downloads
 the file from OCM server and save to local folder ~/Download/
 cec download-file Releases/Projects.pdf -f . Downloads
 the file from OCM server and save to the current local folder
 cec download-file site:blog1/siteinfo.json Downloads
 the file from folder blog1 and save to local folder src/documents/blog1
 cec download-file theme:blog1Theme/designs/default/design.css Downloads
 the css file from folder designs/default of theme blog1Theme and save to
 local folder src/documents/blog1Theme/designs/default/
 cec download-file component:Comp1/assets/render.js Downloads
 the js file from folder assets of component Comp1 and save to local folder
 src/documents/Comp1/assets/

 cec upload-file

Usage: cec upload-file <file>

Uploads file <file> to OCM server. Specify the server with -s <server> or use the one specified in cec.properties file. Optionally specify -f <folder> to set the parent folder on OCM server.

Options:

```
--help, -h      Show Help
--folder, -f    <folder> The parent folder on OCM server
--server, -s    <server> The registered OCM server
```

Examples:

```
cec upload-file ~/Documents/
Projects.pdf           Uploads the file to the
Home folder
cec upload-file ~/Documents/Projects.pdf -s
UAT                   Uploads the file to the Home folder on
the registered server UAT
cec upload-file ~/Documents/Projects.pdf -f Doc/
Plan                 Uploads the file to folder Doc/Plan
cec upload-file ~/Documents/Projects.pdf -f site:blog1/settings/
misc                Uploads the file to folder settings/misc of site blog1
cec upload-file ~/Documents/style1.css -f theme:blog1Theme/designs/
default             Uploads the css file to folder designs/default of theme
blog1Theme
cec upload-file ~/Documents/compl.js -f component:Compl/
assets              Uploads the js file to folder assets of component
Compl
```

```
-----
cec delete-file
-----
```

Usage: cec delete-file <file>

Deletes file on OCM server. Specify the server with -s <server> or use the one specified in cec.properties file. Optionally specify -p to permanently delete the file.

Options:

```
--help, -h      Show Help
--server, -s    <server> The registered OCM server
--permanent, -p Delete the file permanently
```

Examples:

```
cec delete-file docs/Projects.pdf
cec delete-file docs/Projects.pdf -s UAT
cec delete-file docs/Projects.pdf -p
cec delete-file site:blog1/docs/Projects.pdf
cec delete-file theme:blog1Theme/docs/Projects.pdf
cec delete-file component:Compl/docs/Projects.pdf
```

```
-----
cec create-component
-----
```

Usage: cec create-component <name>

Creates the component <name>. By default, it creates a local component. Optionally specify -f <source> to create from a different source.

Valid values for <source> are:

```
local
```

```

local-template
local-iframe
remote
sectionlayout
Sample-File-List
Sample-Folder-List
Sample-Documents-Manager
Sample-Process-Start-Form
Sample-Process-Task-List
Sample-Process-Task-Details
Sample-Stocks-Embedded
Sample-Text-With-Image
Sample-To-Do
Anchor
ContentForm
Document-Search
JET-CCA-Demo-Card
MapFieldEditor
Sample-OPA-Interview
SimpleHTML
SliderFieldEditor
TextFieldEditor

```

Options:

```

--help, -h Show Help
--from, -f <from> Source to create from

```

Examples:

```

cec create-component Comp1
cec create-component Comp2 -f Sample-File-List

```

```

-----
cec copy-component
-----

```

Usage: cec copy-component <source> [<destination>]

Copies an existing component named <source> to <destination>. <source> is a folder name from src/components

Options:

```

--help, -h Show Help

```

Examples:

```

cec copy-component Sample-To-Do 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/Comp1.zip Imports the component Comp1.
```

```
-----  
cec export-component  
-----
```

Usage: cec export-component <name>

Exports the component <name> as a zip file.

Options:

```
--help, -h Show Help
```

Examples:

```
cec export-component Sample-To-Do Exports the component Sample-To-Do.
```

```
-----  
cec download-component  
-----
```

Usage: cec download-component <names>

Downloads the components <names> from the Content Management server. Specify the server with -s <server> or use the one specified in cec.properties file.

Options:

```
--help, -h Show Help  
--server, -s <server> The registered OCM server
```

Examples:

```
cec download-component Sample-To-Do  
cec download-component Sample-To-Do,Sample-To-Do2  
cec download-component Sample-To-Do -s UAT
```

```
-----  
cec upload-component  
-----
```

Usage: cec upload-component <names>

Uploads the components <names> to the Content Management server. Specify the server with -s <server> or use the one specified in cec.properties file. Optionally specify -p to publish the component after deploy. Optionally specify -f <folder> to set the folder to upload the component zip file.

Options:

```
--help, -h Show Help  
--folder, -f <folder> Folder to upload the component zip file  
--publish, -p Publish the component  
--server, -s <server> The registered OCM server
```

Examples:

```
cec upload-component Sample-To-Do Uploads the
```

```

component Sample-To-Do to the server specified in cec.properties.
  cec upload-component Sample-To-Do -s UAT           Uploads the
component Sample-To-Do to the registered server UAT.
  cec upload-component Sample-To-Do -p             Uploads and
publishes the component Sample-To-Do.
  cec upload-component Sample-To-Do,Sample-To-Do2   Uploads component
Sample-To-Do and Sample-To-Do2.
  cec upload-component Sample-To-Do -f Import/Components Uploads file
Sample-To-Do.zip to folder Import/Components and imports the component
Sample-To-Do.

```

```

-----
cec control-component
-----

```

Usage: cec control-component <action>

Perform <action> on components on OCM server. Specify the components with -c <components>. Specify the server with -s <server> or use the one specified in cec.properties file. The valid actions are

publish

Options:

```

--help, -h          Show Help
--components, -c   <components> The comma separated list of components
[required]
--server, -s       <server> The registered OCM server

```

Examples:

```

  cec control-component publish -c Comp1           Publish component
Comp1 on the server specified in cec.properties file
  cec control-component publish -c Comp1 -s UAT   Publish component
Comp1 on the registered server UAT
  cec control-component publish -c Comp1,Comp2 -s UAT Publish component
Comp1 and Comp2 on the registered server UAT

```

```

-----
cec share-component
-----

```

Usage: cec share-component <name>

Shares component with users and groups on OCM server and assign a role. Specify the server with -s <server> or use the one specified in cec.properties file. The valid roles are

manager
contributor
downloader
viewer

Options:

```

--help, -h          Show Help
--users, -u        The comma separated list of user names

```

```
--groups, -g The comma separated list of group names
--role, -r The role [manager | contributor | downloader | viewer]
to assign to the users or groups [required]
--server, -s <server> The registered OCM server
```

Examples:

```
cec share-component Comp1 -u user1,user2 -r manager
Share component Comp1 with user user1 and user2 and assign Manager role
to them
cec share-component Comp1 -u user1,user2 -g group1,group2 -r manager
Share component Comp1 with user user1 and user2 and group group1 and
group2 and assign Manager role to them
cec share-component Comp1 -u user1,user2 -r manager -s UAT
Share component Comp1 with user user1 and user2 and assign Manager role
to them on the registered server UAT
```

```
-----
cec unshare-component
-----
```

Usage: cec unshare-component <name>

Deletes user or group access to a component on OCM server. Specify the server with -s <server> or use the one specified in cec.properties file.

Options:

```
--help, -h Show Help
--users, -u The comma separated list of user names
--groups, -g The comma separated list of group names
--server, -s <server> The registered OCM server
```

Examples:

```
cec unshare-component Comp1 -u user1,user2
cec unshare-component Comp1 -u user1,user2 -g group1,group2
cec unshare-component Comp1 -u user1,user2 -s UAT
```

```
-----
cec create-template
-----
```

Usage: cec create-template <name>

Creates the template <name>. By default, it creates a StarterTemplate. Optionally specify -f <source> to create from different source.

Valid values for <source> are:

```
CafeSupremoLite
JETStarterTemplate
StarterTemplate
BlogTemplate
VBCSSamplesTemplate
search_template
```

To create template based on a site on OCM server, specify -s <site> and specify the server with -r <server> or use the one specified in cec.properties file.

Options:

```
--help, -h          Show Help
--from, -f          <source> Source to create from
--site, -s          <site> Site to create from
--excludecontent, -x  Exclude content
--enterprisetemplate, -e  Enterprise template
--server, -r        <server> The registered OCM server
```

Examples:

```
cec create-template Temp1
cec create-template Temp2 -f CafeSupremoLite
cec create-template Temp1 -s Site1          Create template
Temp1 based on site Site1 on OCM server
cec create-template Temp1 -s Site1 -x      Create template
Temp1 based on site Site1 on OCM server and exclude the content in the site
cec create-template Temp1 -s Site1 -r UAT  Create template
Temp1 based on site Site1 on the registered server UAT
cec create-template EnterpriseTemp1 -s StandardSite1 -e  Create enterprise
template EnterpriseTemp1 based on standard site StandardSite1 on OCM server
```

```
-----
cec create-template-from-site
-----
```

Usage: cec create-template-from-site <name>

Creates the template <name> from site <site> on the Content Management server. Specify the server with -r <server> or use the one specified in cec.properties file. Optionally specify <includeunpublishedassets> to include unpublished content items and digital assets in your template.

Options:

```
--help, -h          Show Help
--site, -s          <site> Site to create from [required]
--includeunpublishedassets, -i  flag to indicate to include unpublished
content items and digital assets in your template
--enterprisetemplate, -e      Enterprise template
--server, -r        <server> The registered OCM server
```

Examples:

```
cec create-template-from-site BlogTemplate -s BlogSite
cec create-template-from-site BlogTemplate -s BlogSite -r UAT
cec create-template-from-site BlogTemplate -s BlogSite -i -r UAT
cec create-template-from-site EnterpriseTemplate -s StandardSite -e
```

```
-----
cec download-template
-----
```

Usage: cec download-template <name>

Downloads the template <name> from the Content Management server. Specify the server with -s <server> or use the one specified in cec.properties file.

Options:

```
--help, -h      Show Help
--server, -s    <server> The registered OCM server
```

Examples:

```
cec download-template BlogTemplate
cec download-template BlogTemplate -s UAT
```

```
-----
cec compile-template
-----
```

Usage: cec compile-template <source>

Compiles all the pages within the site of the template and places the compiled pages under the sites assets folder.

Optionally specify `-s <server>` to make content queries against this server (requires `channelToken`).

Optionally specify `-c <channelToken>` to use this `channelToken` when generating any content URLs.

Optionally specify `-t <contentType>` [draft | published] content to retrieve from the server type, defaults to published.

Optionally specify `-p <pages>` the set of pages to compile.

Optionally specify `-d <debug>` to start the compilation with `--inspect-brk` flag.

Optionally specify `-r <recurse>` recurse through all child pages of specified pages.

Optionally specify `-l <includeLocale>` include default locale when creating pages.

Optionally specify `-a <targetDevice>` [desktop | mobile] target device type when using adaptive layouts.

Optionally specify `-v <verbose>` to display all warning messages during compilation.

Optionally specify `-i <ignoreErrors>` ignore compilation errors when calculating the exit code for the process.

Options:

<code>--help, -h</code>	Show Help
<code>--server, -s</code>	The registered OCM server
<code>--channelToken, -c</code>	The channel access token to use for content URLs
<code>--type, -t</code>	The type of content to retrieve from the serve [published draft]
<code>--pages, -p</code>	The list of pages to compile
<code>--recurse, -r</code>	Compile all child pages of those specified in the page list
<code>--debug, -d</code>	Start the compiler with " <code>--inspect-brk</code> " option to debug compilation
<code>--noDetailPages, -e</code>	Do not generate compiled detail pages
<code>--noDefaultDetailPageLink, -o</code>	Do not generate compiled detail page for items/content lists that use the default detail page
<code>--targetDevice, -a</code>	The target device type when using adaptive layouts [desktop mobile]
<code>--siteName, -n</code>	The target site name to use when compiling the template
<code>--secureSite, -u</code>	The target site is a secure site
<code>--includeLocale, -l</code>	Include default locale when creating pages

--verbose, -v Run in verbose mode to display all warning messages during compilation.
--ignoreErrors, -i Ignore compilation errors when calculating the exit code for the process.

Examples:

cec compile-template Templ Compiles the site in template Templ using content stored in the template.
cec compile-template Templ -c channelToken Compiles the site in template Templ using the given channelToken for any content URLs.
cec compile-template Templ -c channelToken -s UAT -t draft Compiles the site in template Templ retrieving draft content from the specified server.
cec compile-template Templ -p 104,112,183 -r Compiles the specified pages in the site in template Templ including all child pages.
cec compile-template Templ -d Waits for the debugger to be attached. Once attached, compiles the site in template Templ.

cec copy-template

Usage: cec copy-template <source> [<destination>]

Copies an existing template named <source> to <destination>. <source> is a folder name from src/templates

Options:

--help, -h Show Help

Examples:

cec copy-template Templ Temp2 Copies Templ to Temp2.

cec import-template

Usage: cec import-template <zip>

Imports a template from <zip>. Specify the absolute path of the zip file. The zip file name will be used as the template name.

Options:

--help, -h Show Help

Examples:

cec import-template /home/Templ.zip Imports the template Templ.

cec export-template

Usage: cec export-template <name>

Exports the template <name> as a zip file and provides the location of the zip file.

Options:

--help, -h Show Help

--optimize, -o Optimize the template

Examples:

cec export-template Templ Exports the template Templ.

cec upload-template

Usage: cec upload-template <name>

Uploads the template <name> to the Content Management server. Specify the server with -s <server> or use the one specified in cec.properties file. Optionally specify -f <folder> to set the folder to upload the template zip file. Optionally specify -p to publish theme and components after import.

Options:

--help, -h	Show Help
--folder, -f	<folder> Folder to upload the template zip file
--server, -s	<server> The registered OCM server
--optimize, -o	Optimize the template
--excludecontenttemplate, -x	Exclude content template
--excludecomponents, -e	Exclude components
--publish, -p	Publish theme and components

Examples:

cec upload-template StarterTemplate	Uploads the template StarterTemplate.
cec upload-template StarterTemplate -s UAT	Uploads the template StarterTemplate to the registered server UAT.
cec upload-template StarterTemplate -f Import/Templates	Uploads file StarterTemplate.zip to folder Import/Templates and imports the template StarterTemplate.
cec upload-template StarterTemplate -p	Publish the theme and all components in StarterTemplate.zip after import
cec upload-template StarterTemplate -o	Optimizes and uploads the template StarterTemplate.
cec upload-template StarterTemplate -x	Exclude the "Content Template" from the template upload. "Content Template" upload can be managed independently.
cec upload-template StarterTemplate -e	Exclude all components from the template upload. Components can be uploaded independently.

cec delete-template

Usage: cec delete-template <name>

Deletes the template <name> on the Content Management server. Specify the server with -s <server> or use the one specified in cec.properties file. Optionally specify -p to permanently delete the template.

Options:

```
--help, -h      Show Help
--server, -s    <server> The registered OCM server
--permanent, -p  flag to indicate to permanently delete the template
```

Examples:

```
cec delete-template BlogTemplate
cec delete-template BlogTemplate -p
cec delete-template BlogTemplate -s UAT
```

```
-----
cec share-template
-----
```

Usage: cec share-template <name>

Shares template with users and groups on OCM server and assign a role. Specify the server with -s <server> or use the one specified in cec.properties file. The valid roles are

```
manager
contributor
downloader
viewer
```

Options:

```
--help, -h      Show Help
--users, -u     The comma separated list of user names
--groups, -g    The comma separated list of group names
--role, -r      The role [manager | contributor | downloader | viewer] to
assign to the users or groups [required]
--server, -s    <server> The registered OCM server
```

Examples:

```
cec share-template Templatel -u user1,user2 -r manager
Share template Templatel with user user1 and user2 and assign Manager role
to them
cec share-template Templatel -u user1,user2 -g group1,group2 -r manager
Share template Templatel with user user1 and user2 and group group1 and
group2 and assign Manager role to them
cec share-template Templatel -u user1,user2 -r manager -s UAT
Share template Templatel with user user1 and user2 and assign Manager role
to them on the registered server UAT
```

```
-----
cec unshare-template
-----
```

Usage: cec unshare-template <name>

Deletes user or group access to a template on OCM server. Specify the server with -s <server> or use the one specified in cec.properties file.

Options:

```
--help, -h      Show Help
--users, -u     The comma separated list of user names
--groups, -g    The comma separated list of group names
```

--server, -s <server> The registered OCM server

Examples:

```
cec unshare-template Templat1 -u user1,user2
cec unshare-template Templat1 -u user1,user2 -g group1,group2
cec unshare-template Templat1 -u user1,user2 -s UAT
```

cec update-template

Usage: cec update-template <action>

Performs action <action> on a local template. Optionally specify -c for other local content. The valid actions are

rename-asset-id

Options:

```
--help, -h      Show Help
--template, -t  The template [required]
--content, -c   The comma separated list of local content
```

Examples:

```
cec update-template rename-asset-id -t Templat1
cec update-template rename-asset-id -t Templat1 -c Content1,Content2
```

cec describe-template

Usage: cec describe-template <name>

Describes the template <name> package such as theme, components and content types.

Options:

```
--help, -h  Show Help
```

Examples:

```
cec describe-template StarterTemplate  Describes the template
StarterTemplate package
```

cec create-template-report

Usage: cec create-template-report <name>

Generates an asset usage report for the template <name> package. Optionally specify -o to save the report to a json file.

Options:

```
--help, -h      Show Help
--includepagelinks, -i  Include validating page links
--output, -o       Output the report to a JSON file
```

Examples:

```

cec create-template-report StarterTemplate
cec create-template-report StarterTemplate -
o                               The report will be saved to
StarterTemplateAssetUsage.json at the current local location
cec create-template-report StarterTemplate -o ~/
Documents                       The report will be saved to ~/
Documents/StarterTemplateAssetUsage.json
cec create-template-report StarterTemplate -o ~/Documents/
StarterTemplateReport.json The report will be saved to ~/Documents/
StarterTemplateReport.json
cec create-template-report StarterTemplate -
i                               Include validating page links

```

```

-----
cec add-component-to-theme
-----

```

Usage: cec add-component-to-theme <component>

Adds a component to a theme. Optionally specify `-c <category>` to set the component category.

Options:

```

--help, -h      Show Help
--theme, -t     <theme> Theme [required]
--category, -c <category> component category

```

Examples:

```

cec add-component-to-theme Sample-To-Do -t BlogTheme
cec add-component-to-theme Sample-To-Do -t BlogTheme -c Samples

```

```

-----
cec remove-component-from-theme
-----

```

Usage: cec remove-component-from-theme <component>

Removes a component from a theme.

Options:

```

--help, -h     Show Help
--theme, -t    <theme> Theme [required]

```

Examples:

```

cec remove-component-from-theme Sample-To-Do -t BlogTheme

```

```

-----
cec control-theme
-----

```

Usage: cec control-theme <action>

Perform <action> on theme on OCM server. Specify the theme with `-t <theme>`. Specify the server with `-s <server>` or use the one specified in `cec.properties` file. The valid actions are

```

publish

```

Options:

```
--help, -h    Show Help
--theme, -t   <theme> The theme [required]
--server, -s  <server> The registered OCM server
```

Examples:

```
cec control-theme publish -t Theme1          Publish theme Theme1 on
the server specified in cec.properties file
cec control-theme publish -t Theme1 -s UAT   Publish theme Theme1 on
the registered server UAT
```

```
-----
cec share-theme
-----
```

Usage: cec share-theme <name>

Shares theme with users and groups on OCM server and assign a role. Specify the server with -s <server> or use the one specified in cec.properties file. The valid roles are

```
manager
contributor
downloader
viewer
```

Options:

```
--help, -h    Show Help
--users, -u   The comma separated list of user names
--groups, -g  The comma separated list of group names
--role, -r    The role [manager | contributor | downloader | viewer]
to assign to the users or groups [required]
--server, -s  <server> The registered OCM server
```

Examples:

```
cec share-theme Theme1 -u user1,user2 -r manager
Share theme Theme1 with user user1 and user2 and assign Manager role to
them
cec share-theme Theme1 -u user1,user2 -g group1,group2 -r manager
Share theme Theme1 with user user1 and user2 and group group1 and
group2 and assign Manager role to them
cec share-theme Theme1 -u user1,user2 -r manager -s UAT
Share theme Theme1 with user user1 and user2 and assign Manager role to
them on the registered server UAT
```

```
-----
cec unshare-theme
-----
```

Usage: cec unshare-theme <name>

Deletes user or group access to a theme on OCM server. Specify the server with -s <server> or use the one specified in cec.properties file.

Options:

```
--help, -h      Show Help
--users, -u     The comma separated list of user names
--groups, -g    The comma separated list of group names
--server, -s    <server> The registered OCM server
```

Examples:

```
cec unshare-theme Theme1 -u user1,user2
cec unshare-theme Theme1 -u user1,user2 -g group1,group2
cec unshare-theme Theme1 -u user1,user2 -s UAT
```

```
-----
cec create-site
-----
```

Usage: cec create-site <name>

Create Enterprise Site on OCM server. Specify the server with -s <server> or use the one specified in cec.properties file.

Options:

```
--help, -h          Show Help
--template, -t      <template> Template [required]
--repository, -r    <repository> Repository, required for enterprise
site
--localizationPolicy, -l <localizationPolicy> Localization policy
--defaultLanguage, -d <defaultLanguage> Default language, required for
enterprise site
--description, -p    <description> Site description
--sitePrefix, -x    <sitePrefix> Site Prefix
--update, -u        Keep the existing id for assets
--server, -s        <server> The registered OCM server
```

Examples:

```
cec create-site Site1 -t StandardTemplate
Creates a standard site
cec create-site Site1 -t Templatel -r Repository1 -l L10NPolicy1 -d en-US
Creates an enterprise site with localization policy L10NPolicy1
cec create-site Site1 -t Templatel -r Repository1 -d en-US
Creates an enterprise site and uses the localization policy in Templatel
cec create-site Site1 -t Templatel -r Repository1 -d en-US -s UAT
Creates an enterprise site on server UAT
cec create-site Site1 -t Templatel -u -r Repository1 -d en-US -s UAT
Creates an enterprise site on server UAT and keep the existing id for assets
```

```
-----
cec copy-site
-----
```

Usage: cec copy-site <name>

Copy Enterprise Site on OCM server. Specify the server with -s <server> or use the one specified in cec.properties file. If the site uses more than one repository, only the assets from the default repository will be copied.

Options:

```
--help, -h          Show Help
```

```
--target, -t      Target site [required]
--repository, -r  Repository, required for enterprise site
--description, -d Site description
--sitePrefix, -x  Site Prefix
--server, -s      The registered OCM server
```

Examples:

```
cec copy-site Sitel -t SitelCopy           Copies a
standard site
cec copy-site Sitel -t SitelCopy -r Repository1   Copies an
enterprise site
cec copy-site Sitel -t SitelCopy -r Repository1 -x sitelc  Copies an
enterprise site and sets the site prefix to sitelc
```

```
-----
cec update-site
-----
```

Usage: cec update-site <name>

Update Enterprise Site on OCM server using the content from the template. Specify the server with -s <server> or use the one specified in cec.properties file.

Options:

```
--help, -h          Show Help
--template, -t      <template> Template [required]
--excludecontenttemplate, -x  Exclude content template
--server, -s        <server> The registered OCM server
```

Examples:

```
cec update-site Sitel -t Templatel       Updates site Sitel using the
content from template Templatel
cec update-site Sitel -t Templatel -x    Updates site Sitel using the
content from template Templatel excluding the "Content Template"
```

```
-----
cec transfer-site
-----
```

Usage: cec transfer-site <name>

Transfers a site from one OCM server to another. By default all assets are transferred, optionally specify -p to transfer only published assets. Specify the source server with -s <server> and the destination server with -d <destination>. If the site contains assets from other repositories, optionally provide the repository mapping otherwise those assets will not be transferred.

Options:

```
--help, -h          Show Help
--server, -s        The registered OCM server the site is from
[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 Site1 -s DEV -d UAT -r Repository1 -l
L10NPolicy1
Creates site Site1 on server UAT based on site Site1 on server DEV
cec transfer-site Site1 -s DEV -d UAT -r Repository1 -l L10NPolicy1 -
p
Creates site
Site1 on server UAT based on site Site1 on server DEV with published assets
cec transfer-site Site1 -s DEV -d UAT -r Repository1 -l L10NPolicy1 -
x
Creates site
Site1 on server UAT based on site Site1 on server DEV without content
cec transfer-site Site1 -s DEV -d UAT -r Repository1 -l L10NPolicy1 -
e
Creates site
Site1 on server UAT based on site Site1 on server DEV without transferring
components to server UAT
cec transfer-site Site1 -s DEV -d UAT -r Repository1 -l L10NPolicy1 -e -
c
Creates site Site1
on server UAT based on site Site1 on server DEV without transferring
components and theme to server UAT
cec transfer-site Site1 -s DEV -d UAT -r Repository1 -l L10NPolicy1 -m
"Shared Images:Shared Images,Shared Video:Shared Video" Creates site Site1
on server UAT based on site Site1 on server DEV and transfer the assets
from repository Shared Images and Shared Video
cec transfer-site Site1 -s DEV -d UAT -r Repository1 -l L10NPolicy1 -
i
Creates site
Site1 on server UAT based on site Site1 on server DEV with static files from
DEV
cec transfer-site Site1 -s DEV -d
UAT
Updates site Site1 on server UAT based on site Site1 on
server DEV
cec transfer-site StandardSite1 -s DEV -d
UAT
Creates standard site on server UAT based on site StandardSite1 on
server DEV
-----
cec transfer-site-content
-----
Usage: cec transfer-site-content <name>

```

Creates scripts to transfer Enterprise Site content from one OCM server to another. This command is used to transfer large number of content items and

the items are transferred in batches. By default the scripts will not be executed by this command. By default all assets are transferred, optionally specify `-p` to transfer only published assets. Specify the source server with `-s <server>` and the destination server with `-d <destination>`. Optionally specify `-n` for the number of items in each batch, defaults to 500. If the site contains assets from other repositories, optionally provide the repository mapping otherwise those assets will not be transferred.

Options:

```
--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 Site1 -s DEV -d UAT -r
Repository1
Generate script Site1_downloadcontent and Site1_uploadcontent
cec transfer-site-content Site1 -s DEV -d UAT -r Repository1 -
e                               Generate
script Site1_downloadcontent and Site1_uploadcontent and execute them
cec transfer-site-content Site1 -s DEV -d UAT -r Repository1 -n
200                               Set batch size
to 200 items
cec transfer-site-content Site1 -s DEV -d UAT -r Repository1 -
p                               Only the
published assets will be transferred
cec transfer-site-content Site1 -s DEV -d UAT -r Repository1 -
l                               The assets
from the site repository will be added to site default collection on
destination server
cec transfer-site-content Site1 -s DEV -d UAT -r Repository1 -m
"Shared Images:Shared Images,Shared Video:Shared Video"
```

```
-----
cec validate-site
-----
Usage: cec validate-site <name>
```

Validates site `<name>` on OCM server before publish or view publishing failure. Specify the server with `-s <server>` or use the one specified in `cec.properties` file.

Options:

```
--help, -h    Show Help
```

--server, -s <server> The registered OCM server

Examples:

```
cec validate-site Site1           Validate site Site1 on the server
specified in cec.properties file
cec validate-site Site1 -s UAT    Validate site Site1 on the registered
server UAT
```

```
-----
cec control-site
-----
```

Usage: cec control-site <action>

Perform <action> on site on OCM server. Specify the site with -s <site>. Specify the server with -r <server> or use the one specified in cec.properties file. The valid actions are

```
publish
unpublish
bring-online
take-offline
```

Options:

```
--help, -h           Show Help
--site, -s           <site> Site [required]
--usedcontentonly, -u Publish used content only
--compilesite, -c    Compile site after publish
--staticonly, -t     Only publish site static files
--fullpublish, -f    Do a full publish
--server, -r         <server> The registered OCM server
```

Examples:

```
cec control-site publish -s Site1           Publish site Site1 on the
server specified in cec.properties file
cec control-site publish -s Site1 -u       Publish the site and all
assets added to the site's pages
cec control-site publish -s Site1 -c       Compile and publish site
Site1
cec control-site publish -s Site1 -t       Only publish the static
files of site Site1
cec control-site publish -s Site1 -f       Do a full publish of Site1
cec control-site publish -s Site1 -r UAT   Publish site Site1 on the
registered server UAT
cec control-site unpublish -s Site1 -r UAT Unpublish site Site1 on the
registered server UAT
cec control-site bring-online -s Site1 -r UAT Bring site Site1 online on
the registered server UAT
cec control-site take-offline -s Site1 -r UAT Take site Site1 offline on
the registered server UAT
```

```
-----
cec share-site
-----
```

Usage: cec share-site <name>

Shares site with users and groups on OCM server and assign a role. Specify the server with `-s <server>` or use the one specified in `cec.properties` file. The valid roles are

```
manager
contributor
downloader
viewer
```

Options:

```
--help, -h      Show Help
--users, -u     The comma separated list of user names
--groups, -g    The comma separated list of group names
--role, -r      The role [manager | contributor | downloader | viewer]
to assign to the users or groups [required]
--server, -s    <server> The registered OCM server
```

Examples:

```
cec share-site Site1 -u user1,user2 -r manager
Share site Site1 with user user1 and user2 and assign Manager role to
them
cec share-site Site1 -u user1,user2 -g group1,group2 -r manager
Share site Site1 with user user1 and user2 and group group1 and group2
and assign Manager role to them
cec share-site Site1 -u user1,user2 -r manager -s UAT
Share site Site1 with user user1 and user2 and assign Manager role to
them on the registered server UAT
```

```
-----
cec unshare-site
-----
```

Usage: `cec unshare-site <name>`

Deletes user or group access to a site on OCM server. Specify the server with `-s <server>` or use the one specified in `cec.properties` file.

Options:

```
--help, -h      Show Help
--users, -u     The comma separated list of user names
--groups, -g    The comma separated list of group names
--server, -s    <server> The registered OCM server
```

Examples:

```
cec unshare-site Site1 -u user1,user2
cec unshare-site Site1 -u user1,user2 -g group1,group2
cec unshare-site Site1 -u user1,user2 -s UAT
```

```
-----
cec get-site-security
-----
```

Usage: `cec get-site-security <name>`

Gets site security on OCM server. Specify the server with `-s <server>`

or use the one specified in cec.properties file.

Options:

--help, -h Show Help
--server, -s <server> The registered OCM server

Examples:

```
cec get-site-security Site1
cec get-site-security Site1 -s UAT
```

```
-----
cec set-site-security
-----
```

Usage: cec set-site-security <name>

Makes the site publicly available to anyone, restrict the site to registered users, or restrict the site to specific users. Specify the server with -r <server> or use the one specified in cec.properties file. Optionally specify -a <access> to set who can access the site. The valid group names are

```
Cloud users
Visitors
Service users
Specific users
```

Options:

--help, -h Show Help
--signin, -s If require sign in to access site: yes | no [required]
--access, -a The comma separated list of group names
--addusers, -u The comma separated list of users to access the site
--deleteusers, -d The comma separated list of users to remove access from the site
--server, -r <server> The registered OCM server

Examples:

```
cec set-site-security Site1 -s no
make the site publicly available to anyone
cec set-site-security Site1 -s no -r UAT
make the site publicly available to anyone on server UAT
cec set-site-security Site1 -s yes
Require everyone to sign in to access this site and any authenticated user
can access
cec set-site-security Site1 -s yes -a "Visitors,Service users"
Require everyone to sign in to access this site and all service visitors and
users can access
cec set-site-security Site1 -s yes -a "Specific users" -u user1,user2
Require everyone to sign in to access this site and only user1 and user2 can
access
cec set-site-security Site1 -s yes -d user1
Remove user1's access from the site
```

```
-----
cec index-site
-----
```

Usage: cec index-site <site>

Creates content item for each page with all text on the page. If the page index content item already exists for a page, updated it with latest text on the page. Specify -c <contenttype> to set the page index content type. Optionally specify -p to publish the page index items after creation or update. Specify the server with -s <server> or use the one specified in cec.properties file.

Options:

--help, -h Show Help
--contenttype, -c <contenttype> page index content type
--publish, -p publish page index items
--server, -s <server> The registered OCM server

Examples:

```
cec index-site Site1 -c PageIndex
cec index-site Site1 -c PageIndex -p
cec index-site Site1 -c PageIndex -s UAT
```

```
-----
cec create-site-map
-----
```

Usage: cec create-site-map <site>

Creates a site map for site on OCM server. Specify the server with -s <server> or use the one specified in cec.properties file. Optionally specify -p to upload the site map to OCM server after creation. Optionally specify -c <changefreq> to define how frequently the page is likely to change. Optionally specify -t <toppagepriority> as the priority for the top level pages. Also optionally specify <file> as the file name for the site map.

The valid values for <changefreq> are:

```
always
hourly
daily
weekly
monthly
yearly
never
auto
```

Options:

--help, -h Show Help
--url, -u <url> Site URL [required]
--changefreq, -c How frequently the page is likely to change.
--file, -f Name of the generated site map file
--languages, -l <languages> The comma separated list of languages used to create the site map
--publish, -p Upload the site map to OCM server after creation


```

--toppagepriority, -t          Priority for the top level pages, a decimal
number between 0 and 1
--server, -s                   <server> The registered OCM server
--newlink, -n                  Generate new 19.3.3 detail page link
--noDefaultDetailPageLink, -o Do not generate detail page link for items/
content lists that use the default detail page

```

Examples:

```

cec create-site-map Site1 -u http://www.example.com/site1
cec create-site-map Site1 -u http://www.example.com/site1 -s UAT
cec create-site-map Site1 -u http://www.example.com/site1 -t 0.9
cec create-site-map Site1 -u http://www.example.com/site1 -f sitemap.xml
cec create-site-map Site1 -u http://www.example.com/site1 -p
cec create-site-map Site1 -u http://www.example.com/site1 -c weekly -p
cec create-site-map Site1 -u http://www.example.com/site1 -l de-DE,it-IT

```

```

-----
cec create-rss-feed
-----

```

Usage: cec create-rss-feed <site>

Creates RSS feed for site <site> on OCM server. Specify the server with -s <server> or use the one specified in cec.properties file. Optionally specify -x <template> to specify the RSS template. Optionally specify -p to upload the RSS feed to OCM server after creation.

Options:

```

--help, -h                    Show Help
--url, -u                      <url> Site URL [required]
--query, -q                    Query for content items [required]
--limit, -l                    The limit of the items returned from the query
[required]
--orderby, -o                  The order by for the query [required]
--language, -i                 The language for the query
--template, -x                 The RSS xml template
--javascript, -j               Javascript file that contains functions to process
Mustache data
--title, -t                    The RSS feed title
--description, -d              The RSS feed description
--ttl                           How long the data will last in number of minutes
--file, -f                      Name of the generated RSS feed file
--publish, -p                  Upload the RSS feed to OCM server after creation
--server, -s                   <server> The registered OCM server
--newlink, -n                  Generate new 19.3.3 detail page link

```

Examples:

```

cec create-rss-feed Site1 -u http://www.example.com/site1 -q 'type eq
"BlogType"' -l 10 -o name:asc -t "Blog RSS"
cec create-rss-feed Site1 -u http://www.example.com/site1 -q 'type eq
"BlogType"' -l 10 -o name:asc -t "Blog RSS" -x ~/Files/RSSTemplate.xml
cec create-rss-feed Site1 -u http://www.example.com/site1 -q 'type eq
"BlogType"' -l 10 -o name:asc -t "Blog RSS" -x ~/Files/RSSTemplate.xml -i fr-
FR -f rssfrFR.xml

```

```
cec create-asset-report
```

```
-----
```

```
Usage: cec create-asset-report <site>
```

Generates an asset usage report for site <site> on OCM server. Specify the server with -s <server> or use the one specified in cec.properties file. Optionally specify -o to save the report to a json file.

Options:

```
--help, -h    Show Help
--output, -o  Output the report to a JSON file
--server, -s  The registered OCM server
```

Examples:

```
cec create-asset-report Site1
cec create-asset-report Site1 -s UAT
cec create-asset-report Site1 -o                               The
report will be saved to Site1AssetUsage.json at the current local
location
cec create-asset-report Site1 -o ~/Documents                  The
report will be saved to ~/Documents/Site1AssetUsage.json
cec create-asset-report Site1 -o ~/Documents/Site1Report.json The
report will be saved to ~/Documents/Site1Report.json
```

```
-----
cec upload-static-site-files
-----
```

```
Usage: cec upload-static-site-files <path>
```

Uploads files to render statically from a site on OCM server. Specify the site <site> on the server. Specify the server with -r <server> or use the one specified in cec.properties file.

Options:

```
--help, -h    Show Help
--site, -s    The site on OCM server [required]
--server, -r  The registered OCM server
```

Examples:

```
cec upload-static-site-files ~/Documents/localBlog -s BlogSite
cec upload-static-site-files ~/Documents/localBlog -s BlogSite -r UAT
```

```
-----
cec download-static-site-files
-----
```

```
Usage: cec download-static-site-files <site>
```

Downloads the static files from a site on OCM server. Specify the server with -s <server> or use the one specified in cec.properties file. Optionally specify -f <folder> to save the files on the local system.

Options:

```
--help, -h    Show Help
--folder, -f  <folder> Local folder to save the static files
```

```
--server, -s The registered OCM server
```

Examples:

```
cec download-static-site-files BlogSite
Download the files and save to local folder src/documents/BlogSite/static
cec download-static-site-files BlogSite -f ~/Documents/BlogSite/static
Download the files and save to local folder ~/Documents/BlogSite/static
cec download-static-site-files BlogSite -s UAT
```

```
-----
cec delete-static-site-files
-----
```

```
Usage: cec delete-static-site-files <site>
```

Deletes the static files from a site on OCM server. Specify the server with -s <server> or use the one specified in cec.properties file.

Options:

```
--help, -h Show Help
--server, -s The registered OCM server
```

Examples:

```
cec delete-static-site-files BlogSite
cec delete-static-site-files BlogSite -s UAT
```

```
-----
cec refresh-prerender-cache
-----
```

```
Usage: cec refresh-prerender-cache <site>
```

Refreshes pre-render cache for a site on OCM server. Specify the server with -s <server> or use the one specified in cec.properties file.

Options:

```
--help, -h Show Help
--server, -s The registered OCM server
```

Examples:

```
cec refresh-prerender-cache BlogSite
cec refresh-prerender-cache BlogSite -s UAT
```

```
-----
cec migrate-site
-----
```

```
Usage: cec migrate-site <site>
```

Migrates a site from OCI IC server to EC server. Specify the IC server with -s <server> and the EC server with -d <destination>.

Options:

```
--help, -h Show Help
--server, -s The registered IC server the site is from
--destination, -d The registered EC server to create the site [required]
--repository, -r Repository [required]
--template, -t The site template
```

```
--name, -n          Site name
--description, -p   Site description
--sitePrefix, -x   Site Prefix
```

Examples:

```
cec migrate-site Site1 -s ICServer -d ECServer -r
Repol          Migrates site Site1 from ICServer to
ECServer
cec migrate-site Site1 -s ICServer -d ECServer -r Repol -n
newSite        Migrates site Site1 from ICServer to ECServer and
rename to newSite
cec migrate-site Site1 -d ECServer -t ~/Documents/Site1Template.zip -
r Repol Migrates site Site1 to ECServer with template
Site1Template.zip from IC server
```

```
-----
cec download-content
-----
```

Usage: cec download-content

Downloads content from OCM server. By default all assets are downloaded, optionally specify -p to download only published assets. Specify the server with -s <server> or use the one specified in cec.properties file.

Options:

```
--help, -h          Show Help
--publishedassets, -p The flag to indicate published assets only
--collection, -c    Collection name
--repository, -r    Repository name, required when <collection> is
specified
--query, -q         Query to fetch the assets
--assets, -a        The comma separated list of asset GUIDS
--assetsfile, -f    The file with an array of asset GUIDS
--name, -n          The name for this download, default to the
channel or repository name
--server, -s        The registered OCM server
```

Examples:

```
cec download-content
Site1Channel
Download all assets in channel Site1Channel and save to local folder
src/content/Site1Channel
cec download-content Site1Channel -n
Site1Assets          Download all
assets in channel Site1Channel and save to local folder src/content/
Site1Assets
cec download-content Site1Channel -
p                    Download
published assets in channel Site1Channel
cec download-content Site1Channel -s
UAT                  Download all
assets in channel Site1Channel on server UAT
cec download-content Site1Channel -q 'fields.category eq
"RECIPE"'           Download assets from the channel
```

```

Site1Channel, matching the query, plus any dependencies
  cec download-content Site1Channel -r Repol -c
Collection1                               Download assets from the
repository Repol, collection Collection1 and channel Site1Channel
  cec download-content Site1Channel -r Repol -c Collection1 -q
'fields.category eq "RECIPE"' Download assets from repository Repol,
collection Collection1 and channel Site1Channel, matching the query, plus
any dependencies
  cec download-content -a
GUID1,GUID2                               Download
asset GUID1 and GUID2 and all their dependencies
  cec download-content -r
Repol                                       Download
assets from the repository Repol

```

Please specify the channel, repository, query or assets

```

-----
cec upload-content
-----
Usage: cec upload-content <name>

```

Uploads local content from channel <name>, template <name> or local file <name> to repository <repository> on OCM server. Specify -c <channel> to add the template content to channel. Optionally specify -l <collection> to add the content to collection. Specify the server with -s <server> or use the one specified in cec.properties file.

Options:

```

--help, -h          Show Help
--repository, -r <repository> The repository for the types and items
[required]
--template, -t      Flag to indicate the content is from template
--file, -f          Flag to indicate the content is from file
--channel, -c       <channel> The channel to add the content
--collection, -l <collection> The collection to add the content
--server, -s        <server> The registered OCM server
--update, -u        Update any existing content instead of creating new items
--types, -p         Upload content types and taxonomies only

```

Examples:

```

  cec upload-content Site1Channel -r Repol                               Upload
content to repository Repol, creating new items, and add to channel
Site1Channel
  cec upload-content Site1Channel -r Repol -u                           Upload
content to repository Repol, updating existing content to create new
versions, and add to channel Site1Channel
  cec upload-content Site1Channel -r Repol -l Site1Collection           Upload
content to repository Repol and add to collection Site1Collection and
channel Site1Channel
  cec upload-content Site1Channel -r Repol -p                           Upload
content types from content SiteChannel to the server
  cec upload-content Site1Channel -r Repol -s UAT                       Upload
content to repository Repol on server UAT and add to channel Site1Channel
  cec upload-content Templatel -t -r Repol -c channel1                 Upload

```

content from template Templat1 to repository Repol and add to channel
channell

```
cec upload-content ~/Downloads/content.zip -f -r Repol -c channell
Upload content from file ~/Downloads/content.zip to repository Repol
and add to channel channell
```

```
-----
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
Channell
```

Publish all items in channel Channell on the server specified in cec.properties file

```
cec control-content publish -c Channell -a
GUID1,GUID2
```

Publish asset GUID1 and GUID2 in channel Channell

```
cec control-content publish -c Channell -s
UAT
```

Publish all items in channel Channell on the registered server

```
UAT
cec control-content unpublish -c Channell -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 UAT
Add all items in
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 Channel1 -r Repol -a GUID1,GUID2 -s
UAT Add asset GUID1
and GUID2 in repository Repol to channel Channel1
cec control-content remove -c Channel1 -s
UAT
Remove all items in channel Channel1 on the registered server UAT
cec control-content add -l Collection1 -r Repol -s
UAT Add
all items in repository Repol to collection Collection1 on the registered
server UAT
cec control-content remove -l Collection -s
UAT
Remove all items in collection Collection1 on the registered server UAT
cec control-content publish -c C1 -r R1 -s UAT -d "2021/9/21 0:30:00 PST" -
n Name Create a publishing job
called Name to publish all items in channel C1 on the specified date.
Requires server version: 21.2.1

```

```

-----
cec transfer-content
-----

```

Usage: cec transfer-content <repository>

Creates scripts to transfer content from one OCM server to another. This command is used to transfer large number of content items and the items are transferred in batches. By default the scripts will not be executed by this command. By default all assets are transferred, optionally specify `-p` to transfer only published assets. Specify the source server with `-s <server>` and the destination server with `-d <destination>`. Optionally specify `-n` for the number of items in each batch, defaults to 200.

Options:

<code>--help, -h</code>	Show Help
<code>--server, -s</code>	The registered OCM server the content is from
[required]	
<code>--destination, -d</code>	The registered OCM server to transfer the content
[required]	
<code>--channel, -c</code>	The channel
<code>--publishedassets, -p</code>	The flag to indicate published assets only
<code>--number, -n</code>	The number of items in each batch, defaults to 200
<code>--execute, -e</code>	Execute the scripts

Examples:

```

cec transfer-content Repository1 -s DEV -d UAT Generate
script Repository1_downloadcontent and Repository1_uploadcontent
cec transfer-content Repository1 -s DEV -d UAT -e Generate
script Repository1_downloadcontent and Repository1_uploadcontent and execute
them
cec transfer-content Repository1 -s DEV -d UAT -n 1000 Set the
number of items in each batch to 1000

```

```
cec transfer-content Repository1 -s DEV -d UAT -c Channel1
Transfer the items added to channel Channel1 in repository Repository1
cec transfer-content Repository1 -s DEV -d UAT -c Channel1 -p
Transfer the items published to channel Channel1 in repository
Repository1
```

```
-----
cec list-assets
-----
```

Usage: cec list-assets

Lists assets on OCM server. Optionally specify `-c <channel>`, `-r <repository>`, `-l <collection>` or `-q <query>` to query assets. Specify the server with `-s <server>` or use the one specified in `cec.properties` file.

Options:

```
--help, -h          Show Help
--channel, -c       Channel name
--collection, -l    Collection name
--repository, -r    Repository name, required when <collection> is
specified
--query, -q         Query to fetch the assets
--server, -s        The registered OCM server
```

Examples:

```
cec list-assets           List all assets
cec list-assets -s UAT    List all assets on
registered server UAT
cec list-assets -r Repol  List all assets
from repository Repol
cec list-assets -c Channel1 List all assets
from channel Channel1
cec list-assets -r Repol -l Collection1 List all assets
from collection Collection1 and repository Repol
cec list-assets -q 'fields.category eq "RECIPE"' List all assets
matching the query
```

```
-----
cec create-digital-asset
-----
```

Usage: cec create-digital-asset

Creates digital asset on OCM server. Specify the server with `-s <server>` or use the one specified in `cec.properties` file. Specify the asset attributes in JSON file, e.g.

```
{
  "imagetitle": "Logo",
  "copyright": "Copyright © 1995, 2021, Company and/or its affiliates"
}
```

Options:

```
--help, -h          Show Help
--from, -f          The digital asset source file [required]
```



```

--type, -t          The digital asset type [required]
--repository, -r    The repository to add the asset [required]
--slug, -l          The slug for the asset when create a single asset
--attributes, -a    The JSON file of asset attributes
--server, -s        The registered OCM server

```

Examples:

```

cec create-digital-asset -f ~/Documents/logo.jpg -t Image -r
Repol                               Create asset of type
Image
cec create-digital-asset -f ~/Documents/logo.jpg -t Image -r Repol -l
company-logo                         Create asset of type Image and
set slug to company-logo
cec create-digital-asset -f "~/Documents/demo.mp4,~/Documents/demo2.mp4" -
t Video -r Repol                     Create two assets of type Video
cec create-digital-asset -f ~/Documents/logo.jpg -t MyImage -r Repol -a ~/
Documents/logoattrs.json           Create asset of type MyImage with
attributes
cec create-digital-asset -f ~/Documents/logo.jpg -t MyImage -r Repol -l
company-logo -a ~/Documents/logoattrs.json Create asset of type MyImage
with slug and attributes
cec create-digital-asset -f ~/Documents/images -t Image -r
Repol                               Create assets for
all images files from folder ~/Documents/images
Missing required arguments: from, type, repository

```

```

-----
cec update-digital-asset
-----
Usage: cec update-digital-asset <id>

```

Uploads a new version or updates attributes for a digital asset on OCM server. Specify the server with `-s <server>` or use the one specified in `cec.properties` file. Specify the asset attributes in JSON file, e.g.

```

{
  "imagetitle": "Logo2",
  "copyright": "Copyright © 1995, 2021, Company and/or its affiliates"
}

```

Options:

```

--help, -h          Show Help
--from, -f          The digital asset source file for the new version
--slug, -l          The slug for the asset
--attributes, -a    The JSON file of asset attributes
--server, -s        The registered OCM server

```

Examples:

```

cec update-digital-asset CORED129ACD36FCD42B1B38D22EEA5065F38 -l company-
logo                                  Update asset
slug
cec update-digital-asset CORED129ACD36FCD42B1B38D22EEA5065F38 -f ~/
Documents/logo2.jpg                 Upload a
new version
cec update-digital-asset CORED129ACD36FCD42B1B38D22EEA5065F38 -f ~/

```

Documents/logo2.jpg -l company-logo -a ~/Documents/logoattrs2.json
Upload a new version and update slug and attributes

cec copy-assets

Usage: cec copy-assets <repository>

Copies assets to another repository on OCM server. Specify the server with -s <server> or use the one specified in cec.properties file.

Options:

--help, -h	Show Help
--collection, -l	Collection name
--channel, -c	Channel name
--query, -q	Query to fetch the assets
--assets, -a	The comma separated list of asset GUIDS
--target, -t	The target repository [required]
--server, -s	The registered OCM server

Examples:

```
cec copy-assets Repo1 -t
Repo2                               Copy all assets in
repository Repo1 to Repo2
cec copy-assets Repo1 -t Repo2 -s
UAT                                  Copy all assets in
repository Repo1 to Repo2 on server UAT
cec copy-assets Repo1 -a GUID1,GUID2 -t
Repo2                               Copy asset GUID1 and GUID2 and all
their dependencies in Repo1 to Repo2
cec copy-assets Repo1 -q 'fields.category eq "RECIPE"' -t
Repo2                               Copy assets from repository Repo1, matching the
query, plus any dependencies to Repo2
cec copy-assets Repo1 -c Channel1 -t
Repo2                               Copy assets from the repository
Repo1 and channel Channel1 to Repo2
cec copy-assets Repo1 -l Collection1 -t
Repo2                               Copy assets from the repository
Repo1 and collection Collection1 to Repo2
cec copy-assets Repo1 -c Channel1 -q 'fields.category eq "RECIPE"' -t
Repo2                               Copy assets from repository Repo1, channel Channel1, matching
the query, plus any dependencies to Repo2
```

cec create-asset-usage-report

Usage: cec create-asset-usage-report <assets>

Generates an asset usage report for assets on OCM server. Optionally specify -o to save the report to a json file. Specify the server with -s <server> or use the one specified in cec.properties file.

Options:

--help, -h	Show Help
--output, -o	Output the report to a JSON file

--server, -s The registered OCM server

Examples:

```
cec create-asset-usage-report GUID1
cec create-asset-usage-report GUID1 -s UAT
cec create-asset-usage-report GUID1 -o                               The report
will be saved to GUID1AssetUsage.json
cec create-asset-usage-report GUID1,GUID2 -o                       The report
will be saved to GUID1_GUID2AssetUsage.json
cec create-asset-usage-report GUID1,GUID2 -o ItemReport.json      The report
will be saved to ItemReport.json
```

```
-----
cec create-repository
-----
```

Usage: cec create-repository <name>

Creates a repository on OCM server. Specify the server with -s <server> or use the one specified in cec.properties file. Optionally specify -d <description> to set the description. Optionally specify -t <contenttypes> to set the content types. Optionally specify -c <channels> to set the publishing channels. Optionally specify -l <defaultlanguage> to set the default language. Optionally specify -p <type> to set the repository type. The valid repository types are

```
asset
business
```

Options:

```
--help, -h                Show Help
--description, -d          The description for the repository
--type, -p                 The repository type [asset | business]. Defaults to
asset
--contenttypes, -t        The comma separated list of content types for the
repository
--channels, -c             The comma separated list of publishing channels to
use in this repository
--defaultlanguage, -l     The default language
--server, -s               The registered OCM server
```

Examples:

```
cec create-repository Repol
cec create-repository BusinessRepo -p business
cec create-repository Repol -d "Blog Repository" -t BlogType,AuthorType -c
channell,channel2 -l en-US -s UAT
```

```
-----
cec control-repository
-----
```

Usage: cec control-repository <action>

Performs action <action> on repositories on OCM server. Specify the server with -s <server> or use the one specified in cec.properties file. The valid actions are

```
add-type
remove-type
add-channel
remove-channel
add-taxonomy
remove-taxonomy
```

Options:

```
--help, -h          Show Help
--repository, -r    The comma separated list of content repositories
[required]
--contenttypes, -t  The comma separated list of content types
--channels, -c      The comma separated list of publishing channels
--taxonomies, -x    The comma separated list of promoted taxonomies
--server, -s        The registered OCM server
```

Examples:

```
cec control-repository add-type -r Repol -t Blog,Author
cec control-repository add-type -r Repol,Repo2 -t Blog,Author
cec control-repository add-type -r Repol -t Blog,Author -s UAT
cec control-repository remove-type -r Repol -t Blog,Author
cec control-repository add-channel -r Repol -c channell,channel2
cec control-repository remove-channel -r Repol -c channell,channel2
cec control-repository add-taxonomy -r Repol -x Taxonomy1,Taxonomy2
cec control-repository remove-taxonomy -r Repol -x Taxonomy1,Taxonomy2
```

```
-----
cec share-repository
-----
```

Usage: cec share-repository <name>

Shares repository with users and groups on OCM server and assign a role. Specify the server with -s <server> or use the one specified in cec.properties file. Optionally specify -t to also share the content types in the repository with the users. Optionally specify -y <typerole> to share the types with different role. The valid roles for a repository are

```
manager
contributor
viewer
```

The valid roles for a type are

```
manager
contributor
```

Options:

```
--help, -h          Show Help
--users, -u          The comma separated list of user names
--groups, -g         The comma separated list of group names
--role, -r           The role [manager | contributor | viewer] to assign
```

to the users or groups [required]
 --types, -t Share types in the repository
 --typerole, -y The role [manager | contributor] to assign to the users or groups for types
 --server, -s <server> The registered OCM server

Examples:

```
cec share-repository Repol -u user1,user2 -r manager
Share repository Repol with user user1 and user2 and assign Manager role to them

cec share-repository Repol -u user1,user2 -g group1,group2 -r manager
Share repository Repol with user user1 and user2 and group group1 and group2 and assign Manager role to them

cec share-repository Repol -u user1,user2 -r manager -s UAT
Share repository Repol with user user1 and user2 and assign Manager role to them on the registered server UAT

cec share-repository Repol -u user1,user2 -r manager -t
Share repository Repol and all the types in Repol with user user1 and user2 and assign Manager role to them

cec share-repository Repol -u user1,user2 -r manager -t -y contributor
Share repository Repol with user user1 and user2 and assign Manager role to them, share all types in Repol with user user1 and user2 and assign Contributor role to them
```

```
-----
cec unshare-repository
-----
Usage: cec unshare-repository <name>
```

Deletes user or group access to a repository on OCM server. Specify the server with -s <server> or use the one specified in cec.properties file. Optionally specify -t to also delete the user or group access to the content types in the repository.

Options:

```
--help, -h Show Help
--users, -u The comma separated list of user names
--groups, -g The comma separated list of group names
--types, -t Remove the user or group access to types in the repository
--server, -s <server> The registered OCM server
```

Examples:

```
cec unshare-repository Repol -u user1,user2
cec unshare-repository Repol -u user1,user2 -g group1,group2
cec unshare-repository Repol -u user1,user2 -s UAT
cec unshare-repository Repol -u user1,user2 -t
```

```
-----
cec create-channel
-----
Usage: cec create-channel <name>
```

Creates a channel on OCM server. Specify the server with -s <server> or use the one specified in cec.properties file. Optionally specify -t <type> to set the channel type [public | secure], defaults to public. Optionally

specify `-p <publishpolicy>` to set the publish policy [anythingPublished | onlyApproved], defaults to anythingPublished. Optionally specify `-l <localizationpolicy>` to set the localization policy.

Options:

<code>--help, -h</code>	Show Help
<code>--description, -d</code>	The description for the channel
<code>--type, -t</code>	The channel type [public secure]
<code>--publishpolicy, -p</code>	The publish policy [anythingPublished onlyApproved]
<code>--localizationpolicy, -l</code>	The localization policy for the channel
<code>--server, -s</code>	The registered OCM server

Examples:

```
cec create-channel channel1                Create public
channel channel1 and everything can be published
cec create-channel channel1 -s UAT        On registered
server UAT, reate public channel channel1 and everything can be
published
cec create-channel channel1 -l en-fr      Create public
channel channel1 with localization policy en-fr and everything can be
published
cec create-channel channel1 -t secure -p onlyApproved Create secure
channel channel1 and only approved items can be published
```

```
cec share-channel
```

Usage: cec share-channel <name>

Shares channel with users and groups on OCM server and assign a role. Specify the server with `-s <server>` or use the one specified in `cec.properties` file. The valid roles are

```
manager
contributor
viewer
```

Options:

<code>--help, -h</code>	Show Help
<code>--users, -u</code>	The comma separated list of user names
<code>--groups, -g</code>	The comma separated list of group names
<code>--role, -r</code>	The role [manager contributor viewer] to assign to the users or groups [required]
<code>--server, -s <server></code>	The registered OCM server

Examples:

```
cec share-channel Channel1 -u user1,user2 -r
manager          Share channel Channel1 with user user1 and
user2 and assign Manager role to them
cec share-channel Channel1 -u user1,user2 -g group1,group2 -r
manager          Share channel Channel1 with user user1 and user2 and group
group1 and group2 and assign Manager role to them
cec share-channel Channel1 -u user1,user2 -r manager -s
```

UAT Share channel 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 -l <defaultlanguage> to set the default language. Optionally specify -o <optionallanguages> to set the optional languages. Optionally specify -d <description> to set the description.

Options:

```
--help, -h                    Show Help
--requiredlanguages, -r       The comma separated list of required languages
for the localization policy [required]
--defaultlanguage, -l        The default language [required]
--optionallanguages, -o      The comma separated list of optional languages
for the localization policy
--description, -d            The description for the repository
--server, -s                 The registered OCM server
```

Examples:

```
cec create-localization-policy en-us -r en-US -l en-US
cec create-localization-policy en-fr -r en-US,fr-FR -l en-US
cec create-localization-policy multi -r en-US,fr-FR -l en-US -o zh-CN -d
"Policy for Blog" -s UAT
```

```
-----
cec list-server-content-types
-----
```

Usage: cec list-server-content-types

Lists all content types from server.

Options:

```
--help, -h    Show Help
--server, -s  <server> The registered OCM server
```

Examples:

```
cec list-server-content-types
cec list-server-content-types -s UAT
```

```
-----
cec share-type
-----
```

Usage: cec share-type <name>

Shares type with users and groups on OCM server and assign a role. Specify the server with -s <server> or use the one specified in cec.properties file. The valid roles are

```
manager
contributor
```

Options:

```
--help, -h    Show Help
--users, -u    The comma separated list of user names
--groups, -g   The comma separated list of group names
--role, -r     The role [manager | contributor] to assign to the users
or groups [required]
--server, -s  <server> The registered OCM server
```

Examples:

```
cec share-type BlogType -u user1,user2 -r manager
Share type BlogType with user user1 and user2 and assign Manager role
to them
cec share-type BlogType -u user1,user2 -g group1,group2 -r manager
Share type BlogType with user user1 and user2 and group group1 and
group2 and assign Manager role to them
cec share-type BlogType -u user1,user2 -r manager -s UAT
Share type BlogType with user user1 and user2 and assign Manager role
to them on the registered server UAT
```

```
-----
cec unshare-type
-----
```

Usage: cec unshare-type <name>

Deletes user or group access to a type on OCM server. Specify the server with -s <server> or use the one specified in cec.properties file.

Options:

```
--help, -h    Show Help
--users, -u    The comma separated list of user names
--groups, -g   The comma separated list of group names
--server, -s  <server> The registered OCM server
```


Examples:

```
cec unshare-type BlogType -u user1,user2
cec unshare-type BlogType -u user1,user2 -g group1,group2
cec unshare-type BlogType -u user1,user2 -s UAT
```

```
-----
cec download-type
-----
```

Usage: cec download-type <name>

Downloads types from OCM server. The content field editors and forms for the types will also be downloaded. Specify the server with -s <server> or use the one specified in cec.properties file.

Options:

```
--help, -h    Show Help
--server, -s  <server> The registered OCM server
```

Examples:

```
cec download-type BlogType           Download content type BlogType and
save to local folder src/types/BlogType
cec download-type BlogType,BlogAuthor Download content type BlogType and
BlogAuthor and save to local folder
cec download-type BlogType -s UAT
```

```
-----
cec upload-type
-----
```

Usage: cec upload-type <name>

Uploads types to OCM server. The content field editors and forms for the types will also be uploaded. Specify the server with -s <server> or use the one specified in cec.properties file.

Options:

```
--help, -h    Show Help
--server, -s  <server> The registered OCM server
```

Examples:

```
cec upload-type BlogType
cec upload-type BlogType -s UAT
cec upload-type BlogAuthor,BlogType Place the referenced types first
```

```
-----
cec update-type
-----
```

Usage: cec update-type <action>

Performs action <action> on a type in a local template or on OCM server. Specify the server with -s <server> or use the one specified in cec.properties file. The valid actions are

```
add-content-form
remove-content-form
```

Options:

- help, -h Show Help
- objectname, -o the content form [required]
- contenttype, -c the content type [required]
- template, -t The template the content type is from
- contenttemplate, -t Flag to indicate the template is a content template
- server, -s The registered OCM server

Examples:

```
cec update-type add-content-form -o form1 -c BlogPost -t
BlogTemplate Associate content form form1 with content
type BlogPost from local template at src/templates/BlogTemplate
cec update-type add-content-form -o form1 -c BlogPost -t
BlogTemplateContent -n Associate content form form1 with content
type BlogPost from local template at src/content/BlogTemplateContent
cec update-type add-content-form -o form1 -c BlogPost -s
UAT Associate content form form1 with content
type BlogPost on the registered server UAT
cec update-type add-content-form -o form1 -c BlogPost -s
s Associate content form form1 with content
type BlogPost on the server specified in cec.properties file
cec update-type remove-content-form -o form1 -c BlogPost -t
BlogTemplate Change not to use form1 when create or edit
items of type BlogPost from local template at src/templates/BlogTemplate
cec update-type remove-content-form -o form1 -c BlogPost -t
BlogTemplateContent -n Change not to use form1 when create or edit
items of type BlogPost from local template at src/content/
BlogTemplateContent
cec update-type remove-content-form -o form1 -c BlogPost -s
UAT Change not to use form1 when create or edit
items of type BlogPost on the registered server UAT
```

```
-----
cec download-recommendation
-----
Usage: cec download-recommendation <name>
```

Downloads a recommendation from the Content Management server. Specify the server with -s <server> or use the one specified in cec.properties file. Optionally specify repository with -r <repository>. Optionally specify -p to download the published version.

Options:

- help, -h Show Help
- repository, -r The repository
- published, -p The flag to indicate published version
- channel, -c Channel name, required when <published> is set
- server, -s <server> The registered OCM server

Examples:

```
cec download-recommendation Recommendation1 Downloads
Recommendation1
```

```
cec download-recommendation Recommendation1 -p -c Channel1 Downloads
Recommendation1 published to channel Channel1
cec download-recommendation Recommendation1 -s UAT
cec download-recommendation Recommendation1 -r Repol
```

```
-----
cec upload-recommendation
-----
```

Usage: cec upload-recommendation <name>

Uploads a recommendation to repository <repository> on OCM server. Specify the server with -s <server> or use the one specified in cec.properties file.

Options:

```
--help, -h          Show Help
--repository, -r    The repository [required]
--server, -s        <server> The registered OCM server
```

Examples:

```
cec upload-recommendation Recommendation1 -r Repol
cec upload-recommendation Recommendation1 -r Repol -s UAT
```

```
-----
cec create-contentlayout
-----
```

Usage: cec create-contentlayout <name>

Creates a content layout based on a content type from a local template or from OCM server. By default, an "overview" content layout is created. Optionally specify -s <style> to create in a different style.

Valid values for <style> are:

```
detail
overview
```

Options:

```
--help, -h          Show Help
--contenttype, -c    <contenttype> Content layout is based on
[required]
--template, -t      <template> Content type is from
--server, -r        The registered OCM server
--style, -s         <style> Content layout style: detail | overview
--addcustomsettings, -a Add support for custom settings when used in Sites
```

Examples:

```
cec create-contentlayout Blog-Post-Overview-Layout -c Blog-Post -t
BlogTemplate
cec create-contentlayout Blog-Post-Detail-Layout -c Blog-Post -t
BlogTemplate -s detail
cec create-contentlayout Blog-Post-Overview-Layout -c Blog-Post -t
BlogTemplate -a          Add custom settings when used in Sites
cec create-contentlayout Blog-Post-Overview-Layout -c Blog-Post -
r                          Use content type Blog-Post from the server specified
in cec.properties file
```

```
cec create-contentlayout Blog-Post-Overview-Layout -c Blog-Post -r
UAT -s detail          Use content type Blog-Post from the registered
server UAT
```

```
-----
cec add-contentlayout-mapping
-----
```

Usage: cec add-contentlayout-mapping <contentlayout>

Creates content type and content layout mapping. By default, the mapping is set for "Default". Optionally specify -s <layoutstyle> to name the mapping. By default, the mapping is set for desktop. Optionally specify -m to set the mapping for mobile.

Options:

```
--help, -h          Show Help
--contenttype, -c  <contenttype> Content layout is based on
[required]
--template, -t     <template> The mapping is for
--layoutstyle, -s <style> Content layout style
--mobile, -m      mobile mapping
--server, -r      <server> The registered OCM server
```

Examples:

```
cec add-contentlayout-mapping Blog-Post-Detail-Layout -c Blog-Post -t
BlogTemplate
cec add-contentlayout-mapping Blog-Post-Detail-Layout -c Blog-Post -t
BlogTemplate -m
cec add-contentlayout-mapping Blog-Post-Detail-Layout -c Blog-Post -t
BlogTemplate -s Details
cec add-contentlayout-mapping Blog-Post-Overview-Layout -c Blog-Post -
t BlogTemplate -s "Content List Default"
cec add-contentlayout-mapping Blog-Post-Overview-Layout -c Blog-Post -
t BlogTemplate -s Overview
cec add-contentlayout-mapping Blog-Post-Overview-Layout -c Blog-Post -
r UAT          Set "Content Item Default" to
Blog-Post-Overview-Layout for content type Blog-Post on server UAT
```

```
-----
cec remove-contentlayout-mapping
-----
```

Usage: cec remove-contentlayout-mapping <contentlayout>

Removes a content layout mapping. By default, all mappings for the content layout are removed. Optionally specify -s <layoutstyle> to name the mapping and -m to indicate the mobile mapping.

Options:

```
--help, -h          Show Help
--contenttype, -c  Content type, required when <server> is specified
--template, -t     <template> The mapping is from
--layoutstyle, -s <style> Content layout style
--mobile, -m      mobile mapping
--server, -r      <server> The registered OCM server
```

Examples:

```
cec remove-contentlayout-mapping Blog-Post-Detail-Layout -t BlogTemplate
cec remove-contentlayout-mapping Blog-Post-Detail-Layout -t BlogTemplate -m
cec remove-contentlayout-mapping Blog-Post-Detail-Layout -c Blog-Post -r
UAT
cec remove-contentlayout-mapping Blog-Post-Detail-Layout -c Blog-Post -s
Details -r UAT
```

```
-----
cec add-field-editor
-----
```

Usage: cec add-field-editor <name>

Adds a field editor to a field in a content type.

Options:

```
--help, -h          Show Help
--template, -t      The template the content type is from [required]
--contenttype, -c   The content type [required]
--field, -f         The field the field editor is for [required]
--contenttemplate, -n Flag to indicate the template is a content template
```

Examples:

```
cec add-field-editor editor1 -t BlogTemplate -c BlogPost -f
summary          Use editor1 as the appearance for field summary in
content type BlogPost from local template at src/templates/BlogTemplate
cec add-field-editor editor1 -t BlogTemplateContent -n -c BlogPost -f
summary          Use editor1 as the appearance for field summary in content type
BlogPost from local template at src/content/BlogTemplateContent
```

```
-----
cec remove-field-editor
-----
```

Usage: cec remove-field-editor <name>

Removes a field editor from a field in a content type.

Options:

```
--help, -h          Show Help
--template, -t      The template the content type is from [required]
--contenttype, -c   The content type [required]
--field, -f         The field the field editor is for [required]
--contenttemplate, -n Flag to indicate the template is a content template
```

Examples:

```
cec remove-field-editor editor1 -t BlogTemplate -c BlogPost -f
summary          Remove editor1 as the appearance for field summary in
content type BlogPost from local template at src/templates/BlogTemplate
cec remove-field-editor editor1 -t BlogTemplateContent -n -c BlogPost -f
summary          Remove editor1 as the appearance for field summary in content type
BlogPost from local template at src/content/BlogTemplateContent
```

```
-----
cec migrate-content
-----
```

Usage: cec migrate-content <name>

Migrates content from OCI IC server to EC server. Specify the IC server with -s <server> and the EC server with -d <destination>.

Options:

```
--help, -h          Show Help
--server, -s        The registered IC server the content is from
[required]
--destination, -d  The registered EC server to upload the content
[required]
--repository, -r   The repository for the types and items [required]
--channel, -c      The channel to add the content
--collection, -l   The collection to add the content
```

Examples:

```
cec migrate-content collection1 -s ICServer -d ECServer -r
Repol                               Migrates content from collection
collection1 on ICServer to repository Repol on ECServer
cec migrate-content collection1 -s ICServer -d ECServer -r Repol -l
newCollection                       Migrates content from collection collection1
on ICServer to repository Repol and collection newCollection on ECServer
cec migrate-content collection1 -s ICServer -d ECServer -r Repol -l
newCollection -c channel1           Migrates content from collection collection1
on ICServer to repository Repol, collection newCollection and channel
channel1 on ECServer
```

```
-----
cec compile-content
-----
```

Usage: cec compile-content

Compiles all the content items within the publishing job or list of assets and places the compiled renditions under the "dist" folder. Specify -s <server> to make content queries against this server. Optionally specify -a <assets> comma separated lists of assets. Optionally specify -t <contentType> compile all published assets of this content type. Optionally specify -i <repositoryId> Id of the repository for content type queries. Optionally specify -d <debug> to start the compilation with --inspect-brk flag. Optionally specify -v <verbose> to display all warning messages during compilation.

Options:

```
--help, -h          Show Help
--server, -s        The registered OCM server
--assets, -a        The comma separated list of asset GUIDS
--contenttype, -t   Compile all the published assets of this
content 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: <publishingJobId> or <assets> and <server>

cec upload-compiled-content

Usage: cec upload-compiled-content <path>

Uploads the compiled content to OCM server. Specify the site <site> on the server. Specify the server with -r <server> or use the one specified in cec.properties file.

Options:

--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:

<code>--help, -h</code>	Show Help
<code>--createnew, -c</code>	To create new a taxonomy
<code>--name, -n</code>	The name of the new taxonomy
<code>--abbreviation, -a</code>	The abbreviation of the new taxonomy
<code>--description, -d</code>	The description of the new taxonomy
<code>--file, -f</code>	Flag to indicate the taxonomy is from file
<code>--server, -s</code>	The registered OCM server

Examples:

```
cec upload-taxonomy
Taxonomy1                               Create a new
taxonomy or a draft of existing taxonomy on upload
cec upload-taxonomy Taxonomy1 -s
UAT                                       Create a new taxonomy or a
draft of existing taxonomy on upload on the registered server UAT
cec upload-taxonomy Taxonomy1 -
c                                       Create a new taxonomy on
upload
cec upload-taxonomy Taxonomy1 -c -n Taxonomy1_2 -a t12 -d "Taxonomy1
copy" Create a new taxonomy on upload with given name, abbreviation
and description
cec upload-taxonomy ~/Documents/6A6DC736572C468B90F2A1C17B7CE5E4.json
-f Create a new taxonomy or a draft of existing taxonomy on upload
the JSON file
```

```
-----
cec control-taxonomy
-----
```

Usage: cec control-taxonomy <action>

Perform <action> on taxonomy on OCM server. Specify the taxonomy with `-n <name>` or `-i <id>`. Specify the server with `-s <server>` or use the one specified in `cec.properties` file. The valid actions are

```
promote
publish
unpublish
```

Options:


```

--help, -h          Show Help
--name, -n          Taxonomy name
--id, -i            Taxonomy Id
--publishable, -p  Allow publishing of this taxonomy, defaults to true
--channels, -c      List of channels to publish or unpublish, required when
<action> is publish or unpublish
--server, -s        The registered OCM server

```

Examples:

```

cec control-taxonomy promote -n Taxonomy1
Promote taxonomy Taxonomy1 and allow publishing
cec control-taxonomy promote -i 6A6DC736572C468B90F2A1C17B7CE5E4 -p false
Promote the taxonomy and not allow publishing
cec control-taxonomy publish -n Taxonomy1 -c Channel1,Channel2
cec control-taxonomy unpublish -n Taxonomy1 -c Channel1
cec control-taxonomy publish -n Taxonomy1 -c Channel1 -s UAT

```

```

-----
cec list-translation-jobs
-----

```

Usage: cec list-translation-jobs

Lists translation jobs from local or from OCM server.

Options:

```

--help, -h          Show Help
--server, -s        The registered OCM server

```

Examples:

```

cec list-translation-jobs          Lists local translation jobs
cec list-translation-jobs -s       Lists translation jobs on the server
specified in cec.properties file
cec list-translation-jobs -s UAT   Lists translation jobs on the registered
server UAT

```

```

-----
cec create-translation-job
-----

```

Usage: cec create-translation-job <name>

Creates a translation job <name> for a site on OCM server. Specify the server with -r <server> or use the one specified in cec.properties file. Specify -l <languages> to set the target languages, use "all" to select all languages from the translation policy. Optionally specify -c <connector> to set the translation connector. Optionally specify -t <type> to set the content type. The valid values for <type> are:

```

siteAll
siteItems
siteAssets

```

Options:

```

--help, -h          Show Help
--site, -s          <site> Site [required]

```

```
--languages, -l <languages> The comma separated list of languages
used to create the translation job [required]
--connector, -c The translation connector
--type, -t The type of translation job contents
--server, -r The registered OCM server
```

Examples:

```
cec create-translation-job job1 -s Site1 -l all
cec create-translation-job job1 -s Site1 -l all -r UAT
cec create-translation-job job1 -s Site1 -l de-DE,it-IT
cec create-translation-job job1 -s Site1 -l de-DE,it-IT, -t siteItems
cec create-translation-job job1 -s Site1 -l de-DE,it-IT -c Lingotek
```

```
-----
cec download-translation-job
-----
```

Usage: cec download-translation-job <name>

Downloads translation job <name> from OCM server. Specify the server with -s <server> or use the one specified in cec.properties file.

Options:

```
--help, -h Show Help
--server, -s The registered OCM server
```

Examples:

```
cec download-translation-job Site1Job
cec download-translation-job Site1Job -s UAT
```

```
-----
cec submit-translation-job
-----
```

Usage: cec submit-translation-job <name>

Submits translation job <name> to translation connection <connection>.

Options:

```
--help, -h Show Help
--connection, -c <connection> Connection [required]
```

Examples:

```
cec submit-translation-job Site1Job1 -c connector1-auto
```

```
-----
cec refresh-translation-job
-----
```

Usage: cec refresh-translation-job <name>

Refreshes translation job <name> from translation connection.

Options:

```
--help, -h Show Help
--server, -s The registered OCM server
```

Examples:

```
cec refresh-translation-job Site1Job1
cec refresh-translation-job Site1Job1 -s UAT Refresh translation job
Site1Job1 on the registered server UAT
```

```
-----
cec ingest-translation-job
-----
```

Usage: cec ingest-translation-job <name>

Gets translated job <name> from translation connection and ingest.

Options:

```
--help, -h      Show Help
--server, -s    The registered OCM server
```

Examples:

```
cec ingest-translation-job Site1Job1           Ingest local translation job
cec ingest-translation-job Site1Job1 -s DEV    Ingest translation job
Site1Job1 on the registered server DEV
```

```
-----
cec upload-translation-job
-----
```

Usage: cec upload-translation-job <name>

Uploads translation <name> to OCM server, validate and then ingest the translations. Optionally specify -v to validate only. Optionally specify -f <folder> to set the folder to upload the translation zip file. Specify the server with -s <server> or use the one specified in cec.properties file.

Options:

```
--help, -h          Show Help
--folder, -f        <folder> Folder to upload the translation zip file
--validateonly, -v  Validate translation job without import.
--server, -s        The registered OCM server
```

Examples:

```
cec upload-translation-job Site1Job1           File will
be uploaded to the Home folder.
cec upload-translation-job Site1Job1 -s UAT     File will
be uploaded to the Home folder on registered server UAT
cec upload-translation-job Site1Job1 -f Import/TranslationJobs File will
be uploaded to folder Import/TranslationJobs.
cec upload-translation-job Site1Job1 -v        Validate
the translation job without import.
```

```
-----
cec create-translation-connector
-----
```

Usage: cec create-translation-connector <name>

Creates the translation connector <name>. By default, it creates a mockTranslationConnector. Optionally specify -f <source> to create from a different source.

Valid values for <source> are:
mockTranslationConnector

Options:
--help, -h Show Help
--from, -f <source> to create from

Examples:
cec create-translation-connector connector1

cec start-translation-connector

Usage: cec start-translation-connector <name>

Starts translation connector <name>. Optionally specify -p <port> to set the port, default port is 8084.

Options:
--help, -h Show Help
--port, -p Set <port>. Defaults to 8084.
--debug, -d Start the translation connector server with "--inspect" option

Examples:
cec start-translation-connector connector1
cec start-translation-connector connector1 -p 7777
cec start-translation-connector connector1 -d Start the translation connector server with "--inspect" option to allow debugger to be attached.

cec register-translation-connector

Usage: cec register-translation-connector <name>

Registers a translation connector. Specify -c <connector> for the connector. Specify -s <server> for the connector server URL. Specify -u <user> and -p <password> for connecting to the server. Specify -f <fields> for custom fields.

Options:
--help, -h Show Help
--connector, -c <connector> Connector name [required]
--server, -s <server> Server URL [required]
--user, -u <user> User name [required]
--password, -p <password> password [required]
--fields, -f <fields> translation connector custom fields

Examples:
cec register-translation-connector connector1-auto -c connector1 -s http://localhost:8084/connector/rest/api -u admin -p SamplePass1 -f "BearerToken:Bearer token1,WorkflowId:machine-workflow-id,AdditionalData:{}"

```
-----  
cec create-group  
-----
```

Usage: cec create-group <name>

Creates an OCM group on OCM server. Specify the server with -s <server>. Set the group type with -t <type>. The valid group types are

```
PUBLIC_OPEN  
PUBLIC_CLOSED  
PRIVATE_CLOSED
```

Options:

```
--help, -h      Show Help  
--type, -t      The group type [PUBLIC_OPEN | PUBLIC_CLOSED | PRIVATE_CLOSED]  
--server, -s   <server> The registered OCM server
```

Examples:

```
cec create-group Group1          Create group Group1, people can  
add themselves to the group and share content with the group  
cec create-group Group1 -t PUBLIC_CLOSED  Create group Group1, only group  
managers can add members but people can share content with the group  
cec create-group Group1 -t PRIVATE_CLOSED Create group Group1, only group  
managers can add members and only members can share content with the group  
cec create-group Group1 -s DEV
```

```
-----  
cec delete-group  
-----
```

Usage: cec delete-group <name>

Deletes an OCM group on OCM server. Specify the server with -s <server>.

Options:

```
--help, -h      Show Help  
--server, -s   <server> The registered OCM server
```

Examples:

```
cec delete-group Group1  
cec delete-group Group1 -s DEV
```

```
-----  
cec add-member-to-group  
-----
```

Usage: cec add-member-to-group <name>

Adds users and groups to an OCM group and assign a role on OCM server. Specify the server with -s <server>. The valid roles are

```
MANAGER  
MEMBER
```

Options:
 --help, -h Show Help
 --users, -u The comma separated list of user names
 --groups, -g The comma separated list of group names
 --role, -r The role [MANAGER | MEMBER] to assign to the users or groups [required]
 --server, -s The registered OCM server

Examples:
 cec add-member-to-group Group1 -u user1,user2 -g Group2,Group3 -r MEMBER
 cec add-member-to-group Group1 -u user1,user2 -g Group2,Group3 -r MEMBER -s DEV

 cec remove-member-from-group

Usage: cec remove-member-from-group <name>

Removes users and groups from an OCM group on OCM server. Specify the server with -s <server>.

Options:
 --help, -h Show Help
 --members, -m The comma separated list of user and group names [required]
 --server, -s The registered OCM server

Examples:
 cec remove-member-from-group Group1 -m user1,user2,Group2,Group3
 cec remove-member-from-group Group1 -m user1,user2,Group2,Group3 -s DEV

 cec create-encryption-key

Usage: cec create-encryption-key <file>

Create an encryption key to encrypt/decrypt password for servers and save to <file>. Use NodeJS 10.12.0 or later.

Options:
 --help, -h Show Help

Examples:
 cec create-encryption-key ~/.ceckey Create encryption key and save to file ~/.ceckey

 cec register-server

Usage: cec register-server <name>

Registers a OCM server. Specify -e <endpoint> for the server URL. Specify -u <user> and -p <password> for connecting to the server.

Optionally specify `-k <key>` to encrypt the password. Optionally specify `-t <type>` to set the server type. The valid values for `<type>` are:

```
pod_ec
pod_ic
dev_ec
dev_pod
dev_osso
```

and the default value is `pod_ec`.

For `pod_ec` server, optionally specify `<idcsurl>`, `<clientid>`, `<clientsecret>` and `<scope>` for headless commands.

Options:

```
--help, -h          Show Help
--endpoint, -e      <endpoint> Server endpoint [required]
--user, -u          <user> User name [required]
--password, -p      <password> Password [required]
--key, -k           The key file used to encrypt the password
--type, -t          <type> Server type
--idcsurl, -i       <idcsurl> Oracle Identity Cloud Service Instance URL
--clientid, -c      <clientid> Client ID
--clientsecret, -s <clientsecret> Client secret
--scope, -o         <clientsecret> Scope
--timeout, -m       Timeout in millisecond when try to login to the
server. Defaults to 30000ms.
```

Examples:

```
cec register-server server1 -e http://server1.com -u user1 -p samplePass1 -
i http://idcs1.com -c clientid -s clientsecret -o https://primary-audience-
and-scope The server is a tenant on Oracle Public cloud
```

```
cec register-server server1 -e http://server1.com -u user1 -p
samplePass1
```

The server is a tenant on Oracle Public cloud

```
cec register-server server1 -e http://server1.com -u user1 -p samplePass1 -
m
60000
```

The server is a tenant on Oracle Public cloud

```
cec register-server server1 -e http://server1.git.oraclecorp.com.com -u
user1 -p samplePass1 -t
```

```
dev_ec The server
is a standalone development instance
```

```
cec register-server server1 -e http://server1.com -u user1 -p samplePass1 -
k
```

```
~/ .ceckey
```

The password will be encrypted

```
-----
cec set-oauth-token
-----
```

Usage: `cec set-oauth-token <token>`

Set OAuth token for a registered server.

Options:
--help, -h Show Help
--server, -s The registered OCM server [required]

Examples:
cec set-oauth-token token1 -s UAT Set OAuth token for server UAT,
all CLI commands using UAT will be headless

cec list

Usage: cec list

Lists local or server resources such components and templates. Specify the server with -s <server> or use the one specified in cec.properties file. Optionally specify -t <types> to list specific types of resources on the OCM server.

Valid values for <types> on the server are:

channels
components
localizationpolicies
recommendations
repositories
sites
templates
taxonomies
translationconnectors

Options:
--help, -h Show Help
--types, -t <types> The comma separated list of resource types
--server, -s <server> The registered OCM server

Examples:
cec list List all local resources
cec list -s List resources on the server
specified in cec.properties file
cec list -t components,channels -s List components and channels
on the server specified in cec.properties file
cec list -t components,channels -s UAT List components and channels
on the registered server UAT

cec execute-get

Usage: cec execute-get <endpoint>

Makes an HTTP GET request to a REST API endpoint on OCM server. Specify the server with -s <server>.

Options:
--help, -h Show Help


```
--file, -f    The file to save the result [required]
--server, -s  The registered OCM server
```

Examples:

```
cec exeg "/sites/management/api/v1/sites?links=none" -f allsites.json -s
DEV
cec exeg "/content/management/api/v1.1/channels?links=none" -f
allchannels.json -s DEV
cec exeg "/documents/api/1.2/folders/self/items" -f homefolderitems.json -
s DEV
```

```
-----
cec install
-----
```

Usage: cec install

Creates an initial source tree in the current directory.

With cec install, your source can be in a separate directory to the cec command install files, and you no longer need your source to be within a sites-toolkit directory.

The cec.properties file can be used to specify server settings. It will be picked up from the source directory, or can be specified with environment variable CEC_PROPERTIES

Use cec develop to start a dev/test server for your source. Different ports can be used for the server, to enable multiple source trees to exist.

Options:

```
--help, -h  Show Help
```

Examples:

```
cec install
```

```
-----
cec develop
-----
```

Usage: cec develop

Starts a test server in the current folder. Specify the server with -s <server> or use the one specified in cec.properties file. Optionally specify -p <port> to set the port, default port is 8085.

Options:

```
--help, -h    Show Help
--port, -p    Set <port>. Defaults to 8085.
--server, -s  The registered OCM server
--debug, -d   Start the server with "--inspect"
```

Examples:

```
cec develop
cec develop -p 7878
cec develop -p 7878 -s UAT
```

```

-----
cec sync-server
-----

Usage: cec sync-server

Starts a sync server in the current folder to sync changes notified by
web hook from <server> to <destination> server. Specify the source
server with -s <server> and the destination server with -d
<destination>. Optionally specify -p <port> to set the port, default
port is 8086. To run the sync server over HTTPS, specify the key file
with -k <key> and the certificate file with -c <certificate>. Set
authorization option with -a and the valid values are

    none
    basic
    header

Options:
  --help, -h          Show Help
  --server, -s        The registered OCM server for sync source
[required]
  --destination, -d   The registered OCM server for sync destination
[required]
  --authorization, -a The authorization method [none | basic | header]
for the web hook event, defaults to basic
  --username, -u      The username used to authenticate the web hook
event when <authorization> is basic
  --password, -w      The password used to authenticate the web hook
event when <authorization> is basic
  --values, -v        The comma separated list of name-value pairs
used to authenticate the web hook event when <authorization> is header
  --port, -p          Set port. Defaults to 8086.
  --key, -k           The key file for HTTPS
  --certificate, -c   The certificate file for HTTPS

Examples:
  cec sync-server -s DEV -d UAT -u admin -w
samplePass1           Use Basic authorization
  cec sync-server -s DEV -d UAT -u admin -w samplePass1 -p
7878                 Use Basic authorization and port set to 7878
  cec sync-server -s DEV -d UAT
Use Basic authorization and the username and password will be prompted
to enter
  cec sync-server -s DEV -d UAT -u admin
Use Basic authorization and the password will be prompted to enter
  cec sync-server -s DEV -d UAT -a header -v key1:value1,key2:value2
Use Header authorization
  cec sync-server -s DEV -d UAT -a none
No authorization
  cec sync-server -s DEV -d UAT -k ~/keys/key.pem -c ~/keys/cert.pem
The sync server will start over HTTPS
Missing required arguments: server, destination
-----

```

```
cec webhook-server
-----
Usage: cec webhook-server

Starts a server in the current folder to handle events notified by web hook
from <server>. Optionally specify -p <port> to set the port, default port is
8087. The supported event types are

    seo - refresh Detailed page in the Prerender cache

Options:
  --help, -h          Show Help
  --type, -t          The webhook server type [seo] [required]
  --contenttype, -c  The content type [required]
  --detailpage, -d   The full url of the site detail page for this type
[required]
  --server, -s        The registered OCM server [required]
  --port, -p          Set port. Defaults to 8087.

Examples:
  cec webhook-server -t seo -s DEV -c Blog -d "/site/blogsite/detailpage"
  cec webhook-server -t seo -s DEV -c Blog,Author -d "/site/blogsite/
blogdetail,/site/blogsite/authordetail"
  cec webhook-server -t seo -s DEV -c Blog -d "/site/blogsite/detailpage" -p
7878
Missing required arguments: type, contenttype, detailpage, server
```

使用本地测试工具进行测试

在将定制组件、模板和内容布局导入到 Oracle Content Management 之前，先在本地测试工具中运行它们。

要启动本地测试工具，请执行以下操作：

1. 在终端窗口中输入 `cd cec`。
2. 输入 `cec develop &` 或 `cec develop --server <server-name> &`
3. 打开浏览器并访问 `http://localhost:8085`，以查看本地测试工具中运行的组件、模板和内容布局。
4. 您可以在以下目录中查找组件、模板、主题等：
 - `cec/src/main/components`
 - `cec/src/main/templates`
 - `cec/src/main/themes`

升级到 jQuery 3.5.x

作为开发人员，您可以使用 OCE 工具包标识未正确关闭的 html 标记。

升级到 jQuery 3.5.x，以获取 HTML 语法分析器中的安全修复。

<https://blog.jquery.com/2020/04/10/jquery-3-5-0-released>

在 jQuery 3.5.x 中对某些 HTML 字符串进行语法分析与在 3.4.x 中对相同字符串进行语法分析所产生的结果不同。这些字符串包含自关闭标记，这违反了 HTML 标准；例如：“<div />”。此类字符串，特别是作为较大标记序列的一部分时，在 3.5.x 中和 3.4.x 中进行语法分析的方式可能不同。

您可以运行 `cec create-asset-report <site>` 来查找页 JSON 文件以及组件 HTML 和 JS 文件中未正确关闭的 html 标记。

使用开发人员云服务为 Oracle Content Management 进行开发

OCE 工具包（开发人员云服务集成）可帮助您为 Oracle Content Management 开发站点模板、主题、定制组件和内容布局。

使用 OCE 工具包，您可以在 Oracle Content Management 中使用资产资料档案库、文件和文件夹。OCE 工具包提供了一些工具，可用于创建和开发定制组件和站点模板（包括主题和内容布局）。它包含一个 Git 资料档案库、一个用于快速迭代开发的本地测试工具，以及供开始使用的示例单元测试。

借助 OCE 工具包可以执行以下任务：

- 设置本地开发环境以使用 Oracle Content Management 实例在本地开发和测试组件、模板、主题和内容布局
- 基于示例创建组件、站点模板和内容布局，在测试工具中运行它们，浏览它们，以及在开发人员云服务环境中开发组件、模板、主题和内容布局
- 将在 Oracle Content Management 中创建的组件和站点模板导入到开发人员云服务环境以进行源管理和进一步开发
- 从开发人员云服务环境导出组件、模板和内容布局，以在 Oracle Content Management 中使用
- 复制现有组件、模板或内容布局
- 编写单元测试
- 优化组件
- 将组件和模板部署到 Oracle Content Management

以下主题介绍了如何设置开发人员云服务来开发定制组件、站点模板和主题以及内容布局：

1. [关于使用开发人员云服务](#)
2. [登录 Oracle Content Management 的开发人员云服务控制台](#)
3. [在开发人员云服务中创建项目](#)
4. [将 OCE 工具包添加到新的 Git 资料档案库中的项目代码](#)
5. [在本地测试工具中测试定制组件、模板和内容布局](#)

6. 合并更改

以下主题提供关于使用 Oracle Content Management OCE 工具包的信息：

- [使用 cec 命令行实用程序](#)
- [通过开发人员云服务开发定制组件](#)
- [通过开发人员云服务开发模板](#)
- [开发内容布局](#)

关于使用开发人员云服务

Oracle Developer Cloud Service 是基于云的软件开发平台即服务 (Platform as a Service, PaaS)，也是应用程序开发基础结构的托管环境。它提供了基于标准的开源集成，可用于开发、协作处理应用程序并在 Oracle Cloud 中部署应用程序。

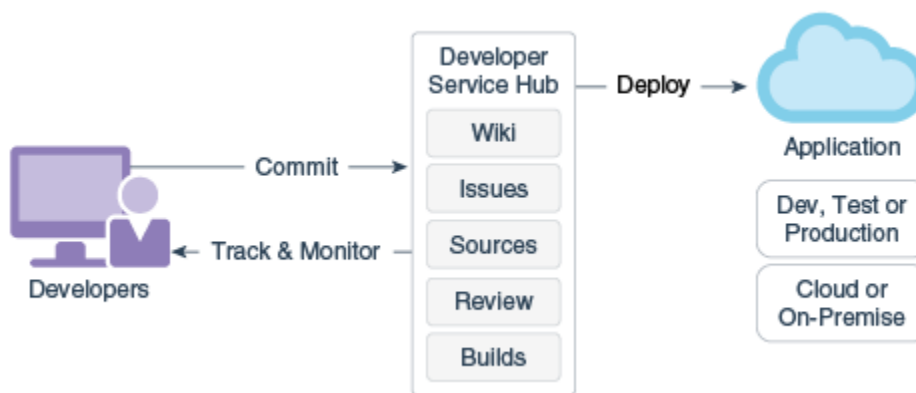
开发人员云服务是托管在 Oracle Cloud 上的软件和服务的集合，可通过与 Git、Maven、问题和 Wiki 集成来帮助您有效地管理应用程序开发生命周期。使用 Oracle Developer Cloud Service，可以将应用程序源代码提交至 Oracle Cloud 上的 Git 资料档案库、联机跟踪分配的问题和缺陷、使用 Wiki 页共享信息、同级复查源代码以及监视项目构建。测试成功后，可以将项目部署到 Oracle Content Management。

登录 Oracle Content Management 的开发人员云服务控制台

开始在开发人员云服务控制台上为 Oracle Content Management 开发定制组件。

作为 Oracle Cloud 服务的管理员，您可以使用我的服务管理来创建和管理您的云服务。如果您是 Oracle Content Management 的服务实例管理员以及标准开发人员服务的实例管理员，则可以设置它们并开始使用它们：

1. 使用为您的账户提供的信息登录 Oracle Cloud。
2. 登录“我的服务管理”，以创建和管理您的 Oracle Content Management 实例和标准开发人员服务。



3. 根据要求验证 Oracle Developer Cloud Service 电子邮件。
4. 使用您的服务的订阅详细信息设置 Oracle Content Management 实例，然后转到实例的 Oracle Content Management URL。
5. 转到标准开发人员服务的 URL。

6. 登录 Oracle Developer Cloud Service 账户。
访问开发人员云服务 URL 并登录控制台。

在开发人员云服务中创建项目

您可以使用“内容体验云”项目模板在开发人员云服务中创建项目，也可以创建具有空 Git 资料档案库的项目，然后从 Oracle Content Management 实例导入 OCE 工具包。

- 使用 [Oracle Content Management 模板创建开发人员云服务项目](#)
- [通过从 Oracle Content Management 下载 OCE 工具包在开发人员云服务中创建项目](#)
- [将 OCE 工具包添加到新的 Git 资料档案库中的项目代码](#)

使用 Oracle Content Management 模板创建开发人员云服务项目

在开发人员云服务中创建项目，以用于开发定制组件、模板、主题和内容布局。

要创建项目，请执行以下操作：

1. 登录到开发人员云服务控制台后，单击新建项目。
2. 在模板列表中，选择内容管理，然后单击下一步。
3. 在“项目属性”下的属性中，选择 Wiki 标记字段中的 **CONFLUENCE**。

通过从 Oracle Content Management 下载 OCE 工具包在开发人员云服务中创建项目

在开发人员云服务中创建项目，以用于开发定制组件、模板、主题和内容布局。

要创建项目，请执行以下操作：

1. 登录到开发人员云服务控制台后，单击新建项目。
2. 命名您的项目，输入或选择所需的其他项目详细信息，然后单击下一步。
3. 在模板列表中，选择初始资源库，然后单击下一步。
4. 在“项目属性”下的属性中，为初始资源库选择空资源库。单击完成。

将 OCE 工具包添加到新的 Git 资料档案库中的项目代码

您可以将 OCE 工具包添加到项目的新的空 Git 资料档案库。

1. 在新项目中的资料档案库下，复制项目 Git 资料档案库的 HTTP URL。
2. 打开终端窗口并输入此命令：`git clone <your-project >.git`
 - a. 在系统询问时，输入您的开发人员云服务密码。
 - b. 如果看到错误 "git is not a command"，请从 <https://git-scm.com/downloads> 安装 Git，然后重新输入 `git clone` 命令。
3. `git clone git@github.com:oracle/content-and-experience-toolkit.git`
也可以从此处下载：<https://github.com/oracle/content-and-experience-toolkit/archive/master.zip>

4. `cp -R content-and-experience-toolkit/sites/cec-components <your-project>`
5. `cd <your-project>`
6. `git add cec-components`
7. `git commit -a -m "<your comments>"`
8. `git push`

在本地测试工具中测试定制组件、模板和内容布局

在将定制组件、模板和内容布局导入到 Oracle Content Management 之前，先在本地测试工具中运行它们。

要启动本地测试工具，请执行以下操作：

1. 在终端窗口中输入 `cd cec-components`。
输入 `npm start &`
2. 打开浏览器并访问 `http://localhost:8085`，以查看本地测试工具中运行的组件、模板和内容布局。

在本地服务器上测试组件时，可以选择使用来自本地模板或来自 Oracle Content Management 服务器的内容。

合并更改

在本地计算机上创建组件、模板或内容布局或者编辑源代码后，需要将新的和更改的组件与模板合并到项目的 Git 资料档案库中。

要将更改合并到 Git 资料档案库中，请在终端窗口中按顺序输入以下命令。

```
cd cec-components git pull
git add .
git status
git commit -a -m "Your comments" git pull
git push
```

使用 OCE 工具包将更改从测试环境传播到生产环境

开发站点模板后，您可以使用 OCE 工具包的命令行界面 (command-line interface, CLI) 在 Oracle Content Management 服务器上将模板从开发传播到测试再到生产。

要传播所做的更改，您可以使用工具包命令在开发、测试和生产服务器上创建站点并管理其生命周期。可以在开发环境中更改站点并将这些更改传播到测试环境和生产环境。您也可以将这组命令行实用程序并入您的脚本环境以管理部署。使用 CLI 实用程序，您可以部署新项（例如资产和组件）以及更新现有内容。

以下步骤说明如何使用 OCE 工具包 CLI 将更改从开发传播到测试再到生产：

1. 使用相同资料档案库和本地化策略设置开发、测试和生产服务器。

要将更改从开发服务器传播到测试服务器、再到生产服务器，需要在三个服务器中的每个服务器上使用相同名称和本地化策略设置资料档案库。资产资料档案库的默认本地化策略为 en-US，但是您可以使用不同的策略，前提是策略在所有三个服务器中都是相同的。

请参见设置资产资料档案库。

2. 在 Oracle Content Management 中注册您的开发、测试和生产服务器。

传播对站点的更改之前，需要注册每个服务器。可以使用 OCE 工具包提供的 `cec register-server` 命令注册服务器：

```
cec register-server <name>
```

指定以下命令选项：

- 服务器 URL 的 `-e <endpoint>`
- 用于连接到服务器的 `-u <user>` 和 `-p <password>`
- `-t <type>`，这是可选的，用于设置服务器类型。默认值为 `pod_eco`。
连接到 Oracle Public Cloud 上的 Oracle Content Management 租户时，仅使用 `pod_eco`。

例如，以下命令注册作为 Oracle Public cloud 上租户的服务器：

```
cec register-server DEV -e https://DEV.example.com -u user1 -p  
<password>
```

下一个命令注册 Oracle Content Management 的独立开发实例：

```
cec register-server DEV -e https://DEV.git.oraclecorp.example.com -  
u user1 -p <password>
```

注册 Oracle Content Management 服务器后，可以使用 `cec list` 工具包命令列出其内容。

以下命令列出开发服务器的内容：

```
cec-components> cec list -s DEV
- Logged in to remote server: <host:port>
Channels:
  Name          Token
  StarterSite   <site-id>

Components:
  Name          Type
  Published
  FooterBar     Component group
  StarterComponent Local component
  StarterFooter  Component group
  StarterNavMenu Local component

Localization policies:
  Name          Required languages
  Optional Languages
  en-US        en-US

Repositories:
  Name
  r
```


Sites:		Theme	Type
Published	Online		
Name		StarterSiteTheme	Enterprise

Templates:		
Name	Theme	Type
StarterTemplate	StarterTheme	Standard

3. 将站点模板上载到开发服务器并从该模板创建站点。

可以使用 `cec create-template` 命令创建站点模板，然后将该模板上载到开发服务器。然后，可以使用 `cec create-site` 命令从该模板创建站点。以下命令创建模板并上载模板：

```
- cec create-template blog -f BlogTemplate
- cec upload-template blog -s DEV
```

下一个命令从上载的模板创建名为 `blog` 的站点：

```
cec-components> cec create-site blog -t blog -r r -l "en-US" -d "en-US" --
server DEV
- Logged in to remote server: <https:<host:<port>
- establish user session
- get template
- get repository
- get localization policy
- creating enterprise site . . .
  name          blog
  template      blog
  site prefix   blog
  repository    r
  localization policy en-US
  default language en-US
- submit create site site
- create site in process: percentage 95
- create site in process: percentage 95
- create site in process: percentage 95
- create site in process: percentage 95
- create site in process: percentage 95
- site created
```

4. 在开发服务器上发布站点并使其联机。

创建站点后，可以使用 `cec control-site` 命令发布该站点并使其联机：

```
cec-components> cec control-site
Usage: cec contrl-site <action>
Perform <action> on site in CEC server. Specify the site with -s <site>
Specify the server with -r <server>.

  publish
  unpublish
  bring-online
  take-offline
```

Options:

```
--site, -s <site> Site
--server, -r <server> The registered CEC server
--help, -h Show help
```

Examples:

```
cec control-site publish -s Site1          Publish site Site1
on the server
cec control-site publish -s Site1 -r UAT   Publish site Site1
on the registered server UAT
cec control-site unpublish -s Site1 -r UAT Inpublish site
Site1 on the registered server UAT
cec control-site bring-online -s Site1 -r UAT Bring site Site1
online on the registered server UAT
cec control-site take-offline -s Site1 -r UAT Take site Site1
offline on the registered server UAT
```

```
Not enough non-option arguments: got 0, need at least 1
cec-components> cec control-site publish --site blog --server DEV
- Logged in to the remote server: https://<host>:<port>
- establish user session
- get site: runtimeStatus: offline publishStatus: unpublished
- submit publish site
- publish in process: percentage 20
- publish in process: percentage 40
- publish in process: percentage 49
- publish in process: percentage 49
- publish in process: percentage 50
- publish in process: percentage 50
```

5. 要将站点从 DEV 移到 UAT，需要将该站点打包。用于在服务器之间移动站点的打包模型是模板。从在开发服务器上创建的站点创建新模板并下载该模板。

以下示例中的 `cec create-template-from-site` 命令从 `blog` 创建名为 `blog2` 的站点模板。

```
cec create-template-from-site blog2 -s blog
```

使用 `cec download-template` 命令下载从开发站点创建的模板：

```
cec-components: cec download-template blog2 --server DEV
- Logged in to remote server: https://<host>:<port>
- establish user session
- export template
- template download to /Users/<user-name>/devenv/git/webclient/
developer/sites-toolkit/cec-components/dist/blog2.zip
- the template will be at /Users/<user-name>/devenv/git/webclient/
developer/sites-toolkit/cec-components/src/main/templates/blog2
- the theme for the template will be at /Users/<user-name>/
devenv/git/webclient/developer/sites-toolkit/cec-components/src/
main/themes/blogTheme
- create link _scs_theme_root_
- create link _scs_design_name_
- override component /Users/<user-name>devenv/git/webclient/
developer/sites-toolkit/cec-components/src/main/components/Starter-
Blog-Author-Summary
```

```

- override component /Users/<user-name>devenv/git/webclient/developer/
sites-toolkit/cec-components/src/main/components/Starter-Blog-Post-Content
- override component /Users/<user-name>devenv/git/webclient/developer/
sites-toolkit/cec-components/src/main/components/Starter-Blog-Post-Header
- override component /Users/<user-name>devenv/git/webclient/developer/
sites-toolkit/cec-components/src/main/components/Starter-Blog-Post-Search-
Result
- override component /Users/<user-name>devenv/git/webclient/developer/
sites-toolkit/cec-components/src/main/components/Starter-Blog-Post-Post-
Sidebar
- override component /Users/<user-name>devenv/git/webclient/developer/
sites-toolkit/cec-components/src/main/components/Starter-Blog-Post-Summary
- set themeName to blogTheme in siteinfo.json
- unzip tmlate content file
*** template is ready to test: https://localhost:8085/templates/blog2
cec upload-template blog2 --server UAT

```

6. 上传模板以创建内容类型和内容布局映射。

```
cec upload-template blog2 --server UAT
```

7. 上传模板，但是从模板中排除内容项（内容模板）。

```
cec upload-template blog2 --server UAT -x
```

您这样做是为了创建一个站点，其中内容与原始站点具有相同 GUID。从包含内容的模板创建站点时，新站点中的所有内容都将具有新的 GUID。因为我们要允许内容更新，而不是创建新内容，所以您需要从模板排除内容。

8. 从模板创建站点。

```
cec create-site blog -t blog2 -r r -l "en-US" -d "end-US" --server UAT
```

9. 将内容模板上载到站点的通道和集合。您需要这样做，因为您在步骤 7 中从模板排除了该内容。

```
cec upload-content blog2 -t -r r -c blog -l "blog site" --server UAT
```

10. 在测试服务器上发布站点并使其联机。

使用 `cec control-site` 命令发布站点并使其联机：

```

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. 如果您之后在 DEV 服务器上对站点 blog 进行更改，您可以将这些更改传播到您已在 UAT 服务器上创建的站点。

12. 从您的站点创建另一个模板，以便该模板将具有您的更改。

```
cec create-template-from-site blog3 -s blog --server DEV
```

13. 下载模板。

```
cec download-template blog3 -s DEV
```

14. 上传模板并从该模板创建站点，以便将更改传播到测试环境。

```
cec upload-template blog3 -s UAT
```

此命令创建或更新已更改的任何组件和主题，但排除内容。

15. 现在使用 update-site 命令获取内容并更新页。

```
cec update-site blog -t blog3 - UAT
```

例如：

```
cec-components> cec update-site blog -t blog3 --server UAT
Updating site: blog3
- Logged in to remote server: https://<host>:<port>
- pages          : updating file# 6   of 6   files
- content        : updating file# 3   of 3   files
- System Files   : updating file# 5   of 5   files
- controller     : no files in update, removing files on
server
- favicons       : no files in update, removing files on
server
- misc           : no files in update, removing files on
server
- seo            : no files in update, removing files on
server
- system         : no files in update, removing files on
server
- created content file /Users/<user-name>/devenv/git/webclient/
developer/sites-toolkit/cec-components/dist/blog3_export.zip
- upload content file
- get CSRF token
- submit import job, updating content
- import job in progress. . .
- import job in progress. . .
- import job in progress. . .
- content imported:
Update Site Results:
- Site Pages     : completed with 0 errors.
- Embedded Content : completed with 0 errors.
```

```
- System Files      : completed with 0 errors.  
- Settings Files   : completed with 0 errors.  
- Content Update   : completed with 0 errors.
```

16. 检查站点以验证是否传播了更改。
17. 执行相同步骤，像您将站点从 DEV 移到 UAT 那样从 UAT 服务器移到 PROD 服务器。
18. 在您的生产服务器上创建站点，使其联机并验证更改。

可以使用 `cec list` 命令查看生产站点的内容，并确保它包括您在开发环境中进行的更改。此外，您还可以检查站点以验证更改是否已传播到生产。

对密码进行加密

使用 OCE 工具包注册服务器时，需要对密码进行加密以使服务器可供本地使用。

1. 使用包含纯文本格式的密码的 `cec register-server` 命令注册 Oracle Content Management 服务器。
2. 使用 `cec create-encryption-key` 命令对密码加密。

```
cec create-encryption-key <file> [alias: cek]  
Create an encryption key to encrypt/decrypt password for servers.
```

3. 使用加密密钥再次注册服务器，这使得服务器可用于本地开发和测试。

加密密码存储在服务器连接文件中。连接到已注册服务器时，将对密码进行解密。

注册服务器

可以在 OCE 工具包中注册服务器。

使用带有加密密钥的 `cec register-server` 命令注册 Oracle Content Management 服务器，以进行本地开发和测试。

注册服务器时，对密码进行加密可以使服务器可用于 OCE 工具包。请参见[对密码进行加密](#)。

为站点创建使用情况和权限报告

可以创建报告来验证和修复站点测试到生产传播的目标服务器成员的权限。

使用 `cec create-asset-report` 并按如下所示对其进行检查：

1. 检查所有站点构件的成员资格和通道分配：
 - 主题
 - 模板
 - 组件
 - 内容类型
2. 标记您可以解决的问题。

例如：

```
cec create-asset-report blog1 -s <registered-server> -o
cec create-asset-report trbcent -s <registered-server> -o
```

报告在 JSON 文件中生成，您可以检查该报告以了解使用情况和权限方面问题。以下命令可用于权限修复：

- `cec share-type`：与 Oracle Content Management 服务器中的用户共享类型。
- `cec unshare-type`：为 Oracle Content Management 服务器中的给定用户删除对类型的访问权限。
- `cec share-repository`：在 Oracle Content Management 服务器中共享资料档案库（和资料档案库使用的类型）。
- `cec unshare-repository`：删除对 Oracle Content Management 服务器中资料档案库的用户访问权限。
- 例如：

```
cec share-repository Repol -u <user-name1>,<user-name2> -r manager -t -s <registered-server>
```

下载和上载文档及文件夹

可以在 Oracle Content Management 服务器中下载和上载文档及文件夹。

以下命令可用于文档和文件夹下载及上载：

```
cec download-folder <path>           Downloads folder from CEC
server. [alias: dlfd]
cec upload-folder <path>           Uploads folder to CEC
server. [alias: ulfd]
cec download-file <file>           Downloads file <file> from CEC
server. [alias: dlf]
cec upload-file <file>           Uploads file <file> to CEC
server. [alias: ulf]
cec-share-folder <name>
cec-unshare-folder <name>
```

对于 `cec-share-folder <name>`，您可以与 Oracle Content Management 服务器上的用户共享文件夹并为用户分配角色。使用 `-s <server>` 指定服务器，也可以使用 `cec.properties` 文件中指定的服务器。有效角色如下：

- manager
- contributor
- downloader
- viewer

对于下载，可以指定文件夹层次。

从模板创建站点并为内容保持相同 GUID

作为开发人员，您可以使用 OCE 工具包命令来从模板创建 Oracle Content Management 站点，并为内容保持相同 GUID。

使用以下 OCE 工具包命令：

```
update create-site-from-template --reuse-content
```

此命令在 Oracle Content Management 服务器中创建站点，并在创建站点时保留内容 ID。为了使多次从测试到生产的运行不致在目标服务器上出现重复的内容项，保留内容 ID 是非常必要的。

从标准站点创建企业模板

作为开发人员，您可以使用 OCE 工具包 CLI 命令从标准站点创建企业模板。

默认情况下，`create-template` 命令创建标准模板（如果站点是标准站点）和企业模板（如果站点是企业站点）。您也可以从标准站点创建企业模板。

使用新的 `-enterprise` 选项运行 `cec create-template`：

```
cec create-template EnterpriseTempl -s StandardSite1 -e
```

导入和导出分类

使用 OCE 工具包命令，可以将分类从本地计算机导入到 Oracle Content Management 服务器，或者将分类从服务器导出到本地计算机。

`cec download-taxonomy <name>` 命令从 Oracle Content Management 导出分类。它从 Oracle Content Management 服务器下载分类。

您可以在此命令中使用以下选项：

- `--status, -t [promoted | published] [required]`：指定分类状态。
- `--id, -i`：如果其他分类具有相同名称，则指定分类 ID。
- `--server, -s`：指定注册的 Oracle Content Management 服务器，或者使用在 `cec.properties` 文件中指定的服务器。
- `--help, -h`：显示命令的帮助。

`download-taxonomy` 命令的一些示例如下：

```
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>` 命令将分类导入到 Oracle Content Management。它将分类上载到 Oracle Content Management 服务器。

您可以在此命令中使用以下选项：

- `--createnew, -c`：创建新分类。
- `--name, -n`：指定新分类的名称。
- `--abbreviation, -a`：指定新分类的缩写。
- `--description, -d`：指定新分类的说明。
- `--file, -f`：指示分类是否来自文件。
- `--server, -s`：指定注册的 Oracle Content Management 服务器，或者使用在 `cec.properties` 文件中指定的服务器。
- `--T2P`：检查目标 Oracle Content Management 实例上是否存在 ID 相同的资产或分类。如果为 `true`，资产将添加为新版本，分类将添加为草稿（覆盖现有草稿）。否则，将使用相同 ID 创建新资产或分类。
- `--New`：始终在目标 Oracle Content Management 实例上创建新资产或分类。
- `--help, -h`：显示命令的帮助。

upload-taxonomy 命令的一些示例如下：

```
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 <filename>.json -f and upload the JSON file

可以使用 OCE 工具包测试到生产 CLI 实用程序，来自动将资产与内容模型及其相关项一起从源 Oracle Content Management 服务器导入或导出到目标 Oracle Content Management 服务器。

管理者或内容管理员可以在测试到生产环境中使用 OCE 工具包命令导入或导出分类。使用管理者权限，可以添加分类的草稿。使用内容管理员权限，可以创建新分类。

分类生命周期操作（例如提升分类、分配到资料档案库和发布分类）可用于测试到生产。可以将资产以及类别信息和分类一起导入到一个导入文件中，也可以从一个导出文件一起导出资产以及类别信息和分类。

导入和导出推荐

使用 OCE 工具包命令，可以将推荐从 Oracle Content Management 服务器导入到本地计算机，或者将推荐从本地计算机导出到服务器。

`cec download-recommendation <name>` 命令从 Oracle Content Management 中导出推荐。它从 Oracle Content Management 服务器下载推荐。

您可以在此命令中使用以下选项：

- `--status, -t [promoted | published] [required]`：指定推荐状态。
- `--id, -i`：如果其他推荐具有相同名称，则指定推荐 ID。
- `--server, -s`：指定注册的 Oracle Content Management 服务器，或者使用在 `cec.properties` 文件中指定的服务器。
- `--help, -h`：显示命令的帮助。

`download-recommendation` 命令的一些示例如下所示：

```
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>` 命令将推荐导入到 Oracle Content Management。它将推荐上载到 Oracle Content Management 服务器。

您可以在此命令中使用以下选项：

- `--createnew, -c`：创建新推荐。
- `--name, -n`：指定新推荐的名称。
- `--abbreviation, -a`：指定新推荐的缩写。
- `--description, -d`：指定新推荐的说明。
- `--file, -f`：指示推荐是否来自文件。
- `--server, -s`：指定注册的 Oracle Content Management 服务器，或者使用在 `cec.properties` 文件中指定的服务器。
- `--help, -h`：显示命令的帮助。

`upload-recommendation` 命令的一些示例如下所示：

```
cec upload-recommendation Recommendation1  
Create a new recommendation or a draft of an existing recommendation on  
upload
```

```
cec upload-recommendation Recommendation1 -s UAT  
Create a new recommendation or a draft of an existing recommendation on  
upload on the registered server UAT
```

```
cec upload-recommendation Recommendation1 -
```

```
c
Create a new recommendation on upload

cec upload-recommendation Recommendation1 -c -n Recommendation 1_2 -a
t12 -d
Create a new recommendation on upload with the given name, abbreviation

"Recommendation1 copy" and description cec upload-
recommendation
Create a new recommendation or a draft of an existing recommendation in
<file-name>.json -f and upload the JSON file
```

添加或删除集合内容

作为开发人员，您可以使用 OCE 工具包命令向集合添加内容并从集合中删除内容。

以下 `cec-control-content` 命令将 **Repo1** 资料档案库中的所有项添加到 **UAT** 已注册服务器上的 **Collection1** 集合：

```
cec control-content add -l Collection1 -r Repo1 -s UAT
```

以下 `cec-control-content` 命令从 **UAT** 已注册服务器上的 **Collection1** 集合中删除所有项：

```
cec control-content remove -l Collection -s UAT
```

使用 OCE 工具包开发定制字段编辑器

OCE 工具包支持开发字段外观类型的组件。开发人员可以创建和管理定制字段编辑器。

对于字段外观类型的组件，可以执行以下任务：

- 打开、复制或删除组件
- 发布或取消发布组件
- 导出或导入组件
- 添加或删除组件上的成员
- 查看属性
- 选择组件标识

可以按字段外观类型筛选组件列表。

以下 OCE 工具包命令可用于开发定制字段编辑器：

```
cec add-field-editor <name>                               Adds a field editor
to a field in a content type.                             [alias: afe]
  cec remove-field-editor <name>                         Removes a field
editor from a field in a content type.                   [alias: rfe]
```

```
cec add-field-editor
-----
Usage: cec add-field-editor <name>

Adds a field editor to a field in a content type.

Options:
  --template, -t          The template the content type is from [required]
  --contenttype, -c      The content type [required]
  --field, -f            The field the field editor is for [required]
  --contenttemplate, -n  Flag to indicate the template is a content template
  --help, -h            Show help [boolean]
```

```
Examples:
  cec add-field-editor editor1 -t BlogTemplate -c BlogPost -f
  summary          Use editor1 as the appearance for field summary in
  content type BlogPost from local template at src/templates/BlogTemplate
  cec add-field-editor editor1 -t BlogTemplateContent -n -c BlogPost -f
  summary          Use editor1 as the appearance for field summary in content type
  BlogPost from local template at src/content/BlogTemplateContent
```

```
-----
cec remove-field-editor
-----
Usage: cec remove-field-editor <name>

Removes a field editor from a field in a content type.

Options:
  --template, -t          The template the content type is from [required]
  --contenttype, -c      The content type [required]
  --field, -f            The field the field editor is for [required]
  --contenttemplate, -n  Flag to indicate the template is a content template
  --help, -h            Show help [boolean]
```

```
Examples:
  cec remove-field-editor editor1 -t BlogTemplate -c BlogPost -f
  summary          Remove editor1 as the appearance for field summary in
  content type BlogPost from local template at src/templates/BlogTemplate
  cec remove-field-editor editor1 -t BlogTemplateContent -n -c BlogPost -f
  summary          Remove editor1 as the appearance for field summary in content type
  BlogPost from local template at src/content/BlogTemplateContent
```

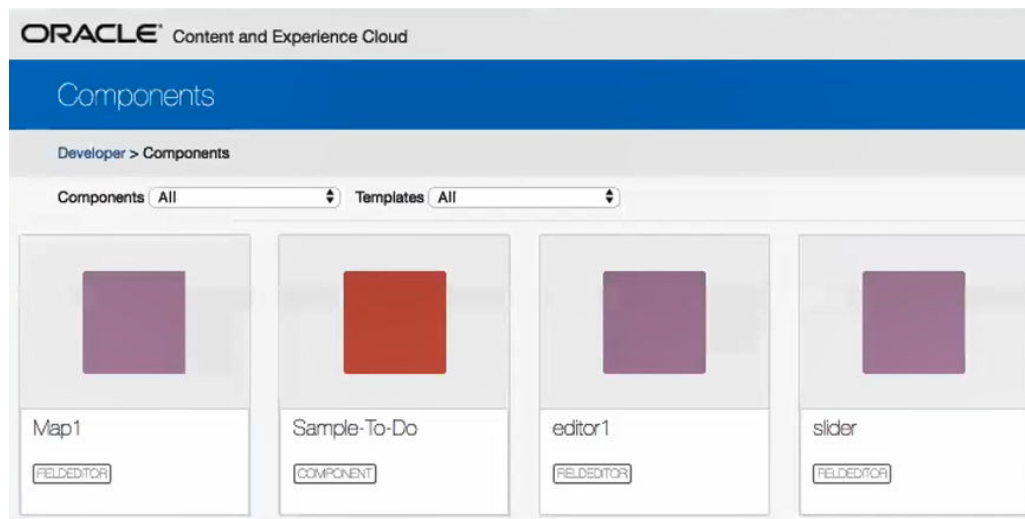
随 OCE 工具包提供以下字段外观组件示例：

- TextFieldEditor
- SliderFieldEditor
- MapFieldEditor

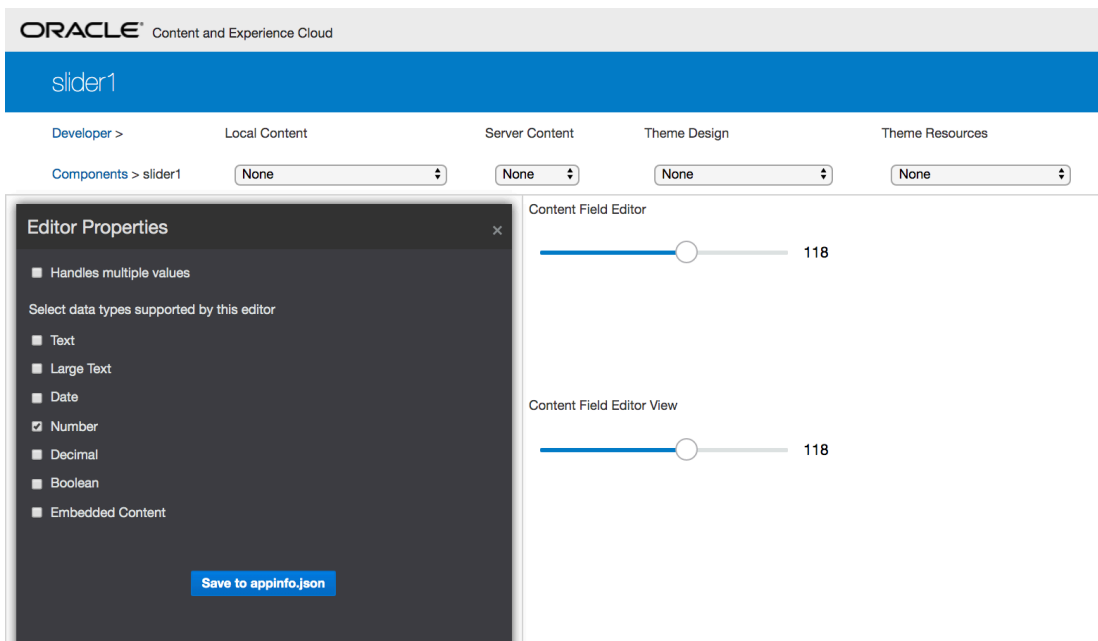
下图显示了可用于开发示例字段外观组件的 OCE 工具包命令。

```
2
3 # create
4 cec cc editor1 -f TextFieldEditor
5
6 cec cc slider -f SliderFieldEditor
7
8 cec cc Map1 -f MapFieldEditor
9
10 #local testing
11
12 # add to content type field
13 cec add-field-editor editor1 -t SimpleContent -n -c SimpleType -f title
14
15 cec add-field-editor slider -t SimpleContent -n -c SimpleType -f value
16
17
18 # upload editors
19 cec ulcp editor1,slider -p -s
20
21 # upload content
22
23 cec cr Repo5 -s
24
25 cec upload-content SimpleContent -r Repo5 -s
26
```

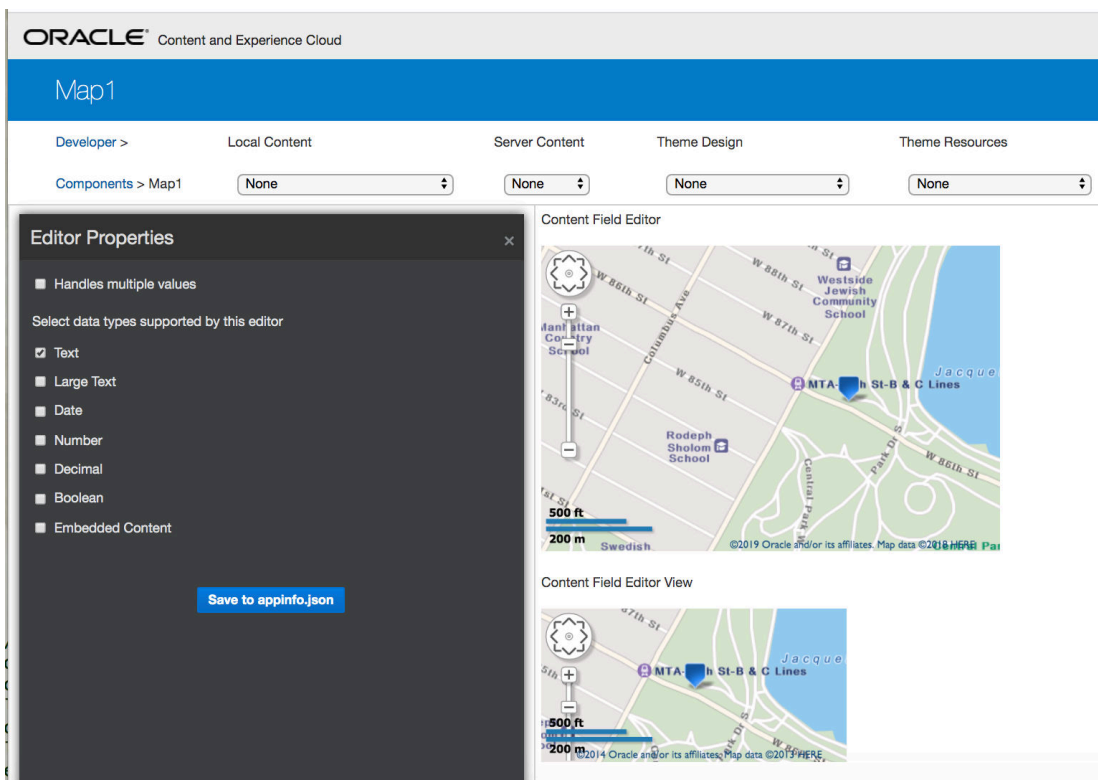
您可以在本地服务器上创建现成的字段外观组件，测试这些组件，然后将其上载到 Oracle Content Management 实例。下图显示了 localhost:8085 上的这些组件。



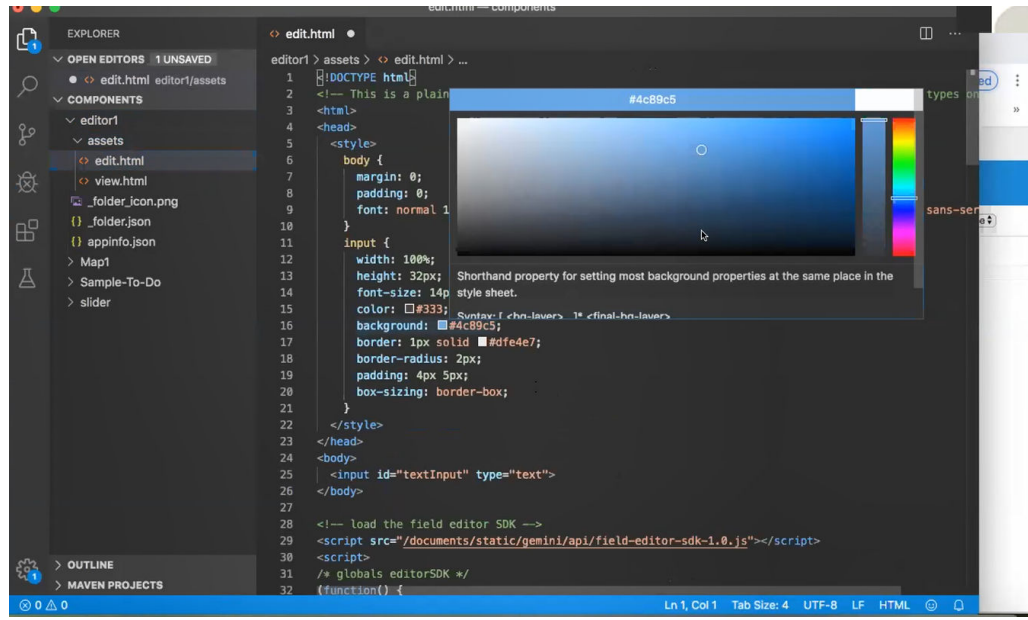
您可以在本地服务器上测试每个组件，例如滑块。可以在那里选择组件的属性，然后保存该组件。



对于地图组件，可以在地图上单击以提供位置作为编辑器值。



您可以编辑组件的 HTML 文件以更改其设置，例如背景颜色。



您可以使用 OCE 工具包命令在本地将字段编辑器与内容类型的字段关联：

```
cec add-field-editor editor1 -t SimpleContent -n -c SimpleType -f title
```

```
cec add-field-editor slider -t SimpleContent -n -c SimpleType -f value
```

配置并测试完定制字段编辑器后，可以使用 OCE 工具包命令将这些编辑器上载到 Oracle Content Management 实例：

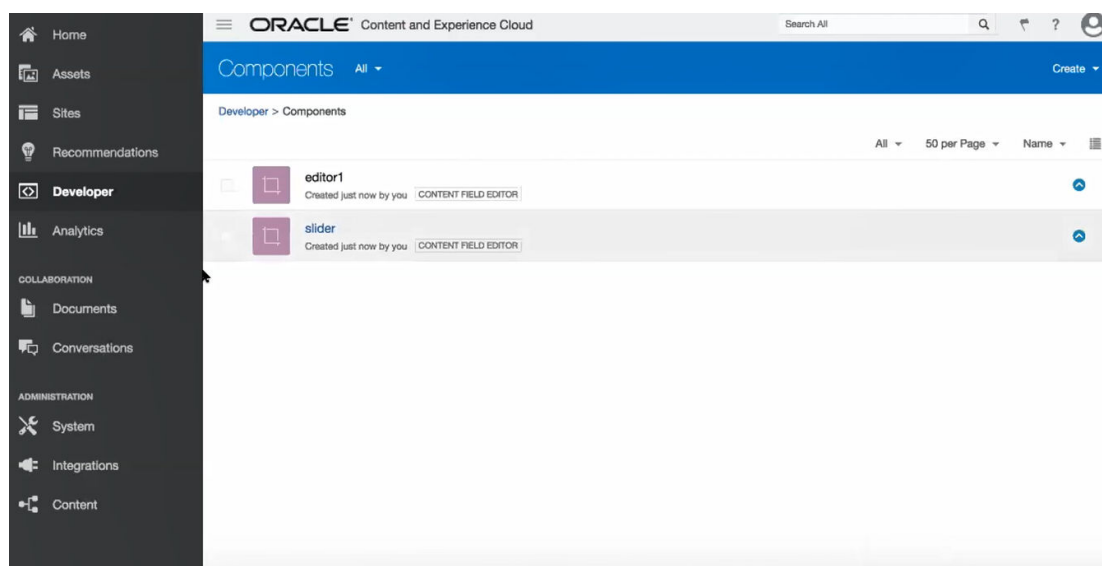
```
# upload editors
cec ulcp editor1,slider -p -s Latest
```

```
# upload content
```

```
cec cr Repo5 -s
```

```
cec upload-content SimpleContent -r Rpo5 -s
```

上载定制字段编辑器时，组件将导入到 Oracle Content Management 实例中。



将站点从一个服务器传输或更新到另一个服务器

作为开发人员，您可以使用 OCE 工具包命令创建站点及其内容或将站点及其内容从服务器 A 更新到服务器 B。

默认情况下会传输所有资产，可以选择指定 `-p` 以仅传输发布的资产。使用 `-s <server>` 指定源服务器，使用 `-d <destination>` 指定目标服务器。如果站点包含来自其他资料档案库的资产，可以选择提供资料档案库映射，否则不会传输这些资产。

您可以使用以下命令将站点从测试环境更新或传输到生产环境：

```
cec transfer-site site --from server --to server --repository r
--localization-policy l
```

有关其他选项，请参见[使用 cec 命令行实用程序](#)。

传输没有内容项的站点

作为开发人员，您可以使用 OCE 工具包将不含内容项的站点从一个 Oracle Content Management 服务器传输到另一个服务器。

要传输具有大量内容资产的站点，需要将站点与其内容分开。您可以将 `--excludecontent` 选项（快捷方式 `-x`）添加到 `transfer-site` 命令。

设置此选项时，将仅传输站点。

例如：

```
cec transfer-site Site1 -s DEV -d UAT -r Repository1 -l LocalizationPolicy1 -x
```

分组下载或上载站点的内容项

作为开发人员，当您将站点从一个 Oracle Content Management 服务器传输到另一个服务器时，您可以分组下载或上载站点的内容项。

要传输具有大量内容资产的站点，需要将站点与其内容分开。传输没有内容的站点 (`cec transfer-site --excludecontent`) 后，使用 `cec transfer-site-content` 传输站点的内容。

例如：

```
cec transfer-site-content <name>
```

此命令创建脚本，用于将企业站点内容从一个 Oracle Content Management 服务器传输到另一个服务器。该命令用于传输大量内容项，将分批传输这些项。默认情况下，此命令将不执行脚本，并且将传输所有资产。可以指定 `-p` 来仅传输发布的资产。

使用 `-s <server>` 指定源服务器，使用 `-d <destination>` 指定目标服务器。

选项

```
--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]
```

如果未设置选项 `--execute`，则在 `cec transfer-site-content` 完成后，执行生成的脚本 `<site name>_downloadcontent` 来从源服务器下载内容，并执行 `<site name>_uploadcontent` 将下载的内容上载到目标服务器。

示例

```
cec transfer-site-content Sitel -s DEV -d UAT -r Repository1 Generate  
script Sitel_downloadcontent and Sitel_uploadcontent  
cec transfer-site-content Sitel -s DEV -d UAT -r Repository1 -e  
Generate script Sitel_downloadcontent and Sitel_uploadcontent and  
execute them  
cec transfer-site-content Sitel -s DEV -d UAT -r Repository1 -n 200  
cec transfer-site-content Sitel -s DEV -d UAT -r Repository1 -p
```

使用 OCE 工具包为站点页编制索引

可以使用 OCE 工具包为站点页上的文本创建内容项并为站点启用页搜索。

以下部分介绍如何使用 OCE 工具包为站点页编制索引：

1. 为站点页文本创建内容类型
2. 使用 OCE 工具包创建页索引内容项

3. 将内容搜索添加到 Oracle Content Management 中的站点

为站点页文本创建内容类型

对于内容类型，为类型指定名称、必需字段值、默认内容布局。

- 类型名称
指定任何有效内容类型名称。
- 字段
以下字段是必需的。

字段名称	字段类型	数据字段值的数目	说明
site	文本	单个	站点名称
pageid	文本	单个	页 ID
pagename	文本	单个	页名称
pageurl	文本	单个	页 URL
pagedescription	文本	单个	页说明
keywords	文本	多个 (无最大值)	页上的所有文本以及来自页上内容项的所有文本字段的值，通过 OCE 工具包 index-site 命令获得

```

{{#fields}}
<div class="indextype"></div>
<div>
  <a href="{{pageFullURL}}" title="{{pagename}}">{{pagename}}</a>
</div>
{{/fields}}

```

```

content.fields.pageFullURL = SCSRenderAPI.getSitePrefix() +
content.fields.pageurl;

```

- 创建类型的内容布局。
内容布局应显示站点名称和 URL 以导航到页。例如，在 layout.html 中：

```

{{#fields}}
<div class="indextype"></div>
<div>
  <a href="{{pageFullURL}}"title="
{{pagename}}">{{pagename}}</a>
</div>
{{/fields}}

```

- 在 render.js 中，生成页完整 URL：

```

content.fields.pageFullURL =
SCSRenderAPI.getSitePrefix() +
content.fields.pageurl;

```

- 将内容布局设置为类型的默认内容布局。

```
content.fields.pageFullURL =  
SCSRenderAPI.getSitePrefix() + content.fields.pageurl;
```

使用 OCE 工具包创建页索引内容项

您可以使用 OCE 工具包命令创建页索引内容项。

先决条件：

- 已在本地计算机上安装和设置 OCE 工具包。
- Oracle Content Management 上的站点已发布。
- 站点页上的内容项已发布到站点通道。

在命令行界面中，键入以下 OCE 工具包命令：

```
cec index-site site name -c content type name -p
```

在命令中，*site name* 是站点的名称，*content type name* 是为页文本创建的内容类型；选项 *-p* 指示在创建后发布页索引内容项。

Usage: cec index-site <site>

Create content item for each page with all text on the page. If the page index content item already exists for a page, updated it with latest text on the page. Specify -c <contenttype> to set the page index content type. Optionally specify -p to publish the page index items after creation or update.

Options:

```
--contenttype, -c <contenttype> page index content type  
--publish, -p      publish page index items  
--help, -h        Show
```

help

[boolean]

Examples:

```
cec index-site Site1 -c PageIndex  
cec index-site Site1 -c PageIndex -p
```

要查看用法，可以键入 `cec index-site -h`

将内容搜索添加到 Oracle Content Management 中的站点

可以使用搜索页和搜索字段将内容搜索添加到 Oracle Content Management 站点。

要将内容搜索添加到站点，请执行以下操作：

1. [将搜索页添加到站点](#)
2. [将搜索字段添加到主题](#)

将搜索页添加到站点

您可以将搜索页添加到站点，并将内容列表组件添加到搜索页。

添加搜索页：

1. 将页添加到站点并将其设置为搜索页。
2. 将内容列表组件添加到搜索页。
3. 将内容类型设置为之前创建的页索引内容类型。

将搜索字段添加到主题

要使搜索字段显示在站点的每个页上，可以将搜索字段添加到主题的布局 HTML 页。

例如：

```
<div align="center">
<input id="searchonpage" type="text" size="30" placeholder="Search on
page. . ."/>
</div>
```

1. 添加输入字段：

```
<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. 在 HTML 正文的末尾添加 JavaScript。

使用 OCE 工具包为多语言站点编制索引

可以使用 OCE 工具包为多语言 (MLS) 站点编制索引，以进行翻译和页及内容项搜索。

可以构建多语言站点索引，并在发布站点之前对其进行测试。使用 OCE 工具包 `cec index-site` 命令为多语言站点编制索引。转到 `cec-components` 目录并发出此命令（不带任何选项），以查看该命令的帮助信息：

```
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
```

页索引项按页和语言而存在。为每种语言创建的页索引内容项创建为默认语言页索引项的翻译。在正在运行的站点中进行查询时，搜索和内容列表从站点 URL 中获取语言。这样将自动筛选搜索。

发布多语言站点之前，需要对其编制索引并进行翻译，为此将需要翻译作业。请参见在 [Oracle Content Management 服务器中创建新站点或资产翻译作业](#)。

默认语言英语是必需的。对于支持的每种语言（必需和可选），执行索引创建并创建索引项的翻译。如果您运行索引两次，其只执行一次更新。

要使用 OCE 工具包命令为多语言站点编制索引，并翻译和发布该站点，请执行以下操作：

1. 为站点创建内容类型，并使其可在资料档案库中使用。请参见 [为站点页文本创建内容类型](#)。
2. 选择验证策略。
 - a. 单击左侧导航菜单中的资产。
 - b. 在资产菜单中选择本地化策略。
 - c. 选择本地化策略。
 - d. 如果需要，修改本地化策略，以包括要用于编制站点索引和翻译站点的语言。例如，如果策略只包含英语，您可以添加法语和西班牙语。

所有翻译都是从英语进行。

3. 下载翻译作业。您只能翻译站点中使用的资产。
4. 翻译站点。可以手动或通过集成连接器完成站点翻译。
5. 翻译完成后上传翻译作业。
6. 使用 `cec index-site` 命令为站点编制索引。指定 `-c <contenttype>` 以设置页面索引内容类型。

还可以指定 `-p` 选项来发布站点。然后，可以在发布对实时站点的更改之前验证索引和翻译。

例如，以下 `cec index-site` 命令为使用英语、法语和西班牙语的站点构建站点索引。站点支持的语言来自分配的 L10n 策略，包括默认语言。

```
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 Navigation
- create page index item for Detail Page
- create page index item for Pages
- create page index item for Page Content
- create page index item for Developing Templates
- create page index item for Themes
- add page index items to site channel
- remove page index items for page Search from site channel
- will create/update translate for fr-FR,es-ES
- query site structure with locale fr-FR
- query page data (fr-FR)
- query content on the pages (fr-FR)
- will create 11 page index items (fr-FR)
- will update 0 page index items (fr-FR)
- will remove 1 page index items (fr-FR)
- create page index item for Themes (fr-FR)
- create page index item for Navigation (fr-FR)
- create page index item for Pages (fr-FR)
- create page index item for Detail Page (fr-FR)
- create page index item for Search (fr-FR)
- create page index item for Page Content (fr-FR)
- create page index item for Components (fr-FR)
```

- create page index item for Developing Templates (fr-FR)
- create page index item for Blog (fr-FR)
- create page index item for Home (fr-FR)
- create page index item for Privacy Policy (fr-FR)
- add page index items to site channel
- set page index items in fr-FR as translated
- remove page index items for page Search from site channel
- query site 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. 发布站点以包括翻译。

创建简化的组件以便于组件开发

使用 OCE 工具包创建简化的组件，使开发更轻松。

OCE 工具包提供了简化的组件 SimpleHTML，让您能够更轻松的开始进行定制组件开发：

```
cec create-component -f SimpleHTML
```

此外，还提供了供您开始使用的 JET 组件示例：

```
cec create-component MyComp -f JET-CCA-Demo-Card
```

设置站点编译服务

您可以为编译服务器创建 Docker 映像，以在 Oracle Content Management 中设置站点编译服务。此服务为您提供在发布之前定义要使用的编译服务器的选项。

编译服务器在管理 > 系统 > 站点和资产页上的编译端点 **URL** 字段中具有 URL 形式的端点。可以在此处输入要向服务器注册的完全限定 URL，然后单击测试以验证端点。请参见“设置编译端点 URL”。

站点编译服务编译站点的页，以便在访问发布的站点时返回 HTML 页。当 Oracle Content Management 服务器发布站点时，如果配置了站点编译服务，它将调用该服务以编译页。

为站点指定端点后，您可以在站点属性对话框的静态站点传送选项卡上启用自动编译。发布站点时，它将通过编译服务器进行编译。

站点编译服务是工具包命令的扩展。您可以自己运行 `cec compilation-server`，但 Docker 映像允许您使用标准模式创建编译服务器，然后调整服务的配置。

要创建 Docker 映像并发布 Docker 文件，请执行以下操作：

1. 从 GitHub 下载 Docker 映像信息。
从 GitHub 下载信息时，有其他三个位于编译服务器下的文件。Docker 映像之一是编译服务器，它包含一个 Docker 文件和自述文件。
2. 创建 Docker 映像后将执行 `run.sh` 命令。
此命令将下载 OCE 工具包，安装它，并创建源目录。
3. 然后，您可以定制环境并运行 `run.sh` 命令。您可以指定要使用的端口、超时时间和其他任何要更改的内容，然后更新并使用 `.sh` 命令。
4. 下载 Docker 映像信息并更改编译服务器目录后，可以构建 Docker 文件。
如果已经下载 OCE 工具包，则可以删除 `no-cache` 选项，这样就不必重新下载所有内容。
5. 列出 Docker 映像。
6. Docker 映像可用后，即可进行注册。
7. 验证它是否有效。
转到可用于查看所支持版本的异常文件 REST API。此测试可用于验证它是否运行。
8. 现在，需要注册服务器。它需要服务器名称的第一部分，并且会处理其中的所有其他内容。
9. 返回到系统目录和站点资产，从中可以注册编译端点并单击保存。
10. 转到站点属性，并指定发布内容和发布时间。

编译站点以提高站点页的运行时性能

在 Oracle Content Management 中编译站点可以提高站点页的运行时性能并改进其行为。编译通过为站点中的每一页创建行为方式与原始页完全相同的静态 HTML 文件来实现这一点。

站点编译概览

组成站点“页”的元数据文件在编译过程中进行组合，从而避免通常在运行时需要的服务器请求。您可以有效地将每次页访问呈现成本转变为一次性编译时成本。

通过减少运行时请求数可实现性能改进。这可以包括避免所有内容查询，因为您可以将这些查询的结果直接编译到页中。

因为页立即呈现，所以可以实现行为改进。因此，可以避免诸如“刷新未设置样式的内容”或组件以不确定的顺序显示在页上（例如页脚立即显示，然后随着页上的其他项呈现而向下移动）的问题。

要在 Oracle Content Management 站点内编译页，您需要将该站点作为模板导出，然后使用 OCE 工具包编译该模板并将生成的静态页上载到原始站点。以下步骤可引导您完成模板编译过程，然后介绍如何使用此模型来编译站点。

与预呈现的交互

预呈现专注于返回适合搜索引擎的结果。预呈现生成的页预期不会像原始页一样运行，并且不会执行 JavaScript。它仅返回浏览器预呈现的 HTML 以支持 SEO 文本搜索。

由站点编译创建的静态页预期将在浏览器中运行，并且其行为与原始动态站点页完全相同。这包括动态运行非编译组件和执行运行时查询，其中用户希望结果中的数据完全是最新的，而不是在编译时修复数据。

当租户管理员启用预呈现解决方案时，还可以通过预呈现服务器传送静态页。这使得索引器和 Crawler 可以处理可能仍动态呈现的部分静态页，如内容列表。

内容查询返回包含多信息文本字段值的内容项，因此当内容布局用于内容列表时，不需要单独提取这些值。

控制器站点页呈现

`cec compile-template` 命令使站点页可以直接在浏览器中呈现，而不必通过 Oracle Content Management 站点页控制器。

Oracle Content Management 站点使用控制器模型呈现页，而不使用编译器创建静态 HTML 页。这涉及使用多个请求获取关于站点、站点内的页以及用于呈现页的模板的信息，然后才能在浏览器中实际显示页。

注：

此外，还涉及获取 `controller.js` 文件等更多请求。

已编译的站点页呈现

使用 `cec compile-template` 命令，可以将所有这些步骤编译到静态 HTML 文件中，并通过减少或消除这些服务器请求来提高运行时站点的性能。此外，还可以进一步减少运行时不需要更改的资源的请求，并将它们编译到静态页中。

默认编译过程执行现有运行时控制器的工作，为站点中的每个 `page.json` 文件创建物理 HTML 页。然后，可以将这些页部署到站点并随站点一起发布，并在点击相应的页 URL 时使用这些页。

模板编译

要编译站点，您需要有权访问网站内的主题和组件。站点模板是站点的打包模型。该模板包含迁移和运行站点所需的所有资源。

您可以使用站点模板将站点导出到 OCE 工具包环境，在该环境中可以编译模板中的站点。以下部分将全面介绍 OCE 工具包中植入的 **BlogTemplate** 中的站点编译。

设置

可通过 OCE 工具包使用 `cec compile-template` 命令，并在安装工具包时安装该命令。请遵循有关安装 OCE 工具包的标准说明。

安装 OCE 工具包后，就可以运行 `cec` 命令以创建模板。

1. 安装 OCE 工具包开发目录：

```
mkdir cec-src
cd ./cec-src
cec install

# Start up the CEC Toolkit Development server for testing
cec develop &
```

2. 创建模板。

已使用定制编译器更新植入 BlogTemplate。从此源创建模板：

```
# create a new template (must be in the cec-src folder if not already
there)
cd ./cec-src
cec create-template BlogTemplate -f BlogTemplate
```

植入 **BlogTemplate** 使用定制编译器扩展以前的模板：

- 内容布局编译器

```
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
```

- 页布局编译器

```
cec-src/src/themes/BlogStarterTheme/layouts/post-detail-compile.js
cec-src/src/themes/BlogStarterTheme/layouts/home-compile.js
cec-src/src/themes/BlogStarterTheme/layouts/authors-compile.js
cec-src/src/themes/BlogStarterTheme/layouts/common-compile.js
cec-src/src/themes/BlogStarterTheme/layouts/about-compile.js
cec-src/src/themes/BlogStarterTheme/layouts/index-compile.js
```

编译模板

编译模板时，会为模板中站点的 `structure.json` 文件中的每个页创建一个静态 HTML 页。

编译步骤将页元数据（例如 `100.json`）与页布局（例如 `about.html`）相组合，这样您就有了一个可以立即显示的 HTML 页，而不需要在运行时使用控制器。使用定制编译器，可以通过将以前的动态项（例如导航和组件）编译到页中来进一步减少运行时 JavaScript 的执行。

`cec compile-template` 命令将对站点中的每个页执行以下操作：

- 读取站点和页元数据
- 读取页布局并应用任何页布局编译器
- 对于页上的每个内容框：
 - 使用在 `page.json` 文件中定义的网格扩展内容框
 - 对于内容框中的每个组件：
 - * 应用任何组件编译器

* 将生成的组件标记插入内容框内的相应位置

- 扩展编译的标记中的任何宏，并插入 Oracle Content Management 呈现器在运行时使用的 SCS JavaScript 对象
- 在 `src/templates/<template>/static` 文件夹下保存编译的页标记

要编译站点，请运行：

```
> cec compile-template BlogTemplate
Compile Template: compiling template BlogTemplate
Oracle Content Management Site Compiler
```

```
createPage: Processing pageId 100. Preview URL:
http://localhost:8085/templates/BlogTemplate/index.html
createPage: Processing pageId 105. Preview URL:
http://localhost:8085/templates/BlogTemplate/post-detail.html
createPage: Processing pageId 401. Preview URL:
http://localhost:8085/templates/BlogTemplate/about.html
createPage: Processing pageId 402. Preview URL:
http://localhost:8085/templates/BlogTemplate/search.html
createPage: Processing pageId 403. Preview URL:
http://localhost:8085/templates/BlogTemplate/authors.html
All page creation calls complete.
```

Creating detail pages:

```
createPage: Processing detail pageId 105. Preview URL:
http://localhost:8085/templates/BlogTemplate/post-detail/1481786063051-developing-content-layout-for-content-and-experience-cloud
createPage: Processing detail pageId 105. Preview URL:
http://localhost:8085/templates/BlogTemplate/post-detail/1481786063052-dynamic-dom-manipulation-in-content-layout
createPage: Processing detail pageId 105. Preview URL:
http://localhost:8085/templates/BlogTemplate/post-detail/1481786063045-getting-media-url-in-the-content-layout
createPage: Processing detail pageId 105. Preview URL:
http://localhost:8085/templates/BlogTemplate/post-detail/1481786063053-getting-reference-items-in-content-layout
createPage: Processing detail pageId 105. Preview URL:
http://localhost:8085/templates/BlogTemplate/post-detail/1481786063048-navigating-to-a-search-page-with-search-query
createPage: Processing detail pageId 105. Preview URL:
http://localhost:8085/templates/BlogTemplate/post-detail/1481786063050-alex-read
createPage: Processing detail pageId 105. Preview URL:
http://localhost:8085/templates/BlogTemplate/post-detail/1481786063047-jerrold-summers
createPage: Processing detail pageId 105. Preview URL:
http://localhost:8085/templates/BlogTemplate/post-detail/1481786063049-kelly-emerson
createPage: Processing detail pageId 105. Preview URL:
http://localhost:8085/templates/BlogTemplate/post-detail/1481786063043-samantha-howard
createPage: Processing detail pageId 105. Preview URL:
http://localhost:8085/templates/BlogTemplate/post-detail/1481786063046-raising-triggers-from-content-layout
createPage: Processing detail pageId 105. Preview URL:
http://localhost:8085/templates/BlogTemplate/post-detail/1481786063044-rendering-the-content-layout-using-mustache-template
All detail page creation calls complete.
```

编译错误

编译过程中，您会得到三种类型的消息：信息、警告和错误。

1. 信息消息是关于您应该注意的问题，但是这些消息可能应作为编译的一部分：
 - 将在运行时呈现的占位符内容项。
 - 缺少页布局编译器。页布局可能没有任何动态元素，因此如果缺少页布局，则不会被视为问题。
 - 标记为“访问时呈现”的项；即，站点开发人员希望在运行时动态呈现此组件，即使页已编译也是如此。
2. 警告消息是关于可能会影响运行中站点性能的问题，应在可能的情况下进行修复。此类别中的项如下：
 - 缺少内容布局或定制组件编译器。如果没有这些项，组件将像以前那样动态地呈现到页中。
 - 缺少内容布局映射。您不太可能希望使用系统默认内容布局来呈现内容项。
3. 错误消息是关于指示编译失败的问题。在可能的情况下，页将继续编译，但是整体编译将退出并显示错误。
 - 这很可能是由定制编译器中的 JavaScript 错误导致的，必须解决所有错误。



注：

为了减少重复消息，同一消息在每次编译中只出现一次，无论它是否出现在多个页上。

上述示例具有以下编译退出状态：

```
Compilation completed with 0 errors and 3 warnings.  
to display warnings, run with --verbose (-v) option.
```

如果使用 "-v" 选项重新运行编译步骤，将看到以下输出：

```
> cec compile-template BlogTemplate --verbose  
Compile Template: compiling template BlogTemplate  
Oracle Content Management Site Compiler  
  
createPage: Processing pageId 100. Preview URL:  
http://localhost:8085/templates/BlogTemplate/index.html  
createPage: Processing pageId 105. Preview URL:  
http://localhost:8085/templates/BlogTemplate/post-detail.html  
Info: no content item specified for placeholder: a890a65c-c0fc-451b-966b-  
e606ca18a1f4  
component will render at runtime.  
Info: no content item specified for placeholder: c90bbc10-  
c9d8-4a54-8dd4-7a8251e8efbb  
component will render at runtime.  
Info: no content item specified for placeholder: f12691e1-79ab-4d1f-  
a8b9-3af8c638dd26
```

```
component will render at runtime.  
createPage: Processing pageId 401. Preview URL:  
http://localhost:8085/templates/BlogTemplate/about.html  
createPage: Processing pageId 402. Preview URL:  
http://localhost:8085/templates/BlogTemplate/search.html  
Info: Component: "fdfd0392-e901-48f6-8044-36803c836aa1" of type "scs-  
contentlist" marked as  
"render on access", will not be compiled.  
Info: Component: "ba9f3711-4367-444e-ae38-71289fc10e73" of type "scs-  
contentlist" marked as  
"render on access", will not be compiled.  
createPage: Processing pageId 403. Preview URL:  
http://localhost:8085/templates/BlogTemplate/authors.html  
All page creation calls complete.
```

```
Creating detail pages:  
createPage: Processing detail pageId 105. Preview URL:  
http://localhost:8085/templates/BlogTemplate/post-detail/1481786063051-developing-content-  
layout-for-content-and-experience-cloud  
createPage: Processing detail pageId 105. Preview URL:  
http://localhost:8085/templates/BlogTemplate/post-detail/1481786063052-dynamic-dom-  
manipulation-in-content-layout  
createPage: Processing detail pageId 105. Preview URL:  
http://localhost:8085/templates/BlogTemplate/post-detail/1481786063045-getting-media-url-in-  
the-content-layout  
createPage: Processing detail pageId 105. Preview URL:  
http://localhost:8085/templates/BlogTemplate/post-detail/1481786063053-getting-reference-  
items-in-content-layout  
createPage: Processing detail pageId 105. Preview URL:  
http://localhost:8085/templates/BlogTemplate/post-detail/1481786063048-navigating-to-a- search-  
page-with-search-query  
createPage: Processing detail pageId 105. Preview URL:  
http://localhost:8085/templates/BlogTemplate/post-detail/1481786063050-alex-read  
Warning: failed to find content layout map entry for: Starter-Blog-  
Author:header. Will compile  
using the system default layout.  
Warning: failed to find content layout map entry for: Starter-Blog-  
Author:content. Will compile  
using the system default layout.  
Warning: failed to find content layout map entry for: Starter-Blog-  
Author:sidebar. Will compile  
using the system default layout.  
createPage: Processing detail pageId 105. Preview URL:  
http://localhost:8085/templates/BlogTemplate/post-detail/1481786063047-jerrold-summers  
createPage: Processing detail pageId 105. Preview URL:  
http://localhost:8085/templates/BlogTemplate/post-detail/1481786063049-kelly-emerson  
createPage: Processing detail pageId 105. Preview URL:  
http://localhost:8085/templates/BlogTemplate/post-detail/1481786063043-samantha-howard  
createPage: Processing detail pageId 105.  
Preview URL: http://localhost:8085/templates/BlogTemplate/post-detail/1481786063046-  
raising-triggers-from- content-layout  
createPage: Processing detail pageId 105. Preview URL:  
http://localhost:8085/templates/BlogTemplate/post-detail/1481786063044-rendering-the-content-  
layout-using-mustache-template  
All detail page creation calls complete.
```

```
Compilation completed with 0 errors and 3 warnings.

*** compiled template is ready to test
*** to render non-compiled pages, remove compiled files from under: /
private/tmp/cec-
src/src/templates/BlogTemplate/static
```

详细信息页编译

详细信息页编译可整理它在编译过程中遇到的所有内容项。然后，它重新编译它找到的每个内容项的详细信息页，使用缩略名值定义新详细信息页的 URL。

前面的编译输出分为两个部分：

1. 页编译
2. 详细信息页编译

在前面的示例中，您将看到有关无 **Starter-Blog-Author** 页的内容布局映射条目的警告。但是，我们不需要 **Starter-Blog-Author** 的详细信息页。详细信息页仅针对 **Starter-Blog-Post** 内容项。要去除这些错误，如果内容项未在其设置中引用显式详细信息页，则可以使用以下选项从详细信息页编译中排除这些内容项：

```
--noDefaultDetailPageLink, -o Do not generate compiled detail page for
items/content lists
    that use the default detail page.
```

因此，在排除使用默认详细信息页的详细信息页创建的情况下，重新运行编译会生成以下输出：

```
> cec compile-template BlogTemplate --noDefaultDetailPageLink
Compile Template: compiling template BlogTemplate
Oracle Content Management Site Compiler
```

```
createPage: Processing pageId 100. Preview URL:
http://localhost:8085/templates/BlogTemplate/index.html
createPage: Processing pageId 105. Preview URL:
http://localhost:8085/templates/BlogTemplate/post-detail.html
createPage: Processing pageId 401. Preview URL:
http://localhost:8085/templates/BlogTemplate/about.html
createPage: Processing pageId 402. Preview URL:
http://localhost:8085/templates/BlogTemplate/search.html
createPage: Processing pageId 403. Preview URL:
http://localhost:8085/templates/BlogTemplate/authors.html
All page creation calls complete.
```

```
Creating detail pages:
createPage: Processing detail pageId 105. Preview URL:
http://localhost:8085/templates/BlogTemplate/post-detail/1481786063051-developing-content-layout-for-content-and-experience-cloud
createPage: Processing detail pageId 105. Preview URL:
http://localhost:8085/templates/BlogTemplate/post-detail/1481786063052-dynamic-dom-manipulation-in-content-layout
```

```
createPage: Processing detail pageId 105. Preview URL:
http://localhost:8085/templates/BlogTemplate/post-detail/1481786063045-getting-media-url-in-the-content-layout
createPage: Processing detail pageId 105. Preview URL:
http://localhost:8085/templates/BlogTemplate/post-detail/1481786063053-getting-reference-items-in-content-layout
createPage: Processing detail pageId 105. Preview URL:
http://localhost:8085/templates/BlogTemplate/post-detail/1481786063048-navigating-to-a-search-page-with-search-query
All detail page creation calls complete.

Compilation completed with no errors.

*** compiled template is ready to test
*** to render non-compiled pages, remove compiled files from under: /
private/tmp/cec-
src/src/templates/BlogTemplate/static
```

编译特定页

您不需要编译模板中的所有页，您可以选择要编译的页。如果您要调试或更新正在处理的特定页，而不必不断地重新编译整个站点，那么这将非常有用。

要编译特定页，请使用 `--pages (-p)` 选项，后跟要编译的页的列表。

注：

注：如果列表中页上的内容项引用了详细信息页，则也会编译详细信息页，即使它们未明确包含在内也是如此。

```
cec compile-template BlogTemplate --pages 401,402
Compile Template: compiling template BlogTemplate
Oracle Content Management Site Compiler

createPage: Processing pageId 401. Preview URL:
http://localhost:8085/templates/BlogTemplate/about.html
createPage: Processing pageId 402. Preview URL:
http://localhost:8085/templates/BlogTemplate/search.html
All page creation calls complete.

Compilation completed with no errors.
*** compiled template is ready to test
*** to render non-compiled pages, remove compiled files from under: /
private/tmp/cec-
src/src/templates/BlogTemplate/static
```

站点编译

前面的步骤概述了如何创建和编译本地模板。通常，您将编译实际的 Oracle Content Management 站点。

要编译 Oracle Content Management 站点，需要将站点导出到模板中，然后编译模板。模板程序包将包含站点以及用于编译与模板一起导出的主题和组件中的站点所需的任何定制代码。

站点编译的先决条件

以下步骤假定您已在 Oracle Content Management 服务器中创建名为 **BlogSite** 的站点。您可以上载上述 **BlogTemplate**，然后从模板创建此站点。

例如：

```
> cec upload-template BlogTemplate --server UAT

> cec create-site BlogSite --template BlogTemplate --repository
<yourRepository> --
localizationPolicy <yourLocalizationPolicy> --defaultLanguage en-US --server
UAT
```

选择或创建站点后，就可以执行以下部分中的步骤以编译站点。

编译站点

OCE 工具包中的 `cec create-template-from-site` 命令可以创建站点的模板，因此可以编译站点。

```
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]
```

要编译站点，请执行以下操作：

1. 从站点创建模板：

```
> cec create-template-from-site BlogTemplate --site BlogSite --
includeunpublishedassets -
server UAT
```

2. 下载模板：

```
> cec download-template BlogTemplate --server UAT
```

3. 编译模板：

```
> cec compile-template BlogTemplate --noDefaultDetailPageLink --verbose --
server UAT
--channelToken e1bb88cdc1e025c8dd278f6b676877a3
```

 注:

您将需要在站点发布通道的服务器中获取站点的通道令牌 (--channelToken (-c) 选项)。然后, 这将于模板中站点内的所有查询。

4. 上载编译的站点页。
将编译的静态文件复制到站点的静态文件夹中:

```
> cec upload-static-site-files src/templates/BlogTemplate/static --site BlogSite --server UAT
```

还原到非编译行为

要还原到非编译行为, 您需要删除已上载到站点中的静态文件:

```
> cec delete-static-site-files BlogSite --server UAT
```

如果该站点当前已发布, 则重新发布该站点。这将删除站点中的“静态”文件夹; 它不会删除站点。

定制编译器

可以通过定制编译器进一步改进编译过程。您可以调用定制编译器, 以将页布局、部分布局、定制组件或内容布局编译到页中, 避免需要在运行时动态添加组件的情况。

如果组件不存在定制编译器或者组件编译器未返回任何标记, 则将在运行时呈现该组件, 就像从未对其进行编译一样。

以下组件支持定制编译器:

- 页布局
- 部分布局
- 定制组件
- 内容布局

以下示例显示页布局和内容布局编译器。部分布局和定制组件编译器遵循与内容布局编译器相同的模型。

如果不存在定制编译器, 则组件通过 `render.js` 文件来呈现。

如果确实存在定制编译器, 则调用该编译器, 生成的 HTML 将插入到页中。定制编译器还可以指示组件在运行时需要 JavaScript 水合。如果是这种情况, 将调用 `render.js` 文件内组件的 `hydrate()` 函数。如果组件不需要水合, 则永远不加载组件的 `render.js` 文件。

 注:

编译是仅运行时 (已发布站点) 功能。在编辑、导航或预览模式下查看站点时, 页像平常一样工作, 所有组件都将始终动态添加到页中。

约束

`cec compile-template` 操作是 NodeJS 应用程序，可在任何浏览器外部运行。由于页不是在浏览器中呈现的，因此没有 DOM 或窗口对象，而且 JQuery、VueJS 或 KnockoutJS 等客户端 JavaScript 库将无法工作。

虽然可以使用 JSDOM 之类的库来创建 DOM 对象并运行这些客户端库，但是这样做没有什么益处。对于简单的 HTML 语法分析，可以使用 cheerio 之类的 NodeJS HTML 语法分析器。

标记返回的 HTML 需要是有效 HTML。它将通过语法分析器进行传递，只有经过语法分析的 HTML 才会添加到页中。这是为了确认编译的 HTML 没有可能中断内容框的不匹配标记。

调试定制编译器

开发定制编译器时，您将需要调试您的代码。

`cec compile-template` 命令附带了 `--debug (-d)` 选项，该选项将启动编译器并设置 `--inspect-brk` 标记，以便可以将调试器连接到进程。然后，您可以遵循标准节点调试来检查您的代码。

```
cec compile-template BlogTemplate --noDefaultDetailPageLink --debug
```

```
Debugger listening on ws://127.0.0.1:9229/8a8eba83-42d2-476b-adc1-  
b29ab4e92642  
For help see https://nodejs.org/en/docs/inspector
```

此外，还可以使用 `--pages (-p)` 选项将要编译的页限定为特定页。

页布局编译器

页布局编译器是用于编译相应页布局的 NodeJS (CommonJS) JavaScript 模块。

特定页布局的页布局编译器通过具有 `-compile.js` 扩展的名称关联进行定义：

- `src`
 - `themes`
 - * `<yourTheme>`
 - * `layouts`
 - * `<yourPageLayout>.html`
 - * `<yourPageLayout>-compile.js`

如果页布局不存在 `-compile.js`，则不应用定制编译。

页布局编译器需要实施 `compile()` 接口，其返回 `Promise`；例如 `about-compile.js`：

```
var mustache = require('mustache');  
  
var PageCompiler = function () {};  
  
PageCompiler.prototype.compile = function (args) {  
  var self = this,  
      layoutMarkup = args.layoutMarkup;
```

```
self.SCSCompileAPI = args.SCSCompileAPI;

return new Promise function (resolve, reject) {
  var compiledPage = layoutMarkup,
      id = self.SCSCompileAPI.navigationRoot;

  // page is compiled so there is no FOUC, can remove the opacity
  workaround
  compiledPage = compiledPage.replace('opacity: 0;', 'opacity:
1;');

  // remove the dynamic menu creation, we'll be compiling it
  here
  compiledPage = compiledPage.replace('<script
src="_scs_theme_root_/assets/js/topnav.js"></script>', '');

  // add link to Home page. . .
  var homePageURL = (self.SCSCompileAPI.getPageLinkData(id) ||
{}).href;
  if (homePageURL) {
    compiledPage = compiledPage.replace('class="navbar-brand"
href="#"', 'class="navbar-brand" href="' + homePageURL + '');
  }

  // build the menu and add it to the page
  var navMenu = self.createNavMenu();
  compiledPage = compiledPage.replace('<!-- navigation menu goes
in
here -->', navMenu);

  // return the compiled page
  resolve(compiledPage);
};

// Create the navigation menu that was previously dynamically generated
on
each page
PageCompiler.prototype.createNavMenu = function () {
  . . .
}

module.exports = new PageCompiler();
```

组件编译器

定制组件编译器都遵循与页编译器相同的模型，可以为部分布局、定制组件和内容布局创建该编译器。

编译过程中，`cec compile-template` 命令将在与组件的 `render.js` 文件相同的位置查找 `compile.js` 文件：

- `src`

```
- components
  * <yourComponent>
    * assets
      * render.js
      * compile.js
```

如果此文件不存在，将不编译组件并将在运行时呈现组件。

如果该文件确实存在，它需要实施 `compile()` 接口，其返回 `Promise`。例如，以下 `Starter-Blog-Author-Summary` 是定制内容布局编译器：

```
var fs = require('fs'),
    path = require('path'),
    mustache = require('mustache');

var ContentLayout = function (params) {
  this.contentClient = params.contentClient;
  this.contentItemData = params.contentItemData || {};
  this.scsData = params.scsData;
};

ContentLayout.prototype = {
  contentVersion: '>=1.0.0 <2.0.0',

  compile: function () {
    var compiledContent = '',
        content = JSON.parse(JSON.stringify(this.contentItemData)),
        contentClient = this.contentClient;

    // Store the id
    content.fields.author_id = content.id;

    if (this.scsData) {
      content.scsData = this.scsData;
      contentType = content.scsData.showPublishedContent === true ?
'published' : 'draft';
      secureContent = content.scsData.secureContent;
    }

    // calculate the hydrate data
    content.hydrateData = JSON.stringify({
      contentId: content.id,
      authorName: content.fields['starter-blog-author_name']
    });

    try {
      // add in style - possible to add to <head> but inline for
simplicity
      var templateStyle = fs.readFileSync(path.join(__dirname,
'design.css'), 'utf8');
      content.style = '<style>' + templateStyle + '</style>';

      var templateHtml = fs.readFileSync(path.join(__dirname,
```

```
'layout.html'), 'utf8');
    compiledContent = mustache.render(templateHtml, content);
  } catch (e) {
    console.error(e.stack);
  }

  return Promise.resolve({
    content: compiledContent,
    hydrate: true // note that we want to hydrate this
component using the render.js hydrate() function. This is required for
when the user clicks on the author
  });
}
};

module.exports = ContentLayout;
```

内容布局编译器

内容布局编译器用于在发布指定内容类型的内容项时输出内容布局的 HTML。

编译过程中，`cec compile-content` 命令将在与组件的 `render.js` 文件相同的位置查找 `compile.js` 文件：

- `src`
 - `components`
 - * `<yourComponent>`
 - * `assets`
 - * `render.js`
 - * **`compile.js`**

如果此文件不存在，将不编译布局并将在运行时呈现布局。

如果该文件确实存在，它需要实施 `compile()` 接口，其返回 `Promise`。例如，下面是一个内容布局编译器，后面是生成的 `layout.html` 输出：

```
var fs = require('fs'),
    path = require('path'),
    mustache = require('mustache');

var ContentLayout = function (params) {
  this.contentClient = params.contentClient;
  this.contentItemData = params.contentItemData || {};
  this.scsData = params.scsData;
};

ContentLayout.prototype = {
  contentVersion: '>=1.0.0 <2.0.0',

  compile: function () {
    var compiledContent = '',
        content = JSON.parse(JSON.stringify(this.contentItemData)),
        contentClient = this.contentClient;
```

```

    // Store the id
    content.fields.author_id = content.id;

    if (this.scsData) {
        content.scsData = this.scsData;
    }

    try {
        // add in style - possibly add to head but inline for simplicity
        var templateStyle = fs.readFileSync(path.join(__dirname,
'design.css'), 'utf8');
        content.style = '<style>' + templateStyle + '</style>';

        var templateHtml = fs.readFileSync(path.join(__dirname,
'layout.html'), 'utf8');
        compiledContent = mustache.render(templateHtml, content);
    } catch (e) {
        console.error(e.stack);
    }

    return Promise.resolve({
        content: compiledContent,
        hydrate: true
    });
}
};

module.exports = ContentLayout;

```

生成的 layout.html 可以插入到 Web 应用程序中，也可以插入需要内容布局的静态 HTML 输出的任何其他地方。layout.html 与 compile.js 文件输出到相同目录。

```

{{{style}}}
{#{fields}}
<div class="author-container">
    <span class="author-name" onclick='{{{scsData.contentTriggerFunction}}}'
("starter-blog-post_author eq \"{{author_id}}\"')>{{{starter-blog-
author_name}}}</span>
{/{fields}}

```

SCSCompileAPI

与 SCSRenderAPI 类似，存在传入到每个编译函数的 SCSCompileAPI。

这包含以下属性和函数。

- 属性：
 - **navigationRoot**: 作为站点根的节点的 ID。
 - **navigationCurr**: 当前页节点的 ID。
 - **structureMap**: 站点层次的所有节点，按 ID 访问。
 - **siteInfo**: 所有站点属性。

- 函数：
 - **getContentClient**：获取要用于内容 API 调用的 contentClient 实例。
 - **getCustomSiteProperty**：在站点编译期间获取定制属性。
您可以在 siteinfo.json 文件中在站点级别存储定制数据和属性，以根据数据和属性定制站点。例如，您可以更改页背景纹理、页脚链接或网站名称。

组件水合

水合是指当 HTML 在浏览器中呈现时，将 JavaScript 行为添加回页中已编译 HTML 的过程。

例如，如果页上有两个要呈现为主项/详细信息的组件，则单击主项中的项需要更新详细信息组件。这都由在页中执行的 JavaScript 处理。要做到这一点，在这两个组件已呈现到页中后，您需要通过将 on click 事件处理程序添加到主项组件中的元素和详细信息组件容器上的监听程序，以针对这两个组件对 HTML 进行水合，从而在发生 on click 事件时根据事件中传递的有效负载重新呈现。

组件编译器将 HTML 插入到页中。如果您的组件需要在运行时执行额外的 JavaScript 来添加诸如事件处理程序之类的内容，则您有两个选项，内嵌 JavaScript 或 Hydrate 函数。您选择的解决方案取决于您的要求。

内嵌 JavaScript

可以将 <script> 标记直接插入到返回的已编译标记。脚本将在页执行时执行。

例如：

```
<script src="/_sitesclouddelivery/renderer/libs/scs-core/jssor-
slider/js/jssor.slider.min.js" type="text/javascript"></script>
<div id="slider_container_c46b122d-978a-429d-aa25-9b5698428f6f"
style="position: relative; top: 0px; left: 0px; height: 400px; width:
600px;
background-color: rgb(68, 68, 68); visibility: visible;" data-jssor-
slider="1">
. . .
</div>
<script>
    (function () {
        // get the required options
        var options =
        {"$FillMode":2,"$AutoPlay":false,"$AutoPlayInterval":3000,"$SlideDuratio
n":50
0,"$ArrowKeyNavigation":true,"$HWA":false,"$BulletNavigatorOptions":
{"$Chance
ToShow":1,"$AutoCenter":1,"$SpacingX":5},"$ArrowNavigatorOptions":
{"$ChanceTo
Show":1,"$AutoCenter":2,"$Steps":1},"$ThumbnailNavigatorOptions":
{"$ChanceToS
how":0,"$DisplayPieces":7,"$SpacingX":8,"$ParkingPosition":240}};

        // select the JSSOR value options
        options.$BulletNavigatorOptions.$Class
= $JssorBulletNavigator$;
        options.$ArrowNavigatorOptions.$Class
```

```

= $JssorArrowNavigator$;
    options.$ThumbnailNavigatorOptions.$Class =
$JssorThumbnailNavigator$;

    // create the slider
    var slider = new $JssorSlider$("slider_container_c46b122d-978a-
429d-aa25-9b5698428f6f", options);

    // resize, maintaining aspect ratio
var container = slider.$Elmt.parentElement;
if (container) {
    slider.$ScaleWidth(container.getBoundingClientRect().width);
}
})();
</script>

```

Hydrate 函数

可以在 `render.js` 文件中包括 `hydrate` 函数（而不是内嵌 JavaScript），并且请注意，当您返回编译的标记时，组件在运行时需要水合。此水合可避免重复 `<script>` 标记，允许您利用现有 JavaScript 代码来管理事件。

即使已加载 `render.js` 文件，在水合期间也不会调用 `render()` 函数。仅调用 `hydrate()` 函数。

注：

如果编译的组件没有说它需要水合，则永远不加载组件的 `render.js` 文件。

例如，定制内容布局编译器将返回 - `{ hydrate: true }`。

```

return Promise.resolve({
    content: compiledContent,

    hydrate: true // note that we want to hydrate this component using the
render.js hydrate() function. This is required for when the user clicks on
the author
});

```

此外，如果需要，定制编译器可以添加它在运行时将查找的 `hydrate` 属性。例如：

```

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}}">

```

最后，如果组件注意到它需要水合，则在运行时将加载组件的 `render.js` 文件，并调用 `hydrate()` 函数，传入包含已编译标记的容器 `<div>`。

例如，`render.js` - 请参见 `hydrate()` 函数：

```
function selectAuthor(parentObj, contentId, authorName) {
  var $parentObj = $(parentObj);
  $parentObj.find(".author-name").click($.proxy(function () {
    $(".author-name").removeClass('author-selected');
    $(event.target).addClass('author-selected');
  }, self));
  if (window.location.href.indexOf("default=" + contentId) >= 0) {
    $(".author-name").each(function () {
      if (this.innerText === authorName) {
        $(this).addClass('author-selected');
      }
    });
  }
}
. . .

hydrate: function (parentObj) {
  var $parentObj = $(parentObj),
      hydrateData = $parentObj.find('.author-
container').attr('data-
hydrate');
  if (hydrateData) {
    var data = JSON.parse(hydrateData);
    selectAuthor(parentObj, data.contentId, data.authorName);
  }
},
render: function (parentObj) {
  . . .
  try {
    // Mustache
    template = Mustache.render(templateHtml, content);

    if (template) {
      $(parentObj).append(template);
    }
    selectAuthor(parentObj, this.contentItemData.id,
content.fields['starter-blog-author_name']);
  } catch (e) {
    console.error(e.stack);
  }
}
}
```

发布

在已生成编译的静态页并将其上载到站点的静态文件夹后，需要发布或重新发布站点，以使这些页变为活动状态。类似地，要还原为非编译站点传送行为，需要在从站点删除静态文件后发布或重新发布。

在发布期间，上载的静态页变为可用于传送。由于这些文件在发布过程中进行复制，因此发布操作的性能可能会随着文件数量的增加而成比例降低。

发布操作获取当前静态文件集，并使其可用于传送。这些文件不一定与动态站点中发生的任何更改同步，也不一定镜像动态站点。在适当的时间更新静态文件集留给站点开发人员完成。

静态站点传送优先级

当站点具有关联的静态文件时，将传送这些文件以与传入到服务器的 URL 匹配。如果传入的 URL 与静态文件不匹配，则将针对请求返回该站点的 `controller.html` 文件。这遵循站点传送的现有动态模型。

Oracle Content Management 站点还可以通过关联的 JSON 文件定义 301 和 302 重定向。当已配置重定向时，重定向的优先级高于静态文件。如果 URL 同时与重定向规则和静态文件匹配，则重定向将从服务器传送。

站点传送的 URL 评估遵循以下流：

1. URL 是否与配置的重定向匹配？
如果匹配，将发出重定向响应。
2. 该 URL 是否对应于静态文件？
如果为站点配置了移动静态用户代理列表，并且请求来自与该列表匹配的浏览器，则会传送移动静态文件。
3. 否则，将传送动态站点 `controller.html` 文件。

注：

如果移动静态文件与站点相关联，并且客户使用 CDN 进行传送，则 CDN（通常是 Akamai）需要配置为分别对移动浏览器请求与标准桌面请求进行缓存。

如果 CDN 未配置单独的移动/标准高速缓存，则移动浏览器可能会收到标准响应，桌面浏览器可能会获得针对移动浏览器的响应。

高速缓存标头

Web 服务器响应中的 HTTP 标头有助于确定浏览器将如何缓存页。静态页还可以通过高速缓存标头进行传送，以帮助简化浏览器高速缓存。

对于安全站点，将随响应发送以下标头：

- Cache-Control: no-store
- Pragma: no-cache

对于标准的非安全站点，将发送以下标头：

- Cache-Control: max-age=300
 - Edge-Control: !no-store,max-age=2592000,downstream-ttl=1800
- Edge-Control 标头有助于简化 CDN 高速缓存行为

如果您已在这两个区域之一中定制标头，则响应将具有定制标头，而非此处列出的标准标头。

您可以在租户级别或站点级别控制这些响应。

详细信息页

Oracle Content Management 站点中的详细信息页允许单个页面显示多个内容项的信息。

例如，可以使用同一详细信息页处理多个 URL。其中的每个 URL 都将显示相同的页结构，但是将分别显示与缩略名值为 `item1.html`、`item2.html` 和 `item3.html` 的内容项相关的内容。对于这种情况，cec 模板编译器可能会创建四个文件：

- `/detail/item1.html`
- `/detail/item2.html`
- `/detail/item3.html`
- `/detail.html`

最终文件允许新发布的材料显示在网站上，而不必重新编译和重新发布站点。在此示例中，缩略名值为 `item4.html` 的内容项在站点联机后发布。静态 `/detail.html` 页允许该新项在站点中动态显示。URL `/detail/item4.html` 将传送 `detail.html` 页，但是显示与 `item4.html` 内容项相关的内容。

cec 编译器生成 `detail.html` 页以显示内容项。因此，已编译 `detail.html` 页内的相对 URL 将具有额外的父段 (`../`)。因此，如果直接引用，`detail.html` 页本身将无法正确显示。因此，不应引用 `detail.html` 页本身或将其添加到页导航。

将内容项添加到通道

可以使用 OCE 工具包 `control-content` 命令将内容项添加到 Oracle Content Management 服务器中的通道。

`control-content <action>` 命令具有操作 `add`，用于将内容项添加到 Oracle Content Management 通道：

```
cec control-content add -c Channel1 -r Repol -s UAT
```

此命令将资料档案库 `Repol` 中的所有项添加到已注册服务器 `UAT` 上的通道 `Channel1`。

可以使用 `-s <server>` 指定服务器，也可以使用 `cec.properties` 文件中指定的服务器。

`content-usage` 命令的有效操作如下：

- `publish`
- `unpublish`
- `add`
- `remove`

`content-usage` 命令的选项如下：

- `--channel, -c Channel` [必需]
- `--repository, -r Repository` [`<action>` 为 `add` 时，此项为必需]
- `--server, -s` 注册的 Oracle Content Management 服务器
- `--help, -h` 显示帮助 [布尔值]

control-content 命令的示例如下：

```
cec control-content publish -c Channel1
```

Publish all items in channel Channel1 on the server specified in the cec.properties file

```
cec control-content publish -c Channel1 -s UAT
```

Publish all items in channel Channel1 on the registered server UAT

```
cec control-content unpublish -c Channel1 -s UAT
```

Unpublish all items in channel Channel1 on the registered server UAT

```
cec control-content add -c Channel1 -r Repo1 -s UAT
```

Add all items in repository Repo1 to channel Channel1 on the registered server UAT.

```
cec control-content remove -c Channel1 -s UAT
```

Remove all items in channel Channel1 on the registered server UAT

为移动设备编译站点

可以使用 OCE 工具包编译站点 Web 页的移动布局。相同内容的移动布局可能与桌面页布局不同。或者移动布局和桌面布局可能相同。

在站点编辑器中，可以为移动设备选择与桌面布局相同的页布局，也可以指定不同的页布局。使用 OCE 工具包，可以为移动设备单独编译静态布局。

您可以在移动设备上以不同方式查看站点页。移动设备上呈现的页可能不会像页在桌面布局中那样具有横幅。

在 OCE 工具包中，cec compile-template 的帮助页显示 targetDevice 选项，用于在您编译站点模板时定位特定设备：

```
C:\git\webclient\developer\test\sites-compiler\cec-install>cec compile-template --help
Usage: cec compile-template <source>

Compiles all the pages within the site of the template and places the compiled pages under the sites assets folder.
Optionally specify -s <server> to make content queries against this server (requires channelToken).
Optionally specify -c <channelTokens> to use this channelToken when generating any content URLs.
Optionally specify -t <contentType> [draft | published] content to retrieve from the server type, defaults to published.
Optionally specify -p <pages> the set of pages to compile.
Optionally specify -d <debug> to start the compilation with --inspect-brk flag.
Optionally specify -r <recurse> recurse through all child pages of specified pages.
Optionally specify -l <includeLocale> include default locale when creating pages.
Optionally specify -a <targetDevice> [desktop | mobile] target device type when using adaptive layouts.
Optionally specify -v <verbose> to display all warning messages during compilation.

Options:
--server, -s           The registered CEC server
--channelToken, -c    The channel access token to use for content URLs
--type, -t            The type of content to retrieve from the serve [published | draft]
--pages, -p          The list of pages to compile
--recurse, -r        Compile all child pages of those specified in the page list
--debug, -d          Start the compiler with "--inspect-brk" option to debug compilation
--noDetailPages, -e   Do not generate compiled detail pages
--noDefaultDetailPageLink, -o Do not generate compiled detail page for items/content lists that use the default detail page
--targetDevice, -a    The target device type when using adaptive layouts [desktop | mobile]
--includeLocale, -l  Include default locale when creating pages
--verbose, -v        Run in verbose mode to display all warning messages during compilation.
--help, -h          Show help

Examples:
cec compile-template Templ                               Compiles the site in template Templ using content stored in the template.
cec compile-template Templ -c channelToken              Compiles the site in template Templ using the given channelToken for any content URLs.
cec compile-template Templ -c channelToken -s UAT -t draft Compiles the site in template Templ retrieving draft content from the specified server.
cec compile-template Templ -p 104,112,183 -r            Compiles the specified pages in the site in template Templ including all child pages.
cec compile-template Templ -d                          Waits for the debugger to be attached. Once attached, compiles the site in template Templ.

C:\git\webclient\developer\test\sites-compiler\cec-install>
```

编译站点时，可以指定要为 desktop 还是 mobile 进行编译。桌面文件放置在 static/_files 下。移动文件放置在 static/_mobilefiles 下。

```
C:\git\webclient\developer\test\sites-compiler\cec-install>cec compile-template Corporate-Site-Template --targetDevice desktop
Compile Template: compiling template Corporate-Site-Template
Oracle Content and Experience Site Compiler

Compiling: desktop pages
-----
createPage: Processing pageId 10. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/index.html
createPage: Processing pageId 100. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/developing-templates.html
createPage: Processing pageId 110. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/developing-templates/themes.html
createPage: Processing pageId 120. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/developing-templates/pages.html
createPage: Processing pageId 130. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/developing-templates/navigation.html
createPage: Processing pageId 140. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/developing-templates/page-content.html
createPage: Processing pageId 150. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/developing-templates/components.html
createPage: Processing pageId 160. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/developing-templates/sign-in.html
createPage: Processing pageId 200. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/privacy-policy.html
All page creation calls complete.

Compilation completed with 0 errors and 5 warnings.
to display warnings, run with --verbose (-v) option.

*** compiled template is ready to test
*** to render non-compiled pages, remove compiled files from under: C:\git\webclient\developer\test\sites-compiler\cec-install\src\templates\Corporate-Site-Template\static

C:\git\webclient\developer\test\sites-compiler\cec-install>cec compile-template Corporate-Site-Template --targetDevice mobile
Compile Template: compiling template Corporate-Site-Template
Oracle Content and Experience Site Compiler

Compiling: mobile pages
-----
createPage: Processing pageId 10. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/index.html
createPage: Processing pageId 100. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/developing-templates.html
createPage: Processing pageId 110. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/developing-templates/themes.html
createPage: Processing pageId 120. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/developing-templates/pages.html
createPage: Processing pageId 130. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/developing-templates/navigation.html
createPage: Processing pageId 140. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/developing-templates/page-content.html
createPage: Processing pageId 150. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/developing-templates/components.html
createPage: Processing pageId 160. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/developing-templates/sign-in.html
createPage: Processing pageId 200. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/privacy-policy.html
All page creation calls complete.

Compilation completed with 0 errors and 6 warnings.
to display warnings, run with --verbose (-v) option.

*** compiled template is ready to test
*** to render non-compiled pages, remove compiled files from under: C:\git\webclient\developer\test\sites-compiler\cec-install\src\templates\Corporate-Site-Template\static
```

您为移动设备编译模板后，OCE 工具包命令 upload-static-site-files 将支持移动文件。

站点生命周期和已编译页

在从站点创建模板时，模板中不包括已编译页。这是为了避免出现在站点开发人员预期传送动态页时传送了静态页的问题。

如果后续从模板创建站点，则需要编译站点页并将其上载到新站点。

在 Oracle Content Management 服务器中创建新站点或资产翻译作业

使用 OCE 工具包为 Oracle Content Management 中的站点或资产创建翻译作业。

对多语言站点编制索引之前，您需要翻译作业。要创建翻译作业，请执行以下操作：

1. 单击“站点”页顶部菜单上的翻译。
2. 在创建翻译作业对话框中为作业输入名称，并选择默认源语言、目标语言和翻译作业内容。

您可以选择使翻译包包括所有站点内容和目标资产、仅站点内容或仅定向到站点发布通道的资产。

从翻译中排除配置了不翻译文本设置的任何内容项。例如，产品名称通常不翻译。

3. 单击创建以创建翻译作业。
4. 使用 OCE 工具包命令列出可用作业：

```
cec components> cec list-translation-jobs
Asset translation jobs:
Name                               Status      Source
Language Target Languages                Pending Languages

Site translation jobs:
Name                               Status      Source
Language Target Languages                Pending Languages

demol                               READY      en-
US                                 fr-FR,es-ES

searchdemol                         TRANSLATED en-
US                                 fr-FR,es-ES
```

5. 下载翻译作业：

```
cec components> cec download-translation-job demol
- translation job downloaded to /Users/<user-name>/Dev/webclient/
  developers/sites-toolkit/cec-components/demo.zip
- update the translation job status to INPROGRESS.
cec components> cec translate dmol.zip -l all -t demol-xlate.zip
- target languages: fr-FR,ex-ES
- translation finished: /Users/<user-name>/Dev/webclient/developers/
  sites-toolkit/cec-components/demol-xlate.zip
```

6. 打开翻译包并针对要翻译到的语言构建资源的文件夹：

```
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
inflating

```

7. 导入翻译作业:

```

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

```

通过语言服务提供商翻译站点

您可以通过 OCE 工具包命令行界面和语言服务提供商 (Language Service Provider, LSP) 管理多语言站点的翻译。

站点的本地化策略指定默认语言，如英语（美国）(en-US)，以及站点的一种或多种替代语言，如德语和法语。站点的文本字符串可以翻译为指定的替代语言。如果在翻译之前更改站点的语言，则文本字符串仍然会以默认语言显示。

OCE 工具包在命令行界面中提供以下翻译选项：

Translation	
cec list-translation-jobs	Lists translation
jobs.	[alias: ltj]
cec create-translation-job <name>	Creates a translation
job <name> for a site on CEC server.	[alias: ctj]
cec download-translation-job <name>	Downloads translation
job <name> from CEC server.	[alias: dtj]
cec submit-translation-job <name>	Submits translation job
<name> to translation connection <connection>.	[alias: stj]
cec ingest-translation-job <name>	Gets translated job
<name> from translation connection and ingest.	[alias: itj]
cec upload-translation-job <name>	Uploads translation job
<name> to CEC server.	[alias: utj]
cec create-translation-connector <name>	Creates translation
connector <name>.	[alias: ctc]
cec start-translation-connector <name>	Starts translation
connector <name>.	[alias: stc]
cec register-translation-connector <name>	Registers a translation
connector.	[alias: rtc]

可以使用 `cec list-translation-jobs` 命令列出服务器上已经存在的翻译作业。例如：

```
cec ltj -s
Server: <server-name>
Asset translation jobs:
Name                               Status           Source Language
Target Languages                   Pending Languages
testHash                           INPROGRESS      en-US
fr-FR,de-DE                        fr-FR,de-DE
Site translation jobs:
Name                               Status           Source Language
Target Languages                   Pending Languages
demoTest                           TRANSLATED      en-US
de-DE,fr-FR
```

键入任何不带参数或者带 `-h` 的 `cec` 命令将提供关于命令的帮助。请参见[使用 cec 命令行实用程序](#)。

以下部分提供了有关通过 LSP 翻译站点的信息：

1. [使用 OCE 工具包创建翻译作业](#)
2. [列出翻译作业](#)
3. [创建翻译连接器](#)
4. [为多语言站点生成站点地图](#)
5. [将翻译作业提交给语言服务提供商](#)
6. [将翻译作业上传到服务器](#)

使用 OCE 工具包创建翻译作业

您可以使用 OCE 工具包命令在本地系统上创建站点翻译作业。

要为站点创建新翻译作业，请使用 `cec create-translation-job` 命令。此命令可查找站点的所有资产，并为需要从该站点翻译的所有内容创建 zip 文件。

```
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
```

有关翻译选项，请参见在 [Oracle Content Management 服务器中创建新站点或资产翻译作业](#)。

列出翻译作业

您可以在服务器上列出翻译作业，以验证您的作业是否已创建并可以使用。

```
cec list-translation-jobs -s
Server: <server-name>
Asset translation jobs:
Name                               Status           Source Language Target
Languages                           Pending Languages
testHash                             INPROGRESS      en-US           fr-
FR,de-DE                             fr-FR,de-DE
Site translation jobs:
Name                               Status           Source Language Target
Languages                           Pending Languages
demoTest                             TRANSLATED      en-US           de-
DE,fr-FR
FridayDemo                           READY           en-US           de-
DE,fr-FR                             de-DE,fr-FR
```

请注意，FridayDemo 作业处于 READY 状态。

创建翻译连接器

语言服务提供商 (language service provider, LSP) 可帮助您翻译站点。使用连接至 LSP 的翻译连接器，可以提交并摄取翻译作业。

提交翻译作业前，需要创建翻译连接器。要在不使用 LSP 的情况下翻译站点，可以创建要针对其运行的模拟翻译连接器。使用 `cec create-translation-connector` 命令创建翻译连接器，并使用 `cec start-translation-connector` 命令启动该连接器：

```
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
```

使用 OCE 工具包通过预期的 API 运行翻译连接器以对其进行测试：

1. 使用 OCE 工具包注册连接器。

```
>cec register-translation-connector
```

2. 打开工具包并转到“翻译连接”页。

```
>http://localhost:8085/public/translationconnections.html
```

3. 运行翻译连接器验证页上的所有步骤。这些步骤使用连接器环境中 /data 文件夹中的 translationBundle.zip 文件验证连接器。

可以使用翻译连接器 SDK 为 Oracle Content Management 开发翻译连接器。此 SDK 是翻译连接器 API 的示例 NodeJS 实施。该示例接受 Oracle Content Management 翻译作业 zip 文件，翻译文件中的所有资源，然后返回包含所有翻译的新 zip 文件。

SDK 要求用户能够访问 LSP 来进行实际的字符串翻译。SDK 中包含模拟 LSP 服务器，用于通过仅将目标区域设置预置到字符串来模拟来自 LSP 的响应。

翻译连接器 SDK 包含三个主要模块。

- **连接器：**实施所需 Oracle Content Management 翻译连接器 API 的翻译连接器。
- **作业管理器：**基于文件系统的示例作业管理器，在语言服务提供商翻译连接器作业时维护作业的状态。
- **提供商：**您的 LSP 提交待翻译文档和检索已翻译文档所需的特定 API 集的实施。

您可以复制模拟翻译提供商 JS 并实施其中的所有方法。

为多语言站点生成站点地图

使用 OCE 工具包为多语言站点生成站点地图，并将该地图发布到该站点。

您可以使用 `cec create-site map <site>` 命令为 Oracle Content Management 服务器上的多语言站点创建站点地图。例如：

```
cec create-site-map Site1 -u http://www.example.com/site1
```

此命令遍历站点结构，生成与站点页层次匹配的站点地图层次，并在 Oracle Content Management 服务器上的指定站点 URL 创建站点地图。

命令选项如下所示：

```
--url, -u          <url> Site
URL                                                         [required]
--changefreq, -c  How frequently the page is likely to change
--file, -f        Name of the generated site map file
--publish, -p     Upload the site map to CEC server after creation
--help, -h       Show
help                                                     [boolean]
```

<changefreq> 选项的有效值如下所示：

- always
- hourly
- daily
- weekly
- monthly
- yearly
- never
- auto

cec create-site-map 命令的示例如下所示：

```
cec create-site-map Site1 -u http://www.example.com/site1
cec create-site-map Site1 -u http://www.example.com/site1 -f sitemap.xml
cec create-site-map Site1 -u http://www.example.com/site1 -p
cec create-site-map Site1 -u http://www.example.com/site1 -c weekly -p
```

要发布站点地图，请创建站点更新，更新站点地图，然后提交更新。

将翻译作业提交给语言服务提供商

OCE 工具包提供可发送到语言服务提供商的 zip 文件，用于开始进行翻译作业。

您可以通过翻译连接器将翻译作业提交给 LSP。提交会需要一段时间，因为连接器需要解压缩文件并将所有单个文件提交给 LSP。然后，LSP 可以为您的翻译作业创建项目。文件导入到项目中后，即可开始选择文件进行翻译。然后，LSP 开始监视翻译的状态。

要检查状态，请使用不带任何选项的 `cec list-translation-jobs` 命令在本地列出翻译作业。当作业的状态为 `READY TO INGEST` 时，您可以从 LSP 下载 zip 文件以摄取翻译作业。翻译连接器已将您的 zip 文件提交给 LSP，LSP 已翻译所列的文件，连接器已从 LSP 检索回您可以下载并摄取的 zip 文件中的这些文件。

```
cec list-translation-jobs
Local translation jobs:
Name                               Status           Source Language
Target Languages
FridayDemo                          READY TO INGEST en-US
de-DE,fr-FR
demoTest                             READY TO INGEST en-US
de-DE,fr-FR
```

摄取 zip 文件可以将翻译作业从连接器拉回到 OCE 工具包中。

```
cec ingest-translation-job FridayDemo
- use connection <lsp name>
- query translation connection to get job status
- get translation
- translation saved to <sites-toolkit folder>/cec-components/dist/
FridayDemo-translated.zip
- validate translation file
- translation job ingested to <sites-toolkit folder>/cec-components/src/
main/translationJobs/FridayDemo
```

摄取 zip 文件后，在本地列出翻译作业时，翻译作业的状态为 `TRANSLATED`。

```
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
```

您可以将已翻译作业上载到 Oracle Content Management 服务器。通常，作业将经过初步的快速翻译，然后发送回以供您审核。站点的翻译可能需要几周时间才能完成，其中包括摄取 LSP 返回的翻译作业、对翻译进行更正和重新提交翻译作业。

将翻译作业上载到服务器

摄取翻译作业后，可以将其上载到 Oracle Content Management 服务器，然后在您的站点上检查翻译。

使用 `cec upload-translation-job` 命令将翻译 zip 文件上载到服务器。

```
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
```

上载翻译作业后，服务器上作业的状态为 INPROGRESS：

```
cec list-translation-jobs -s
Server: <server-name>
Asset translation jobs:
Name                               Status      Source Language
Target Languages                    Pending Languages
testHash                            INPROGRESS    en-US
fr-FR,de-DE                         fr-FR,de-DE
Site translation jobs:
Name                               Status      Source Language
Target Languages                    Pending Languages
demoTest                            TRANSLATED    en-US
de-DE,fr-FR                         de-DE,fr-FR
FridayDemo                          INPROGRESS    en-US
de-DE,fr-FR                         de-DE,fr-FR
```

要验证翻译，可以检查所翻译站点上资产中的文本字符串。

第 VI 部分

附录

提供了以下附录：

- [教程：使用 Knockout 开发组件](#)
- [Sites SDK 参考](#)
- [故障排除](#)

教程：使用 Knockout 开发组件

本教程将指导您使用一组 JavaScript 对象（这些对象利用标准 Knockout ViewModel 和模板功能）来创建存储在 Oracle Content Management 组件目录中的组件。

- [使用 Knockout 进行组件开发的简介和先决条件](#)
- [步骤 1: 创建组件](#)
- [步骤 2: 查看本地组件呈现的结构](#)
- [步骤 3: 查看本地组件设置的结构](#)
- [步骤 4: 显示组件中的新属性](#)
- [步骤 5: 注册触发器](#)
- [步骤 6: 引发触发器](#)
- [步骤 7: 注册操作](#)
- [步骤 8: 执行操作](#)
- [步骤 9: 为组件的每个实例创建不同的标题](#)
- [步骤 10: 结合使用嵌套组件和内嵌编辑](#)
- [步骤 11: 支持不同的布局](#)
- [步骤 12: 定义定制样式](#)
- [步骤 13: 在内嵌框架中呈现组件](#)
- [步骤 14: 在内嵌框架中呈现组件时使用定制样式](#)
- [步骤 15: 与页撤消和重做行为集成](#)
- [步骤 16: 资产管理](#)
- [教程回顾](#)

使用 Knockout 进行组件开发的简介和先决条件

本教程提供步骤和验证过程以使用 JavaScript 对象创建示例组件，这些对象利用标准 Knockout JS ViewModel 和模板功能。

您应该能够采用这些步骤中引用的代码（在您创建组件时植入的文件中提供），并使用您自己的代码仅更新 .html 模板和 JavaScript viewModel。

注：

虽然 Oracle Content Management 不指示您用于创建组件的 JavaScript 技术，但是，无论选择哪个 JavaScript 框架，通常工厂 JavaScript 函数对于组件的每次实施都是相同的。

先决条件

本教程仅着重讲述组件的实施。有关组件的更多一般信息，请参见“开发组件”。

要完成本教程中的步骤，必须满足以下要求：

- 您必须有权访问 Oracle Content Management 实例并具有创建站点和组件的权限。
- 已使用 Oracle Content Management 桌面或定制组件将 Oracle Content Management 实例服务器同步到您的本地计算机。请参见“通过开发人员云服务开发定制组件”。

此外，您还应该熟悉以下 JavaScript 概念和框架：

- JavaScript 浏览器调试
- JavaScript 闭包
- JavaScript 异步模块定义 (Asynchronous Module Definition, AMD) 开发
- RequireJS 和 KnockoutJS 框架

继续执行[步骤 1: 创建组件](#)。

步骤 1: 创建组件

此步骤说明如何在 Oracle Content Management 中创建定制组件。

创建定制组件时，必须注册该组件以便可由 Oracle Content Management 使用。要告知 Oracle Content Management 有关您组件的信息，请使用站点构建器中的“组件”页注册该组件。

有两种类型的组件要注册。

- **本地组件：**
 - 此组件的文件存储在 Oracle Content Management 实例服务器上。
 - 主要优点是您不必担心跨域或跨协议问题，因为文件位于您的站点中。
 - 缺点是您无法在 Oracle Content Management 服务器中执行任何中间层逻辑，因此您必须对支持 CORS 的远程服务器使用 REST API。
 - 此类型的组件可以直接嵌入到页中，或者您可以选择使用内嵌框架在页上呈现该组件。
- **远程组件：**
 - 该组件中的文件存储在远程服务器上，您只需将 URL 注册到该组件的“呈现和设置”面板。
 - 为组件创建内容时，如果您具有必须执行的服务器端逻辑，则远程组件会具有优势。
 - 缺点是您必须确保解决所有跨域和安全问题才能访问这些 URL。
 - 远程组件始终使用内嵌框架在页上进行呈现。

要创建和注册本地组件，请执行以下操作：


1. 在 Oracle Content Management 主页上，单击开发人员。
此时将显示开发人员页。

2. 单击查看所有组件。
3. 从菜单中，选择创建本地组件。
4. 为组件输入名称；例如 **A_Local_Component**。
5. 输入可选说明。
6. 单击创建。


完成此操作后，您将在组件列表中看到名为 `A_Local_Component` 的组件。

检查步骤 1 的结果

现在，您已成功创建组件，在您创建的任何站点的组件选项板中应该可看到该组件。使用以下步骤验证组件创建情况：

1. 创建名为 `localComponentTest` 的站点。
2. 选择站点并单击打开。
3. 单击编辑。
4. 为站点创建更新并为其提供名称和（可选）说明。
5. 选择站点上的页。
6. 单击侧选项板中的 ，并选择定制以显示定制组件列表。
7. 从定制组件列表中选择 `A_Local_Component`，并将其拖放到页上。

现在应该会看到创建的本地组件的默认呈现效果。

8. 选择刚放到页上的组件横幅中的 .
9. 选择设置。
10. 更改对齐方式并设置组件的样式。
11. 关闭“设置”面板。

以下步骤说明如何构建定制组件，以及如何针对您自己的目的对其进行修改。继续执行 [步骤 2: 查看本地组件的结构](#)。

步骤 2: 查看本地组件呈现的结构

在此步骤中，我们查看为本地组件创建的默认文件的结构。

对于简单的 `Hello World` 示例，四个 JavaScript 对象和代码行数可能看起来过多，但这为您提供构建更复杂组件以及处理与 Oracle Sites Cloud Service 页生命周期交互的基础。

要查看本地组件的结构，请执行以下操作：

1. 在 Oracle Content Management 主页上，单击开发人员。
此时将显示开发人员页。
2. 单击查看所有组件。
3. 从菜单中，选择创建本地组件。
4. 为组件输入名称；例如 **A_Local_Component**。
5. 输入可选说明。

6. 单击创建。

完成此操作后，您将在组件列表中看到名为 `A_Local_Component` 的组件。

1. 使用 Oracle Content Management 桌面同步客户端，找到组件并将其与文件系统同步。

如果您没有桌面客户端，您可以在 Oracle Content Management 界面的“组件”页中查看所有组件并选择组件，然后下钻以查看文件。

2. 如果列出组件下的文件，将看到以下文件：

```
assets
  render.js
  settings.html
appinfo.json
_folder_icon.jpg
```

3. 打开 `/assets` 目录下的 `render.js` 文件。

`render.js` 文件的要点为：

- 它结构化为 JavaScript AMD 模块，因此它可以“必需”进入到页中。
- 它还包括对已经作为 Oracle Content Management 页一部分加载的 KnockoutJS 和 JQuery 的引用。

考虑 `render.js` 文件的结构。

在 `render.js` 文件的内容中，存在两个实施所需 Oracle Content Management 组件 API 的 JavaScript 对象：`sampleComponentFactory` 和 `SampleComponentImpl`。这些对象是用于创建任何基于 KnockoutJS 的组件实施的示例。这些对象的实施将根据所使用的技术而改变。

- `sampleComponentFactory`
 - 此对象由 `render.js` AMD 模块返回。
 - 这是非常简单的工厂对象，可实施单个 `createComponent()` 接口。
 - 更复杂的实施可以使用传递的 `args` 值来根据 `viewMode` 参数返回组件的不同实施。您可以通过这种方式为运行时与站点构建器获得显著的更轻量级组件实施。
- `SampleComponentImpl`
 - 此对象中的主要函数为 `render` 函数，其用于将组件呈现到页上。
要将 Knockout 组件呈现到页中，`render` 函数会将模板动态地添加到页中，然后将 `viewModel` 绑定应用于模板。
 - 其余的实施处理 `viewModel` 参数和模板的初始化，以及处理页和组件之间的消息传送。

`render.js` 文件中的最后两个对象（即 `sampleComponentTemplate` 和 `SampleComponentViewModel`）提供组件的定制实施。它们的实施将根据您的要求而有所不同。

- `sampleComponentTemplate`
 - 此对象提供 KnockoutJS 模板创建。在尝试显示任何内容之前，它将等待组件初始化所有数据。
- `SampleComponentViewModel`

- viewModel 检索由 Oracle Content Management 代表组件存储的信息，然后选择如何根据该数据适当地布局组件
- 模板使用的一般 Knockout 可观察项（用于处理对代表组件存储的元数据的访问）：

```
self.imageWidth = ko.observable('200px');
self.alignImage = ko.observable();
self.layout = ko.observable();
self.showTopLayout = ko.observable();
self.showStoryLayout = ko.observable();
```

- 触发器和操作集成：

触发器：用于从组件引发 Oracle Content Management 触发器的函数，该函数可以绑定到页上其他组件的操作。

```
self.imageClicked = function (data, event) {
    self.raiseTrigger("imageClicked"); // matches appinfo.json
};
```

操作：当组件被告知使用给定的有效负载执行操作时用于处理回调的函数。

```
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);
            }
        });
    }
};
```

用于按需执行任何已注册操作的回调。

```
SitesSDK.subscribe(SitesSDK.MESSAGE_TYPES.EXECUTE_ACTION,
$.proxy(self.executeActionsListener, self));
```

- 组件生命周期的订阅：

- * **组件初始化：**确保在提取所有数据之前不呈现组件。这通过 Knockout 可观察项进行处理。

```
self.componentLayoutInitialized = ko.observable(false);
self.customSettingsDataInitialized = ko.observable(false);
```

获取任何所需属性的初始值。这通过回调进行处理以检索数据。

```
SitesSDK.getProperty('componentLayout',
self.updateComponentLayout);
```

```
SitesSDK.getProperty('customSettingsData',  
self.updateCustomSettingsData);
```

- * 元数据更新: 每当代表组件存储的组件元数据更改时进行回调; 例如, 当用户调用“设置”面板并更新数据时。

```
SitesSDK.subscribe(SitesSDK.MESSAGE_TYPES.SETTINGS_UPDATED,  
$.proxy(self.updateSettings, self));
```

注:

由于 Oracle Content Management 服务器始终设置 mime 类型 .html 文件, 您无法上传 .html 文件, 而是使用所需的 "text!" 插件才能加载它。因此, 对于模板, 您需要使用不同的扩展来通过 "text!" 插件加载它, 或者直接在 JavaScript 中以内嵌方式加载它, 如植入数据中所示。

检查步骤 2 的结果

现在, 您应该大致了解了如何创建定制组件呈现器的结构。要验证它是否有效, 请执行以下操作:

1. 更新 `render.js` 文件中的 `sampleComponentTemplate` 对象以更改以下行。更改以下代码:

```
'<!-- ko if: initialized -->'+
```

改用以下代码:

```
'<!-- ko if: initialized -->'+  
'<div data-bind="text:\'image width is: \' + imageWidth()"></div>' +
```

2. 将组件同步或上载到 Oracle Content Management 实例服务器。
3. 编辑站点中的页并将 `A_Local_Component` 定制组件放在页上。

此时, 您应该看到组件中的 `image width is: 260px`。

4. 打开“设置”面板并单击定制设置按钮。
5. 将图像宽度字段更改为 `300px`。
6. 此时, 组件中将发生两件事:
 - a. 默认图像的大小将从 `260px` 扩展到 `300px`。
 - b. 您添加的文本将更新为 `image width is 300px`。

继续执行 [步骤 3: 查看本地组件设置的结构](#)。

步骤 3: 查看本地组件设置的结构

在此步骤中, 我们查看为本地组件指定的设置的结构。

与 `/assets` 目录中的 `render.js` 文件类似, 同一目录中存在预先创建的 `settings.html` 文件。`settings.html` 文件呈现组件的任何定制设置数据。在默认实施中, 定制设置数据中有单个属性 `imageWidth`。

要查看本地组件的结构, 请执行以下操作:

1. 使用 Oracle Content Management 桌面同步客户端, 找到组件并将其与文件系统同步。
如果您没有桌面同步客户端, 您可以在 Oracle Content Management Web 界面的组件选项卡上选择组件, 然后下钻以查看文件。
2. 如果列出组件下的文件, 将看到以下文件:

```
assets
  render.js
  settings.html
appinfo.json
_folder_icon.jpg
```

打开 `/assets` 目录下的 `settings.html` 文件, 并查看内容。与 `render.js` 文件不同, `settings.html` 文件使用站点构建器的“设置”面板中的内嵌框架, 这就是为什么它还需要访问支持文件才能在内联框架内正确呈现。需要站点构建器来管理站点, 以便 JavaScript 代码中的任何错误都可以与站点构建器隔离, 这就是为什么 `settings.html` 文件使用内嵌框架。

以下是 `settings.html` 文件的主要区域:

- 用于呈现“设置”面板的 Knockout 模板。

```
<!-- ko if: initialized() -->
<div class="scs-component-settings">
  <div>
    <!-- Width -->
    <label id="widthLabel" for="width" class="settings-heading" data-
bind="text: 'Image Width'"></label>
    <input id="width" data-bind="value: width" placeholder="example:
200px or 33%" class="settings-text-box">
  </div>
</div>
<div data-bind="setSettingsHeight: true"></div>
<!-- /ko -->
```

- 定制绑定处理程序, 用于在呈现“设置”面板后调整内嵌框架的高度。

```
ko.bindingHandlers.scsCompComponentImpl
```

- 应用于 Knockout 模板的 Knockout ViewModel。

```
SettingsViewModel
```

以下是 `SettingsViewModel` 的主要元素:

- 组件生命周期的订阅。
- 组件初始化:

- 确保在提取所有数据之后才呈现组件。这通过 Knockout 可观察项进行处理。

```
self.initialized = ko.observable(false);
```

- 确保在就绪之前不尝试更新数据。

```
self.saveData = false;
```

- 获取任何所需属性的初始值。这通过回调进行处理以检索数据。

```
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;
});
```

- 将任何属性更改保存到定制设置数据中。

```
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);
```

要添加要捕获的另一个属性，需要执行几个步骤：

1. 更新用户界面以显示新值。
2. 将值初始化为针对组件存储的当前值。
3. 将对值的任何更改保存回组件。

要向定制组件添加另一个属性，请对 settings.html 文件进行以下更改：

1. 添加另一个可观察项以处理新属性。更改以下代码：

```
self.width = ko.observable();
```

改用以下代码：

```
self.width = ko.observable();
self.imageBannerText = ko.observable();
```

- 首次显示设置面板时，获取新属性的任何当前值。更改以下代码：

```
self.width(data.width);
```

改用以下代码：

```
self.width(data.width);  
self.imageBannerText(data.imageBannerText);
```

- 保存对此新属性所做的任何更改。更改以下代码：

```
'width': isNaN(self.width()) ? self.width() : self.width() + 'px'
```

改用以下代码：

```
'width': isNaN(self.width()) ? self.width() : self.width() + 'px',  
'imageBannerText': self.imageBannerText()
```

- 添加用户界面以显示新字段。更改以下代码：

```
<label id="widthLabel" for="width" class="settings-heading" data-  
bind="text: 'Image Width'"></label>  
<input id="width" data-bind="value: width" placeholder="example: 200px or  
33%" class="settings-text-box">
```

改用以下代码：

```
<label id="widthLabel" for="width" class="settings-heading" data-  
bind="text: 'Image Width'"></label>  
<input id="width" data-bind="value: width" placeholder="example: 200px or  
33%" class="settings-text-box">  
  
<label id="imageBannerTextLabel" for="imageBannerText" class="settings-  
heading" data-bind="text: 'Image Banner'"></label>  
<input id="imageBannerText" data-bind="value: imageBannerText"  
placeholder="Text to display above an image" class="settings-text-box">
```

- 同步或上载 settings.html 文件。

如果要立即运行此项，则将显示该字段。但是，“设置”面板的大小不会自动更改。由于您增大了面板的大小，因此还必须将 components.json 注册条目更新为新大小。

- 下载 appinfo.json 文件（它与组件的 assets/ 目录处于同一级别），然后更新设置面板的大小。更改以下代码：

```
"settingsHeight": 90,
```

改用以下代码：

```
"settingsHeight": 160,
```

- 同步或上载 appinfo.json 文件。

检查步骤 3 的结果

现在, 您应该能够看到并输入添加到“设置”面板的新属性。

1. 刷新站点中的页, 以便站点构建器能够获取对组件的更改。
2. 让页进入编辑模式。
3. 将组件拖放到页上。
4. 针对组件打开“设置”面板。
5. 单击定制设置按钮。

您将看到为 `settings.html` 文件中的每个属性显示的两个字段。

继续执行 [步骤 4: 显示组件中的新属性](#)。

步骤 4: 显示组件中的新属性

在本节的末尾, 您将能够在“设置”面板中为新属性输入值, 并看到可反映新值的定制组件更改。对该属性的更新也会随页自动保存。

在 `render.js` 文件中, 必须更新组件中的两个 JavaScript 对象:

- `SampleComponentViewModel`
- `sampleComponentTemplate`

编辑 `render.js` 并更新 `SampleComponentViewModel` 组件, 以包含新属性。更改以下属性:

```
self.showStoryLayout = ko.observable();
```

改用以下内容:

```
self.showStoryLayout = ko.observable();  
self.imageBannerText = ko.observable();
```

更新 `SampleComponentViewModel` 以获得对值的任何更改。更改以下属性:

```
self.imageWidth(customData && customData.width);
```

改用以下内容:

```
self.imageWidth(customData && customData.width);  
self.imageBannerText(customData && customData.imageBannerText);
```

更改 `sampleComponentTemplate` 以显示新属性。更改以下属性:

```
'<div data-bind="text: \'image width is: \' + imageWidth()'></div>' +
```

改用以下内容:

```
'<div data-bind="text: imageBannerText"></div>' +
```

将组件同步或上传到 Oracle Content Management 服务器。

现在已变更组件以显示新属性。与嵌入在页的内嵌框架中的“设置”面板不同，因为组件直接插入到页中，所以随着它的大小增大，它可用的区域将自动增加。

检查步骤 4 的结果

要查看显示的新属性，请执行以下操作:

1. 刷新站点中的页，以便站点构建器能够获取对组件的更改。
2. 让页进入编辑模式。
3. 将组件拖放到页上。
4. 针对组件打开“设置”面板。
5. 单击定制设置按钮。
6. 将 Image Banner 更改为 Workspace。

您将在页上看到组件更新，Workspace 显示在图像上方。

继续执行步骤 5: 注册触发器。

步骤 5: 注册触发器

在此步骤中，您将回顾如何注册 Oracle Content Management 触发器，您可以使用组件“设置”面板中“链接”选项卡下的“触发器操作”选项进行选择。

触发器是 Oracle Content Management 组件间通信的一部分。任何组件都可以引发任意数量的触发器。组件可以为触发器提供有效负载，有效负载将传递给引发触发器时执行的任何操作。用户可以选择应为每个触发器执行的操作。最后，构建为一起工作的组件可以自动引发触发器，以对其他组件执行操作，而无需用户定义组件之间的交互。

对于您添加的组件，将作为该组件的注册数据的一部分来注册触发器。要添加触发器，请使用组件支持的每个触发器更新 "triggers" 属性数组。还必须指定触发器支持的有效负载，以便可以创建用户界面来允许用户将有效负载内的值映射到操作支持的属性。

打开 appinfo.json 文件并检查 "triggers": [], 条目。

```
"triggers": [{
  "triggerName": "imageClicked",
  "triggerDescription": "Image clicked",
  "triggerPayload": [{
    "name": "payloadData",
    "displayName": "Trigger Payload Data"
  }]
}],
```

在此条目中，您将看到以下内容:

- 一个 triggerName (即 "imageClicked")，它应该是唯一值，通常根据定制组件 ID 生成名称空间。

- 一个 triggerDescription (即 "Image clicked")，它由用户界面对话框用于显示触发器。
- 用于触发器的单一值 triggerPayload (即 "payloadData")。用户将能够在此有效负载中选择条目，并将它们映射到操作中的字段。

检查步骤 5 的结果

当转到组件“设置”面板的链接选项卡时，您可以看到并选择触发器：

1. 刷新站点中的页，以便站点构建器能够获取对组件的更改。
2. 让页进入编辑模式。
3. 将组件拖放到页上。
4. 针对组件打开“设置”面板。
5. 选择“设置”面板顶部的链接选项卡。
6. 单击触发器操作作为链接类型。
7. 单击您注册的 **Image clicked** 触发器。
8. 在对话框中，从“页操作”部分中拖动显示预警操作。（页操作是由 Oracle Content Management 提供的内置操作。）
9. 在消息字段中，选择触发器有效负载数据值，该值是您在注册触发器时看到的有效负载中条目的名称。您可以将其更改为所需的任何名称。

现在，您可以注册触发器并将触发器映射到内置操作（通过传递值）。在下一步中，我们将回顾如何引发触发器来执行操作。

继续执行[步骤 6: 引发触发器](#)。

步骤 6: 引发触发器

在此步骤中，我们将向您说明如何引发您看到的已注册的触发器。

组件可以在任何点引发触发器。通常，触发器由用户交互（通过单击按钮或选择表中的行）引发。但是，组件可以基于任何标准引发触发器；例如，当数据由于 REST 调用而更改时。

对于此示例，当您单击图像时，它将引发触发器，传递 whoAreYou 属性的当前值。

检查 render.js 文件并查看 SampleComponentViewModel 对象。

要引发触发器，请执行以下操作：

1. 检查 SampleComponentViewModel 对象中用于调用 Sites SDK 来引发触发器的函数。

```
self.raiseTrigger = function (triggerName) {
    SitesSDK.publish(SitesSDK.MESSAGE_TYPES.TRIGGER_ACTIONS, {
        'triggerName': triggerName,
        'triggerPayload': {
            'payloadData': 'some data here'
        }
    });
};
```

- 现在需要用户界面中的某些信息来调用函数以引发触发器。检查 `render.js` 文件并更新 `sampleComponentTemplate` 对象以具有以下条目：

```
'<div data-bind="attr: {style: imageStyle, \'data-layout\':
alignImage()}, click: imageClicked">' +
```

在 `SampleComponentViewModel` 对象中，您查看单击图像时调用的 JavaScript 函数。此函数调用 Sites SDK 以告知它触发为触发器 `imageClicked` 定义的所有操作，这是从步骤 2 中的 `click` 绑定传入的值。它还传递具有单个字段 `payloadData` 的 `triggerPayload`，并传递静态值 `'some data here'`。这些值 (`imageClicked` 和 `whoAreYou`) 与 (在前面步骤中) 注册触发器的 `appinfo.json` 文件中的那些值匹配。

在示例代码中，触发器由 `click` 绑定的 `data-bind` 引发，并传入触发器名称 `imageClicked`。根据用户选择的布局，当前存在 `<scs-image>` 组件的三种呈现。要确保为每种布局引发触发器，请编辑 `render.js` 文件以进行以下更改。

- 从不同布局引发触发器。查找以下代码的两个条目：

```
'<div data-bind="attr: {style: imageStyle, \'data-layout\':
alignImage()}">' +
```

将代码更改为以下内容：

```
'<div data-bind="attr: {style: imageStyle, \'data-layout\':
alignImage()}, click: imageClicked">' +
```

- 指定要传递给触发器的有效负载。更改以下代码：

```
self.raiseTrigger = function (triggerName) {
  SitesSDK.publish(SitesSDK.MESSAGE_TYPES.TRIGGER_ACTIONS, {
    'triggerName': triggerName,
    'triggerPayload': {
      'payloadData': 'some data here'
    }
  });
};
```

改用以下代码：

```
self.raiseTrigger = function (triggerName) {
  SitesSDK.publish(SitesSDK.MESSAGE_TYPES.TRIGGER_ACTIONS, {
    'triggerName': triggerName,
    'triggerPayload': {
      'payloadData': self.imageBannerText() // pass banner text as
payload
    }
  });
};
```

- 将 `render.js` 文件同步或上载到 Oracle Content Management 实例服务器。

现在您已查看了所需的代码，您可以挂接触发器，以便在单击按钮时定制组件将引发该触发器。

检查步骤 6 的结果

您现在应该能够注册要针对您的触发器执行的操作，并在引发触发器时执行该操作：

1. 刷新站点中的页，以便站点构建器能够获取对组件的更改。
2. 让页进入编辑模式。
3. 将组件拖放到页上。
4. 针对组件打开“设置”面板。
5. 选择“设置”面板顶部的链接选项卡。
6. 选择触发器操作作为链接类型。
7. 单击您看到的已注册的 **imageClicked** 触发器。
8. 在对话框中，从页操作部分中拖动显示预警操作。
9. 在消息字段中，选择 **payloadData** 值，其是您注册触发器时输入的有效负载。
10. 关闭“设置”面板，并将站点构建器切换到预览模式。
11. 单击组件中的图像。

此时将出现预警，显示 no message defined，因为您未指定 imageBannerText 值。

12. 让页进入编辑模式，并再次为组件打开“设置”面板。
13. 单击定制设置，并输入 Workplace。
14. 关闭“设置”面板，并将页切换到预览模式。
15. 单击组件中的图像。

现在它应该显示更新的有效负载 Workplace，这是从您对 click 绑定进行的更改调用的。

引发触发器时，您可以执行任意数量的操作。

注：

执行操作时没有预定义的顺序。虽然每个操作将按其列出的顺序进行调用，但是在调用下一个操作之前不需要等待该操作完成。如果操作进行异步调用，则执行下一个操作之前该操作可能未完成。

继续执行 [步骤 7: 注册操作](#)。

步骤 7: 注册操作

引发触发器时会对组件调用 Oracle Content Management 操作。

组件可以注册任意数量的操作，并且还可以定义操作支持的有效负载。用户选择操作时，他们可以填充要传递到相应操作的有效负载。

与注册触发器一样，可以在 `appinfo.json` 文件注册数据中注册组件支持的操作。要查看组件中示例操作的注册，请打开 `appinfo.json` 文件并查找 `"actions"` 代码。

```
"actions": [{
  "actionName": "setImageWidth",
  "actionDescription": "Update the image width",
  "actionPayload": [{
    "name": "imageWidth",
    "description": "Image Width in pixels",
    "type": {
      "ojComponent": {
        "component": "ojInputText"
      }
    },
    "value": ""
  }]
}]
```

此注册的操作将在“操作”对话框中可见，当您在组件的“设置”面板中的链接选项卡中单击触发器时，将调用该操作。

检查步骤 7 的结果

1. 刷新站点中的页，以便站点构建器能够获取对组件的更改。
2. 让页进入编辑模式。
3. 将组件拖放到页上。
4. 将按钮组件放到页上。
5. 针对按钮组件打开“设置”面板。
6. 在“一般信息”选项卡上，将按钮的标签更改为单击我。
7. 选择“设置”面板上的链接选项卡。
8. 选择触发器操作作为链接类型。
9. 针对按钮组件单击单击按钮触发器。
10. 在对话框中，展开左侧选项板中的 `A_Local_Component` 组件。
11. 将更新图像宽度操作从 `A_Local_Component` 组件拖放到页上。
12. 在图像宽度（像素）字段中输入 **300px**。

现在您已了解了如何注册操作以及该操作将如何在用户界面中显示。在下一步中，您将了解如何在调用组件时处理组件中的操作。

继续执行 [步骤 8: 执行操作](#)。

步骤 8: 执行操作

本主题结束时，您能够将在组件中执行操作的组件放在页上。这利用您在上一步中创建的操作注册。

要使组件执行操作，它必须监听 `EXECUTE_ACTION` 消息。此消息还包括传递到操作的有效负载，您将从其提取预期值。

要监听 EXECUTE_ACTION 消息, 请编辑 render.js 文件并使用以下条目更新 SampleComponentViewModel 对象:

```
SitesSDK.subscribe('EXECUTE_ACTION', $.proxy(self.executeActionsListener, self));
```

当收到 EXECUTE_ACTION 消息时, 将执行关联的回调函数:

```
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);  
      }  
    });  
  }  
}
```

这段代码创建一个执行操作的 JavaScript 函数, 然后在发出 EXECUTE_ACTION 消息时使用 Sites SDK 调用该函数。

每当发出 EXECUTE_ACTION 消息时都会调用该组件, 并且由组件决定仅处理设计为由其处理的操作。为此, 您必须检查操作的名称以确保它是组件可以处理的操作。

操作的有效负载是值数组。通常, 您将需要从数组中查找您关注的有效负载值。

注:

由于操作监听程序是回调, 因此您应使用 JavaScript 闭包或适当地绑定函数, 以确保在执行函数时, 您有权访问 viewModel。

检查步骤 8 的结果

1. 刷新站点中的页, 以便站点构建器能够获取对组件的更改。
2. 让页进入编辑模式。
3. 将组件拖放到页上。
4. 将按钮组件拖放到页上。
5. 针对按钮组件打开“设置”面板。
6. 在“一般信息”选项卡上, 将按钮的标签更改为单击我
7. 选择“设置”面板顶部的链接选项卡。
8. 选择触发器操作作为链接类型。
9. 针对按钮组件单击单击按钮触发器。
10. 在对话框中, 展开左侧的 A_Local_Component 组件。

11. 将 A_Local_Component 组件中的更新图像宽度操作拖放到右侧。
12. 在图像宽度（像素）字段中输入 300px。
13. 将页切换到预览模式。
14. 单击单击我按钮。

此时，图像的大小将增加到 300px。

注：

触发器和操作设计为支持组件间通信。它们并非设计为创建或管理状态。如果刷新页，页将还原到其原始状态，就像未引发触发器或者未执行操作一样。

继续执行步骤 9：为组件的每个实例创建不同的标题。

步骤 9：为组件的每个实例创建不同的标题

此步骤说明如何为组件的不同实例创建不同的标题。

将组件放到页上时，您将注意到组件的横幅显示为：A_Local_Component。如果用户只将您的组件之一放到页上，虽然这是可以的，但是您可能希望创建不同的标题，以使用户能够区分组件的不同实例。

可以使用 Sites SDK 更新组件的标题。在此步骤中，将根据 "imageBannerText" 属性更新标题。

要更新标题，请编辑 render.js 文件，并将以下代码添加到您的 SampleComponentViewModel 对象：

```
self.updateDescription = ko.computed(function () {
    SitesSDK.setProperty('description', self.imageBannerText());
});
```

只要 imageBannerText 可观察项更改，此 Knockout 计算就将更新组件的说明。

检查步骤 9 的结果

1. 刷新站点中的页，以便站点构建器能够获取对组件的更改。
2. 让页进入编辑模式。
3. 将组件放到页上。
4. 针对组件打开“设置”面板。
5. 单击定制设置按钮。
6. 将 **Image Banner** 更改为 **Workplace**。
7. 关闭“设置”面板，并将光标悬停在您的组件上以显示横幅。

现在应该看到显示 A_Local_Component Workplace。

继续执行步骤 10：结合使用嵌套组件和内嵌编辑。

步骤 10: 结合使用嵌套组件和内嵌编辑

Oracle Content Management 组件使用 KnockoutJS 组件体系结构进行实施。这意味着如果您使用 KnockoutJS 实施组件, 则可以将 Oracle Content Management 内置组件直接包含到模板中。

注:

由于 Oracle Content Management 内置组件只能在 Oracle Content Management 页中运行, 因此当组件在内嵌框架中呈现时无法使用嵌套组件。

要利用嵌套组件, 请执行以下操作:

1. 使用 KnockoutJS 实施组件。
2. 使用 RequireJS 包含您的组件, 并使用 Oracle Content Management 创建的相同 Knockout "ko" 实例变量。

这是必需的, 因为 Oracle Content Management 使用组件扩展 Knockout, 如果您使用自己的 KnockoutJS 实例, 则这些组件将不可用。

在此步骤中, 您将回顾如何在您的定制组件中呈现 Oracle Content Management 图像、段落和标题组件。用户将能够在页中直接编辑它, 并访问嵌套组件的“设置”面板。

要了解这些组件如何包含在模板中, 请编辑 `render.js` 文件并查看 `sampleComponentTemplate` 对象。默认呈现的部分如下所示:

```
'<!-- ko if: alignImage() !== \'right\' -->' +  
'<div style="display:flex;">' +  
'<div data-bind="attr: {style: imageStyle, \'data-layout\':  
alignImage()}, click: imageClicked">' +  
'<scs-image params="{ scsComponent: { \'renderMode\': mode,  
\'parentId\': id, \'id\': \'imageId\', \'data\': imageData } }"></scs-  
image>' +  
'</div>' +  
'<div data-bind="attr: {style: paragraphStyle}">' +  
'<scs-title params="{ scsComponent: { \'renderMode\': mode,  
\'parentId\': id, \'id\': \'titleId\', \'data\': titleData } }"></scs-  
title>' +  
'<scs-paragraph params="{ scsComponent: { \'renderMode\': mode,  
\'parentId\': id, \'id\': \'paragraphId\', \'data\':  
paragraphData } }"></scs-paragraph>' +  
'</div>' +  
'</div>' +  
'<!-- /ko -->' +
```

查看 `<scs-image>` 嵌套组件时, 您将看到以下条目:

```
'<scs-image params="{ scsComponent: { \'renderMode\': mode,  
\'parentId\': id, \'id\': \'imageId\', \'data\': imageData } }"></scs-  
image>' +
```

传递到 `params` 模板绑定的 `scsComponent` 数据包含以下内容:

- `renderMode`: 这是指站点构建器所处的模式。可以使用此项启用和禁用功能。例如, 当它由 `<scs-title>` 组件使用时, 它将在 `edit` 模式下运行时添加多信息文本编辑器。
- `parentId`: 这是必需的, 以便 Oracle Content Management 组件了解它将作为嵌套组件呈现。对嵌套组件的所有更改都将保存在定制组件的数据中。
- `id`: 嵌套组件的唯一 ID。这将进一步根据定制组件的 ID 生成名称空间。
- `data`: 嵌套组件的初始数据。如果未进一步修改组件, 它将使用此初始数据呈现。

引用的 `id` 和 `mode` 值传入到 `SampleComponentViewModel` 对象中的定制组件中, 因此您不需要修改该对象来获取这些值:

```
// Store the args
self.mode = args.viewMode;
self.id = args.id;
```

所有其他受支持嵌套组件的语法遵循与 `<scs-paragraph>` 相同的模式; 例如: `<scs-image>`, `<scs-title>`, `<scs-button>`。

检查步骤 10 的结果

1. 刷新站点中的页, 以便站点构建器能够获取对组件的更改。
2. 让页进入编辑模式。
3. 将组件拖放到页上。
4. 单击组件中的 `As a page author, you can edit. . .` 文本, 然后使用多信息文本编辑器更新说明。
5. 切换到预览模式以查看更新。
6. 切换回编辑模式。
7. 针对组件打开“设置”面板。
8. 单击现在显示的组件链接, 因为它已找到嵌套组件。
9. 单击段落, 这是它找到的嵌套组件。

现在可以针对组件中的段落组件更新属性。

注:

在组件已实例化之前, Oracle Content Management 不了解模板中可能存在的任何嵌套组件。要将隐藏的嵌套组件告知 Oracle Content Management, 可以使用 `SitesSDK.setProperty('visibleNestedComponents', []);` API。要在默认情况下显示隐藏的嵌套组件, 必须更新组件注册中的 `"nestedComponents": []` 数组。

继续执行 [步骤 11: 支持不同的布局](#)。

步骤 11: 支持不同的布局

在此步骤中, 我们将查看允许用户变更组件显示方式的布局。

定制组件可以支持您要允许用户选择的任意数量的布局。这些布局中的每个布局都将变更定制组件的显示方式。布局是对注册数据的另一个扩展。

要查看示例代码中支持的三种布局, 请查看 `appinfo.json` 文件中的 `"componentLayouts"` 条目

```
"componentLayouts": [  
  {  
    "name": "default",  
    "displayName": "IMAGE_LEFT_LAYOUT"  
  },  
  {  
    "name": "right",  
    "displayName": "IMAGE_RIGHT_LAYOUT"  
  },  
  {  
    "name": "top",  
    "displayName": "IMAGE_TOP_LAYOUT"  
  }  
],
```

如果针对定制组件打开“设置”面板, 则会看到一个用于在布局之间进行切换的选项。为了使组件能够对选择中的更改做出反应, `render.js` 文件提供了相应代码来获取当前所选值并监听对此值的更改。

编辑 `render.js` 文件并查看 `SampleComponentViewModel` 对象。

- 有一个 `layout` 可观察项 (在模板对其进行引用):

```
self.layout = ko.observable();
```

- 每当此值更改时, 都有一个更新函数要处理:

```
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);
```

- 初始化代码获取布局的原始值并调用更新函数:

```
SitesSDK.getProperty('componentLayout', self.updateComponentLayout);
```

属性更改监听程序检查对此属性的任何更改并调用更新函数:

```
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));

```

最后, `sampleComponentTemplate` 模板对象提供了相应代码来反映此值的更改:

```

'<!-- ko if: alignImage() === \'right\' -->' +

```

总之, 通过这些更改, 您可以在“设置”面板中选择布局并更新组件。

检查步骤 11 的结果

1. 刷新站点中的页, 以便站点构建器能够获取对组件的更改。
2. 让页进入编辑模式。
3. 将组件拖放到页上。
4. 针对组件打开“设置”面板。
5. 从布局属性中选择图像居右。

此时, 组件将更新来显示 "`<scs-image>`" 组件。

继续执行 [步骤 12: 定义定制样式](#)。

步骤 12: 定义定制样式

对于您创建的组件, 将像用于站点的主题中 `design.json` 和 `design.css` 文件中的任何其他组件那样进行处理。

要为定制组件添加您自己的样式, 请确认您注册组件时使用的 `id` 值。在 `appinfo.json` 文件中, 此项是 `"id": "hello-world"`。

使用该值, 编辑主题的 `design.json` 文件, 并针对该 `id` 添加要支持的新样式。例如, 编辑主题中的 `/designs/default/design.json` 文件并添加以下代码:

```

"hello-world": {
  "styles": [{
    "name": "Plain",
    "class": "hello-world-default-style"
  },
  {
    "name": "Gothic",
    "class": "hello-world-gothic-style"
  }
  ]
},

```

如果您针对组件打开“设置”面板, 现在应看到 **Plain** (默认值) 和 **Gothic** 作为“样式”选项卡中列出的两个选项。不过, 在这些选项之间切换将不执行任何操作, 直到您实际定义 `design.css` 文件中列出的样式类。

编辑主题的 `design.css` 文件，并添加样式的级联样式表 (cascading style sheet, CSS) 类。例如，编辑主题中的 `/designs/default/design.css` 文件并添加以下代码：

```
.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; }
```

将文件保存并同步到 Oracle Content Management 实例服务器。

检查步骤 12 的结果

1. 刷新站点中的页，以便站点构建器能够获取对组件的更改。
2. 让页进入编辑模式。
3. 将组件拖放到页上。
4. 针对组件打开“设置”面板。
5. 转到“样式”选项卡。
6. 在 `design.json` 文件中定义的 **Gothic** 和 **Plain** 样式之间切换。

您将注意到，在每个选择的已应用 CSS 类之间切换时，组件内的字体大小将进行调整以反映更改。

继续执行步骤 13: 在内嵌框架中呈现组件。

步骤 13: 在内嵌框架中呈现组件

到目前为止，示例已经显示了在页中内嵌呈现的本地组件。您还可以选择在内嵌框架中呈现组件。

例如，如果您的组件对页进行次要更新，则您可以选择在内嵌框架中呈现组件，这要求您在属性更改时重新创建页。此外，远程组件始终在内嵌框架中呈现。

本节中的示例取自在创建本地组件时您选择创建在 **iFrame** 中呈现的组件选项的情况下为您创建的文件。但是，您可以获取这些文件集并将其托管在远程服务器上，以便它们可以平等地应用于远程组件。

内嵌框架和非内嵌框架组件之间的相似性

“设置”面板

由于“设置”面板总是放置在内嵌框架中的页中，因此无论组件是否使用内嵌框架，“设置”面板的代码都不会更改。您将这两种用例创建相同的“设置”面板代码。

Sites SDK API

SDK API 对于这两种用例是相同的。您将使用相同代码来引发触发器，监听操作以及获取和设置属性值。虽然某些属性可能不适用于这两种情况（例如，无法为不使用内嵌框

架的组件设置 "height" 属性)，但 API 保持不变。因此，您可以在这两种类型的组件之间复制代码，本教程中讨论的示例代码将适用于这两种情况。

内嵌框架和非内嵌框架组件之间的差异

文件结构和相关项

如果在创建本地组件时选择创建在 **iFrame** 中呈现的组件，您将看到为您创建以下文件：

```
<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
```

通过创建这些文件，您可以立即在页上的内嵌框架中运行您的组件。此结构和标准本地组件的结构之间的主要差异为：

- JavaScript 相关项：
 - 您将获得这些文件的完整副本，以便组件能够运行。这些文件是运行示例内嵌框架组件所必需的。您可以根据您的要求添加和删除此目录的内容。
 - 由于在发布组件时该组件的 `assets` 目录下的所有内容都将推送到公共站点，因此 `js` 目录中的所有内容在站点构建器和运行时均可用。
 - 注：创建这些文件是为了易于使用。您应该考虑在主题中或在其他公共位置合并这些文件，而不是为每个内嵌框架组件创建这些文件的单独版本。
- `render.html`：
 - 这是完整的 HTML 文档，与标准组件的 `render.js` 文件相反，该文件是 AMD 模块。

组件“高度”管理

使用内嵌框架的问题之一是内嵌框架自身的高度管理。如果此项出错，您将看到在页上为组件显示滚动条，您可能需要该滚动条，也可能不需要该滚动条。

为了管理内嵌框架的高度，组件必须告诉页它需要内嵌框架的高度是多少。使用远程组件，您可能要处理跨域问题，因此在页上呈现了组件后，您必须使用 Sites SDK 消息传送来要求页将内嵌框架设置为所需的高度。通过使用 `SitesSDK.setProperty('height', {value})` API 完成此操作。（请参见 Oracle Content and Experience SDK。）

例如，创建 `setHeight` 函数和定制绑定处理程序，以便在页上呈现了组件时调用它。

- 更新高度函数：

```
// 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');
};
```

- Knockout 定制绑定处理程序，只要在页上呈现组件或属性更改就调用 `setHeight`:

```
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();
  }
};
```

- 模板更新以调用绑定处理程序:

```
<div data-bind="sampleAppSetAppHeight: true"></div>
```

触发器和操作注册

不在内嵌框架中的组件的触发器/操作注册位于 `appinfo.json` 文件中时，对于内嵌框架组件，组件自身负责提供此信息。使用以下两个 API 完成此操作:

```
SitesSDK.subscribe('GET_ACTIONS', self.getAppActions);
SitesSDK.subscribe('GET_TRIGGERS', self.getAppTriggers);
```

下面是使用这些 API 的示例。

```
// 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;
};

```

访问主题样式

由于组件在内嵌框架中呈现，因此它无权访问主题中提供的样式。Sites SDK 提供 API 来检索这些样式，以便它们可以应用于内嵌框架中的元素。

[步骤 14: 在内嵌框架中呈现组件时使用定制样式](#)中进一步探讨了该主题。

混合 HTTPS 与 HTTP 协议

由于 Oracle Content Management 使用 HTTPS 协议，因此页内引用的所有资源也必须使用 HTTPS。资源包括将在内嵌框架中呈现的基本 .html 文件以及它引用的所有文件。

此资源要求主要适用于远程组件，不过，您必须知道此约束。使用内嵌框架的本地组件的资源由 Oracle Content Management 服务器提供，因此，这些组件已经使用匹配的协议。

继续执行[步骤 14: 在内嵌框架中呈现组件时使用定制样式](#)。

步骤 14: 在内嵌框架中呈现组件时使用定制样式

在内嵌框架中呈现的组件无法直接访问 design.css 文件。相反，有其他步骤来获取组件中 design.css 的 URL 并将其添加到页。然后，您必须更新组件来反映用户选择的样式。

要在组件中包括和使用 design.css 文件，需要在 render.html 文件中进行更改：

1. 找到并包括 design.css 文件的 URL
2. 每当选择样式类更改时，获取其值
3. 更新模板来反映所选的 styleClass
4. 反映对组件中所选样式类的更改
5. 确保样式更改时内嵌框架会调整大小

下面是用于编辑 render.html 文件的详细说明：

1. 找到并包括 design.css 文件的 URL。

将 design.css 文件动态添加到页的 <head> 部分。进行加载后，设置内嵌框架的高度，因为它可能已通过应用样式进行了变更。

将以下代码添加到 viewModel 对象中:

```
// Dynamically add any theme design URL to the <head> of the page
self.loadStyleSheet = function (url) {
    var $style,
        styleSheetDeferred = new $.Deferred(),
        attempts = 100,
        numAttempts = 0,
        interval = 50,
        pollFunction = function () {
            // try to locate the style sheet
            for (var i = 0; i < document.styleSheets.length; i++) {
                try {
                    // locate the @import sheet that has an href
                    based on our expected URL
                    var sheet = document.styleSheets[i],
                        rules = sheet && sheet.cssRules,
                        rule = rules && rules[0];
                    // check whether style sheet has been loaded
                    if (rule && (rule.href === url)) {
                        styleSheetDeferred.resolve();
                        return;
                    }
                } catch (e) {}
            }
            if (numAttempts < attempts) {
                numAttempts++;
                setTimeout(pollFunction, interval);
            } else {
                // didn't find style sheet so complete anyway
                styleSheetDeferred.resolve();
            }
        };

    // add the themeDesign stylesheet to <head>
    // use @import to avoid cross domain security issues when
    determining when the stylesheet is loaded
    $style = $('<style type="text/css">@import url("'" + url + "'">
    style>');
    $style.appendTo('head');

    // kickoff the polling
    pollFunction();

    // return the promise
    return styleSheetDeferred.promise();
};

// update with the design.css from the Sites Page
SitesSDK.getSiteProperty('themeDesign', function (data) {
    if (data && data.themeDesign && typeof data.themeDesign ===
    'string') {
        // load the style sheet and then set the height
        self.loadStyleSheet(data.themeDesign).done(self.setHeight);
    }
});
```

```
    }
  });
```

2. 每当选择样式类更改时，获取其值。

`styleClass` 属性的值更改时，创建要跟踪的可观察项：

```
self.selectedStyleClass = ko.observable();
```

请注意，只有在有了样式类之后才能呈现。更改以下代码：

```
self.customSettingsDataInitialized = ko.observable(false);
self.initialized = ko.computed(function () {
    return self.customSettingsDataInitialized();
}, self);
```

改用以下代码：

```
self.customSettingsDataInitialized = ko.observable(false);
self.styleClassInitialized = ko.observable(false);
self.initialized = ko.computed(function () {
    return self.customSettingsDataInitialized() &&
self.styleClassInitialized();
}, self);
```

通过添加以下内容，获取所选样式类的初始值：

```
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. 更新模板来反映 `styleClass`。更改以下代码：

```
<p data-bind="attr: {id: 'titleId'}, text: titleText"></p>
```

改用以下代码：

```
<p data-bind="attr: {id: 'titleId'}, text: titleText, css:
selectedStyleClass"></p>
```

4. 反映对组件中所选样式类的更改：更改以下代码：

```
if (settings.property === 'customSettingsData') {
    self.updateCustomSettingsData(settings.value);
}
```


改用以下代码:

```
if (settings.property === 'customSettingsData') {
    self.updateCustomSettingsData(settings.value);
}
if (settings.property === 'styleClass') {
    self.updateStyleClass(settings.value);
}
```

5. 确保样式更改时内联框架会调整大小。更改以下代码:

```
// 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();
```

改用以下代码:

```
// create dependencies on any observables so this handler is called
whenever it changes
var imageWidth = viewModel.imageWidth(),
    imageUrl = viewModel.imageUrl(),
    titleText = viewModel.titleText(),
    userText = viewModel.userText(),
    selectedStyleClass = viewModel.selectedStyleClass();
```

6. 将文件保存并同步到 Oracle Content Management 实例服务器。

检查步骤 14 的结果

1. 刷新站点中的页, 以便站点构建器能够获取对组件的更改。
2. 让页进入编辑模式。
3. 将组件拖放到页上。
4. 针对组件打开“设置”面板。
5. 转到“样式”选项卡。
6. 在 design.json 文件中定义的 **Gothic** 和 **Plain** 样式之间切换。

您将注意到, 在每个选择的已应用 CSS 类之间切换时, 组件内的字体大小将进行调整以反映更改。

继续执行步骤 15: 与页撤消和重做行为集成。

步骤 15: 与页撤消和重做行为集成

由于 Oracle Content Management 代表定制组件存储属性, 因此对这些属性的更改会自动成为页的撤消和重做行为的一部分。

为了确保当用户单击撤消或重做时所发生的事情是清楚的, 这些“撤消事件”只应在用户实际对页执行了某些操作时发生。例如, 在用户实际更改属性之前, 打开定制组件的“设置”面板不应更新页内的属性。单纯地初始化“设置”面板中的属性不应导致更新事件。

如果不注意确保此行为，则可能发生意外行为。页仍将运行，但会损害用户体验。例如，可能会发生以下行为：

- 只要打开“设置”面板，保存按钮就会变为活动状态。
- 用户必须多次单击撤消才能看到任何效果。
- Redo 堆栈被删除，因为组件回写了意外的更改并使用新值更新了 Redo 堆栈。

本教程中提供的“设置”面板示例代码演示了如何确保只在准备实际调用 `saveData` 时而不是在初始化时进行回写。除非组件本身涉及用户交互，否则在组件本身内也应同样引起注意，以免更新 `customSettingsData`，尽管通常不那么令人担心。

继续执行步骤 16: 资产管理。

步骤 16: 资产管理

此步骤描述并说明了如何管理组件使用的资产。

资产包括 Oracle Content Management 在管理资产的生命周期时必须了解的组件和定制组件。

Oracle Content Management 内容文件夹

您在 Oracle Content Management 中创建的每个站点都有其自己的 `content` 文件夹。这是您通常看不到的隐藏文件夹。发布站点时，`content` 文件夹中的所有文件也将发布到文件系统。

例如，当您使用图像组件选择图像时，Oracle Content Management 将生成您所选图像的副本，并将其放在 `content` 文件夹中。您的 URL 始终指向此复制版本的图像，以便在您删除原始图像时站点不会中断。这也适用于 Oracle Content Management 提供的其他组件：图库、图库网格、文档、社交栏、文件下载，以及用于内容框和 `componentGroup` 的背景图像。

要让定制组件参与到此资产生命周期中，定制组件必须告知 Oracle Content Management 有关它需要服务代表其管理的任何资产。因为这涉及到生成资产的副本，所以定制组件还必须使用 Oracle Content Management API 选择资产，以便我们了解如何管理资产。

管理 URL

资产的 URL 会基于多个标准而改变。

- 组件的运行时 URL 与组件的站点构建器 URL 不同
- 如果您复制页，则 Oracle Content Management 还会在 `content` 文件夹中生成所有引用资产的副本，因此在 `content` 文件夹中从不会有多个组件指向相同的资产
- 将 `componentGroup` 放到页上可以为 `componentGroup` 中的组件引用的任何资产生成新副本

此外，尽管相对 URL 可能适合本地组件，但是远程组件需要您希望 Oracle Content Management 代表您管理的任何资产的完全限定 URL，这样它们就可以使用完整 URL 呈现其内嵌框架内容。

由于您不能依赖于保持静态的 URL，因此只能在代码中保存对资产 ID 的引用，并在要呈现资产时检索资产的 URL。

管理资产

以下 Sites SDK API 可用于管理资产。

```
SitesSDK.getProperty('componentAssets', callback);
```

- 这将获取当前资产的数组

- 每个资产条目包含：
 - **id**: 资产的唯一 ID。
 - **title**: Oracle Content Management 标题元数据。
 - **description**: Oracle Content Management 说明元数据。
 - **fileName**: 所选文件的原始名称。可用于在定制组件的“设置”面板中显示，以使用户知道所选的文件。这不是复制到 content 文件夹的文件的名称。
 - **source**: 资产的启用宏的 URL。此值将随时间而改变，并且不应由组件引用，但它必须保存为资产的一部分。
 - **url**: 基于从中调用 `getProperty()` 的上下文的资产的完全限定 URL。

```
SitesSDK.setProperty('componentAssets', [assets]);
```

- 调用此项可以保存希望 Oracle Content Management 代表您管理的所有资产。
- 如果不调用此项，则不会保存资产。
- 发布站点时，将删除不在此数组中的任何资产。
- `assets` 参数是资产数组，其格式与 `getProperty` 以及 `filePicker` 返回的格式相同。

 **注:**

不存储任何 `url` 值。该值将在您请求资产时动态创建。

```
SitesSDK.filePicker(options, callback);
```

- 用于打开文件选择器以选择资产列表的 API。
- 它在成功选择传入所选资产数组的资产时调用回调。
- 此时不保存任何内容，并且根据组件来决定是否调用 `setProperty('componentAssets', [assets]);` 以通过此选择保存项以及任何要保存的其他资产。

选择资产的示例

本节向您展示如何选择资产、存储其 ID 并从存储的资产中重新提取实际值。

1. 编辑 `settings.html` 文件。
2. 更改模板对象以包含 `Image selection`。

```
<div>
  <!-- Image selection -->
  <label id="imageLabel" for="imageAsset" class="settings-
heading" data-bind="text: 'Image'"></label>
  <input id="imageAsset" data-bind="value: imageName" readonly
class="settings-text-box">
  <button id="imageSelect" type="button" class="selectbutton"
data-bind="click: showFilePicker">Select Image</button>
</div>
```

3. 更改 viewModel 以添加用于存储所选资产的 ID 的可观察项。

```
self.imageID = ko.observable();
```

4. 更改 viewModel 以通过打开文件选择器并显示所选资产的名称管理资产选择。

```
//  
// handle component assets  
//  
self.assets = []  
  
// bring up a file picker to select the assets  
self.showFilePicker = function () {  
    // select an image  
    SitesSDK.filePicker({  
        'multiSelect': false,  
        'supportedFileExtensions': ['jpg', 'png']  
    }, function (result) {  
        if (result.length === 1) {  
            // update the array of assets  
            self.assets = result;  
  
            // update the image in customSettingsData  
            self.imageID(result[0].id);  
        }  
    });  
};  
  
// update the display name based on the assets  
self.imageName = ko.computed(function () {  
    var imageName = '',  
        imageID = self.imageID();  
    for (var i = 0; i < self.assets.length; i++) {  
        if (self.assets[i].id === imageID) {  
            imageName = self.assets[i].fileName;  
            break;  
        }  
    }  
  
    return imageName  
}, self);
```

5. 更新 viewModel 以在获取 customSettingsData 前先检索资产。此代码还将引起在 self.ImageID() 可观察项更改时调用 self.imageName。

```
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. 最后, 更新 save 函数以保存 imageID, 并确保使用所引用资产的列表更新 componentAssets。

```
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);
```

检查选择资产的结果

1. 刷新站点中的页, 以便站点构建器能够获取对组件的更改。
2. 让页进入编辑模式。
3. 将组件拖放到页上。
4. 打开“设置”面板。
5. 单击选择图像按钮。
6. 浏览 (或上载) 并选择图像。
请注意, 将存储图像的名称以显示所选图像。
7. 关闭“设置”面板。
8. 重新打开“设置”面板。
请注意, 将重新反映图像的名称。

呈现资产的示例

本节向您展示如何检索资产并在组件中呈现它们, 以及如何在设置面板中的值发生更改时使组件动态更新。

注:

虽然这展示的是页上内嵌框架中本地组件的示例, 但是类似代码适用于页中以内嵌方式呈现的组件。

1. 编辑 `render.html` 文件。
2. 更新模板以包含资产：

```
<!-- ko if: imageURL -->
<div style="flex-shrink:0;">
  <img data-bind="attr: {style: imageStyle, id: 'imageId', src:
imageURL, alt: '', title: ''}, click: imageClicked" />
</div>
<!-- /ko -->
```

3. 在 `viewModel` 中，创建两个可观察项，以从 `customSettingsData` 获取 `imageID`，并存储从存储的资产列表检索的 `imageURL`。

```
self.imageID = ko.observable();
self.imageURL = ko.observable();
```

4. 更新 `viewModel`，以便每当 `imageID` 更改时都获取相应的图像资产 URL。

```
self.imageID.subscribe(function (imageID) {
  // whenever the image changes get the updated referenced asset
  SitesSDK.getProperty('componentAssets', function (assets) {
    for (var i = 0; i < assets.length; i++) {
      if (assets[i].id === imageID) {
        self.imageURL(assets[i].url);
        break;
      }
    }
  });
});
```

5. 更新 `viewModel` 以从 `customSettingsData` 检索 ID。

检查呈现资产的结果

1. 刷新站点中的页，以便站点构建器能够获取对组件的更改。
2. 让页进入编辑模式。
3. 将组件拖放到页上。
4. 打开“设置”面板。
5. 单击选择图像按钮。
6. 浏览（或上载）并选择图像。

请注意，将存储图像的名称以显示所选图像。

7. 关闭“设置”面板。

此时，您应该会看到在组件中呈现的所选图像。

继续执行[教程回顾](#)。

教程回顾

本教程概述了如何使用 Knockout 组件工厂创建定制组件。

本教程的主要目的是，使用此模式，您只需更新 `SampleComponentViewModel` 和 `sampleComponentTemplate` JavaScript 对象即可创建任何定制组件。您浏览本教程时，`sampleComponentFactory` and `SampleComponentImpl` 对象未进行更改。您能够实施这些更改，而不必处理与页的通信，并能够执行以下任务：

- 将更改从“设置”面板传送到组件，并保持这些更改。
- 执行触发器和操作，并与页上的其他组件交互。
- 创建布局并利用嵌套组件。
- 定义特定于组件的样式。

虽然此示例将定制组件拆分为许多文件，但这是为了使本教程更清晰。为了进行优化，您应该考虑适当地打包您的文件，以避免多次下载。

最后，虽然本教程适用于基于 Knockout 的组件，但是如果您要使用其他 JavaScript 技术堆栈（例如 AngularJS）创建定制组件，则必须重新实施 `SampleComponentImpl` 对象来创建与该框架的相应通信，以及实际组件的特定于技术的实施。此工作不在本教程的范围内。

27

Sites SDK 参考

Oracle Content Management JavaScript SDK (Sites SDK) 具有一组函数，通过这些函数，组件可以与 Oracle Content Management 更为紧密的集成。

可从 Oracle Content Management 服务器下载 Sites SDK：

```
http://{server}/_sitesclouddelivery/renderer/app/sdk/js/sites.min.js
```

站点应用程序编程接口 (Application Programming Interface, API) 定义涵盖全局对象 SitesSDK 的所有已定义名称空间：

- [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](#)

Sites SDK 的主要用途是让用户能够构建和管理应用程序。它提供多个功能：

- 跨内嵌框架和域边界的通信
 - 允许组件与页通信，而不管组件的托管位置
 - 包括参与页模型生命周期：
 - * 创建组件
 - * 使用存储的属性呈现组件
 - * 更新组件并保存那些更改
 - * 从页中删除组件
 - 存储并检索用于配置组件以用于页的定制设置
 - 使用组件间通信以参与触发器和操作
- 对页的几何提示（不适用于定制组件）
 - 在组件要求页动态调整内嵌框架大小时允许页执行此操作
 - 导致组件显示为页的一部分，并且页响应组件调整大小事件而移动（无滚动条）

- Oracle Content Management 调用组件时组件可意识到该调用
 - 组件和页之间的通信包含由私有密钥签名的令牌。
 - 通过 Sites SDK，组件可以对令牌进行解码并帮助对调用方进行验证。

通过提供脚本的已知路径包括 Sites SDK 函数。例如：

```
<script type="text/javascript"
src="<sdk_install_dir_path_prefix>/sites.min.js">
</script>
```

Sites SDK 具有以下全局对象：

- SitesSDK
- SitesSDK.Utils

SitesSDK

SitesSDK 全局对象可用于呈现和设置端点。

命令	任务
SitesSDK.getProperty(propertyName, callback)	从主机站点获取指定属性的值。
SitesSDK.setProperty(propertyName, propertyValue)	将指定属性的值设置为 <i>propertyValue</i> 。
SitesSDK.getSiteProperty(propertyName, callback)	从主机站点获取指定站点属性值。例如，它可以提取主机站点正在使用的当前主题设计。
SitesSDK.publish(messageType, payload)	向服务器发送消息，并接受 <i>messageType</i> 和 JSON 对象作为有效负载。
SitesSDK.subscribe(messageType, callback)	从主机站点发出的消息的消息监听程序。调用是异步进行的。这些是 JavaScript 回调，因此，该函数应使用 JavaScript 闭包或适当地绑定该函数，以确保它有权访问适当的上下文。

SitesSDK.Utils

SitesSDK.Utils 名称空间具有可由 Oracle Content Management 组件的所有端点调用的实用程序函数。此名称空间可用于远程组件的“设置”面板，但不可用于内嵌本地组件。

▲ 注意：

Oracle 官方不支持这些函数；它们用作示例实施。使用它们时需自担风险。

命令	任务
SitesSDK.Utills.Logger	返回 Sites SDK 正在使用的日志记录程序对象。日志级别可以设置为以下级别之一： <ul style="list-style-type: none"> • 调试 • 日志 • 信息 • 错误 • 警告
SitesSDK.Utills.addSiteThemeDesign(cssUrl)	将当前站点主题设计添加到组件。
Utills.getStyle(elem,styleProp)	用于计算内嵌框架高度的实用程序方法。

SitesSDK.getProperty('componentAssets', callback)

此函数返回当前代表站点内的组件而存储的资产列表。

参数

名称	类型	说明
componentAssets	字符串	调用列表创建器。
callback	函数	回调函数

用法

```
// 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
//
// returns a list of selected assets. If user cancels out of the picker,
// the callback is never called
SitesSDK.filePicker({options}, callback);

//
// example 'componentAssets' JSON returned:
//
'componentAssets': [{
  'name': <nameOfAssetInContentFolder>, // this is used to uniquely and
  persistently identify the asset. It is typically prefixed by the
  ;originalName; followed by a 16 digit string
  'originalName': <originalName>, // name of the asset as selected
```

```

from DOCS
  'description': <description>,           // description/other
attributes that are available from DOCS
  'url': <fully qualified url to the asset>
}]

```

SitesSDK.getProperty(propertyName, callback)

此函数从主机站点获取指定属性的值。

参数

名称	类型	说明
propertyName	字符串	属性名称。
callback	函数	回调函数。

相关属性

属性名称	说明	设置	呈现
height	框架高度	NA	是
width	框架宽度	NA	是
customSettingsData	设置数据	是	是
styleClass	选定的样式类	是	是
任何其他样式属性，例如 borderColor、 borderRadius、alignment、 borderStyle、margin 等。			

用法

```

SitesSDK.getProperty('customSettingsData', function (propertyData) {
  // store settings data
  self.configuration(
    {
      'id': propertyData.id || self.id,
      'url': propertyData.url || self.params.url,
      'limit': propertyData.limit || self.params.limit
    }
  );
});

```

从呈现 URL 提取的其他相关属性

除了使用 getProperty() 提取属性值外，还可从呈现 URL 提取的其他相关属性包括如下属性。

属性名称	说明
id	组件 ID
instance	加密令牌
width	iframe 宽度
height	iframe 高度
viewMode	呈现模式值
locale	站点或浏览器中设置的语言
settingsId	当前设置 ID

从设置 URL 提取的其他属性

除了使用 `getProperty()` 提取属性值外，还可从设置 URL 提取的其他属性包括如下属性。

属性名称	说明
currCompId	正在编辑的组件的 ID
instance	加密令牌
width	设置 iframe 的宽度
locale	站点或浏览器中设置的语言
OrigSettingsId	组件的原始设置 ID
settingsId	新设置 ID

SitesSDK.getSiteProperty(propertyName, callback)

此函数从主机站点获取指定站点属性值。例如，提取主机站点正在使用的当前主题设计。

参数

名称	类型	说明
propertyName	字符串	站点属性的名称
callback	函数	用于读取属性值的回调函数

站点属性

属性名称	说明	设置	呈现
theme	当前主题设计的 URL	NA	是

用法

```
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])

此函数用于更新代表定制组件存储的站点资产列表。

参数

名称	类型	说明
componentAssets	字符串	创建组件资产列表。
assets	JSON 对象	返回选定资产的列表。

参数

属性名称	说明
filetypes	支持的文件类型列表。
multiSelect	选择单个或多个文件。

用法

```

// 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
//
// returns a list of selected assets. If user cancels out of the
picker,
// the callback is never called
SitesSDK.filePicker({options}, callback);

//
// example 'componentAssets' JSON returned:
//
'componentAssets': [{
    'name': <nameOfAssetInContentFolder>, // this is used to uniquely
and persistently identify the asset. It is typically prefixed by the
originalName; followed by a 16 digit string

```

```

    'originalName': <originalName>,           // name of the asset as selected
    from DOCS
    'description': <description>,           // description/other attributes
    that are available from DOCS
    'url': <fully qualified url to the asset>
  }]

```

SitesSDK.setProperty(propertyName, propertyValue)

此函数将指定属性的值设置为 *propertyValue*。

参数

名称	类型	说明
propertyName	字符串	属性名称。
propertyValue	JSON 对象	属性的值。

相关属性

此表列出了可使用 `setProperty()` 提取的一组相关属性。

属性名称	说明	设置	呈现
height	框架高度	NA	是
width	框架宽度	NA	是
customSettingsData	设置数据	是	是
styleClass	选定的样式类	是	是
description	用于页上本地组件实例的横幅中		
layout	仅为本地组件返回当前选择的布局。可以根据 <code>appinfo.json</code> 文件中的值从“设置”面板进行更新。		
renderStatus	允许定制组件报告呈现已完成。	true	
任何其他样式属性，例如 borderColor、borderRadius、 alignment、borderStyle、 margin 等。			

用法

```

// configuration that can be saved
var saveconfig = {
    //current settings id
    'id': self.appSettingsProperties['settingsId'],
    'url': self.url(),
    'limit': self.limit()
};
// save property 'customSettingsData'
SitesSDK.setProperty('customSettingsData',saveconfig);

```

SitesSDK.filePicker({options}, callback)

此函数返回选定文件列表。

参数

名称	类型	说明
options	字符串	用于指定要返回的选定文件的选项。
callback	函数	回调函数。如果用户从选择器中取消，则永远不会调用回调。

用法

```
// 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
//
// returns a list of selected assets. If user cancels out of the
// picker,
// the callback is never called
SitesSDK.filePicker({options}, callback);

//
// example 'componentAssets' JSON returned:
//
'componentAssets': [{
  'name': <nameOfAssetInContentFolder>, // this is used to uniquely
and persistently identify the asset. It is typically prefixed by the
;originalName; followed by a 16 digit string
  'originalName': <originalName>, // name of the asset as
selected from DOCS
  'description': <description>, // description/other
attributes that are available from DOCS
  'url': <fully qualified url to the asset>
}]
```

SitesSDK.openDocumentPicker(options)

此函数返回选定文件列表。

SitesSDK.openDocumentPicker() 函数采用单个参数并返回 Promise，其解析为所选文档。可用选项与《*Integrating and Extending Oracle Content Management*》中

"Documents View Embed UI (documentsView Component) API V2" 中的 documentsView 组件的那些选项匹配。

用法

```
SitesSDK.openDocumentPicker({
  selectable: "any",
  layout: "grid"
}).then(function (selection) {
  console.log(selection);
});
```

一旦文档被检索到，将使用 [SitesSDK.getProperty\('componentAssets', callback\)](#) 和 [SitesSDK.setProperty\('componentAssets', \[assets\]\)](#) 属性告知 Oracle Content Management 必须将这些文档的 ID 存储在站点中。否则，它们将不会发布到站点上，且可能会删除。

SitesSDK.openAssetPicker (可选)

此函数返回选定文件列表。

`SitesSDK.openAssetPicker()` 函数采用单个参数并返回 Promise，其解析为所选资产。可用选项与《*Integrating and Extending Oracle Content Management*》中 "Assets View Embed UI (assetsView Component) API V2" 中的 assetsView 组件的那些选项匹配。

用法

```
SitesSDK.openAssetPicker({
  select: "single"
}).then(function (selection) {
  console.log(selection);
});
```

一旦检索到资产，系统将使用 [SitesSDK.getProperty\('componentAssets', callback\)](#) 和 [SitesSDK.setProperty\('componentAssets', \[assets\]\)](#) 属性指示 Oracle Content Management 必须将这些资产的 ID 存储在站点中。否则，它们将不会发布到站点上，且可能会删除。

SitesSDK.publish(messageType, payload)

此函数向服务器发送消息。它接受 messageType 和 JSON 对象作为有效负载。消息类型在页中进行识别和处理。如果没有处理程序，将忽略传递的消息类型。

参数

名称	类型	说明
messageType	字符串	消息类型
payload	JSON 对象	消息有效负载

消息类型

此表列出了本地（使用内嵌框架）或远程组件已知的且可以使用 `SitesSDK.publish()` 函数发送的消息类型。

消息类型	说明
SETTINGS_UPDATED	用于“设置”面板以将设置更新的消息发送到呈现端点。
TRIGGER_ACTIONS	用于组件呈现端点以触发主机站点中的操作。

用法

```
// raise trigger
SitesSDK.publish(SitesSDK.MESSAGE_TYPES.TRIGGER_ACTIONS, {
    'triggerName': 'scsChangeSettings',
    'triggerPayload': { 'url':
this.configuration()['url'],'feedcount': this.configuration()['limit'] }
});
```

SitesSDK.subscribe(messageType, callback)

此函数实际上是消息监听程序，监听主机站点发出的消息。调用是异步进行的。

在 SDK 收到特定类型的消息时将调用注册的回调。如果回调返回值，则该值将重新返回到页。如果未向此函数传递任何回调，则将删除该特定消息类型对应的任何已注册监听程序。

这些是 JavaScript 回调，因此，该函数应使用 JavaScript 闭包或适当地绑定该函数，以确保它有权访问适当的上下文。

参数

名称	类型	说明
messageType	字符串	消息类型： <ul style="list-style-type: none"> SETTINGS_UPDATED TRIGGER_ACTIONS EXECUTE_ACTION GET_ACTIONS GET_TRIGGERS COPY_CUSTOM_DATA PASTE_CUSTOM_DATA
callback	函数	从主机站点收到消息时的函数。

消息类型

消息类型	说明
SETTINGS_UPDATED	订阅此消息以在“设置”面板中更改了任何值时收到通知。通常，组件通过更新其在用户界面中的重现来响应。
TRIGGER_ACTIONS	发布此消息将引发触发器并传递有效负载。
EXECUTE_ACTION	订阅此消息以处理定制操作。监听程序通常通过执行给定操作来处理此消息。
GET_ACTIONS	订阅此消息以声明组件可以执行的一组操作。返回一组操作。

消息类型	说明
GET_TRIGGERS	订阅此消息以声明组件可以引发的一组触发器。返回一组触发器。
COPY_CUSTOM_DATA	订阅此消息以处理将定制设置数据复制到剪贴板。返回表示要放在剪贴板上的数据的对象。
PASTE_CUSTOM_DATA	订阅此消息以处理从剪贴板粘贴定制设置数据。

用法

有些监听程序要求传递数据，而有些监听程序应返回数据。

示例 1:

为消息类型 EXECUTE_ACTION 注册消息监听程序。

```
SitesSDK.subscribe(SitesSDK.MESSAGE_TYPES.EXECUTE_ACTION,
self.executeAction,self);
```

注册的 callback 函数应接受参数，如下所示：

```
// 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
}
```

示例 2:

COPY_CUSTOM_DATA 监听程序通常采用类似如下所示的代码实现：

```
// listen for COPY_CUSTOM_DATA request
SitesSDK.subscribe(SitesSDK.MESSAGE_TYPES.COPY_CUSTOM_DATA,
```

方法 copyCustomDataListener() 会返回表示要复制到剪贴板的数据的对象，例如：

```
// Handle Copy Style (save customSettingsData to the clipboard)
self.copyCustomDataListener = function() {
    return {
        prop1: this.prop1(),
        prop2: this.prop2()
    };
};
```

示例 3:

PASTE_CUSTOM_DATA 请求接受数据，但不必返回任何内容：

```
// listen for PASTE_CUSTOM_DATA request
SitesSDK.subscribe(SitesSDK.MESSAGE_TYPES.PASTE_CUSTOM_DATA,
pasteCustomDataListener);
```

以及：

```
// Handle Paste Style (apply customSettingsData from the clipboard)
self.pasteCustomDataListener = function(data) {
    ...
};
```

SitesSDK.Utils.addSiteThemeDesign(cssUrl)

此函数在当前页 HTML <head> 标记中创建 link 元素。源设置为 cssUrl 的路径。

▲ 注意：

Oracle 官方不支持 Utils 名称空间中的函数。它们将用作示例实施。使用它们时需自担风险。

参数

名称	类型	说明
cssUrl	字符串	当前主题设计的 URL 路径。仅用于远程组件。本地组件（在 iframe 中实现）必须从页（而不是 URL）提取属性。

用法

此函数通常与从主机站点提取当前主题设计操作结合使用，如以下示例代码中所示：

```
// 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

此函数返回 Sites SDK 正在使用的日志记录程序对象。

 **注意：**

Oracle 官方不支持 Utils 名称空间中的函数。它们将用作示例实施。使用 Utils 函数需要您自担风险。

日志级别可设置为 debug、log、info、error 或 warn。

故障排除

您可能会遇到下面这些与创建站点有关的问题。此处提供了一些答案。

- [我尝试创建站点，但没有模板](#)
- [我无法删除站点](#)
- [无法打开站点树或编辑页](#)
- [我添加了组件，但没有在页中显示](#)
- [我的文件夹、文件和对话组件无法工作](#)
- [我上载了图像的新版本，但没有在页中显示](#)
- [我更改了页布局，但一些内容消失了](#)
- [我基于另一个服务添加了组件，但它无法正常工作](#)
- [我的企业站点显示一条警告](#)

我尝试创建站点，但没有模板

必须先安装和共享模板，然后才能创建站点。

随 Oracle Content Management 提供的模板必须由管理员安装和共享，然后才能使用。请与管理员联系。

与此类似，在您创建模板时，不论是通过导入、复制还是从某个站点创建的方式，除非您明确共享，否则任何其他人都无法使用该模板。

请参见 *Administering Oracle Content Management* 中的“配置站点设置”。

我无法删除站点

下面列举了您无法删除站点的一些原因：

- 站点处于联机状态。
首先，请使站点脱机，然后再删除。您必须具有站点的管理者角色才能使该站点脱机。
- 您不具有所需权限。
如果您创建了站点（您是站点所有者）或者其他人与您共享了站点并授予您内容提供者或管理者角色，则您可以删除该站点。

有关详细信息，请参见[管理站点和站点设置](#)。

无法打开站点树或编辑页

默认情况下站点构建器以预览模式打开，因此您可以查看站点，但不能编辑站点。您只能在编辑器处于编辑模式时编辑站点。

请确保  设置为编辑。

我添加了组件，但没有在页中显示

这可能有多种原因。

- 确保与远程组件（应用程序）关联的 URL 有效。请参见[注册远程组件](#)。
- 编辑器将远程组件封装在称为内嵌框架的 HTML 元素中（iframe 标记）。并非所有远程组件都允许封装在内嵌框架中。

向组件提供者确认它是否能封装在内嵌框架中。

我的文件夹、文件和对话组件无法工作

与 Oracle Content Management 通信的组件需要特定资源和设置。

以下组件需要访问 Oracle Content Management 中的资源：

- 文件夹列表
- 文件列表
- 文档管理器
- 对话

文件夹和文件组件需要访问 REST 应用程序编程接口 (API)。

验证以下内容：

- 组件必须具有对 API 端点的访问权限，才能访问文件夹和文件 REST 操作。

默认情况下，在您使用为站点提供的标准 URL 时，Oracle Content Management REST API 端点可用。如果使用定制 URL，则可能需要明确提供对 /documents REST API 端点的访问权限。

请参见[如何映射站点 URL?](#)

我更改了页布局，但一些内容消失了

如果选择数量较少或者不同名称的内容框，则其他内容框中的现有内容将不会显示在新布局中。

内容不会被删除，但仅当您选择的布局具有同名内容框时才能够显示。

请确保您选择了具有相同数量的内容框的布局。如果您选择的布局具有相同数量的内容框，但一些内容仍未显示，则可能是因为内容框的名称不同。请与主题设计人员联系以解决不符之处。

我上载了图像的新版本，但没有在页中显示

选择要用于组件的图像时，将创建唯一副本并将它存储在站点中。

上载文件时，文件存储在站点资料档案库中。上载与现有文件同名的文件时，将创建该文件的新版本。

您在页上看到的图像文件是从资料档案库中选择的文件的副本，不会自动更新。其设计如此，用于防止在资料档案库中更新图像文件时无意更改了现有页。

要更新用于组件的图像，您必须从组件属性页中明确选择该图像。有关详细信息，请参见[图像](#)。

我基于另一个服务添加了组件，但它无法正常工作

可以将站点与若干不同的进程和服务集成。

如果您基于与 Process Cloud Service、Oracle Intelligent Advisor（以前称为 Oracle Policy Automation）以及其他服务的集成创建了组件，这两个服务必须使用同一身份域。请与设置该集成的人员核实，以验证这两个服务是否使用同一个身份域。

我的企业站点显示一条警告

如果一个企业站点中包含 v1.0 内容布局，您将在站点构建器中看到一条警告，在查看该站点时也会看到一条警告，这是由于这些内容布局将无法呈现。需要将这些布局更新到 v1.1 内容布局。

在以下任一情况下可能会出现此问题：

- 如果您使现有站点可翻译，则可以通过为该站点分配默认的语言和本地化策略来自动将该站点转换为企业站点。如果站点模板中包含 v1.0 内容布局，则这些内容布局将无法呈现。
- 如果您从使用 v1.0 内容布局的旧模板中创建企业站点，则这些内容布局将再次无法重现。

要更正此问题，请将内容布局更新到 v1.1 内容布局。可以在组件目录中创建新内容布局来查看差异。您将需要添加一行来注册内容布局版本并将 `data.fieldname` 更改为 `fields.fieldname`。