

# Oracle® Cloud

## 運用 Oracle Content Management 建置網站



F26689-21  
2021 June



Oracle Cloud 運用 Oracle Content Management 建置網站,

F26689-21

版權所有 © 2018, 2021, Oracle 和 (或) 其關係公司。

主要作者：Bruce Silver

特約作者：Jean Wilson, Sarah Bernau, Kalpana N, Ron van de Crommert

特約作者：Bonnie Vaughan

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

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

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

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

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

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

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

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

# 目錄

## 前言

---

適用對象	xvi
說明文件協助工具	xvi
多元共融	xvi
相關資源	xvi
印刷慣例	xvii

## 部分 I 簡介

---

### 1 Oracle Content Management 簡介

---

存取 Oracle Content Management	1-1
瞭解角色	1-1
管理資產	1-2
管理內容	1-2
協作內容	1-2
建置網站	1-3
整合與擴充 Oracle Content Management	1-3
開始使用	1-3
移轉至 Oracle Cloud Infrastructure	1-4

### 2 開始建置網站

---

開始建立網站	2-1
瞭解網站建立程序	2-2
瞭解網站開發	2-3
瞭解網站治理	2-5

### 3 建立您的第一個網站

---

開始之前	3-1
------	-----

步驟 1：設定環境	3-2
匯入基本導覽功能表自訂元件	3-2
發布基本導覽功能表自訂元件	3-3
新增範例影像	3-4
步驟 2：設定網站	3-5
建立網站	3-5
編輯網站	3-6
步驟 3：發布網站	3-25
執行其他作業	3-27

## 部分 II 建立和編輯網站

---

### 4 建立網站

---

建立網站	4-1
複製網站	4-3
管理網站要求	4-4
變更網站要求詳細資訊	4-5
檢視網站要求原則	4-5

### 5 編輯網站

---

瞭解網站產生器頁面	5-1
瞭解網站更新	5-3
使用更新	5-5
編輯秘訣與技巧	5-5
使用樣式和格式	5-9
自訂網站設定值	5-10
新增自訂網站特性	5-11
使用表格	5-13
上傳網站檔案	5-14

### 6 在網站中使用樣板和主題

---

瞭解樣板	6-1
從網站建立樣板	6-4
變更樣板詳細資訊	6-5
變更樣板規定	6-6
變更樣板狀態或對象	6-7

管理樣板	6-7
匯出及匯入樣板	6-10
瞭解主題	6-11
管理主題	6-12
發布主題	6-14

## 7 管理自訂元件與版面配置

---

瞭解自訂元件	7-1
瞭解版面配置	7-2
使用自訂元件和版面配置	7-4
註冊遠端元件	7-6
建立本機元件、版面配置、內容欄位編輯器或內容表單	7-7
匯出/匯入元件或版面配置	7-8

## 8 使用網站頁面

---

瀏覽至頁面	8-1
檢視頁面	8-1
新增頁面	8-3
移動頁面	8-5
刪除頁面	8-6
變更頁面設定值	8-6
變更頁面版面配置	8-9
變更背景或主題	8-10

## 9 排列頁面內容

---

新增元件和區段版面配置	9-1
使用資產和內容項目	9-3
使用觸發程式和動作	9-3
使用水平區段版面配置	9-5
使用兩欄和三欄版面配置	9-6
使用垂直區段版面配置	9-7
使用頁籤式區段版面配置	9-8
使用滑桿區段版面配置	9-8

## 10 使用內建元件

---

基本元件	10-2
標題	10-2
段落	10-4
純文字	10-5
按鈕	10-6
結構元件	10-7
分隔線	10-8
空白間隔	10-8
媒體元件	10-9
影像	10-9
圖庫	10-11
圖庫格線	10-14
YouTube 視訊	10-17
視訊	10-18
文件元件	10-19
文件	10-19
資料夾清單	10-20
檔案清單	10-21
文件管理程式	10-22
專案程式庫	10-24
社群元件	10-25
社群列	10-26
Facebook 按讚與推薦	10-27
Twitter 分享與跟隨	10-27
對話元件	10-28
對話清單	10-30
處理作業元件	10-31
處理作業開始表單	10-31
處理作業工作清單	10-33
工作詳細資訊表單	10-34
內容項目	10-36
內容項目元件	10-36
內容預留位置	10-37
內容清單	10-38
內容搜尋	10-42
建議	10-43
其他元件	10-44

地圖	10-45
大標題	10-45
文章	10-47
影像 (含文字)	10-48
元件群組	10-50
在頁面上使用 Cobrowse 功能	10-50
Oracle Intelligent Advisor	10-52
Oracle Visual Builder	10-52

## 部分 III 發布和管理網站

---

### 11 管理網站

---

瞭解網站頁面	11-1
管理網站和網站設定值	11-2
變更網站描述、標誌或內嵌特性	11-6
設定搜尋引擎特性	11-6
啟用搜尋引擎最佳化的預先呈現服務	11-7
設定靜態網站傳遞選項	11-9
啟用發布時自動編譯	11-9
覆寫編譯網站的預設快取控制標頭	11-9
指定行動使用者代理程式以支援經過編譯的調適型版面配置	11-10
指定並設定虛名 URL	11-10
啟用 Cobrowse 整合	11-12
新增分析追蹤功能	11-14

### 12 發布網站

---

讓網站上線或離線	12-1
發布網站變更	12-1

### 13 保護網站

---

瞭解網站安全性	13-1
變更網站安全性	13-3

### 14 使用多語言網站

---

多語言網站總覽	14-1
翻譯網站	14-2

管理網站翻譯工作	14-4
翻譯的地區設定	14-4
翻譯的自訂地區設定	14-5
設定 URL 重新導向的地區設定別名	14-5

## 15 使用網站重新導向或 URL 對應

---

重新導向計畫	15-1
簡單的字串對字串比對	15-1
簡化的萬用字元比對	15-2
新增網站重新導向	15-2
在 JSON 檔案指定重新導向規則	15-2
將重新導向規則檔上傳至網站	15-8
對應網站 URL	15-8

## 16 改善網站效能

---

利用快取功能改善效能	16-1
程式實際執行快取功能	16-1
網站產生器快取功能	16-3
網頁頭版區域 (ATF) 呈現	16-3

## 部分 IV 網站開發

---

### 17 自訂設計與樣式

---

關於設計	17-1
設計檔案	17-1
響應式表格設計	17-4
自訂對話清單樣式	17-7
自訂資料夾清單和檔案清單樣式	17-9
自訂社群列圖示	17-10
設定 Oracle Intelligent Advisor 互動式提問樣式擴充功能	17-11

### 18 瞭解背景使用

---

關於背景與主題	18-1
背景的實行方式	18-1

設定值的儲存位置	18-2
----------	------

## 19 設定觸發程式和動作

---

關於觸發程式和動作	19-1
設定觸發程式	19-1
設定動作	19-3

## 20 開發樣板

---

關於樣板	20-1
基本樣板結構	20-2
建立樣板	20-3
匯出樣板	20-4
匯入樣板	20-5
使用入門樣板	20-6
從 Bootstrap 或網站設計樣板建立網站樣板	20-9
使用 Developer Cloud Service 開發樣板	20-14
登入 Oracle Content Management 的 Developer Cloud Service 主控台	20-15
在 Developer Cloud Service 中建立專案	20-15
在 Developer Cloud Service 中建立樣板	20-16
在 Developer Cloud Service 中複製樣板	20-16
將樣板匯入 Developer Cloud Service	20-17
合併變更	20-17
從 Developer Cloud Service 匯出樣板	20-17

## 21 開發主題

---

關於主題	21-1
基本主題結構	21-2
網站導覽	21-4
建立主題	21-6
隱藏主題的元件和區段版面配置	21-11
隱藏主題的元件對齊方式、寬度或間距選項	21-12
建立元件與主題的關聯	21-13
Sites Rendering API	21-15

## 22 開發版面配置

---

關於版面配置	22-1
搜尋引擎最佳化 (SEO)	22-3
瞭解 components.json 檔案與格式	22-4
自訂網站產生器中的工具列群組	22-8
限制槽位中的元件	22-11
將版面配置內容設為可編輯	22-13
建立區段版面配置	22-14
建立支援遲緩載入的區段版面配置	22-15
使用 API 開發自訂區段版面配置	22-16
開發內容版面配置	22-20
使用 Oracle Content Management 建立內容版面配置	22-21
將版面配置檢視傳送給內容版面配置	22-27
使用 API 產生網站詳細資訊頁面 URL	22-28
運用 Developer Cloud Service 在本機開發內容版面配置	22-29
建立 Developer Cloud Service 的內容版面配置	22-30
定義 RequireJS 模組	22-31
設定建構子函數參數	22-31
呈現內容版面配置	22-32
編輯 Mustache 樣板中的內容版面配置	22-33
新增動態 DOM 操控	22-33
在 design.css 檔案中定義樣式	22-33
取得參照項目	22-33
取得媒體 URL	22-34
引發觸發程式	22-34
以搜尋查詢方式瀏覽至搜尋頁面	22-34
擴充巨集及呈現 RTF 內容	22-35
連結至詳細資訊頁面	22-36
擴充內容清單查詢巨集	22-36
開發完善的內容版面配置	22-39
呈現內容項目	22-39
內容版面配置的資料結構標準化	22-40
建立範例部落格樣板	22-44
將內容版面配置對應新增至樣板	22-44
使用本機測試控制程式測試內容版面配置	22-45
使用本機測試控制程式進行測試	22-45
將含有內容版面配置的樣板匯入 Oracle Content Management	22-45

## 23 開發元件

---

關於元件	23-1
關於開發元件	23-2
建立元件	23-5
使用 Developer Cloud Service 開發自訂元件	23-6
開發 Oracle Content Management 自訂元件	23-6
開發您的自訂元件	23-7
撰寫並執行單元測試	23-8
最佳化元件 (精簡) 以提升效能	23-9
執行連續整合工作	23-10
開發多語言網站適用的可翻譯元件	23-10
建置含設定值面板的 H1 元件	23-11
建立新本機元件	23-11
建置基本 H1 元件	23-12
新增元件的 CSS	23-15
新增設定值面板以變更標題文字	23-16
更新主題以讓其他人挑選 H1 元件樣式	23-17
使用 HTML 樣板建立簡單元件	23-19
建立 HTML 元件	23-19
將使用者提供的資料新增至 HTML 元件	23-23
本機元件與遠端元件的比較	23-27
呈現元件設定值	23-28
本機元件實行	23-29
元件的樣式類別	23-30
如何設定內建元件的樣式	23-32
元件樣式設定基本資訊	23-32
元件特定樣式設定	23-34
設定元件特性	23-40
內嵌框架中呈現的元件	23-41
關於內嵌框架中呈現之元件的執行處理 ID 和結構	23-42
遠端元件安全性	23-44
註冊遠端元件	23-45
刪除元件	23-46
Sites SDK	23-46

## 24 自訂控制器檔案

---

關於控制器檔案	24-1
預設控制器檔案	24-1

修改預設 controller.html 檔案	24-2
關於 SCS 物件	24-3
SCS.sitePrefix	24-3
SCS.preInitRendering	24-3
SCS.getDeviceInfo	24-3
不應自訂的控制器檔案區段	24-4
使用權杖允許自訂控制器檔案可攜性	24-5
自訂控制器檔案範例	24-5
變更網站前置碼	24-6
自訂等待圖形	24-7
自訂 Favicon	24-9
自訂非 JavaScript 蒐集程式的 <noscript> 和 <meta> 標記	24-9
預先擷取 JavaScript 檔案	24-10
使用其他標記驗證網站擁有權	24-10
增強裝置偵測	24-11
使用權杖增強 controller.htm 可攜性	24-12

## 部分 V 使用其他工具開發網站

---

### 25 使用 OCE 工具程式進行開發

---

在本機機器上設定 OCE 工具程式	25-1
透過 npm 安裝相依項	25-2
使用 cec 命令行公用程式	25-2
使用本機測試控制程式進行測試	25-67
升級至 jQuery 3.5.x	25-67
使用 Developer Cloud Service 開發 Oracle Content Management	25-68
關於使用 Developer Cloud Service	25-69
登入 Oracle Content Management 的 Developer Cloud Service 主控台	25-69
在 Developer Cloud Service 中建立專案	25-70
使用 Oracle Content Management 樣板建立 Developer Cloud Service 專案	25-70
使用從 Oracle Content Management 下載的 OCE 工具程式在 Developer Cloud Service 中建立專案	25-70
將 OCE 工具程式新增至新 Git 儲存區域中的專案程式碼	25-70
在本機測試控制程式中測試自訂元件、樣板及內容版面配置	25-71
合併變更	25-71
使用 OCE 工具程式將變更從測試環境傳輸到生產環境	25-71
加密密碼	25-77
註冊伺服器	25-77

建立網站的使用狀況和權限報表	25-78
下載與上傳文件和資料夾	25-78
從樣板建立網站並保持相同的內容 GUID	25-79
從標準網站建立企業樣板	25-79
匯入及匯出分類	25-79
匯入及匯出建議	25-81
新增或移除集合內容	25-82
使用 OCE 工具程式開發自訂欄位編輯器	25-82
將網站從一個伺服器移轉或更新至另一個伺服器	25-87
傳輸無內容項目的網站	25-87
下載或上傳群組中網站的內容項目	25-88
使用 OCE 工具程式編製網站頁面索引	25-88
建立網站頁面文字的內容類型	25-89
使用 OCE 工具程式建立頁面索引內容項目	25-90
在 Oracle Content Management 中新增網站的內容搜尋	25-91
新增搜尋頁面至網站	25-91
新增搜尋欄位至主題	25-91
使用 OCE 工具程式編製多語言網站的索引	25-92
建立簡化的元件以便於元件開發	25-94
設定網站編譯服務	25-95
編譯網站以提昇網站頁面的程式實際執行效能	25-95
網站編譯總覽	25-96
與預先呈現互動	25-96
呈現控制器網站頁面	25-96
呈現編譯的網站頁面	25-96
樣板編譯	25-97
設定	25-97
編譯您的樣板	25-98
網站編譯	25-103
網站編譯先決條件	25-103
編譯網站	25-104
自訂編譯器	25-105
限制條件	25-105
除錯自訂編譯器	25-106
頁面版面配置編譯器	25-106
元件編譯器	25-107
內容版面配置編譯器	25-109
SCSCompileAPI	25-110
元件填入	25-111

發布	25-113
靜態網站傳遞優先順序	25-114
快取標頭	25-114
詳細資訊頁面	25-115
新增內容項目至通道	25-115
編譯行動裝置適用的網站	25-116
網站生命週期和編譯過的頁面	25-117
在 Oracle Content Management 伺服器中建立新網站或資產翻譯工作	25-118
使用語言服務提供者翻譯網站	25-120
使用 OCE 工具程式建立翻譯工作	25-121
列出翻譯工作	25-122
建立翻譯連線器	25-122
產生多語言網站的網站地圖	25-123
將翻譯工作送至語言服務提供者	25-124
將翻譯工作上傳至伺服器	25-125

## 部分 VI 附錄

---

### 26 教學課程：使用 Knockout 開發元件

---

透過 Knockout 開發元件的簡介和先決條件	26-1
步驟 1：建立元件	26-2
步驟 2：複查本機元件呈現的結構	26-3
步驟 3：複查本機元件設定值的結構	26-7
步驟 4：在元件中顯示新特性	26-10
步驟 5：註冊觸發程式	26-11
步驟 6：引發觸發程式	26-12
步驟 7：註冊動作	26-15
步驟 8：執行動作	26-16
步驟 9：為元件的每個執行處理建立不同的標題	26-17
步驟 10：使用具有內嵌編輯的巢狀元件	26-18
步驟 11：支援不同的版面配置	26-20
步驟 12：定義自訂樣式	26-22
步驟 13：在內嵌框架中呈現元件	26-23
步驟 14：在內嵌框架中呈現元件時使用自訂樣式	26-26
步驟 15：與頁面還原和重做行為的整合	26-29
步驟 16：資產管理	26-29

## 27 Sites SDK 參考資料

---

SitesSDK.getProperty('componentAssets', callback)	27-3
SitesSDK.getProperty(propertyName, callback)	27-4
SitesSDK.getSiteProperty(propertyName, callback)	27-5
SitesSDK.setProperty('componentAssets', [assets])	27-6
SitesSDK.setProperty(propertyName, propertyValue)	27-7
SitesSDK.filePicker({options}, callback)	27-8
SitesSDK.openDocumentPicker(options)	27-9
SitesSDK.openAssetPicker(options)	27-9
SitesSDK.publish(messageType, payload)	27-9
SitesSDK.subscribe(messageType, callback)	27-10
SitesSDK.Utils.addSiteThemeDesign(cssUrl)	27-12
SitesSDK.Utils.Logger	27-13

## 28 疑難排解

---

我嘗試建立一個網站，但是沒有任何樣板可用	28-1
我無法刪除網站	28-1
我無法開啟網站樹狀結構或編輯頁面	28-1
頁面中未顯示已新增的元件	28-2
我的資料夾、檔案和對話元件無法運作	28-2
部分內容在頁面版面配置變更後消失不見	28-2
頁面中未顯示已上傳的新版影像	28-2
我使用其他服務新增了元件，但該元件無法運作	28-3
我的企業網站顯示警告	28-3

# 前言

本文件描述如何建立及管理體驗，包括網站、資產以及與 Oracle Content Management 關聯的所有結構和原則。

## 適用對象

本書的適用對象為想要使用 Web 瀏覽器、透過桌面同步應用程式或行動裝置處理檔案的 Oracle Cloud 使用者。

## 說明文件協助工具

如需有關 Oracle 對於協助工具的承諾資訊，請瀏覽 Oracle Accessibility Program 網站，網址為 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>。

### 取用 Oracle 支援

已經購買客戶支援的 Oracle 客戶可從 My Oracle Support 取得網路支援。如需相關資訊，請造訪 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info>；如果您在聽力上需要特殊服務，請造訪 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs>。

## 多元共融

Oracle 全面致力於推行多元共融。擁有多元的勞動力有助於提升思惟領導力與創新力，Oracle 對此相當重視。Oracle 主動打造對員工、客戶及合作夥伴有正向影響的共融文化，而將漠視他人感受的不當詞彙從 Oracle 產品與文件中移除，便是 Oracle 為此努力的其中一項任務。在 Oracle 產品和業界標準往前推進的同時，Oracle 也深切注意必須與客戶的現有技術保持相容，確保服務連續不間斷。由於受到這些技術因素的限制，移除不當詞彙需要時間、須與外部合作，是一項持續性的任務。

## 相關資源

如需詳細資訊，請參閱下列 Oracle 資源：

- 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*

## 印刷慣例

本文件使用下列印刷慣例。

印刷慣例	意義
<b>粗體</b>	粗體類型表示與動作相關的圖形使用者介面元素，或是以文字定義或在辭彙中定義的用語。
<i>斜體</i>	斜體類型表示書名、強調或您可以提供特定值的預留位置變數。
<b>等寬字型</b>	等寬字型類型表示區段中的命令、URL、範例中的程式碼、顯示在畫面上的文字，或是您輸入的文字。

# 第 I 部分

## 簡介

此部分提供有關運用 **Oracle Content Management** 建置網站的概念資訊，並包含易於使用的教學課程來建立第一個網站。其中包含下列章節：

- [開始建置網站](#)
- [建立您的第一個網站](#)

# 1

## Oracle Content Management 簡介

Oracle Content Management 是一種雲端內容中心，可駕馭全通路的内容管理，並加快服務體驗的傳遞。它提供強大的協作與工作流程管理功能，可簡化建立及交付內容的作業，並且改善客戶和員工的互動體驗。

Oracle Content Management 可協助您在任何裝置上迅速進行內部及外部協作，以核准內容並建立情境式體驗。具業務便利性的內建工具，讓建置優質內容的新 Web 體驗輕而易舉。您可以使用相同的内容平台和相同的流程，加速與所有相關人員的數位互動。技術與組織的瓶頸都已消除，再也沒有什麼能夠阻礙您建立互動式體驗。

### 備註：

Oracle Content Management 入門版提供有限的功能集。若要使用完整功能集，請升級至進階版。

## 存取 Oracle Content Management

授予您存取 Oracle Content Management 的權限後，您會收到一封歡迎使用電子郵件，內含執行處理 URL 和您的使用者名稱等相關詳細資訊。您將需要此資訊來登入服務，因此建議您保留供日後參考。

您可以透過不同方式與 Oracle Content Management 進行互動：

- 透過 Web 介面，就能從您常用的 Web 瀏覽器輕鬆存取。您可以管理雲端的内容、與其他人共用檔案和資料夾、開始並參與對話、建立網站 (如果允許的話) 等。
- 桌面應用程式可讓您將雲端與電腦之間的檔案和資料夾保持同步。您可以同步自己的檔案和與您共用的檔案，確保您存取的都是最新版本。
- Microsoft Office 附加元件可讓您直接從 Microsoft Word、Excel、PowerPoint 和 Outlook 存取 Oracle Content Management 功能。
- 只要透過手機或其他行動裝置上適用於 Android 和 iOS 的行動應用程式，您就能夠輕鬆存取。這些行動應用程式的外觀與操作方式就像 Web 瀏覽器中的服務一樣，因此可立即上手。您可以存取雲端内容、搜尋及排序您的檔案和資料夾、共用内容，以及使用對話。
- REST API 和 SDK 提供開發人員功能強大的工具，讓他們可以程式設計方式將 Oracle Content Management 功能併入 Web 應用程式和行動應用程式中。

## 瞭解角色

您可以存取的 Oracle Content Management 功能取決於已指定給您的角色。端視您的使用者角色而定，您將會見到不同的選項。標準使用者可以使用文件、對話和網站。企業使用者還可以存取資產。而開發人員則可以看到建置和自訂樣板、主題、元件及版面配置之類的網站項目選

項。管理員可以看見設定服務、將服務與其他商業應用程式整合，以及建立資產儲存區域的選項。

Oracle Content Management 提供不同類型的角色：

- **組織角色** — 您在本身組織中可決定需要執行哪些工作以及如何使用功能的角色。
- **使用者角色** — 使用者角色可控制您在 Oracle Content Management 中看到哪些功能。
- **資源角色 (權限)** — 您在使用資源 (例如文件、內容項目、網站或樣板) 時可以見到哪些內容及執行哪些作業，取決於與您共用該資源時指定給您的角色。

進一步瞭解...

## 管理資產

Oracle Content Management 提供強大的功能，可管理您所有的資產以用於不同的通道中，包括網站、行銷資料、電子郵件行銷活動、線上商店以及部落格。它針對資產提供中央內容中心，您可在此將所有的資產整理至儲存區域和集合中，並建立規則來定義如何使用資產以及在何處使用資產。

它還具備完整的管理與工作流程功能，可引導資產完成其建立與核准程序，並確保只有已授權的版本才可供使用。

標記和篩選資產相當簡單，因此您可以快速找到所需的資產，然後在允許的位置使用這些資產。而且，如果使用智慧內容功能，它將會自動標記並建議資產。

建立資產類型，以定義使用者建立資產時所需收集的資訊。*數位資產類型*定義您數位資產 (檔案、影像與視訊) 所需的自訂屬性。*內容類型*將不同的內容片段群組成可重複使用的單元。接著，使用者可根據這些資產類型建立數位資產和內容項目，如此便能在不同通道間保持一致的使用狀況和呈現方式。

進一步瞭解...

## 管理內容

有了 Oracle Content Management，在雲端中管理內容時，您只要在單一位置即可管理全部內容，而且可以從任何地點進行存取。

您可以將檔案群組在資料夾中並執行一般檔案管理作業 (複製、移動、刪除等)，作業與本機電腦的方式大致相同。由於所有檔案都位於雲端，因此您也可以透過行動裝置隨時隨地存取這些檔案。如果您安裝了桌面應用程式，所有的內容便能自動同步到本機電腦，因此您隨手即可取得最新版本的內容。

進一步瞭解...

## 協作內容

由於所有內容都在雲端，因此您可以輕鬆地與組織內外的人員共用檔案或資料夾以進行協作。與您共用內容的所有人員不論何時何地皆可存取最新資訊。您可以授予整個資料夾的存取權或提供特定項目連結。系統將會保留共用項目的所有存取記錄，以便您監督每個共用項目的存取方式和存取時間。

Oracle Content Management 中的對話可讓您以即時方式討論主題和張貼註解，與他人進行協作。您可以針對任何主題開始獨立對話，並視需要新增檔案。您也可以開始關於某個特定檔案、資料夾、資產或網站的對話，以取得快速簡便的意見。

與對話關聯的所有訊息、檔案和附註將會保留，以方便您追蹤及複查討論。而且對話都放在雲端，因此您也可以從行動裝置隨時隨地檢視並參與這些對話。

進一步瞭解...

## 建置網站

有了 Oracle Content Management，從概念到啟動網站的過程中，您可以迅速地建置及發布行銷與社群網站，以提供互動式線上體驗。此為完全整合的過程：內容、協作和創造性合併於單一編寫與發布環境中。

若要快速開始使用，請使用內建樣板、拖放式元件、範例頁面版面配置，以及網站主題，從預先定義的建置區塊來組合成一個網站。開發人員也可以建立自訂樣板、自訂主題或自訂元件，打造獨一無二的線上體驗。

只要將元件拖放至頁面上的指定槽位中，即可新增 YouTube 視訊、串流視訊、影像、大標題、段落、社群媒體連結，以及其他網站物件。只要按下一個按鈕，即可切換主題並為網站重新加上商標，在整個組織中提供最佳化且一致的外觀與感覺。

您可以準備及處理一或多個更新、在網站中預覽更新，然後當一切就緒後，只要按一下就能發布更新。

除了在「網站產生器」中建立及發布網站之外，Oracle Content Management 還支援使用 REST API、React JS、Node JS 和其他技術的開發無頭 (headless) 網站。

進一步瞭解...

## 整合與擴充 Oracle Content Management

Oracle Content Management 為 Oracle 平台即服務 (PaaS) 產品方案，能與其他 Oracle Cloud 服務緊密合作。

您可以內嵌 Web UI 到 Web 應用程式中，讓使用者可直接與內容進行互動。透過自訂動作，Application Integration Framework (AIF) 可用來將第三方服務與應用程式整合至 Oracle Content Management 介面。您也可以開發內容連線器，將已經在其他地方建立的內容導入 Oracle Content Management 中、集中管理內容，並且用於可跨多個通道的新體驗中。

您可以利用一組豐富的 REST API 和 SDK 來管理、傳遞及協作內容與網站，將 Oracle Content Management 功能併入 Web 應用程式中。

建立與雲端內容 SDK 及資產互動的從屬端應用程式。使用協作物件開發自訂整合或擷取資產以便需要時就能使用。無論是透過網站、內容傳遞網路 (CDN) 或行動應用程式，您都能存取及傳遞已針對每個通道最佳化的所有內容與資產。

進一步瞭解...

## 開始使用

為了協助您開始使用 Oracle Content Management，請瀏覽 [Oracle Help Center](#)，其中含有多項資源，包括 [文件](#)、[視訊](#)、[導覽](#) 和 [開發人員資訊](#)。

另外，若有需要的話，還有[支援](#)和[社群](#)可提供協助。

## 移轉至 Oracle Cloud Infrastructure

如果 Oracle Content Management 訂閱尚未在含有基礎架構主控台的 Oracle Cloud Infrastructure (OCI) 上執行，Oracle 建議您移轉至該原生 OCI 環境。這可確保您享有未來 Oracle 雲端平台的優點與進階功能。

移轉並不是自動的；您必須送出服務要求，才能起始移轉處理作業。

[進一步瞭解...](#)

# 2

## 開始建置網站

Oracle Content Management 是一種雲端內容中心，可駕馭全通路的内容管理，並加快服務體驗的傳遞。Oracle Content Management 可協助您在任何裝置上迅速進行內部及外部協作，以核准內容、管理數位資產並使用具業務便利性的內建工具建立情境式體驗。

- [Oracle Content Management 簡介](#)
- [開始建立網站](#)
- [瞭解網站建立程序](#)
- [瞭解網站開發](#)
- [瞭解網站治理](#)
- [建立您的第一個網站](#)



## 開始建立網站

所有具備適當權限的人都能使用 Oracle Content Management 建置網站。您並不需要使用任何專屬工具、程式碼或軟體。使用者介面是圖形介面，直觀且易於使用。

### 備註：

使用 Oracle Content Management 入門版，您只能使用 1 個網站，而且不能進行網站治理。如需完整功能集，請升級至 Oracle Content Management 進階版。

誰能夠建置網站取決於數個因素：

- 服務管理員是否已啟用網站建立功能。
- 服務管理員是否已啟用網站治理。請參閱[瞭解網站治理](#)。
- 網站建立功能是否僅限網站管理員使用 (在停用網站治理的情況下可供使用)。
- 網站管理員是否已將樣板設為可供使用。

建立網站時，是利用樣板來開始進行。樣板含有您開始建置網站所需的一切項目，包括網站程式碼架構、含有範例頁面和內容的預設網站、含有樣式設定的主題、資源 (例如影像)，甚至是自訂元件。請參閱[瞭解網站建立程序](#)。

如果您是建立企業網站，網站將會與儲存區域相關聯，而且必須要有已定義指定預設語言的本地化規定。在建立網站之前，必須要先建立儲存區域和本地化規定。您將網站所需的資產和文件儲存在儲存區域中，儲存區域規定則控管可對資產執行的動作。請參閱[瞭解資產儲存區域瞭解如何使用儲存區域的詳細資料](#)。

編輯網站時，可以建立新的更新項目，或是使用現有更新項目。在更新項目內，您可以編輯和新增內容、調整樣式設定值、新增和刪除頁面、變更頁面版面配置，以及組織頁面。請參閱[瞭解網站產生器頁面](#)瞭解您可以使用編輯器進行哪些作業。

您不需要一次備妥所有更新項目。您和團隊成員可以同時或個別處理多個更新項目。例如，您可能正在處理具有每週新聞的更新項目，而另一個團隊成員正在新增即將舉行的銷售會議頁面。您可以依需要的頻率編輯、複查和儲存更新項目的變更，並在任何時間將更新項目與基本網站合併。

當您就緒時，就可以啟動您的網站。只要按一下，就可以將網站發布至 Web。簡言之，從概念到啟動網站的過程就是這樣簡單。

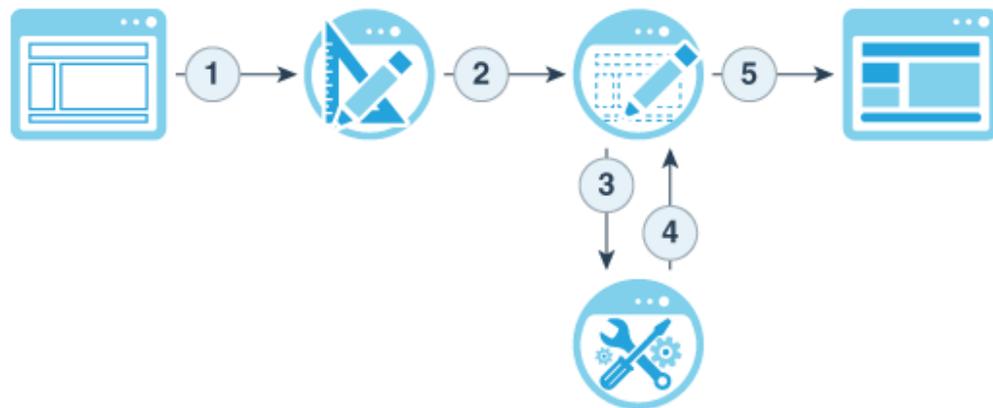
## 瞭解網站建立程序

就讓我們來瞭解建立並發布網站的整個過程。

在您開始之前，您的服務管理員應該已啟用網站建立功能、設定可建立網站的人員，以及安裝並提供將供您用來建立網站的樣板。服務管理員至少必須授予您樣板的「下載者」權限。請參閱[瞭解使用者角色](#)、[開始建立網站](#)以及[瞭解網站治理](#)。

如果您要建立企業網站(可使用資產和多語言網站)，則網站將會與儲存區域建立關聯。您將網站所需的資產和文件儲存在儲存區域中，儲存區域規定則控管可對資產執行的動作。儲存區域必須在您建立企業網站之前先建立。請參閱[瞭解資產儲存區域](#)。

以下為網站建立程序總覽，並且提供詳細資訊的連結。



- 1. 選取樣板並為網站命名：**如果已啟用網站治理且您的網站需要核准，您就必須等待核准之後，才能完成下一個步驟。網站會保持離線狀態，直到您準備好發布它為止。
- 2. 建立網站：**建立可以進行自訂的空白架構。
- 3. 建立新的更新項目，或是使用現有更新項目：**更新項目是目前基本網站的具名變更集合。每次在編輯器中檢視或編輯網站時，都會使用更新項目。請參閱[瞭解網站更新](#)。
- 4. 在編輯器中開啟網站：**使用拖放式編輯器，將頁面和內容新增至網站。請參閱[瞭解網站更新](#)。
  - **將頁面新增至網站：**選取預先定義的版面配置以快速定義頁面類型。請參閱[新增頁面](#)。

- **新增元件至頁面**：將文字、影像、文件和其他項目拖放至頁面。請參閱[使用內建元件](#)。
  - **變更元件的內容或特性**：微調元件的間距、對齊方式和其他特性。請參閱[使用樣式和格式](#)。
  - **選擇性地切換頁面所用的版面配置**：可快速變更頁面設計。請參閱[變更頁面版面配置](#)。
  - **選擇性地切換網站所用的主題**：快速變更整個網站的外觀與感覺。請參閱[在網站中使用樣板和主題](#)。
  - **設定搜尋引擎特性**：幫助搜尋引擎找到您的網站，以增加流量。請參閱[設定搜尋引擎特性](#)。
5. **套用網站更新項目**：使用更新項目中的變更來更新網站。請參閱[發布網站變更](#)。
- **與其他團隊成員共用網站**：與指定的使用者共用網站，並為所有使用者指定角色，以決定他們可對網站執行的作業。請參閱[瞭解網站安全性](#)以瞭解共用的相關資訊。
  - **設定網站安全性**：選擇可存取已發布網站的使用者。請參閱[變更網站安全性](#)。
  - **發布網站**：將變更發布至網站。請參閱[發布網站變更](#)。
  - **讓網站上線**：依據您設定的網站安全性將網站提供給使用者使用。請參閱[讓網站上線或離線](#)。

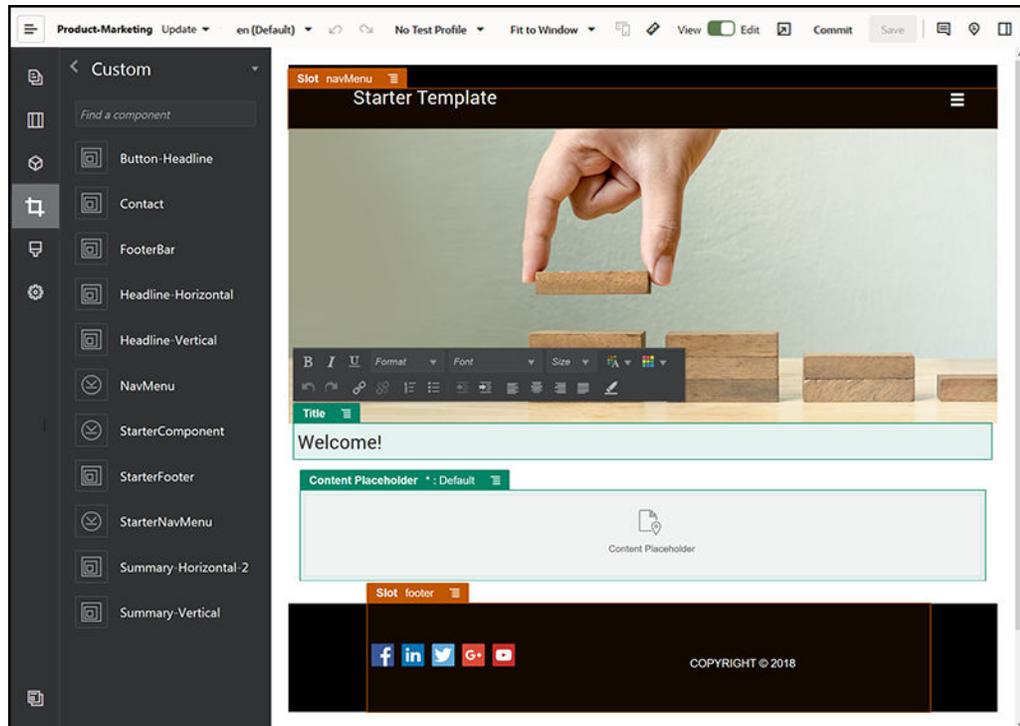
## 瞭解網站開發

若您是網站開發人員，您可以定義網站建立者用來建置網站的架構，例如：

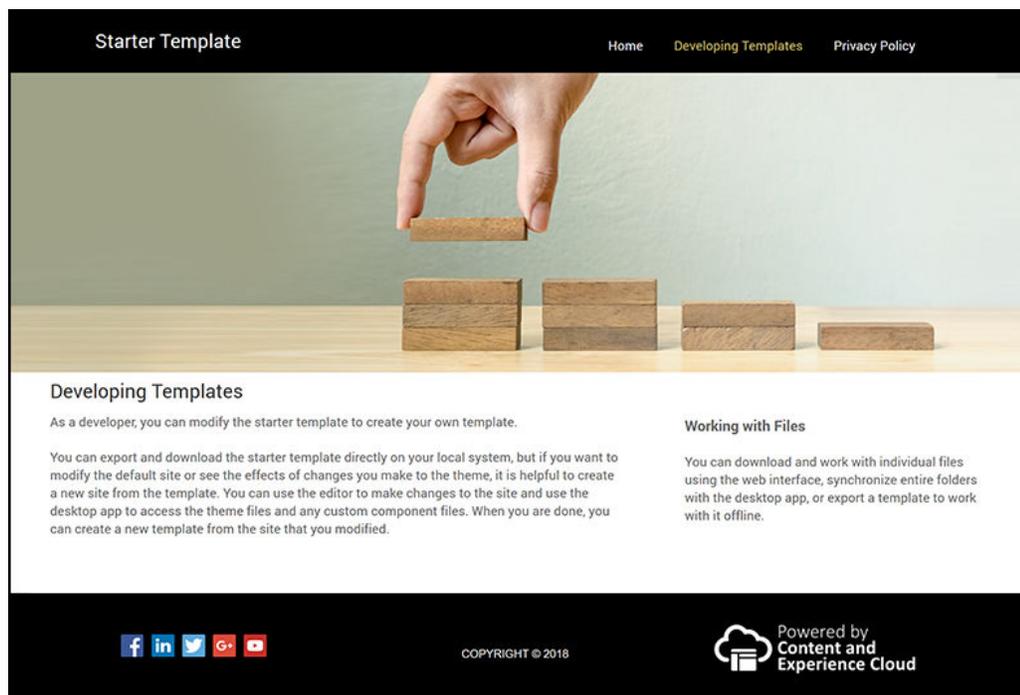
- **標準樣板**，當作網站的架構使用，根據主題含有範例頁面和內容、自訂元件，以及開始建置網站所需的其他資源
- **主題**，用來定義網站的整體外觀與感覺，包含標誌、樣式表、組態檔以及定義網站導覽的背景程式碼
- **自訂元件**，用來將特定類型的內容新增至網站頁面，可讓您使用任何選擇的頁面技術，開發可內嵌於網站頁面內的複合元素

將網站呈現方式與網站內容做區隔，您可以確保任何建立的網站都會維護您組織的標準和商標，而且可讓內容建立者專注於內容，更有效且更有效率地進行開發和網站建立。

以下是內容建立者在將內容新增至網站時可使用的簡單樣板範例。請注意頁面上所配置的一般預留位置：



以下是使用簡單樣板所建置的網站，在新增內容之後預覽時的外觀範例。請注意如何使用特定內容和導覽元素來自訂預留位置：



此外，網站開發人員還可以建立和維護樣式表、透過 Web 介面或使用 OCE 工具程式建置網站以及設定 Oracle Content Management 與其他服務之間的整合。而且，就像任何其他員工一樣，他們也可以共用內容、開始或參與對話或是使用桌面或行動應用程式，來與其他人進行協作。

開發人員必須具備標準使用者或企業使用者角色，才能夠使用 Oracle Content Management。具有標準使用者角色的開發人員可以建立元件、主題和標準範本。具有企業使用者角色的開發人員也可以建立版面配置，以及儲存網站作為標準或企業樣板。

除了網站開發之外，Oracle Content Management 也可以用於無頭 (headless) 環境，作為雲端中功能強大且具有彈性的後端內容管理系統 (CMS)。

## 瞭解網站治理

網站治理除了便於業務使用者根據公司政策建立網站外，也讓網站管理員能輕鬆地集中控管及追蹤網站。

### 備註：

使用 Oracle Content Management 入門版，您只能使用 1 個網站，而且不能進行網站治理。如需完整功能集，請升級至 Oracle Content Management 進階版。

### 問題

業界目前面臨到一個問題。各家公司正經歷著體驗爆發。這些體驗必須能夠快速地建立並部署，而且即時掌握公司的品牌和訊息的傳遞。企業一般感受到開發及提供新的體驗都需要很長久的時間。若想要成功，企業需要有強大且能與時俱進的技術，才能夠迅速擴展以採用一致的方式建立及管理許多體驗，進而推動產品或品牌創新。您必須快速且有效率地提供新的體驗，不然就會錯失市場先機，也將無法吸引住客戶，更無法掌握最佳前景。

體驗爆發通常是歷經多年資訊孤立且訊息封閉所形成的結果，而且各部門、品牌和小組的 Web 內容管理 (WCM) 系統也都沒有受到中央監控。這造就了一個情況複雜的環境 — 上市緩慢、容易受到安全威脅，以及造成急速攀升且不可預測的成本。

缺乏一致的建置體驗流程，以及使用迥然相異的 WCM 技術混合，都會讓組織面臨巨大的安全風險。當多個相關人員都有其各自的網站小問題時，核心職責 (例如安全性) 就會支離破碎，因而企業也就很容易受到攻擊。這些體驗全都是從不同的 WCM 系統搭配不同的 Web 應用程式，再加上 IT 基礎架構難以確保一切都獲得妥善保護而來。

此外，維護多個高價位的商業 WCM 系統也會造成重複的成本支出 — 每一年的授權續約費以及內部或外部夥伴的支援成本。

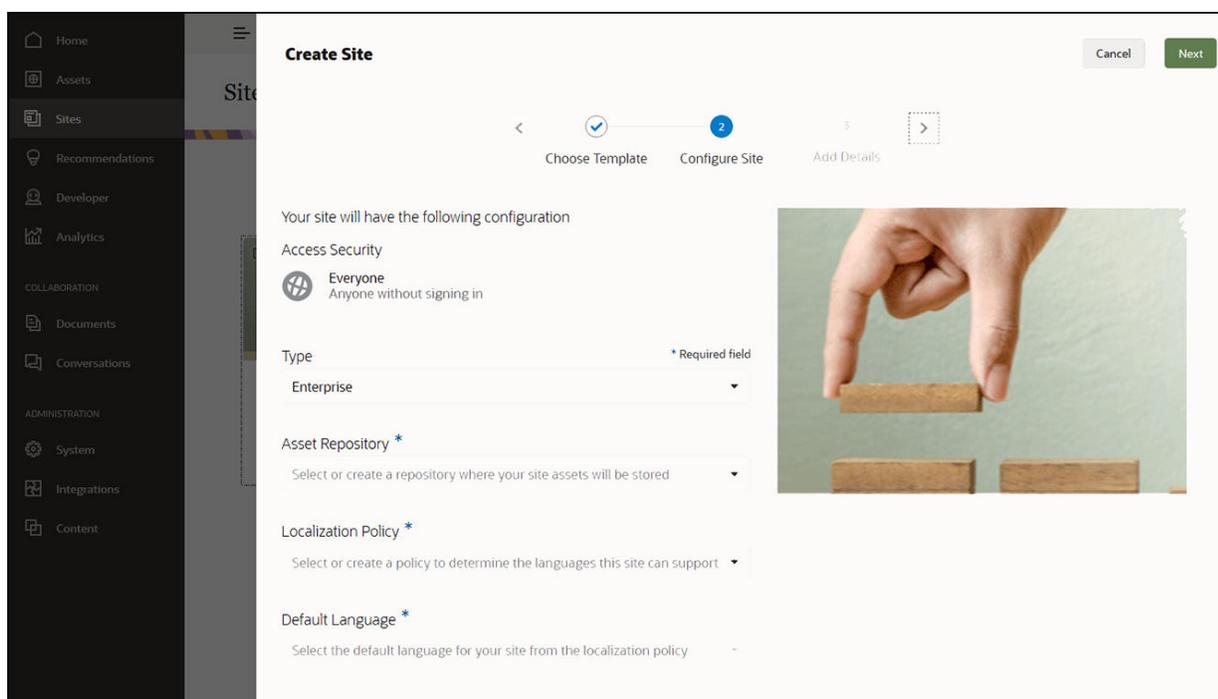
這造成各方面的限制：

- 業務方面的限制：
  - 沒有自助服務，需仰賴 IT 或昂貴的外包
  - 必須藉由技術協助才能進行更新
  - 沒有具業務便利性的工具可管理工作
  - 無法從單一中心瞭解所有的體驗與活動
- IT 方面的限制：
  - 每一項體驗都需要 IT 資源
  - 缺乏對業務所建立體驗的治理，包括未監管的外包體驗
  - 使用片段化的堆疊建置體驗

- 必須管理及部署數以百計體驗的內容與版面配置變更
- 使用者方面的限制：
  - 訊息傳遞通道不一致
  - 資訊過時
  - 效能低落，因而導致通道棄用

### 解決方案

Oracle Content Management 的治理功能除了可簡化並加快業務使用者的體驗傳遞外，IT 部門也可輕鬆地集中控管及追蹤體驗，並且能夠完全管理整個體驗生命週期，降低公司對所需之每項新體驗的建立與維護成本。治理功能內建於 Oracle Content Management 核心，而且只需要由管理員啟用即可。請參閱 *Administering Oracle Content Management* 中的設定網站和資產設定值。



### 功能

- **已核准樣板目錄** — IT 開發人員可以在樣板目錄中提供一組網站樣板，以滿足不同業務別的需求。它們可以限制可用的樣板以及可使用這些樣板的使用者。他們可以套用樣板規定，以指定新網站必須遵守的安全型態，以及新網站是否需要核准。如此除了可以透過既快速又簡單的方式要求新網站外，並且可同時確保讓業務使用者遵守品牌與安全準則。請參閱 [變更樣板狀態或對象](#) 和 [變更樣板規定](#)。
- **簡化要求、核准與佈建** — 讓業務使用者可以快速要求獲得必要核准的新網站並自動佈建。請參閱 [建立網站](#) 和 [管理網站要求](#)。
- **網站管理** — IT 部門可以從單一位置管理所有網站，完全與誰建立及部署網站無關。IT 使用者可以監督網站狀態，以及變更任何已部署網站的狀態。請參閱 [管理網站](#) 和 [網站設定值](#)。

## 優點

使用治理功能後，各項新體驗的建立與維護成本都可因此降低。

- 業務上的優點：
  - 不需依賴 IT 即可迅速佈建
  - 將內容與設計分開，讓內容可重複使用
  - 從單一主控台管理全域的體驗、使用者以及許可權
  - 透過報表深入瞭解體驗作業
- IT 方面的優點：
  - 簡化導入 (雲端原生解決方案)
  - 可自動擴展以因應成長和季節性高峰
  - 治理體驗，確保體驗安全無虞且符合公司的商標與規範標準
- 使用者方面的優點：
  - 訊息傳遞通道一致
  - 獲得最新資訊
  - 效能達到最佳
  - 體驗安全無虞

## 程序

以下是使用網站治理來建立和管理網站時所涉及的步驟：

1. 管理員啟用治理功能。請參閱 *Administering Oracle Content Management* 中的設定網站和資產設定值。
2. 網站管理員將已核准的樣板提供給使用者用於建立網站。請參閱[變更樣板狀態或對象](#)和[變更樣板規定](#)。
3. 使用者從已核准的樣板建立網站要求。請參閱[建立網站](#)。
4. 網站管理員核准網站要求並自動建立網站。網站管理員也可以拒絕網站要求，並提供附註說明拒絕原因，網站建立者則可以更正問題，然後重新送出要求。請參閱[管理網站要求](#)。
5. 網站建立者繼續進行網站建立處理作業、編輯及發布網站。請參閱[瞭解網站建立程序](#)。
6. 進行中的網站管理與更新都能在網站頁面中執行。網站管理員可以在網站頁面上看見所有網站；其他使用者則可以看見他們所建立或自己是成員的網站。請參閱[管理網站和網站設定值](#)。

# 3

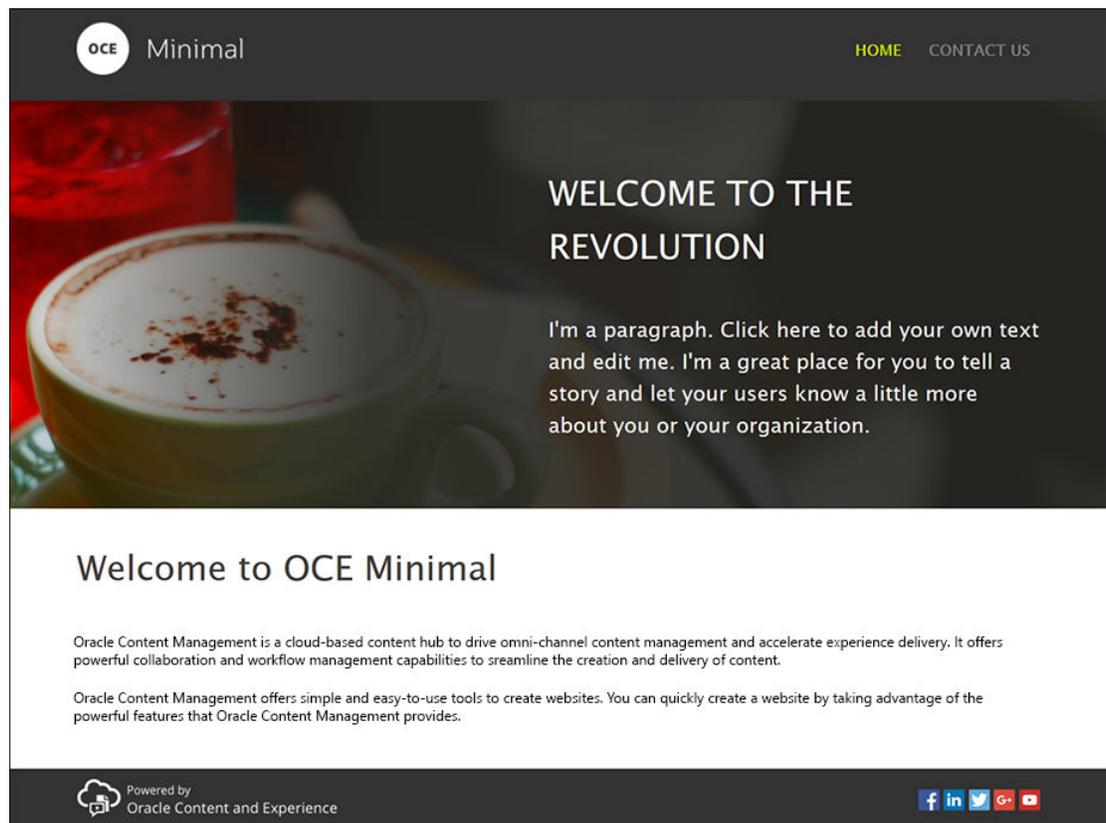
## 建立您的第一個網站

使用 Oracle Content Management 可輕鬆又快速地建置您的第一個網站。

在本教學課程中，您將瞭解如何利用 Oracle Content Management 所提供的內建 **Blank-Template** 樣板設定及發布網站。您將瞭解透過「網站產生器」建置第一個網站有多麼的簡單，完全不需要任何額外的編碼作業。

本教學課程包含三個基本步驟：

1. 設定環境
2. 設定網站
3. 發布網站



在開始之前，讓我們先確認一些事項。

## 開始之前

若要開始使用，您需要使用下列應用程式角色來存取 Oracle Content Management 執行處理：

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

若您未具備上述角色，請要求服務管理員將這些角色指定給您。

取得 Oracle Content Management 執行處理的存取權之後，請以內容管理員身分登入。

[開始使用](#)。

## 步驟 1：設定環境

首先，您必須設定環境，方法為匯入基本導覽功能表自訂元件並上傳範例影像。

因此，請下載資產套件 **OCESamplesAssetPack.zip**，網址為 <https://www.oracle.com/middleware/technologies/content-experience-downloads.html>。在下載的資產套件中，找到 **OCECreateYourFirstSite\_data.zip** 檔案，其中包含下列兩個壓縮檔：

- **Minimal-NavMenu.zip**：包含基本導覽功能表自訂元件。
- **Minimal-Images.zip**：包含網站中使用的範例影像。

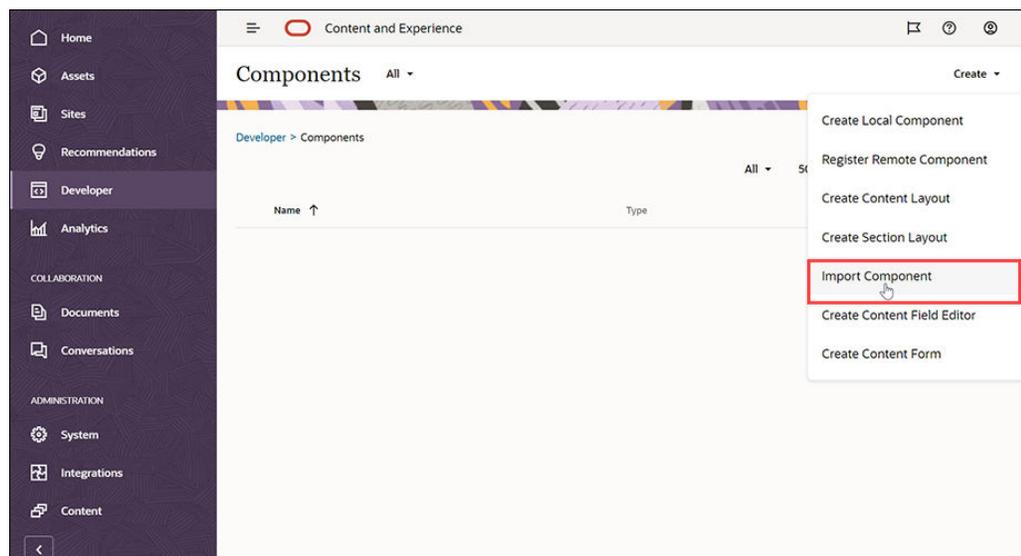
開始設定環境：

1. [匯入基本導覽功能表自訂元件](#)
2. [發布基本導覽功能表自訂元件](#)
3. [新增範例影像](#)

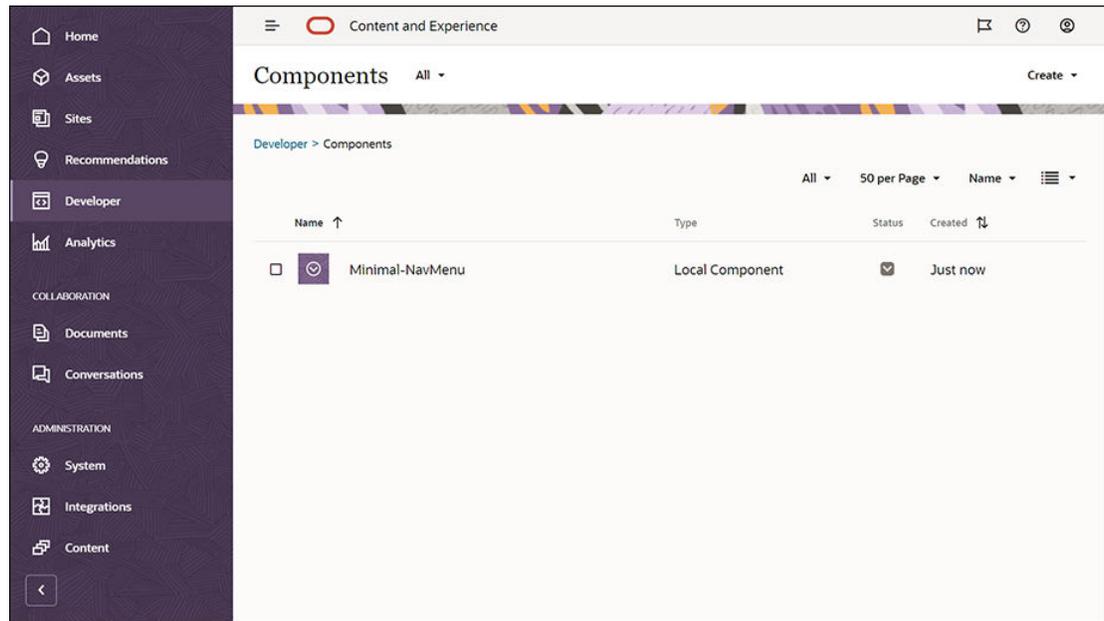
## 匯入基本導覽功能表自訂元件

登入 Oracle Content Management Web 介面後，按一下左側導覽功能表中的**開發人員**，然後選取**檢視所有元件**。若未見到**開發人員**選項，代表您沒有必要的使用者**角色**。

在「元件」頁面上，按一下**建立**，然後選擇**匯入元件**。上傳 **Minimal-NavMenu.zip** 檔案 (可從 **OCECreateYourFirstSite\_data.zip** 檔中取得)，然後將它選取。



基本導覽功能表元件便會列在「元件」頁面上。

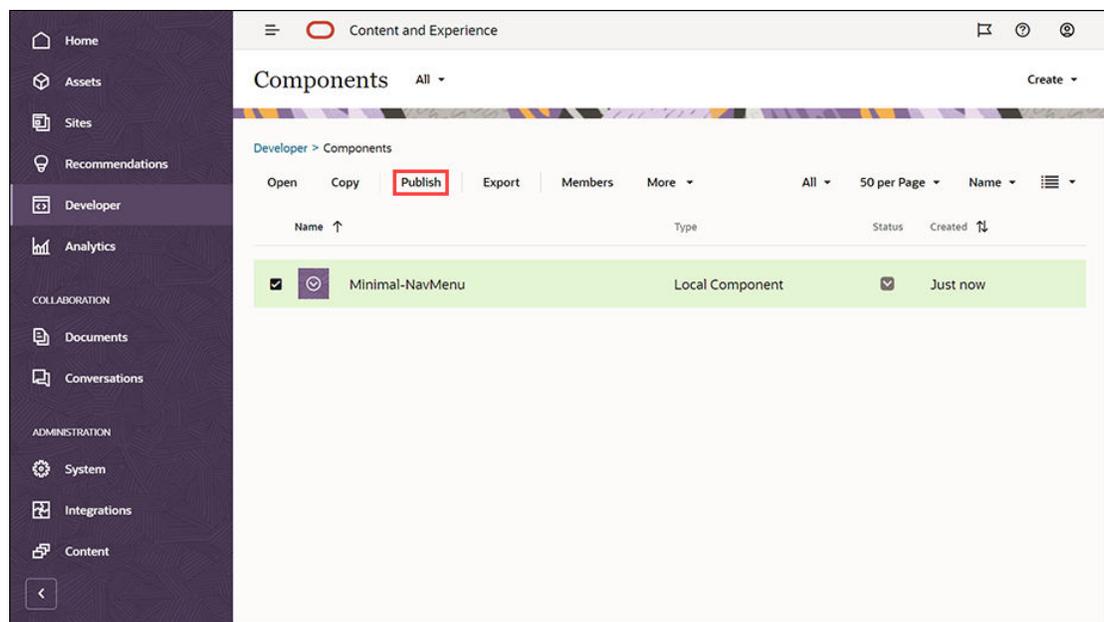


下一步：發布基本導覽功能表自訂元件

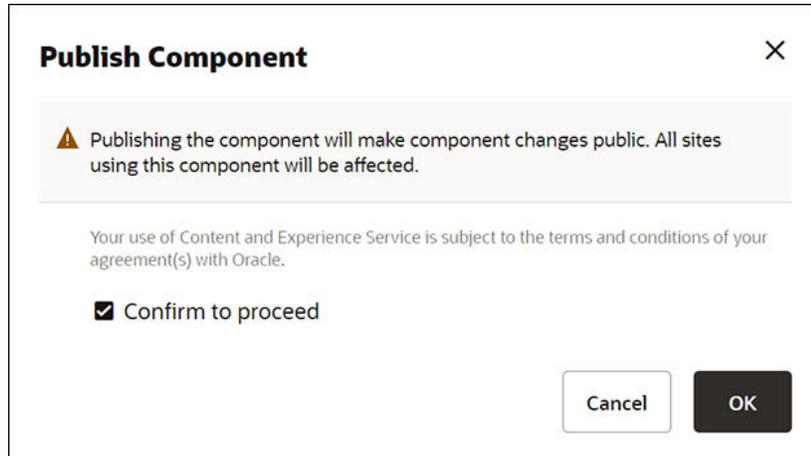
## 發布基本導覽功能表自訂元件

現在，您需要發布已匯入的基本導覽功能表元件。

在「元件」頁面上選取基本導覽功能表元件，然後從功能表列或滑鼠右鍵功能表中按一下**發布**。



在「發布元件」對話方塊中，選擇**確認繼續進行**並按一下**確定**。



基本導覽功能表元件發布之後，頁面頂端便會顯示一則通知，指出該元件已發布。

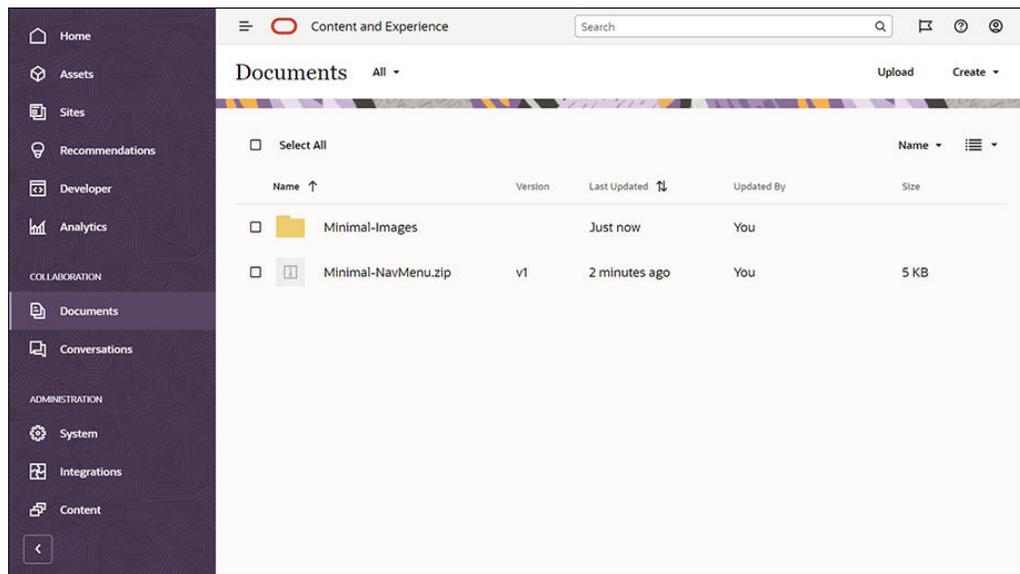
下一步：新增範例影像

## 新增範例影像

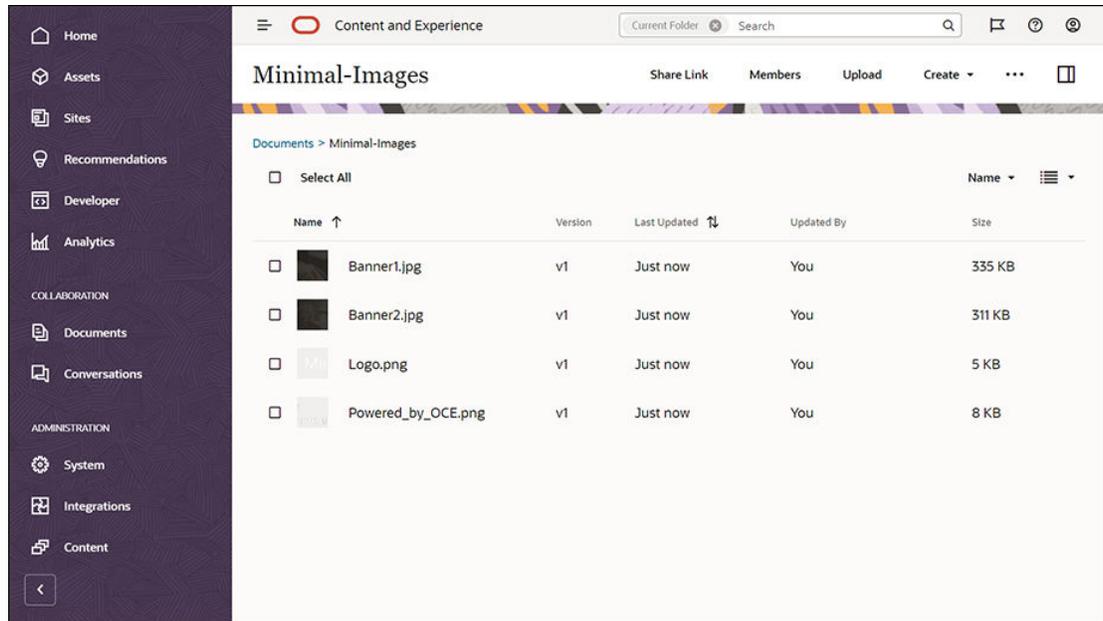
現在，您需要將範例影像 (將會在所建立的網站中使用) 新增至**文件**。

將 **Minimal-Images.zip** 檔案 (可從 [OCECreateYourFirstSite\\_data.zip](#) 檔中取得) 的內容擷取至本機電腦上名為**基本影像**的資料夾。

在左側導覽功能表中，依序按一下**文件**和**建立**，建立名為**基本影像**的資料夾。



將本機電腦上**基本影像**資料夾中的所有影像上傳到 Oracle Content Management 中新建立的**基本影像**資料夾。



 **備註：**

樣板和網站中所使用的是 Shutterstock 影像。若要重複使用這些影像，必須經過 Shutterstock 授權。您也可以將這些影像取代為自己的影像。

下一步：設定網站

## 步驟 2：設定網站

現在您已具備建立網站所需的一切項目，您就可以開始設定網站了。

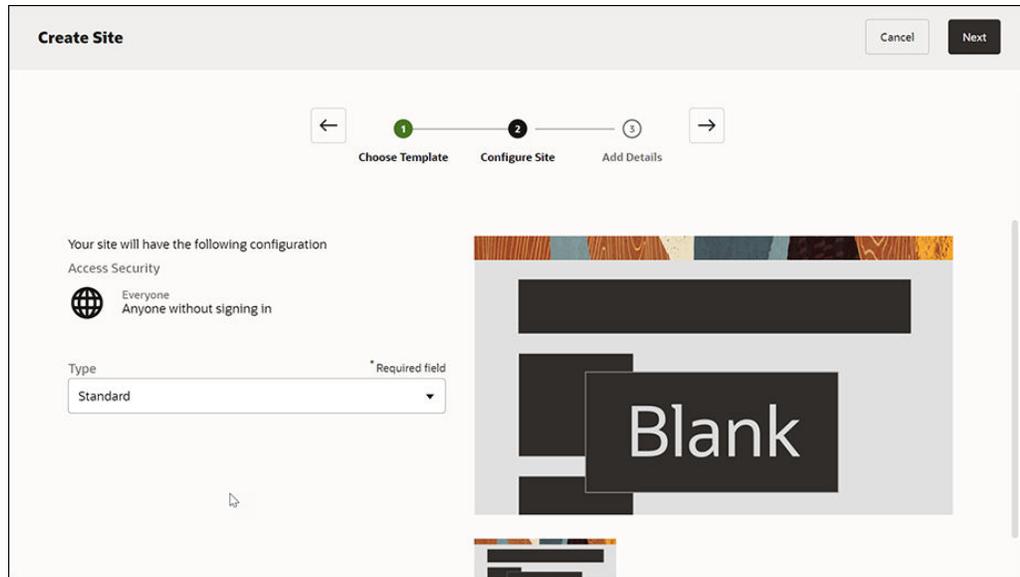
1. 建立網站
2. 編輯網站

### 建立網站

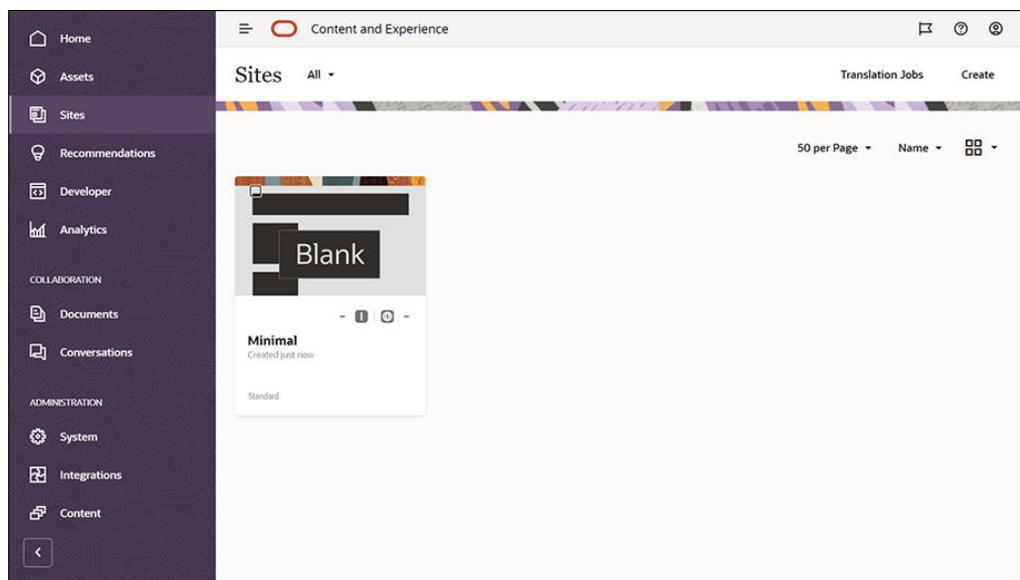
在左側導覽功能表中，依序按一下 **網站** 和 **建立**。選擇 **Blank-Template**，然後按 **下一步**。

如果您未見到 **Blank-Template**，請洽詢服務管理員。服務管理員必須讓內建樣板可供使用。服務管理員通常會在佈建服務後 **安裝內建樣板**，而內建樣板是由 Oracle Content Management 所提供。

確定您選擇建立標準網站，然後按一下 **下一步**。輸入網站的名稱 (例如 **基本**)。按一下 **完成**。



您的新網站已建立，並列在「網站」頁面上。

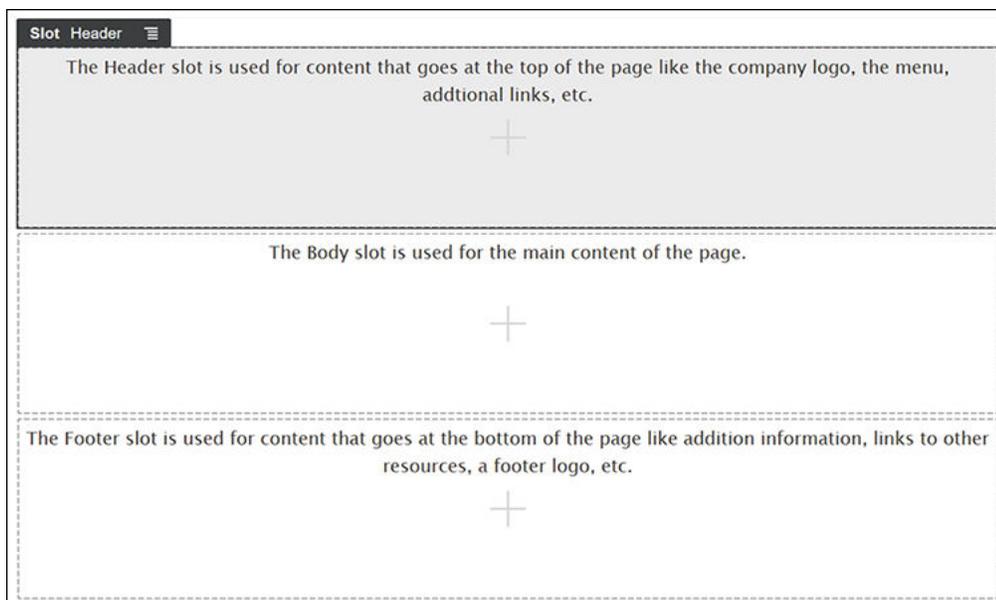


[下一步：編輯網站](#)

## 編輯網站

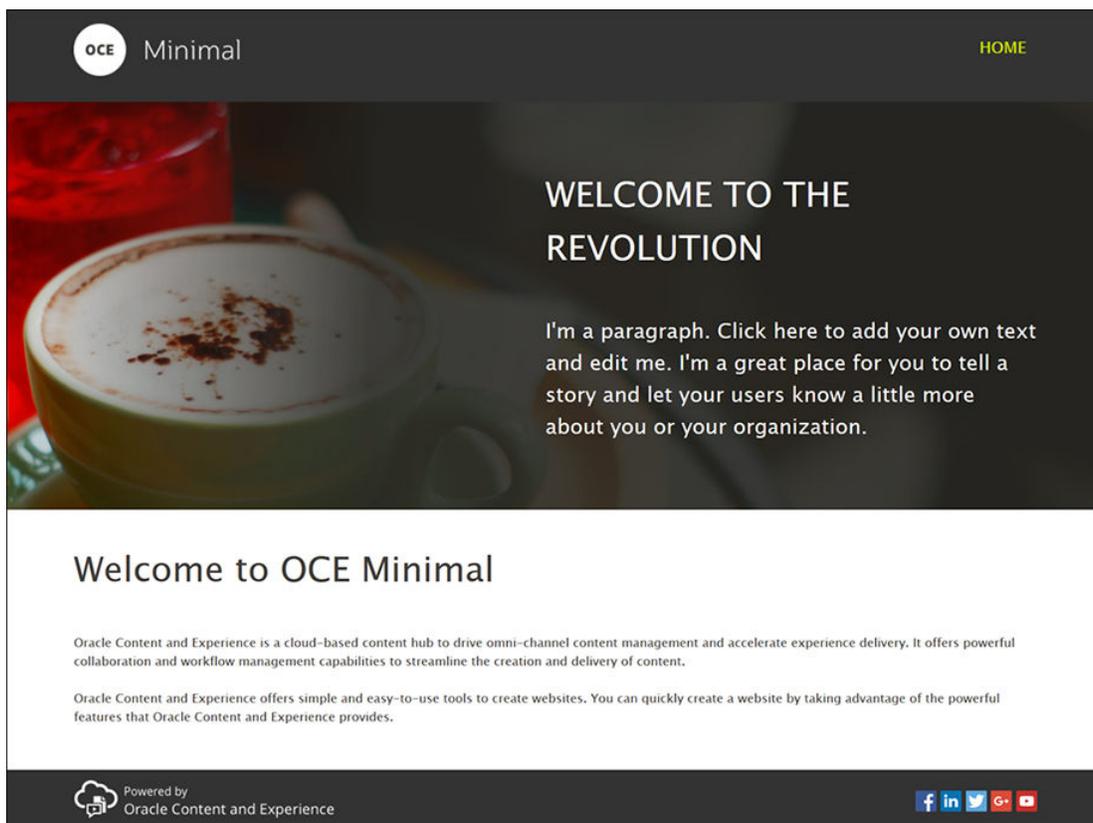
選取「網站產生器」中新建立的網站，然後從功能表列或滑鼠右鍵功能表中選擇**開啟**來開啟該網站。在「網站產生器」中，將開關  設為**編輯**模式。輸入更新的名稱，然後按一下**確定**。

在編輯模式中，您會看見網站包含三個槽位，這些是頁面上的可用區域 (視頁面版面配置而定)。將滑鼠游標停駐在頁面上的每個 + 號，即可檢視這些槽位 (例如標頭、主體及頁尾)。



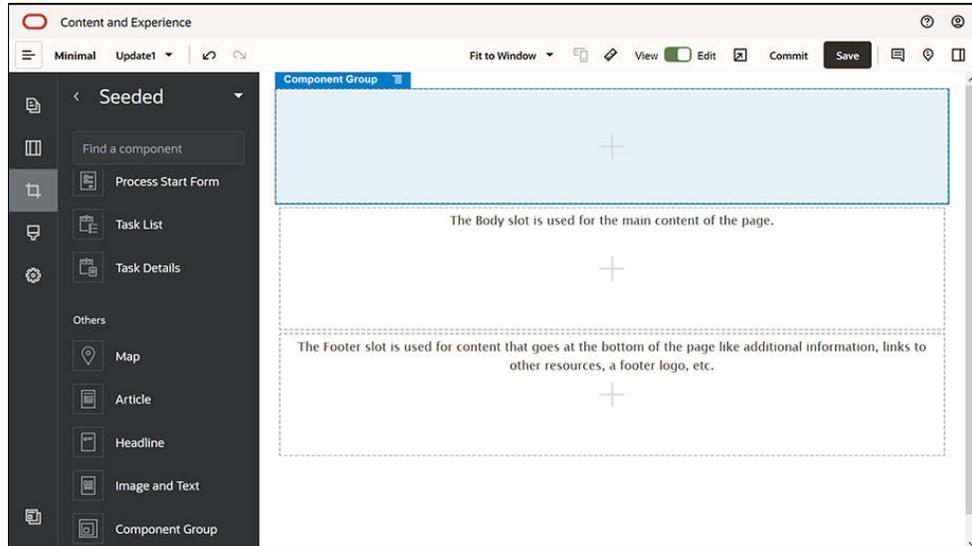
「標頭」槽位通常用來顯示公司的標誌、導覽功能表等等。「主體」槽位用於顯示頁面的主內容，「頁尾」槽位則顯示著作權資訊、社群媒體連結以及所有其他資訊。

首先，讓我們來建置首頁。這是首頁完成後的外觀：

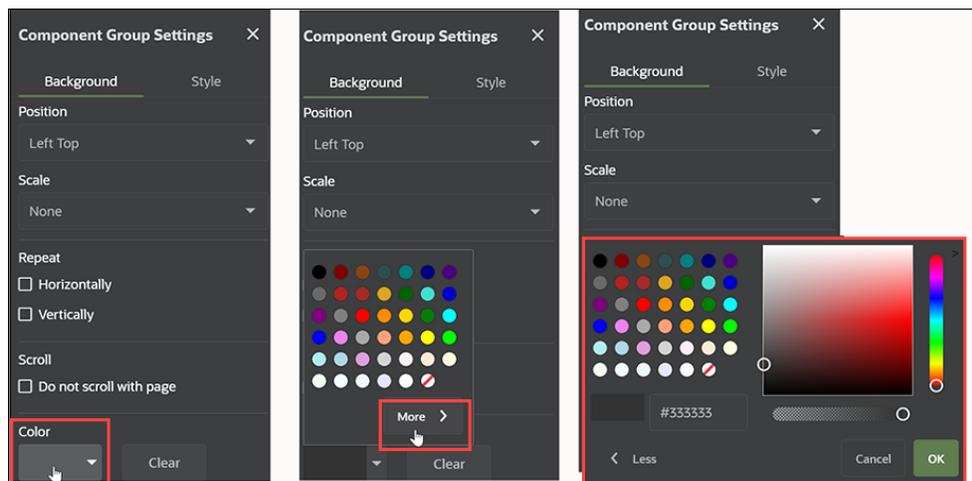


使用內建元件來填入「標頭」槽位：

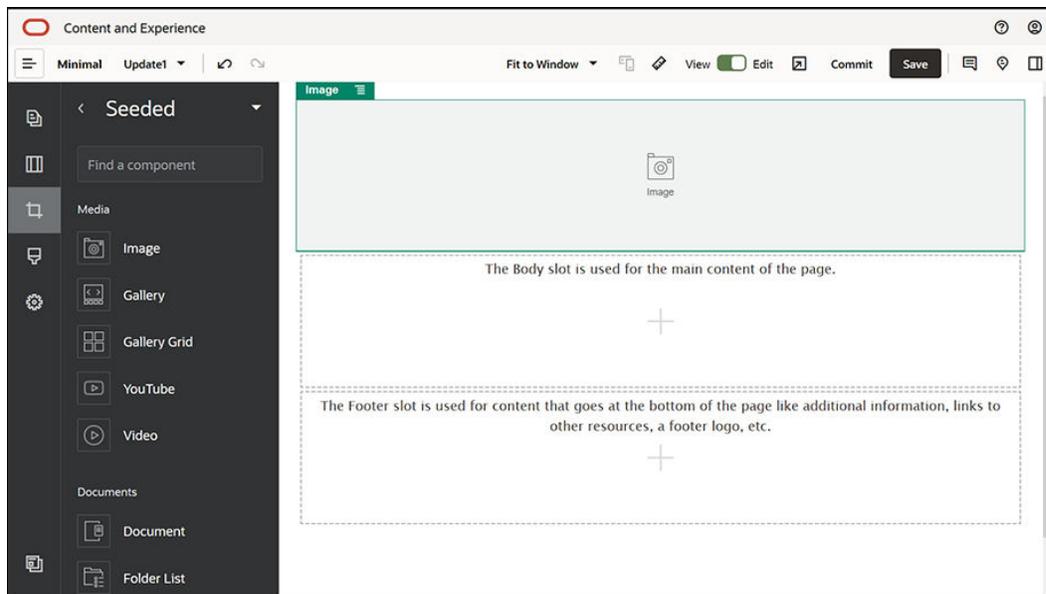
1. 在左側提要欄位中，依序按一下  和 **內建**，即可顯示 Oracle Content Management 提供的內建元件清單。
2. 在左側提要欄位中，尋找名為「元件群組」的內建元件。將此元件拖放至「標頭」槽位中。



3. 按一下元件群組的功能表圖示 ，然後按一下 **設定值**。在設定值中，請按一下 **色彩** 下拉式清單 (顯示在設定值清單的底部)，然後按一下 **其他**。輸入 **#333333** 然後按一下 **確定**。

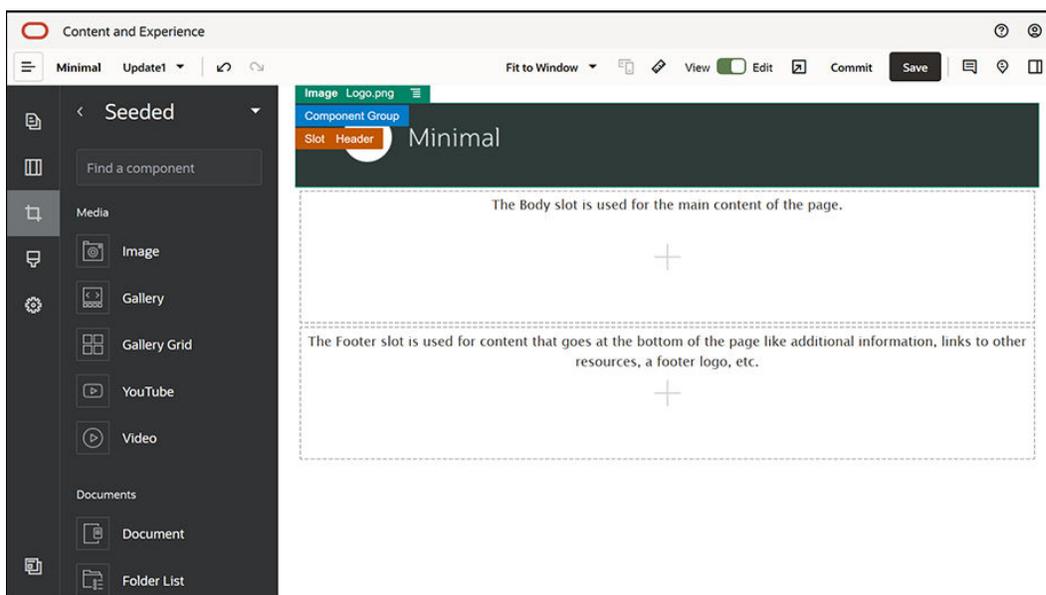


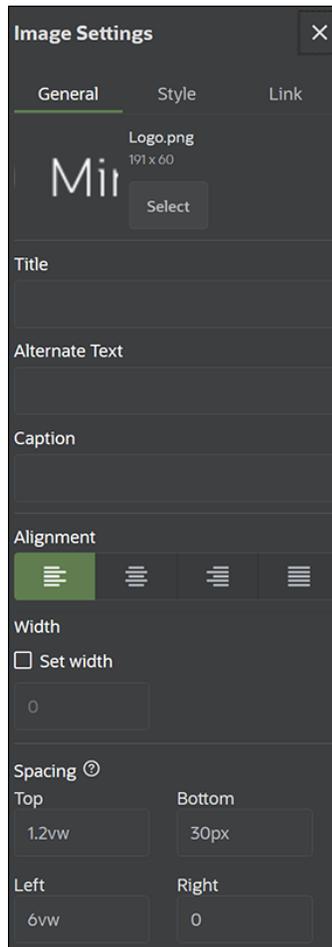
4. 將「影像」元件拖放至元件群組中。



5. 按一下「影像」元件的功能表圖示 ，然後按一下**設定值**。完成輸入**一般**頁籤中的設定值。

特性	值
選取	「基本影像」資料夾中的 Logo.png
對齊方式	靠左
寬度	取消選取 <b>設定寬度</b>
頂端	1.2vw
底端	30px
靠左	6vw
右側	0

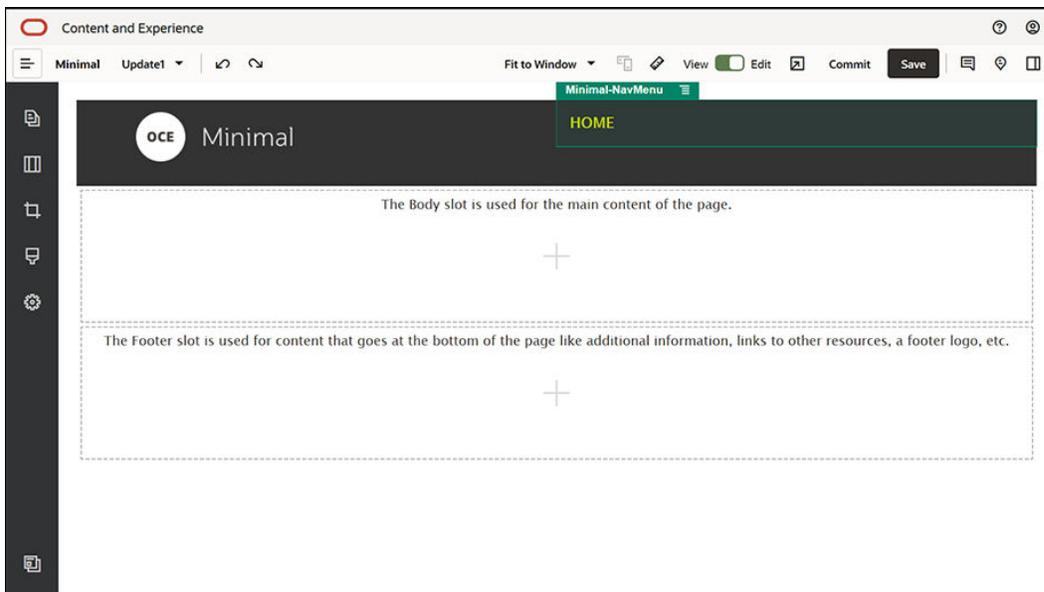




- 將此標誌影像連結至首頁。然後完成輸入**連結**頁籤中的影像元件設定值。

特性	值
選取連結類型	網站頁面
頁面	首頁
目標	在同一個視窗中開啟

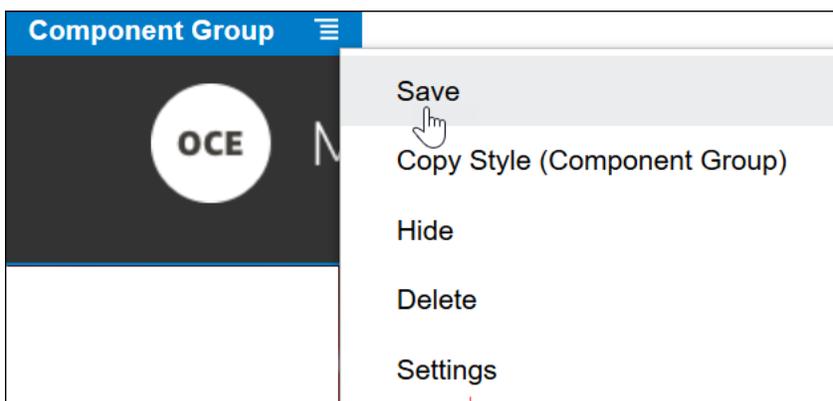
- 在左側提要欄位中，依序按一下  和**自訂**，即可顯示自訂元件清單。
- 現在，您可以使用基本導覽功能表自訂元件，將導覽功能表新增至首頁。將基本導覽功能表元件拖放至元件群組中「影像」元件的右側。按一下基本導覽功能表元件的標題，確定其父項就是您稍早所新增的元件群組。對於查看任一元件在網頁結構中的所在位置，這是很有用的方法。



然後完成一般頁籤中此元件的設定值。

特性	值
對齊方式	右側
頂端	1.2vw
底端	0
靠左	0
右側	6vw

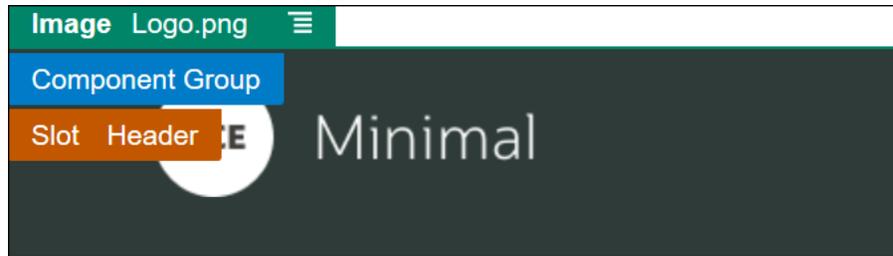
- 現在，標頭已就緒。請將此元件群組另存為自訂元件群組，以便之後能夠將它用於其他網站頁面中。依序按一下元件群組的標題和功能表圖示 ，然後按一下**儲存**。在「儲存元件群組」對話方塊的**名稱**欄位中輸入「基本標頭」，然後按一下**儲存**。



 **提示：**

按一下元件群組後，如果顯示的是「影像」或「基本導覽功能表」元件標題，而不是元件群組標題，請再按一下「影像」或「基本導覽功能表」元件標題，就會顯示元件群組的標題。現在，您可以按一下元件群組的標題並執行以上步驟。

以下圖像顯示「標頭」槽位中「影像」元件的父項結構：

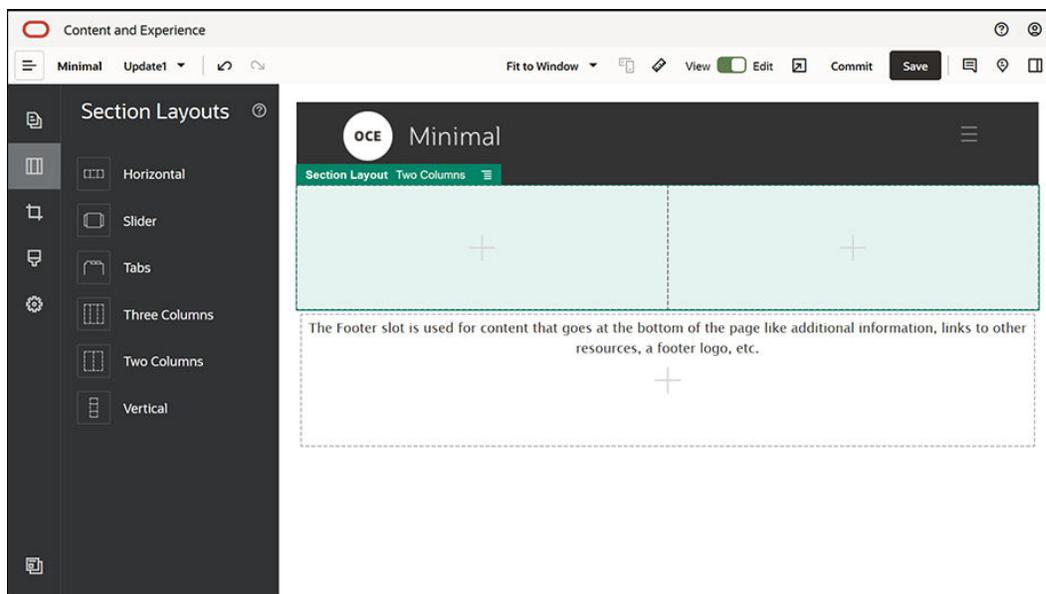


10. 按一下「網站產生器」右上方的**儲存**，以儲存您所做的變更。「標頭」槽位現在看起來應該像下圖這樣：



繼續設定「主體」槽位：

1. 在左側提要欄位中，依序按一下  和**內建**。
2. 將左側提要欄位中的「元件群組」拖放至「主體」槽位中。此元件群組與元件 (將新增至元件群組) 將會用來建立橫幅。
3. 在左側提要欄位中，按一下 。
4. 將「兩欄」區段版面配置拖放至元件群組中。



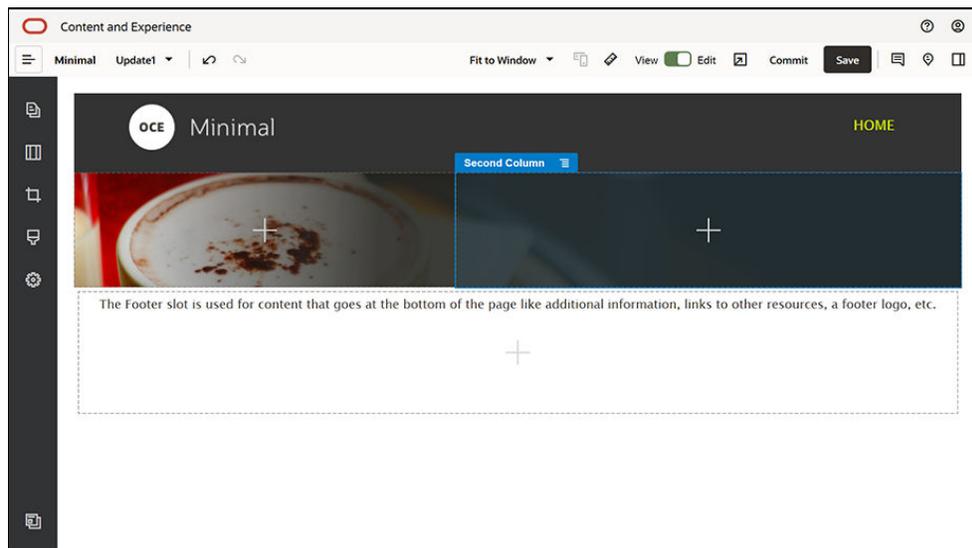
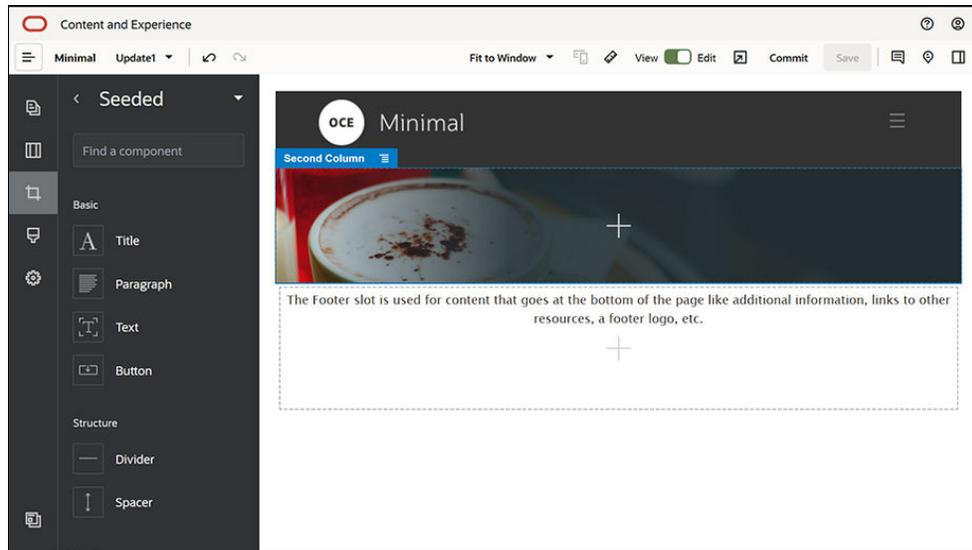
5. 然後完成輸入**一般**頁籤中的區段版面配置設定值。按一下**自訂設定值**，指定下列設定值。

特性	值
第一個資料欄寬度 (%)	43
第二個資料欄寬度 (%)	57
響應式中斷點 (像素)	1,023
響應式行為	隱藏第一個資料欄

完成輸入**背景**頁籤中的設定值：

特性	值
影像	「基本影像」資料夾中的 <b>Banner1.jpg</b>
位置	中間置中
比例	延伸

6. 在左側提要欄位中，按一下  即可查看內建元件清單。
7. 將「標題」元件從內建元件清單拖放至「兩欄」版面配置的第二個資料欄中。



- 按一下「標題」元件並輸入「歡迎加入改革行列」。選取文字，並透過文字編輯器將其字型色彩設為**白色**。然後完成輸入**一般**頁籤中的標題元件設定值。

特性	值
頂端	6vw
底端	1.8vw
靠左	6vw
右側	6vw

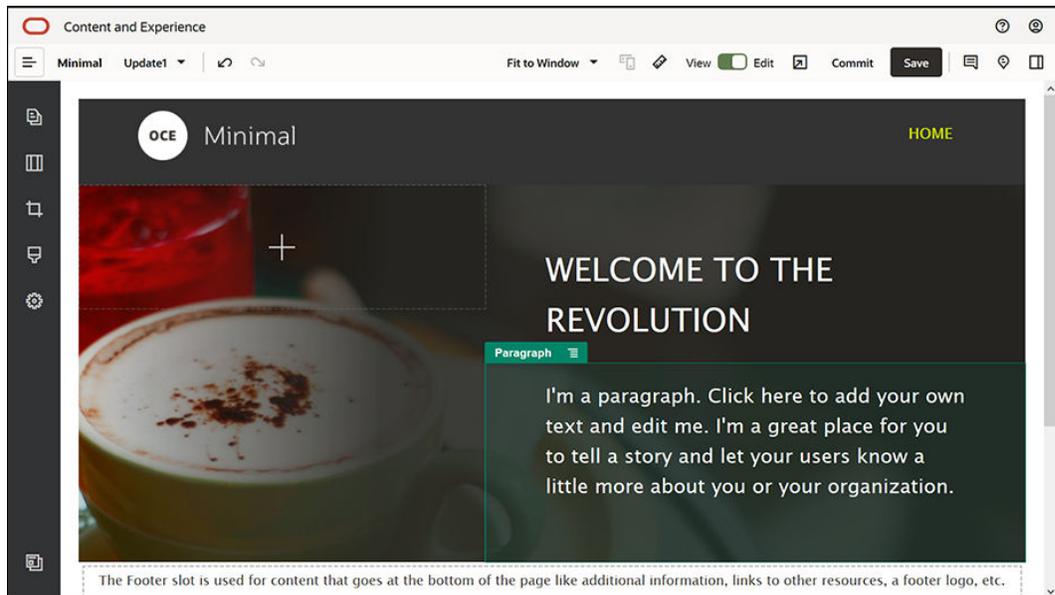
- 將「段落」元件從左側提要欄位中的內建元件清單，拖放至「兩欄」版面配置第二個資料欄中的「標題」元件下方。然後完成**一般**頁籤中此元件的設定值。

特性	值
頂端	1.8vw

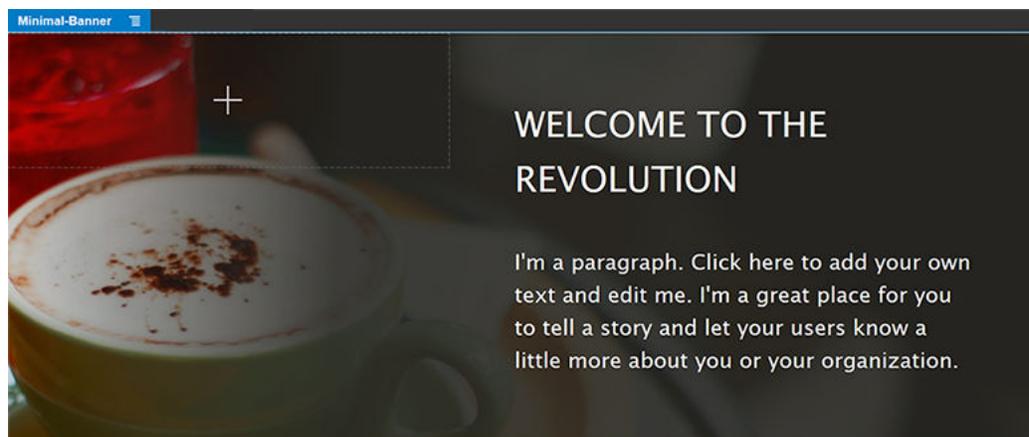
特性	值
底端	6vw
靠左	6vw
右側	6vw

10. 按一下「段落」元件並輸入下列文字：  
「我是一個段落。按一下此處即可加入您自己的文字並對我進行編輯。您可以善用這個絕佳位置來陳述一則案例，讓您的使用者對您或您的組織有更多的瞭解。」

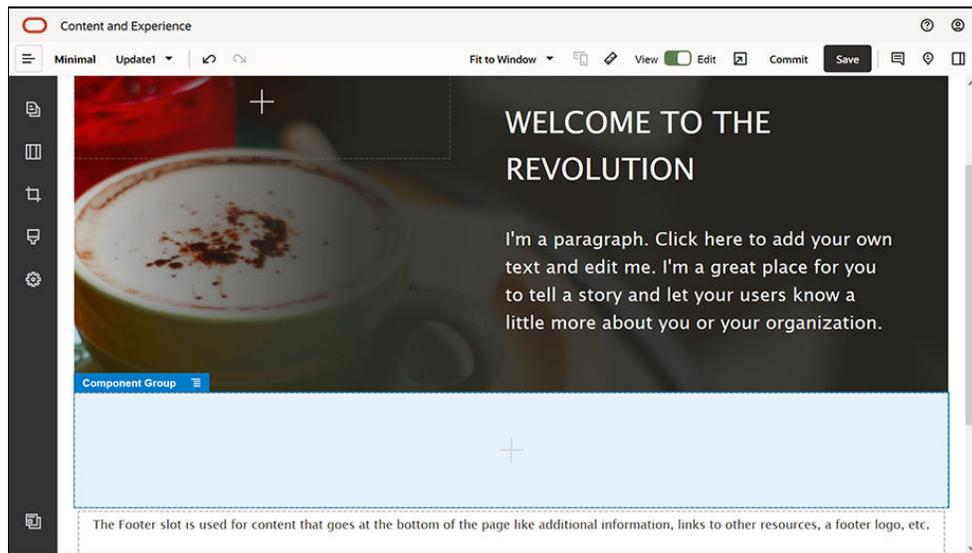
選取文字，並透過文字編輯器將文字大小設為 24。此外，請透過文字編輯器將其字型色彩設為白色。



11. 現在，橫幅已就緒。請將此元件群組另存為自訂元件群組，以便之後能夠將它用於其他網站頁面中。按一下元件群組的功能表圖示 ，然後按一下**儲存**。在「儲存元件群組」對話方塊的**名稱**欄位中輸入「基本橫幅」，然後按一下**儲存**。您會發現這個名稱 (**基本橫幅**) 現在已顯示為元件群組的名稱。

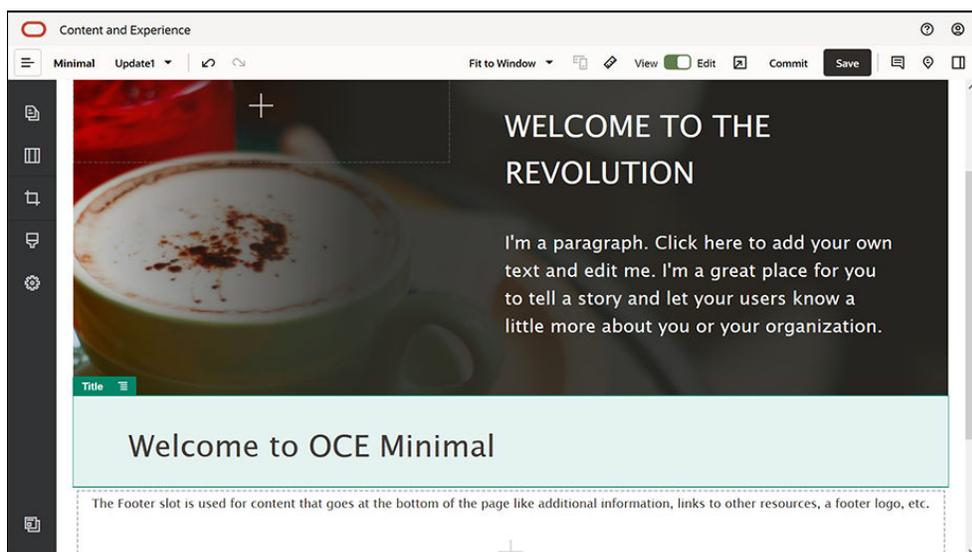


- 從左側提要欄位中，將另一個「元件群組」拖放至「主體」槽位中已新增的基本橫幅元件群組下方。



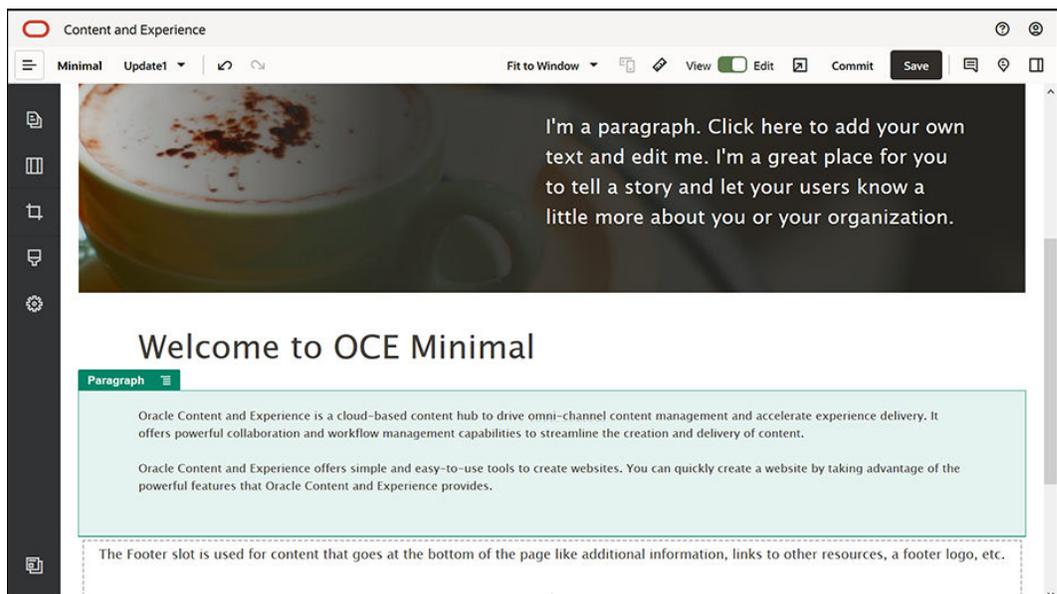
- 將「標題」元件拖放至此元件群組中。
- 按一下「標題」元件並輸入「歡迎使用基本 OCE」。
- 然後完成輸入一般頁籤中的標題元件設定值。

特性	值
頂端	3vw
底端	1.8vw
靠左	6vw
右側	6vw

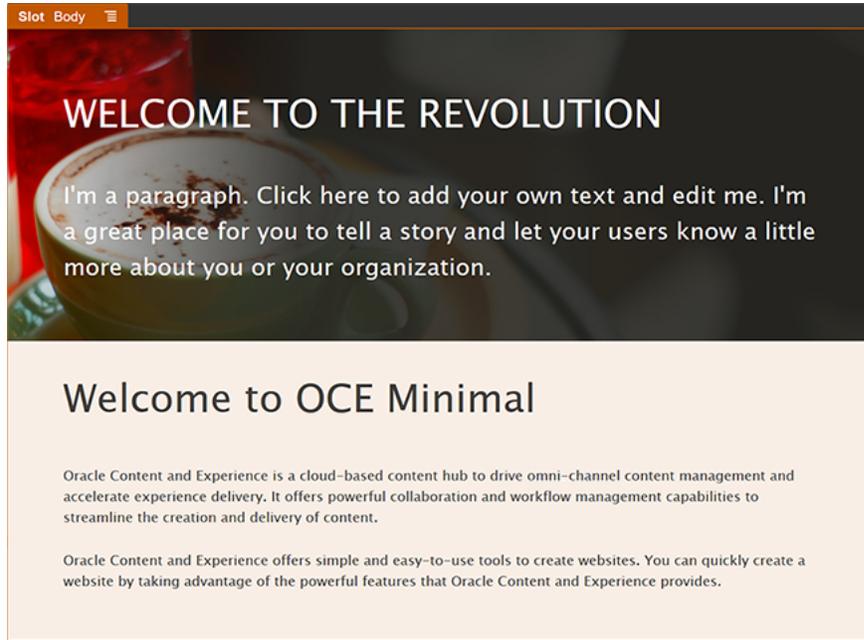


- 將「段落」元件從左側提要欄位拖放至元件群組中的「標題」元件下方。按一下「段落」元件並輸入下列文字：  
「Oracle Content Management 是一種雲端內容中心，可駕馭全通路的内容管理，並加快服務體驗的傳遞。它提供強大的協作與工作流程管理功能，可簡化建立及交付內容的作業。」  
「Oracle Content Management 提供簡單且易於使用的工具來建立網站。您可以利用 Oracle Content Management 提供的強大功能，快速建立網站。」
- 完成輸入一般頁籤中的段落元件設定值。

特性	值
頂端	20px
底端	50px
靠左	6vw
右側	6vw

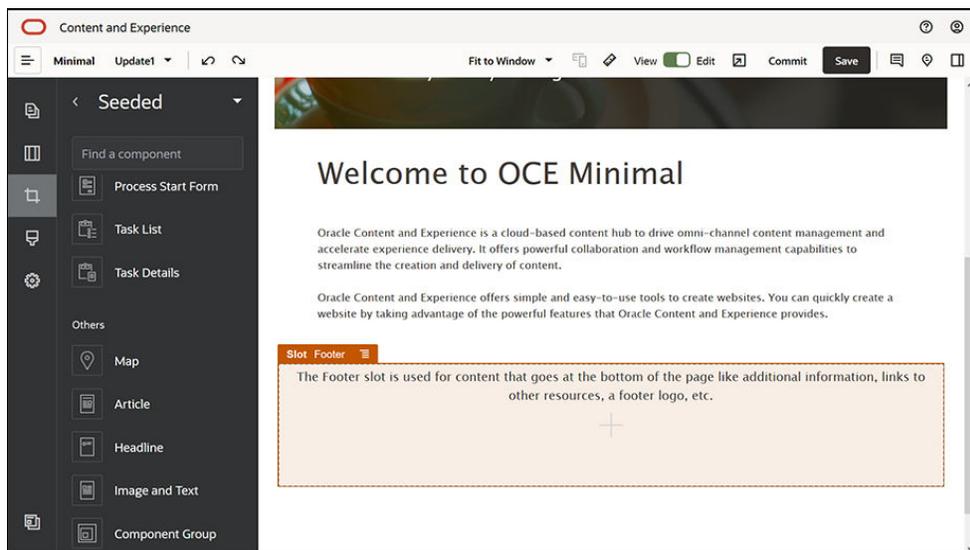


- 主體槽位已經完成。請將此元件群組另存為自訂元件群組，以便之後能夠將它用於其他網站頁面中。按一下元件群組的功能表圖示 ，然後按一下**儲存**。在「儲存元件群組」對話方塊的**名稱**欄位中輸入「基本主體」，然後按一下**儲存**。
- 按一下「網站產生器」右上方的**儲存**，以儲存您所做的變更。「主體」槽位現在看起來應該像下圖這樣：



接著繼續完成「頁尾」槽位：

1. 將左側提要欄位中的「元件群組」拖放至「頁尾」槽位中。將元件群組設定值中的**色彩**欄位設為 **#333333**。

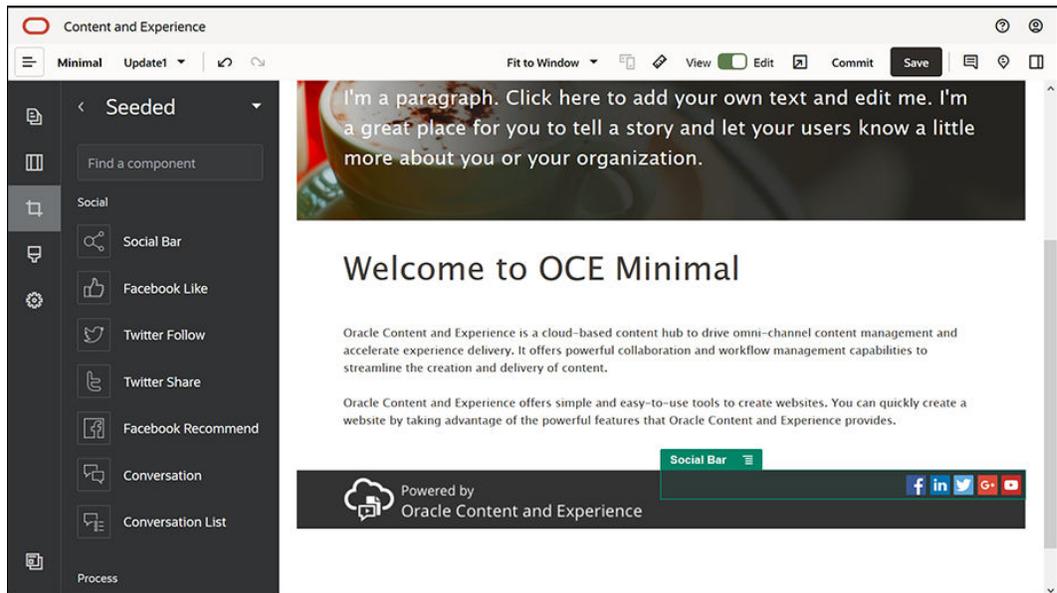


2. 將「影像」元件拖放至元件群組中，然後完成輸入**一般**頁籤中的影像元件設定值。

特性	值
選取	「基本影像」資料夾中的 Powered_by_OCE.png
對齊方式	靠左
寬度	取消選取 <b>設定寬度</b>
頂端	0.9vw

特性	值
底端	0.9vw
靠左	6vw
右側	0

3. 將「社群列」元件從左側提要欄位拖放至元件群組中「影像」元件的右側。



完成輸入**一般**頁籤中的社群列元件設定值。

特性	值
頂端	1.8vw
底端	1.8vw
靠左	0.3vw
右側	6vw

在**一般**頁籤中，依序按一下圖示和其中一個圖示名稱，以完成輸入設定值。

特性	值
URL	<ul style="list-style-type: none"> <li>· <a href="https://www.facebook.com/Oracle/">https://www.facebook.com/Oracle/</a> (Facebook)</li> <li>· <a href="https://www.linkedin.com/company/oracle/">https://www.linkedin.com/company/oracle/</a> (LinkedIn)</li> <li>· <a href="https://twitter.com/Oracle">https://twitter.com/Oracle</a> (Twitter)</li> <li>· <a href="https://www.youtube.com/oracle/">https://www.youtube.com/oracle/</a> (YouTube)</li> </ul>
目標	在新視窗開啟

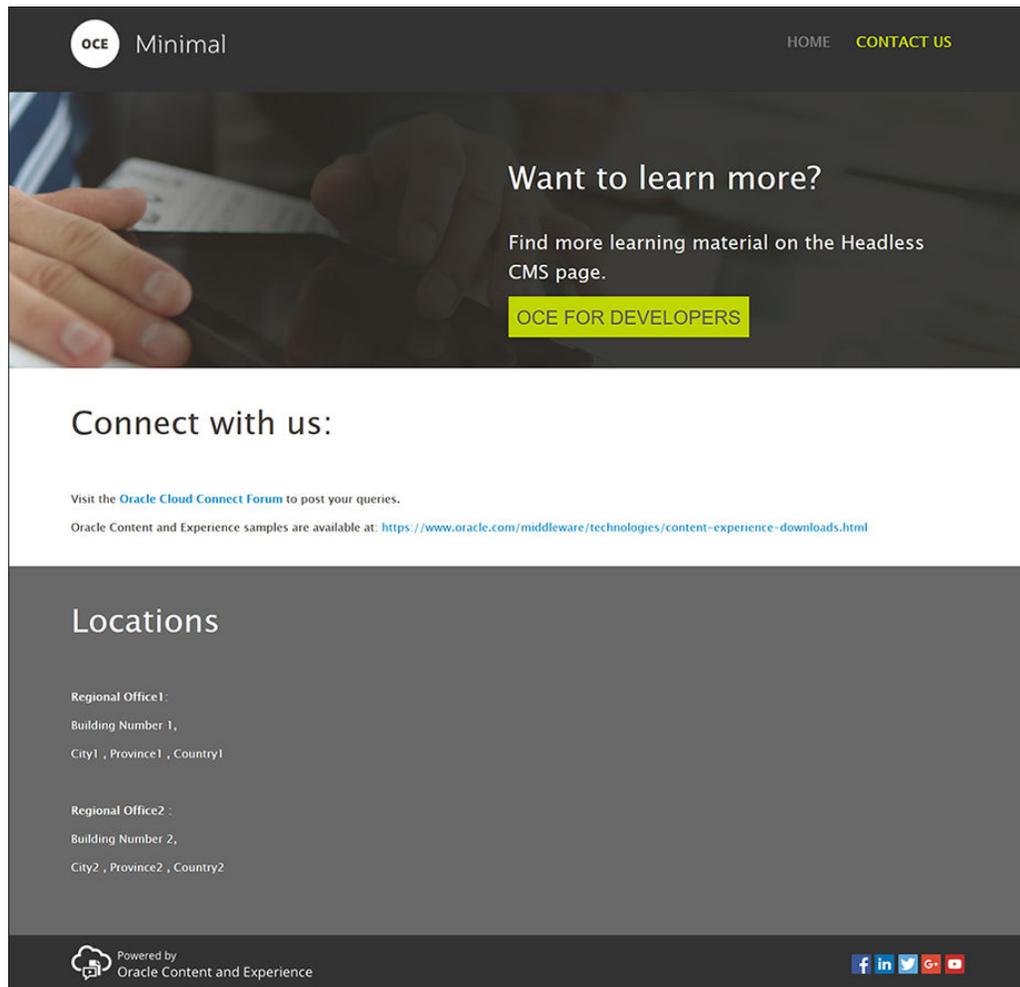
4. 現在，頁尾已就緒。請將此元件群組另存為自訂元件群組，以便之後能夠將它用於其他網站頁面中。按一下元件群組的功能表圖示 ，然後按一下**儲存**。在「儲存元件群組」對話方塊的**名稱**欄位中輸入「基本頁尾」，然後按一下**儲存**。此元件群組看起來應該像這樣：



5. 按一下「網站產生器」右上方的**儲存**，以儲存您所做的變更。

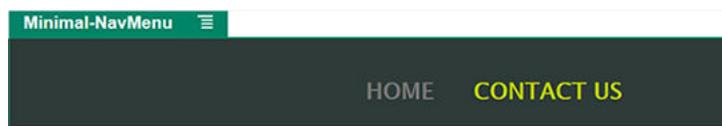
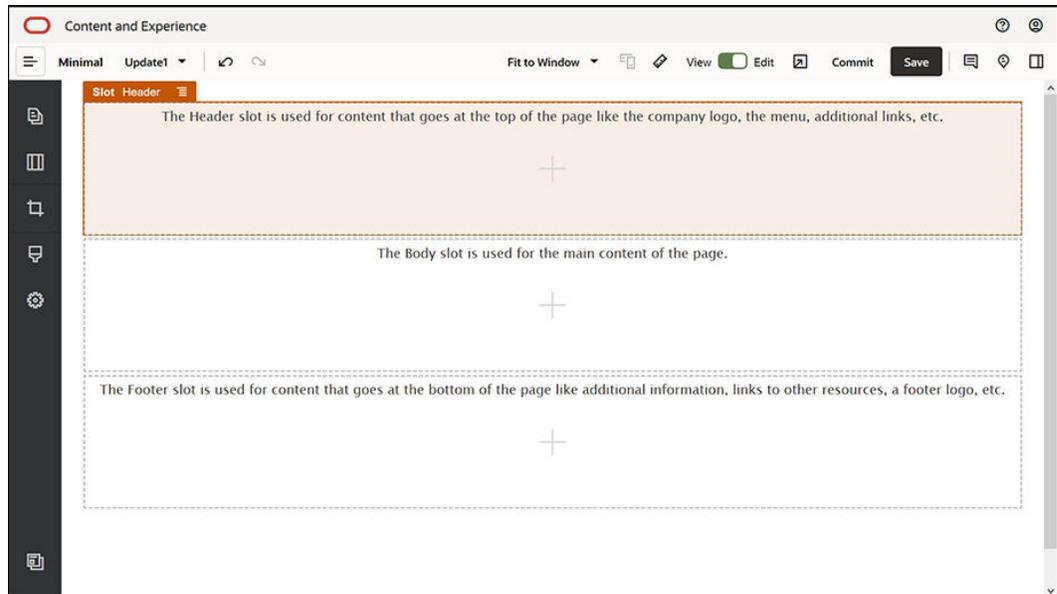
按一下「網站產生器」右上方的  即可預覽您網站的第一個頁面。網站仍未發布，因此其他人目前無法檢視。

您已完成建立首頁。接著讓我們來建置「與我們聯絡」頁面。這是聯絡方式頁面完成後的外觀：

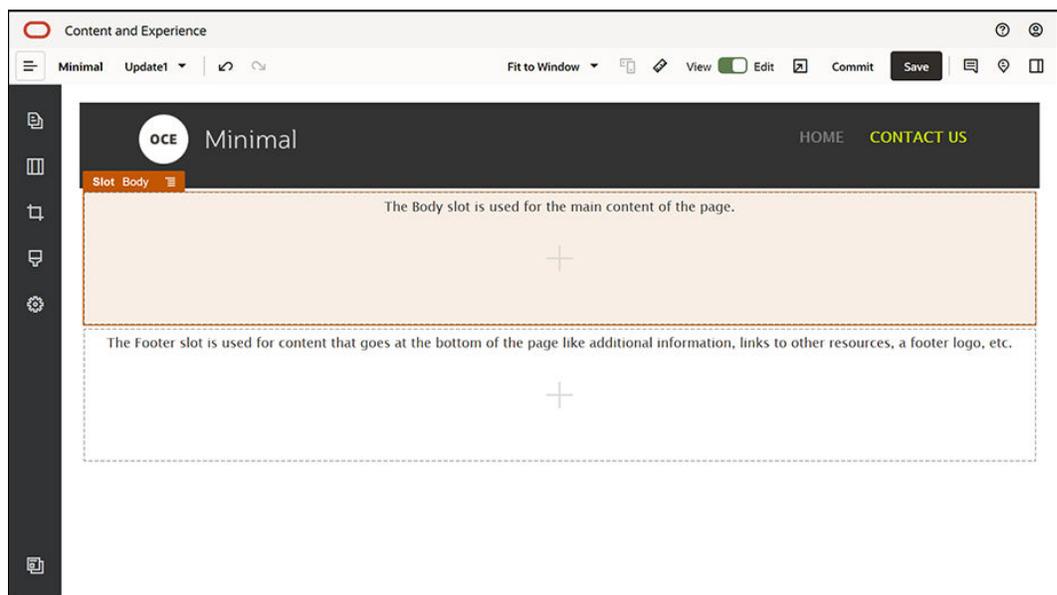


在各種不同的槽位中加入元件：

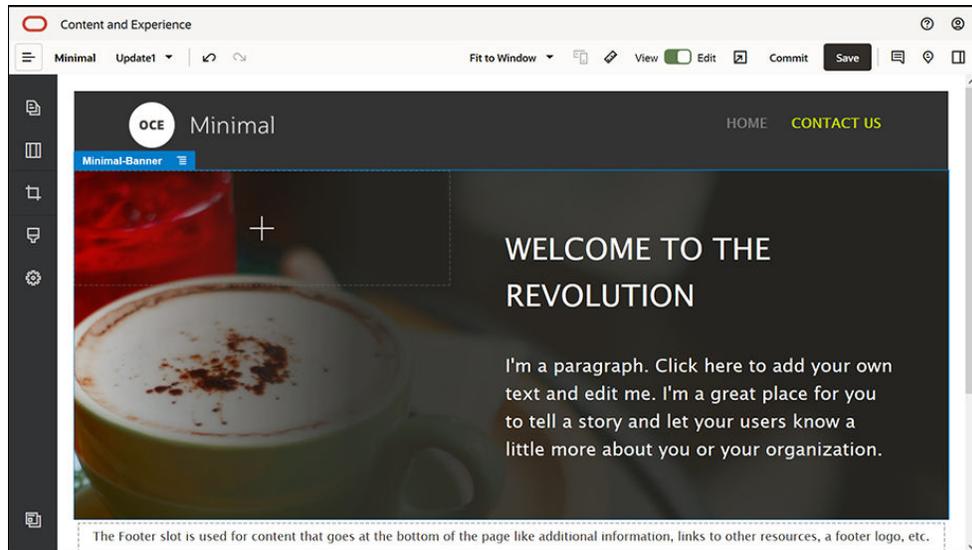
1. 在左側提要欄位中，依序按一下  和**新增頁面**。
2. 在**頁面名稱**欄位中輸入「與我們聯絡」並按一下**關閉**。頁面已新增至您的網站。
3. 在左側提要欄位中，依序按一下  和**自訂**。
4. 將基本標頭元件 (您稍早所建立並儲存) 拖放至「標頭」槽位中。



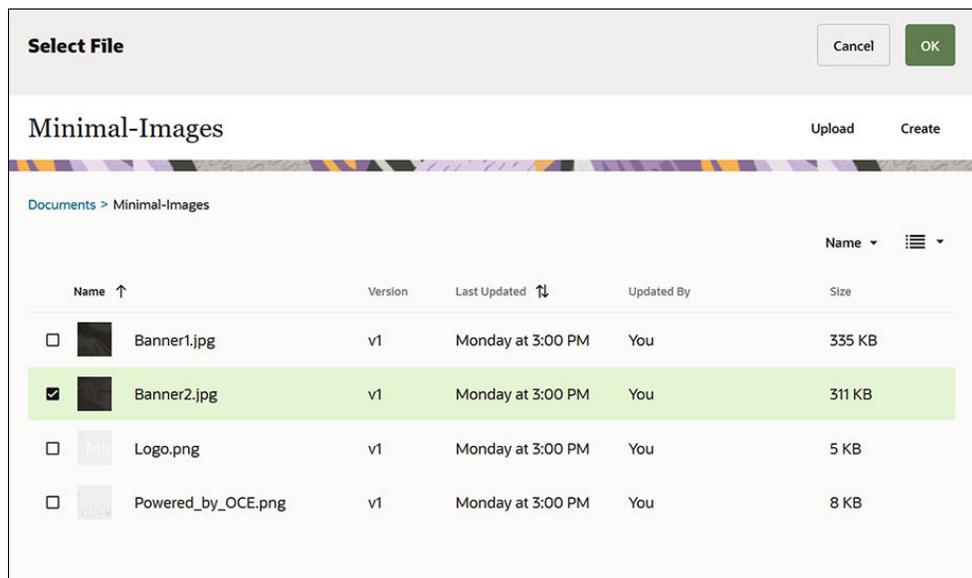
請注意，基本導覽功能表元件已自動選擇您剛剛建立的「與我們聯絡」頁面。



5. 將左側提要欄位中的基本橫幅元件拖放至「主體」槽位中。



6. 修改橫幅，讓它看起來與首頁的橫幅不同，並符合「與我們聯絡」頁面的需求。在元件群組內，修改「兩欄」區段版面配置的設定值：  
在背景頁籤中，按一下影像欄位中的**選取影像**，然後從**基本影像**資料夾 (您稍早在文件頁籤中所建立) 選取 **Banner2.jpg**。



7. 在「兩欄」區段版面配置中，修改「標題」與「段落」元件中的文字。
  - 在「標題」元件中輸入「想要進一步瞭解嗎？」。
  - 在「段落」元件中輸入「尋找無頭 (headless) CMS 頁面的其他學習教材。」。
8. 在左側提要欄位中，依序按一下  和**內建**。
9. 「與我們聯絡」頁面的影像將一個稱為**適用於開發人員的 OCE** 的按鈕包含在橫幅中，因此請將此按鈕加入橫幅中。在基本橫幅元件群組內，將「按鈕」元件拖放到

「兩欄」區段版面配置的第二個資料欄 (在最近新增的「段落」元件下方)。然後完成輸入一般頁籤中的按鈕元件設定值。

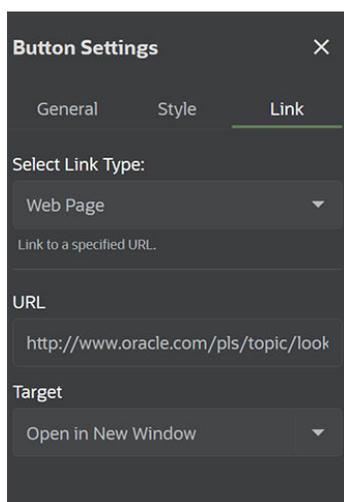
特性	值
標籤	適用於開發人員的 OCE
頂端	0.3vw
底端	3vw
靠左	6vw
右側	0.3vw

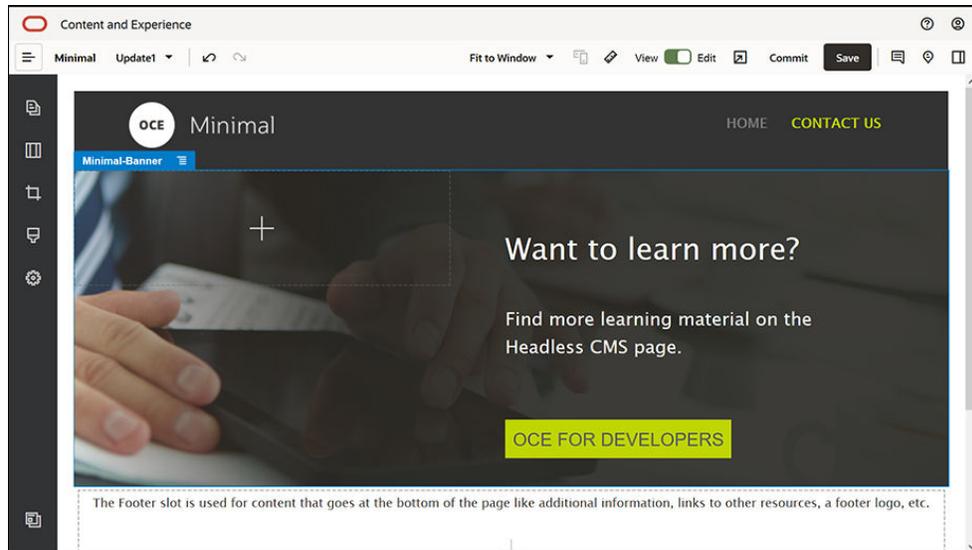
在**樣式**頁籤中，選取**自訂**並完成設定值。

特性	值
背景色彩	#c0d600
字型	<ul style="list-style-type: none"> <li>· 輸入 24 作為字型大小。</li> <li>· 輸入色彩 #58595b。</li> </ul>
框線	無
超連結色彩	<ul style="list-style-type: none"> <li>· 將<b>背景</b>設為 #e1fa00。</li> <li>· 將<b>字型</b>設為 #58595b。</li> <li>· 將<b>框線</b>設為 #2222dd。</li> </ul>
角	0

完成輸入**連結**頁籤中的設定值。

特性	值
選取連結類型	網頁
URL	<a href="http://www.oracle.com/pls/topic/lookup?ctx=cloud&amp;id=content-cloud-headless">http://www.oracle.com/pls/topic/lookup?ctx=cloud&amp;id=content-cloud-headless</a>
目標	在新視窗開啟





10. 在左側提要欄位中，依序按一下  和 **自訂**。
11. 將基本主體元件拖放至「主體」槽位中基本橫幅元件的下方。
12. 在基本主體元件中，修改「標題」與「段落」元件中的文字。
  - a. 在「標題」元件中輸入「聯絡我們：」。
  - b. 在「段落」元件中輸入詳細資訊，例如電子郵件地址和其他支援相關的連結：「請造訪 [Oracle Cloud 聯繫論壇](#)張貼您的詢問。」

「Oracle Content Management 範例位於：<https://www.oracle.com/middleware/technologies/content-experience-downloads.html>」
13. 從左側提要欄位中，將另一個基本主體元件拖放至「主體」槽位中稍早新增的基本主體元件下方。
14. 在元件群組設定值的**背景**頁籤中，將**色彩**欄位設為 **#696969**。
15. 在基本主體元件中，修改「標題」與「段落」元件。
  - a. 在「標題」元件中輸入「位置」。選取文字，並透過文字編輯器將其字型色彩設為**白色**。
  - b. 在「段落」元件中輸入下列文字。

「辦事處 1：  
大樓號碼 1，  
城市 1，省 1，國家/地區 1  
辦事處 2：  
大樓號碼 2，  
城市 2，省 2，國家/地區 2」

選取文字，並透過文字編輯器將其字型色彩設為**白色**。
16. 現在，請在「與我們聯絡」頁面中加入頁尾。將基本頁尾元件拖放至頁尾槽位中。
17. 按一下「網站產生器」右上方的**儲存**，以儲存您所做的變更。

按一下「網站產生器」右上方的  來預覽網站，確定所有項目都沒問題。確定按一下「與我們聯絡」頁面上的標誌影像後，可將您帶回到首頁。測試功能表，以確定網站頁面之間的導覽可正確運作。

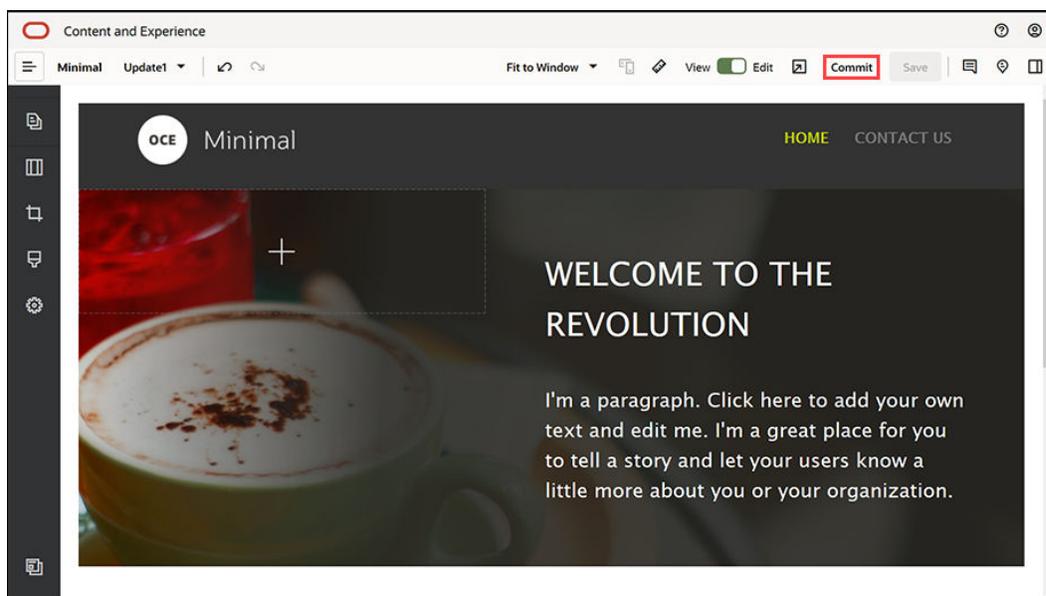
您的網站已可進行發布了。

下一步：發布網站

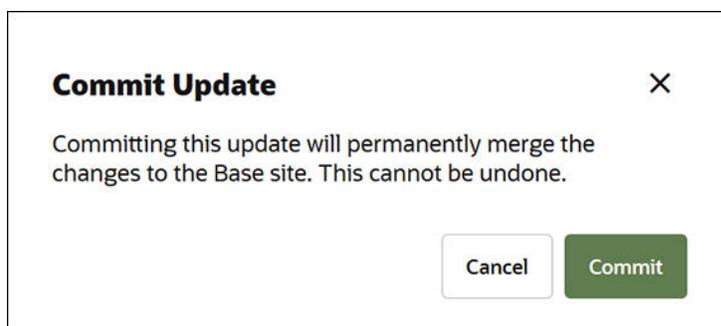
## 步驟 3：發布網站

現在您已順利建立自己的網站，您可以發布網站，讓它上線給您的使用者看見。

1. 所有項目都沒問題後，您可以按一下「網站產生器」右上方的**確認**，確認您對基本網站所做的變更。

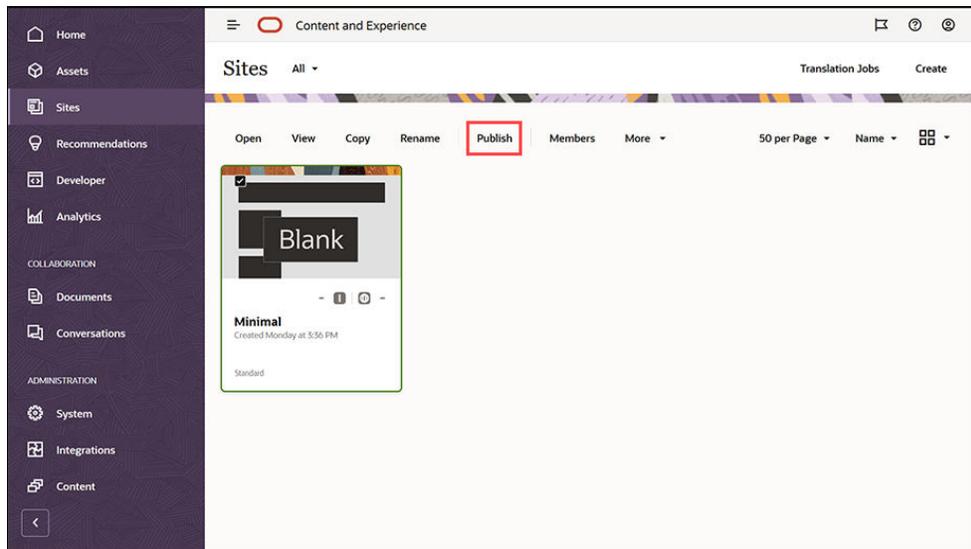


2. 在「確認更新」對話方塊中，按一下**確認**。

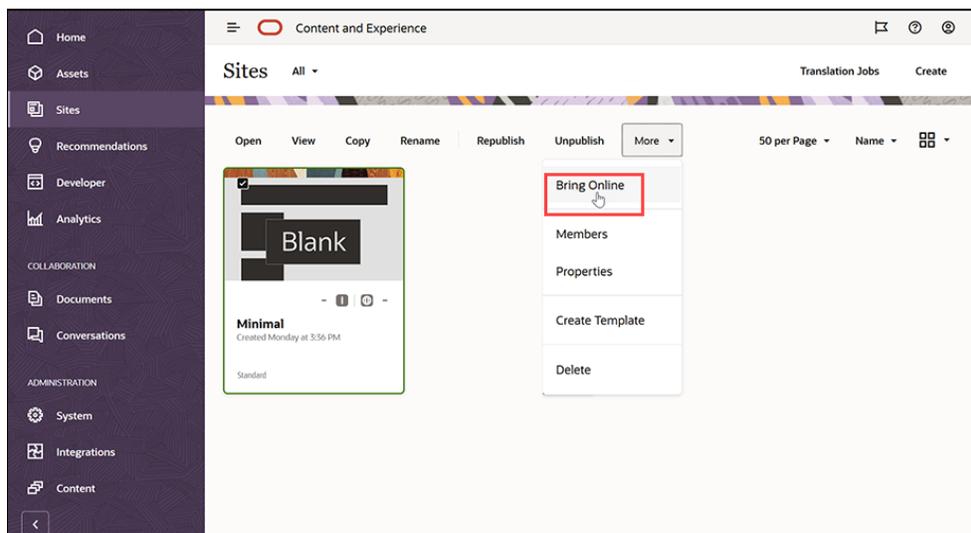


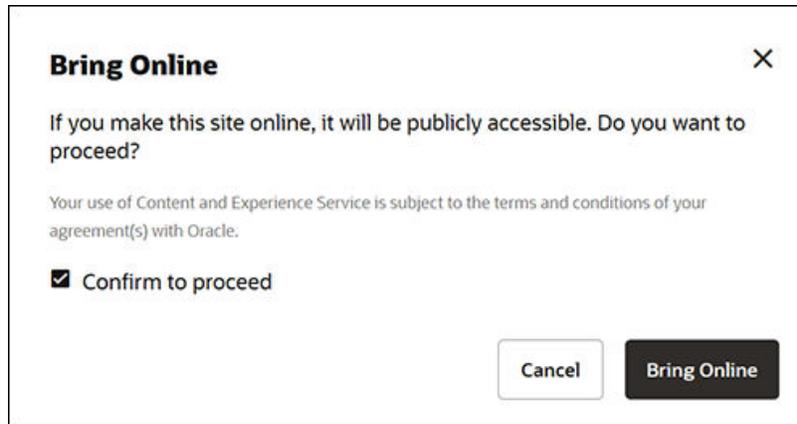
3. 確認變更後，網站就可以進行發布了。
4. 關閉「網站產生器」。

5. 在 Oracle Content Management Web 介面中，按一下左側導覽功能表中的**網站**，然後選取網站。
6. 從功能表列或滑鼠右鍵功能表中選擇**發布**。

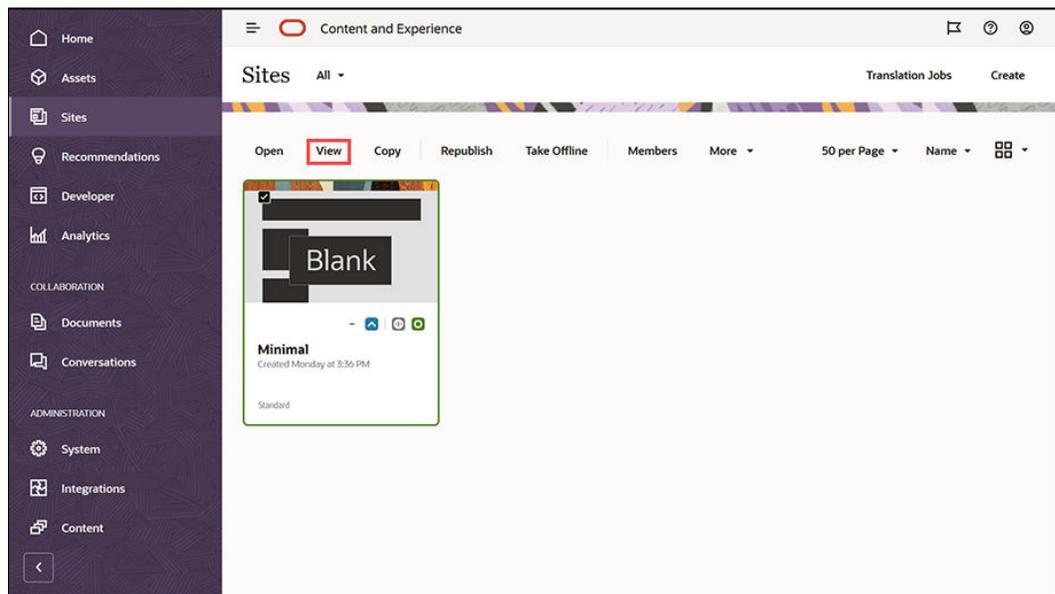


7. 網站發布之後，頁面頂端便會顯示一則通知，指出該網站已發布。
8. 在「網站」頁面上，再次選取該網站，然後從功能表列或滑鼠右鍵功能表中選擇**上線**。在「上線」對話方塊中，選擇**確認繼續進行**並按一下**上線**。





9. 就是這麼簡單。您的網站已是線上狀態，因此其他人也能進行檢視。  
若要檢視您的公用網站，請從功能表列或滑鼠右鍵功能表中選取**檢視**。



## 執行其他作業

您可以根據您組織的需求來自訂網站。以下提供幾個實用連結，可協助您開始使用：

- [使用樣式和格式](#)
- [編輯秘訣與技巧](#)
- [管理網站和網站設定值](#)
- [上傳網站檔案](#)
- [新增頁面](#)
- [移動頁面](#)
- [排列頁面內容](#)

- 變更頁面設定值
- 變更頁面版面配置
- 使用表格
- 設定搜尋引擎特性

# 第 II 部分

## 建立和編輯網站

此部分詳細說明如何使用樣板和主題開始建立網站，以及如何使用元件和版面配置編輯網站以組織並新增內容。其中包含下列章節：

- [建立網站](#)
- [編輯網站](#)
- [在網站中使用樣板和主題](#)
- [管理自訂元件與版面配置](#)
- [使用網站頁面](#)
- [排列頁面內容](#)
- [使用內建元件](#)

# 4

## 建立網站

所有具備適當權限的人都能使用 **Oracle Content Management** 建置網站。您並不需要使用任何專屬工具、程式碼或軟體。使用者介面是圖形介面，直觀且易於使用。

建立網站時，是利用樣板來開始進行。樣板含有您開始建置網站所需的一切項目，包括網站程式碼架構、含有範例頁面和內容的預設網站、含有樣式設定的主題、資源 (例如影像)，甚至是自訂元件。請參閱[瞭解網站建立程序](#)。

**Oracle Content Management** 的網站治理功能除了可簡化並加快業務使用者的體驗傳遞外，IT 部門也可輕鬆地集中控管及追蹤體驗，並且能夠完全管理整個體驗生命週期，降低公司對所需之每項新體驗的建立與維護成本。治理功能內建於 **Oracle Content Management** 核心，而且只需要由管理員啟用即可。請參閱 *Administering Oracle Content Management* 中的設定網站和資產設定值。

- [瞭解網站建立程序](#)
- [建立網站](#)
- [複製網站](#)
- [管理網站要求](#)
- [變更網站要求詳細資訊](#)
- [檢視網站要求原則](#)

## 建立網站

若要建立網站，請選取樣板、為網站命名，以及新增內容。若已啟用治理，則在新增內容之前，您必須先送出網站要求，然後在要求獲得核准後新增內容。

您的管理員必須先啟用網站建立功能並提供樣板給您使用，您才能建立網站。如果您在網站頁面上沒有看到**建立**選項，或沒有任何樣板可供使用，請洽詢您的管理員。請參閱[開始建立網站](#)和[瞭解網站治理](#)。

在建立可使用資產和多語言網站功能之企業網站的過程中，您的網站會與儲存區域、本地化規定以及預設語言建立關聯。如果沒有儲存區域和本地化規定可使用，請建立它們，或要求您的內容管理員建立它們。

### 備註：

**Oracle Content Management** 入門版無法提供翻譯、無法啟用治理，而且只能建立一個網站。如需完整功能集和無限制網站，請升級至 **Oracle Content Management** 進階版。

1. 按一下**網站**。
2. 按一下**建立**。

3. 在選擇樣板頁面中，選取您網站的樣板。
  - 若已啟用治理，則樣板將判斷您建立的網站是標準網站 (未使用資產儲存區域或本地化規定)，還是企業網站 (使用資產儲存區域和本地化規定)。
4. 在設定網站頁面中，您會見到網站建立之前所需經過的核准、網站所需的最低層級安全性，以及網站頁面的影像。如果只能由特定人員進行核准，請按一下**顯示核准者**來查看哪些人可以核准網站要求。
  - 如果未啟用治理，則請選取**標準**或**企業**作為您要建立的網站類型。
  - 若是要建立標準網站，請按**下一步**以前往下一頁。
  - 若是要建立企業網站，則請完成下列步驟：
    - a. 選擇用來管理網站內容的預設儲存區域。如果您是儲存區域管理員，但沒有可用的儲存區域，或是您想要使用新的儲存區域，請選取**建立新的儲存區域**，然後依照步驟建立儲存區域。如需在網站使用多個儲存區域的相關資訊，請參閱 *運用 Oracle Content Management 管理資產* 中的讓網站存取多個儲存區域。
    - b. 選取網站的本地化規定。如果您是儲存區域管理員，但沒有可用的規定，或是您想要使用新的規定，請選取**建立新的本地化規定**，然後依照步驟建立規定。
    - c. 樣板的本地化規定若沒有預設語言，請立即選取。您只會看到本地化規定所需的語言。
    - d. 按**下一步**以前往下一頁。
5. 在新增詳細資訊頁面中，輸入下列資訊。
  - 輸入網站的名稱。網站 URL 中會使用此名稱。名稱中可以包含字母、數字、底線 ( \_ ) 以及連字號 ( - )。URL 區分大小寫。如果您輸入空格，系統會自動以連字號取代。請勿使用與現有網站相同但大小寫不同的網站名稱。例如，若現有網站名為 **ABC**，請不要建立名為 **Abc** 的另一個網站。

 **備註：**

請勿使用下列作為樣板、主題、元件、網站、或網站頁面的名稱：  
authsite、content、pages、scstemplate\_\*、\_comps、  
\_components、\_compsdelivery、\_idcservice、\_sitescloud、  
\_sitesclouddelivery、\_themes、\_themesdelivery。您可以使用下列  
名稱作為網站頁面的名稱，但不可以作為樣板、主題、元件或網站的  
名稱：文件、網站。

- 若是建立企業網站，樣板規定允許您編輯易記 URL 的預設網站前置碼。此前置碼將附加至內容項目網址代稱值 (特定頁面或資產 URL 的部分)。
  - (選擇性) 輸入網站的描述。
  - (選擇性) 輸入此網站要求的理由。這將有助於網站管理員判斷是否應該核准要求。
6. 按一下**完成**。

如果網站要求需要核准，系統會傳送一封電子郵件給需要核准的人員。如經核准，您會收到一封電子郵件，通知您要求已經核准，將會建立網站。如果您的要求被拒絕，您會收到一封電子郵件，通知您要求已被拒。您可以在網站頁面上，從篩選功能表中選取**要求**來查看擱置中的要求。如果要求被拒絕，您應該會看到說明拒絕原因的訊息，以便您可以更正問題和重新送出要求。

如果網站要求已設定成自動核准，網站就會自動建立。畫面會出現一個進度列，顯示新網站的名稱和建立狀態。在網站建立後，其名稱會出現在網站清單中，且其狀態為離線。

如果建立的是企業網站，就會在您選取的儲存區域中建立相對應的網站集合。若您和某個使用者共用網站，該使用者對於關聯的集合會具有相同權限。

網站建立後，就可以共用該網站、變更安全性，新增和編輯內容、發布網站，以及讓網站上線。請參閱[管理網站和網站設定值](#)和[編輯網站](#)。

## 複製網站

您可以利用複製網站方式，快速建立新的網站。原始網站中的所有項目 (包括主題、所有未處理更新項目、頁面、頁面內容、建議、所有其他資產 (例如影像) 以及規定) 都會被複製到所提供之新名稱的網站底下。

### 備註：

如果您複製的網站使用來自多個儲存區域的內容，則需要使用 **OCE** 工具程式進行此作業。請參閱[使用 OCE 工具程式進行開發](#)和[使用 cec 命令行公用程式](#)

1. 按一下**網站**。
2. 選取要複製的網站，然後從滑鼠右鍵功能表中選擇**複製**，或按一下動作列中的 。
3. 在設定網站頁面中，您會見到網站建立之前所需經過的核准、網站所需的最低層級安全性，以及網站頁面的影像。請完成下列步驟：
  - a. 從**複製**下拉式清單中選取是只要複製基本網站，還是要包括更新項目。如果選擇包括更新項目，複製的網站將會包含原始網站的所有未處理更新項目。若您是複製標準網站，請按一下箭號以前往下一頁，並跳到步驟 4。
  - b. 若是複製企業網站，系統會根據原始網站設定預設資產儲存區域、本地化規定、必要的語言以及預設語言。您可以選擇使用其他儲存區域來管理此網站的內容。
  - c. 按**下一步**以前往下一頁。
4. 在新增詳細資訊頁面中，輸入下列資訊。
  - 輸入網站的名稱。網站 **URL** 中會使用此名稱。名稱中可以包含字母、數字、底線 (`_`) 以及連字號 (`-`)。URL 區分大小寫。如果您輸入空格，系統會自動以連字號取代。請勿使用與現有網站相同但大小寫不同的網站名稱。例如，若現有網站名為 **ABC**，請不要建立名為 **Abc** 的另一個網站。

### 備註：

請勿使用下列作為樣板、主題、元件、網站、或網站頁面的名稱：`authsite`、`content`、`pages`、`scstemplate_*`、`_comps`、`_components`、`_compsdelivery`、`_idcservice`、`_sitescloud`、`_sitesclouddelivery`、`_themes`、`_themesdelivery`。您可以使用下列名稱作為網站頁面的名稱，但不可以作為樣板、主題、元件或網站的名稱：文件、網站。

- 若是複製企業網站，您可以選擇編輯易記 **URL** 的預設網站前置碼值。此前置碼將附加至內容項目網址代稱值 (特定頁面或資產 **URL** 的部分)。

- 輸入網站的選擇性描述。
- 輸入此網站要求的選擇性理由。這將有助於網站管理員判斷是否應該核准要求。

#### 5. 按一下**完成**。

如果網站要求需要核准，就會在網站管理員核准要求之後建立網站。您可以在網站頁面上，從篩選功能表中選取**要求**來查看擱置中的要求。如果您的要求被拒絕，您應該會看到說明拒絕原因的訊息，以便您可以更正問題和重新送出要求。

如果網站要求已設定成自動核准，網站就會自動建立。畫面會出現一個進度列，顯示新網站的名稱和建立狀態。在網站建立後，其名稱會出現在網站清單中，且其狀態為離線。

若是複製企業網站，系統會在您所選的儲存區域中建立相對應的網站集合。若您和某個使用者共用網站，該使用者對於關聯的集合會具有相同權限。

網站建立後，就可以共用該網站、變更安全性，新增和編輯內容、發布網站，以及讓網站上線。如果您共用的網站可存取多個儲存區域，只會共用預設儲存區域。任何其他儲存區域都必須個別共用。請參閱[管理網站和網站設定值](#)和[編輯網站](#)。

## 管理網站要求

如果已啟用網站治理，建立網站之前，可能必須先核准網站。您可以在「網站」頁面使用「要求」篩選來檢視擱置中的網站要求。

### 備註：

使用 Oracle Content Management 入門版，無法啟用治理。如需完整功能集和無限制網站，請升級至 Oracle Content Management 進階版。

請參閱[瞭解網站治理](#)。

若要檢視擱置中的網站要求，請從網站頁面上的篩選功能表中，選取**要求**。您若是**網站管理員**，便能見到系統中的所有網站要求，包括需要其他人核准的要求。您若是**核准者**，將能見到您本身為核准者的所有網站要求。*所有使用者*可見到自己送出的網站要求。

在要求清單中，您可以看到網站的縮圖、網站的名稱、要求者、要求時間、網站描述、要求狀態，以及顯示網站是否需要登入的圖示。

視您的角色而定，您可以執行下列動作：

- 若要檢視要求的更多詳細資訊 (例如所需的最低安全性、使用的樣板、選擇性的網站理由，以及縮圖)，請按一下網站名稱或選取要求，然後按一下**檢視**。
- 您若是網站管理員或核准者，便能夠選取要求，然後按一下**核准**來核准要求。網站會在獲得核准後自動建立。
- 您若是網站管理員或核准者，便能夠檢視詳細資訊，然後按一下**拒絕**來拒絕要求。請輸入拒絕要求的原因，然後按一下**拒絕**。雖然您也可以從要求清單中選取要求，然後按一下**拒絕**來拒絕要求，但將無法新增拒絕要求的原因。

- 如果要求失敗或被拒絕，您可以檢視詳細資訊、視需要編輯要求，然後**重新送出**該要求。
- 您若是網站建立者，可以選取要求，然後按一下**刪除**來刪除要求。

 **備註：**

網站管理員可以刪除已遭網站要求者刪除的網站要求。

網站建立後，就可以共用該網站、變更安全性，新增和編輯內容、發布網站，以及讓網站上線。請參閱[管理網站和網站設定值](#)和[編輯網站](#)。

## 變更網站要求詳細資訊

「網站要求詳細資訊」頁籤中會顯示網站名稱、描述、使用的樣板、理由、擁有者 (提出要求的人員) 以及網站影像。

您若是建立要求的要求擁有者，就可以更新網站要求詳細資訊。

檢視或變更網站要求詳細資訊：

1. 按一下**網站**，然後按一下網站功能表中的**要求**。
2. 按一下想要檢視或編輯的要求。
3. 您若為核准者，即可**核准**或**拒絕**要求。您若為要求者，可視需要編輯資訊，然後**重新送出**要求。

## 檢視網站要求原則

「網站要求原則」頁籤中會顯示網站類型 (標準或企業)、可以存取已發布網站的人員，以及網站建立之前需經過的所有核准。若為企業網站，您還會見到資產儲存區域、本地化規定、必要語言以及網站所使用的預設語言。

檢視網站要求原則：

1. 按一下**網站**，然後按一下網站功能表中的**要求**。
2. 按一下想要檢視或編輯的要求，然後按一下**原則**頁籤。
3. 您若為核准者，即可**核准**或**拒絕**要求。您若為要求者，可視需要變更資產儲存區域，然後**重新送出**要求。

# 5

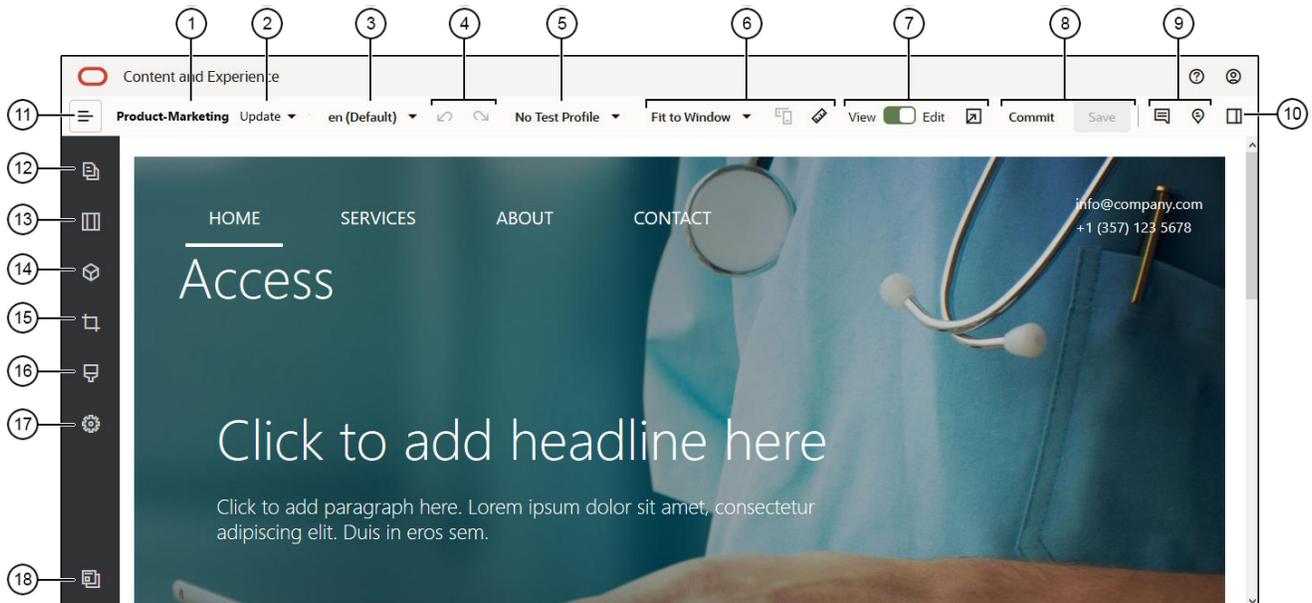
## 編輯網站

讓我們深入瞭解編輯器，並查看它的功能。

- [瞭解網站產生器頁面](#)
- [瞭解網站更新](#)
- [使用更新](#)
- [編輯秘訣與技巧](#)
- [使用樣式和格式](#)
- [新增自訂網站特性](#)
- [自訂網站設定值](#)
- [使用表格](#)
- [上傳網站檔案](#)

### 瞭解網站產生器頁面

編輯網站的現有更新或建立更新時，會在「網站產生器」中開啟該更新。因此，請花一些時間熟悉頁面版面配置與工具。



以下是注意事項：

註標	描述
1	列出網站的 <b>名稱</b> 。
2	查看 <b>更新項目</b> 以瞭解目前處理之更新項目的名稱。若有多個可用的更新，您可以切換至其他更新項目。
3	查看 <b>語言</b> 以瞭解目前所處理的是哪個版本。
4	使用 <b>還原</b>  可回復上次在編輯器中所做的編輯或變更。使用 <b>重做</b>  可重新套用使用還原所回復的最近變更。您可以多次重複使用還原，回復目前更新中的一系列變更，包括對內容、樣式和頁面組織的變更。
5	<b>測試設定檔</b> 提供可建立具備各種受眾屬性設定檔的機會，並使用這些設定檔代表不同的網站訪客來測試建議。測試設定檔是在網站管理員建立建議時所設定。
6	可使用 <b>預覽選項</b> 查看內容在不同情況下所顯示的樣子。您可以瀏覽任何頁面，並查看基本網站在套用目前更新後的變化。按一下 <b>調整成視窗大小</b> 並選擇尺寸，即可檢視頁面以該畫面大小在裝置上顯示的樣子。系統已提供幾種大小，您也可以自行建立裝置大小。按一下  即可查看標記。按一下尺規上的間隔，可快速查看網站以不同大小顯示時的樣貌。您也可以選取  來查看網站在行動裝置上不同方向的顯示情形。具有響應式設計的主題會自動根據所選螢幕大小排列頁面內容，讓使用者享有最佳的閱讀體驗。
<div style="border: 1px solid #0070c0; padding: 10px; background-color: #e6f2ff;"> <p> <b>備註：</b> 網站頁面的連結在預覽模式中沒有作用。</p> </div>	
7	可以使用頁面頂端的 <b>開關</b> 在檢視與編輯之間做切換。按一下  即可預覽網站。這會顯示網站訪客將看到的頁面，但不會顯示槽位和元件框線，以及編輯時所使用的其他視覺化輔助。您可以使用頁面上的連結 (包括導覽功能表中的連結、文字中的連結等等) 前往網站中的各個位置。連至其他網站的連結會在新視窗中開啟。視您指定的連結目標而定，連至目前網站中頁面的連結會在同一個視窗或新視窗中開啟。
8	使用可用的 <b>選項</b> 處理您的變更。若要將變更合併至 <b>基本網站</b> ，請按一下 <b>確認</b> ，或是若要儲存對 <b>更新項目</b> 所做的變更，請按一下 <b>儲存</b> 。
9	可以使用 <b>附註控制項</b> 新增附註或檢視現有附註。 若要新增附註，請按一下  ，接著按一下要新增附註的元件，然後在文字方塊中輸入註解，再按一下 <b>張貼</b> 。請參閱 <i>Collaborating on Documents with Oracle Content Management</i> 中的新增附註。 若要檢視附註，請按一下  。按一下附註後，附註便會成為對話中的焦點。
10	按一下  以顯示 (或隱藏) <b>對話面板</b> ，您可以在此面板建立或參與此網站的相關討論。請參閱 <i>Collaborating on Documents with Oracle Content Management</i> 中的使用對話。

註標	描述
11	按一下  可顯示 (或隱藏) 提要欄位中的工具。

 **備註：**

您必須先將  設為**編輯**，才能使用提要欄位中的工具。

按一下提要欄位中的圖示 (12–18)，即可管理頁面、新增元件、變更主題等等。

註標	按一下...	進行...
12		編輯與新增頁面、管理巢狀頁面結構、重新組織頁面，以及變更頁面設定值。
13		新增區段版面配置，例如水平或垂直版面配置。
14		將資產新增至您的網站。您可以選取建立網站時所選儲存區域中的任何資產。
15		將元件插入您的網站。您可選擇不同類型的元件，例如主題、自訂、內建，或是查看所有可用元件。
16		為網站選取不同的主題，或是變更頁面或槽位的背景。
17		調整網站設定值，例如用於搜尋引擎最佳化 (SEO) 的關鍵字以及網站重新導向。
18		返回「網站」頁面以管理網站。

## 瞭解網站更新

更新項目是目前基本網站的具名變更集合。在您確認變更並永久地更新基本網站之前，變更會保留在更新項目中。

每次在編輯器中檢視或編輯網站時，都會使用更新項目。您在編輯器中進行的任何變更都會成為該更新項目的一部分。您可以有一或多個更新項目，而且隨著時間持續將變更新增至更新項目。更新項目可讓您彈性管理網站的編輯方式。例如，有多位人員處理網站不同部分的專屬更新項目。您可以複查及修改個別的更新項目，當您準備就緒後，就可以將更新項目確認到基本網站。您仍需要發布網站，才能在線上提供這些更新項目。

您可以使用許多方式來組織更新項目：

- 更新項目中的頁面特定變更以該頁面命名
- 由特定使用者在更新項目中進行變更，則含有該使用者的名稱

- 在指定日期或針對特定專案里程碑所進行的變更

 **備註：**

更新項目會顯示該更新項目中針對基本網站的變更。雖然在編輯器內切換可用的更新項目十分容易，但是您一次只能檢視一個更新項目。如果指定的頁面有多個更新項目，則您可能不會知道相同內容區域是否有多個變更。若要防止頁面上的變更衝突，請將個別更新項目的目標設為特定頁面或網站區域。

確認目前更新的變更時，系統會對基本網站進行變更，然後將更新刪除。您仍需要發布網站 (讓網站上線或重新發布)，才能在線上對所有存取網站的人員顯示那些變更。

編輯網站的步驟如下：

1. 從清單中選取網站，然後從滑鼠右鍵功能表中選擇**開啟**，或按一下動作列中的 。
2. 編輯器是以預覽模式開啟。若要在提要欄位中進行變更或使用導覽選項，請確定「編輯」開關  已設為**編輯**。
3. 如果這是網站的第一個更新，請輸入更新的名稱和選擇性的描述，然後按一下**確定**。網站名稱可包含字母、數字、底線 ( \_ ) 以及連字號 ( - )。如果您輸入空格，系統會自動以連字號取代。如果已經有網站的更新，請由清單中選取一個更新，然後按一下 。
4. 若要編輯特定頁面，請使用左側提要欄位中的網站樹狀結構來搜尋頁面，或使用網站的專屬導覽來尋找頁面。若要顯示網站樹狀結構，請按一下左邊的 ，然後按一下 。
5. 視需要新增及變更頁面內容。選取**調整成視窗大小**，以使用版面配置選項來查看頁面在不同裝置與不同尺寸的外觀。系統已提供幾種大小，您也可以自行建立裝置大小。按一下  即可查看標記。按一下尺規上的間隔，可快速查看網站以不同大小顯示時的樣貌。您也可以選取  來查看網站在行動裝置上不同方向的顯示情形。
6. 完成編輯網站之後，請使用下列其中一種方式儲存變更：
  - 按一下**儲存**，將變更儲存到目前的更新。您可以繼續對目前的更新進行作業，或者稍後再返回更新。
  - 按一下**確認**，將變更套用至基本網站。

您仍需要發布網站，網站使用者才能見到變更。請參閱[讓網站上線或離線](#)或[發布網站變更](#)。

確認目前更新的變更時，系統會對基本網站進行變更，然後將更新刪除。每次在編輯器中檢視或編輯網站時，都必須使用作用中的更新項目，因此系統會讓您回到網站清單，以便建立新的更新項目。

使用更新項目時，會以線性方式來處理作業：

- 原始網站 (基本網站) 已經存在。我們將它稱為「第 1 版」。
- 您建立更新項目。當您合併更新項目 (使用「確認」) 時，更新會永久寫入現有網站。您現在有基本網站的新版本 (第 2 版)。

- 如果您建立和合併另一個更新，則會將更新項目永久地寫入至現有網站。您現在有基本網站的新版本 (第 3 版)。

使用更新項目時，請記住：

目前基本網站 + 合併的更新項目 = 基本網站的新版本

## 使用更新

每次在編輯器中編輯網站時，都會用到更新項目。您在編輯器中進行的任何變更都會成為該更新項目的一部分。

建立並使用更新的步驟如下：

1. 在「網站」頁面中，選取網站並從滑鼠右鍵功能表中選擇**開啟**，或按一下動作列中的 。
2. 將「編輯」開關  設定為「編輯」。
3. 如果這是網站的第一個更新，請輸入更新的名稱和選擇性的描述，然後按一下**確定**。名稱中可以包含字母、數字、底線 ( \_ ) 以及連字號 (-)。如果您輸入空格，系統會自動以連字號取代。若要使用現有的更新項目，請按一下清單中的更新項目。
4. 網站名稱和目前的更新會顯示在編輯器的頂端列中。若有多個更新，您可以從更新清單中選取不同的更新來進行切換。
5. 視需要新增及變更頁面內容。可使用不同的顯示大小選項，查看頁面在不同裝置上的顯示情形。
6. 完成編輯後，請按一下**儲存**。準備好將變更併入基本網站時，便可確認更新。您還是需要發布網站，網站使用者才能見到變更。請參閱[發布網站變更](#)。

## 編輯秘訣與技巧

以下資訊可協助您開始使用編輯器。

- [隱藏或顯示提要欄位](#)
- [瀏覽](#)
- [管理頁面](#)
- [拖放編輯](#)
- [元件](#)
- [調整元件大小和間距](#)
- [可編輯的版面配置內容](#)
- [樣式和格式](#)
- [還原您的變更](#)

### 隱藏或顯示提要欄位

按一下  即可顯示可供管理及編輯頁面和頁面內容的選項。

將  設為**編輯**即可存取提要欄位。按一下  即可隱藏提要欄位，以增加預覽頁面時的檢視區域。

## 瀏覽

若要使用網站樹狀結構選取另一個頁面，請按一下提要欄位中的 。使用頁面清單頂端的搜尋方塊來搜尋頁面。您也可以使用頁面本身的網站導覽或連結。

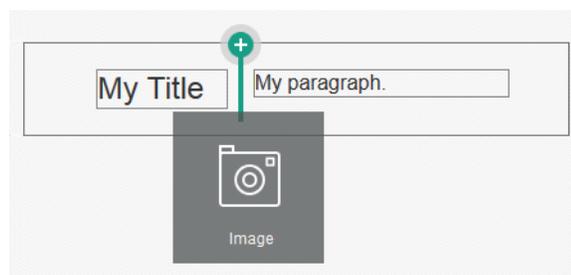
當您切換至另一個頁面時，會儲存您對頁面進行的任何變更。您也可以按一下**儲存**，將變更儲存在目前的更新項目中。

## 管理頁面

- 若要新增頁面，請選取要新增頁面的層級或分支，然後按一下「**新增頁面**」；若要新增子項頁面，請按一下 ，再按一下 。「新頁面」會新增至網站樹狀結構底端，而且系統會提示您命名頁面以及指定其他設定值。
- 若要刪除頁面，請選取頁面，然後按一下 。
- 若要移動頁面，請選取頁面，再將其拖曳至新的位置。您也可以將頁面剪下並貼到新的位置。若要剪下頁面，請按一下 ，再按一下 。若要貼上頁面，請選取要貼上頁面的分支，然後按一下 ，再按一下 。
- 若要複製頁面，請選取頁面，然後按一下 ，再按一下 。若要貼上頁面，請選取要貼上頁面的分支，然後按一下 ，再按一下 。
- 若要變更頁面設定值，例如描述資料、標頭、頁尾和其他選項，請按一下 。請參閱[變更頁面設定值](#)。

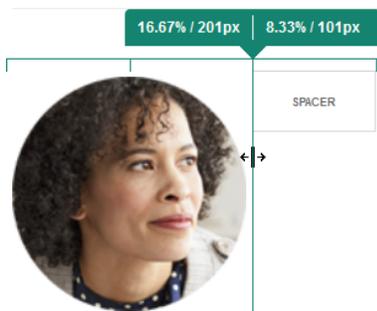
## 拖放編輯

若要新增提要欄位中的元件或要移動頁面中的元件，請按一下該項目，然後將其拖放到頁面上的位置。當您將項目拖曳至頁面時，會顯示可用槽位界限和任何現有項目。位置列  可指出新內容的位置 (上、下、左、右)：



一個槽位可以有多個項目，而且只要將頁面上的項目拖曳至新位置即可移動項目。您也可以調整某個槽位中之兩個元件的相對寬度，方法為在兩個元件之間的邊界按一下並拖曳就可以調整。元件會貼齊至元件上方顯示之「尺規」中指示的下一個格線。

每個元件的大小都會以像素和槽位中可用空間的百分比表示。若要將寬度調整為格線定義值以外的值，請按住 **Ctrl** 鍵，然後按一下並拖曳元件的邊界。



## 元件

將元件置於頁面中之後，就可以調整對齊方式、間距以及其他特性，方法為選取項目、按一下該項目的功能表圖示 ，然後選擇**設定值**。如果您按一下元件名稱，而非功能表圖示，則可以看見與選取適用於槽位和元件群組 (如果目前的元件是元件群組的一部分) 的功能表圖示。如果您選取其他頁籤，就可以見到該元素的功能表圖示及設定它的特性：



主題設計者可以指定主題所包含的元件，因此某網站上可使用的部分元件在另一個網站上可能會無法使用。主題設計者也可以指定特定頁面版面配置的特定槽位中可以有哪一些元件。如果特定槽位中不允許有某個元件，位置列的色彩和符號都會改變 (減號) ，並且顯示類似下面的訊息：



## 調整元件大小和間距

您可以將許多元件的大小 (例如圖庫或影像) 調整為不同的 **CSS** 單位，也可以調整元件周圍的間距。例如：

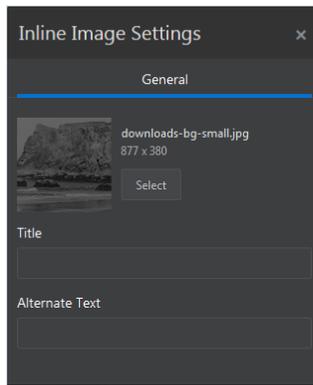
- **px (像素)**：預設值。如果只有指定數值，會假設以像素為單位。
- **% (百分比)**：將項目設為其父項 **HTML** 元素大小的百分之幾。範例：**25%**
- **em**：使用 **em** 空格設定元件大小。範例：**20em**
- **vw**：將大小設為檢視區域寬度的百分之幾，取決於檢視區域的大小。範例：**10vw**

## 可編輯的版面配置內容

主題設計者可以將「內建」內容 (例如著作權聲明) 新增至無法在編輯器中編輯的頁面版面配置。主題設計者也可以將簡單文字和影像內容指定為可編輯，包括數位資產。這可讓提供者變更文字或影像內容，但不能更改位置或其他版面配置屬性。可編輯的文字元素具有選項功能表，其中包含指定粗體、斜體和底線文字以及變更或移除連結的選項：



可編輯的圖形元素具有設定值圖示 ，以及可供指定影像、標題及替代文字的面板。



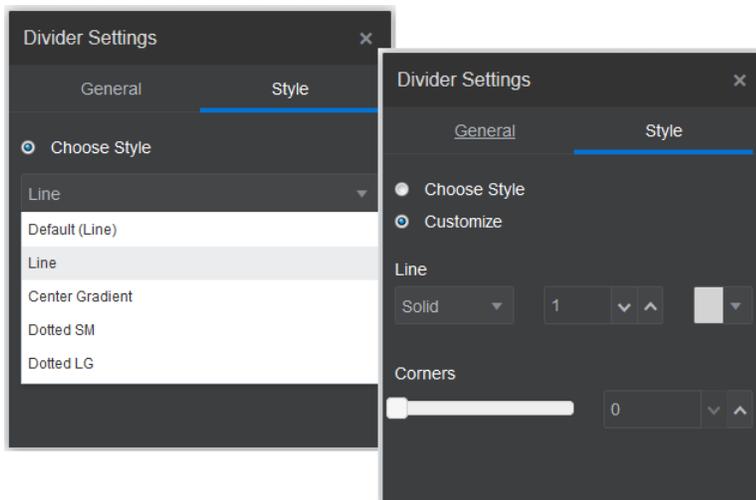
包含此文字或影像的框架會進行調整，以容納文字的長度或影像的大小。

### 備註：

您所做的變更只會套用在目前的頁面。原始內容與版面配置會一併儲存在主題中，原始內容將會是第一次將此版面配置套用至頁面時的預設值。

## 樣式和格式

大多數的元件都有一或多個由主題定義的基本樣式，以指定元件外觀的各個層面。您可以輕鬆地切換樣式或覆寫樣式。若要選擇可用的樣式，請開啟元件的「設定值」面板，依序按一下**樣式**頁籤和**選擇樣式**，然後從功能表中選擇一個樣式。若要針對樣式定義中所指定的特性指定您自己的值，請按一下**自訂**，然後指定格式選項。



### 還原您的變更

使用  即可還原您最後在編輯器中所做的編輯或變更。您可以多次重複使用還原，以還原一系列的變更。

使用  即可重新套用使用還原所回復的最近變更。若您連續多次使用還原，可多次使用重做。

您可以還原目前更新中的內容、樣式和頁面組織變更。部分動作不會包含在還原鏈結中：

- 切換至不同的更新時，會重設還原鏈結，您便無法還原在先前的更新中所做的變更。在同一個更新中，即使您已儲存變更，仍然可以將其還原。
- 若在編輯器中變更檢視，例如切換頁面或變更指定頁面的大小，您必須手動還原那些類型的變更。
- 若您編輯的是文字元件，例如標題或段落，則文字編輯器有自己的還原鏈結。離開文字編輯器後，就無法再還原那些變更。

## 使用樣式和格式

大多數的元件都有一或多個由主題定義的基本樣式，以指定元件外觀的各個層面。您可以輕鬆地切換樣式，或者使用您選擇的選項來覆寫樣式。

1. 瀏覽至您要編輯的頁面，並確定  已設為**編輯**。
2. 若要設定元件的基本樣式，請按一下元件的功能表圖示 ，然後選擇**設定值**。按一下**樣式**頁籤。
  - 若要使用網站主題的樣式，請按一下**選擇樣式**，然後從功能表中選擇所需樣式。樣式是用來定義個別元件，因此樣式清單會因元件而有所不同。例如，影像的樣式與段落的樣式不同。
  - 若要指定您自己基本格式選項，請按一下**自訂**並指定格式選項。
3. 若要複製基本樣式並貼到一或多個相似的元件，請按一下元件的功能表圖示 ，然後選擇**複製元件樣式**。按一下類似元件的功能表圖示 ，然後選擇**貼上元件樣式**。

- 若要設定標題或段落元件內文字的格式，請按一下文字元件。含有格式設定選項的工具列會隨即顯示。選取您要設定格式的文字，然後選取任何一個選項，例如字型、色彩或對齊方式。系統會立即套用您所做的變更。若要移除格式設定，請選取要移除格式設定的文字並按一下 。格式變更會覆蓋基本樣式。如果您變更基本樣式，覆寫的樣式依舊會保留。

## 自訂網站設定值

您可以自訂網站設定值，以新增網站圖示、控制器檔案、網站地圖、機器人程式檔案和輔助檔案，以及指定地圖提供者。

這些設定值都會儲存在某項更新中，直到您確認該項更新為止。發布之後，這些檔案會儲存在主題的根資料夾中，讓使用該主題的所有網站都可使用這些檔案。

變更整個網站的圖示和連結行為：

- 開啟網站進行編輯。
- 先按一下提要欄位中的 ，然後按一下  網站。
- 在**我的最愛圖示**區段中，選擇一個當網站在瀏覽器中或不同平台 (例如行動裝置) 上最小化時，用來代表該網站的影像。此圖示必須儲存成您可以存取的數位資產。按一下**選取要上傳的檔案**，瀏覽該圖示並選取它，然後按一下**確定**。您需要自訂我的最愛圖示，讓圖示能夠在所有瀏覽器上運作：
  - 對於 **Chrome 和 Safari**，控制器檔案中必須有我的最愛圖示的參照 (包括網站前置碼)，如此範例所示：

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

對於 **Internet Explorer 11 和 Firefox**，頁面樣板中必須包含類似的項目。但是，頁面樣板是主題的一部分，可能會在多個網站中使用該樣板。所以不能使用固定的網站前置碼，必須改為使用權杖。請參閱此範例：

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

權杖將會在頁面發布時和網站前置碼交換。

請注意，**Internet Explorer 和 Firefox** 會從控制器載入我的最愛圖示，然後再從頁面樣板載入一次。因此，除非從控制器和頁面樣板參照相同的圖示，否則圖示將會閃爍。**Chrome 和 Safari** 只會從控制器載入參照的圖示。

- 在**控制器檔案**區段中，您可以新增一個更改瀏覽器處理連結要求方式的檔案。除了參照我的最愛圖示之外，也可以新增 **OpenGraph** 標記或描述資料標記，以用於網站管理員網站驗證，或用於在社群媒體中公用網站。您可以下載預設控制器檔案並進行編輯，或上傳您自己的檔案。此檔案必須儲存成您可以存取的數位資產。按一下**選取要上傳的檔案**，瀏覽該檔案並選取它，然後按一下**確定**。

例如，如果網站包含「Facebook 分享」按鈕，您可能會想要提供 Facebook 可以用來在 Facebook 中顯示您網站相關詳細資訊的描述資料，如此範例所示：

```
<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"/>
```

您可以新增 Google 網站管理員驗證標記，類似此範例所示：

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

5. 在**網站地圖和機器人程式檔案**區段中，您可以上傳自訂網站地圖和機器人程式檔案。*網站地圖*是一個 XML 檔案，可用來列出網站的 URL 與每個 URL 的相關資訊 (例如上次更新時間)。*機器人程式檔案*是一個文字檔，您可以建立此檔案來指示搜尋引擎機器人程式如何編製您網站上頁面的索引。這些檔案必須儲存成您可以存取的數位資產。按一下**選取要上傳的檔案**，瀏覽該檔案並選取它，然後按一下**確定**。
6. 在**輔助檔案**區段中，您可以視需要上傳其他檔案，例如驗證網站擁有權所需的檔案。此檔案必須儲存成您可以存取的數位資產。按一下**選取要上傳的檔案**，瀏覽該檔案並選取它，然後按一下**確定**。
7. 您可以為地圖元件和使用的連結選擇提供者。選取 **Oracle Maps** 或 **Google 地圖**。
8. 當您發布更新時，系統會發布變更並應用。

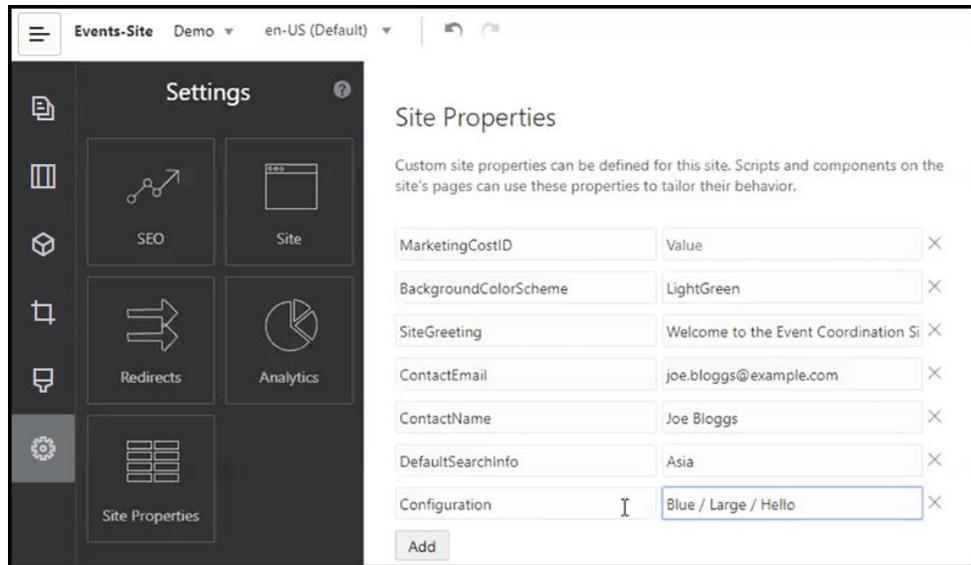
## 新增自訂網站特性

您可以使用名稱/值組形式將自訂特性新增至網站。這些特性會儲存在網站中，而且可用於網站頁面上的命令檔和元件。這些特性可協助參數化或自訂網站，而不需要變更基礎命令檔和元件程式碼。

例如，自訂特性可用來變更頁面背景色彩、縮小搜尋結果範圍、填入清單，以及在一般控制網站相依變數中。

編輯網站時，會使用設定值面板來新增自訂網站特性。

1. 開啟網站進行編輯。
2. 先按一下提要欄位中的 ，然後按一下  **網站特性**。
3. 按一下**新增**。
4. 輸入自訂網站特性的名稱和值。您最多可以增加 50 個自訂網站特性。名稱欄位有 200 個字元限制，而值欄位有 2000 個字元限制。



5. 按一下名稱/值組旁邊的 **X** 即可予以刪除。
6. 完成新增或移除您的自訂網站特性時，請按一下**確認**。

 **備註：**

變更不會與現有自訂網站特性合併。確認自訂網站特性變更時會覆寫基本網站中的任何現有自訂網站特性。

定義之後，自訂網站特性可用於網站和網站元件的所有命令檔，例如頁尾中或支援 **SCSMacro** 擴充之元件 (例如內容清單) 的「其他查詢字串」欄位，或透過標題和段落元件中的權杖。

在程式實際執行以及設計時，此命令檔會與 **SCSRenderAPI** 搭配使用。此外也可以透過 **SCSCompileAPI** 將自訂網站特性值用於樣板編譯器中的版面配置和元件程式碼。此 **API** 的新函數 **getCustomSiteProperty** 與 **SCSRenderAPI** 類似，允許版面配置和元件程式碼讀取自訂區段特性的值。

例如，假設您要定義使用 **SCSRenderAPI** 的自訂區段版面配置，來呼叫和擷取標頭、聯絡人名稱和聯絡人電子郵件的自訂網站特性。下列命令檔使用附加至 **DOM** 的自訂網站特性來建置 **HTML** 字串。

```
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 );
    }
},
};
};

```

您已新增自訂網站特性的網站建立樣板時，以及從具有自訂網站特性的樣板建立網站時，都會保留自訂特性。

## 使用表格

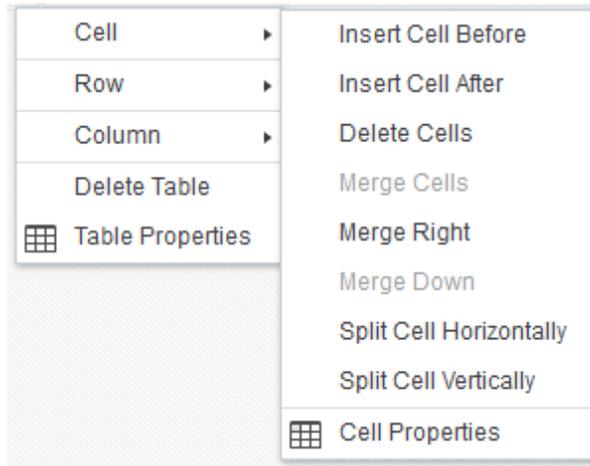
您可以在段落元件內包括您所建立或從現有 HTML 來源貼上的表格。

1. 前往您要編輯的頁面，並確定  已設為**編輯**。
2. 新增元件至頁面，或在現有段落元件按一下。
3. 若要新增另一個 HTML 頁面的表格，只要按一下並拖曳即可選取該表格，然後將其複製並貼到段落元件中。若要在目前的游標位置建立新表格，請按一下 。選擇資料列和資料欄數目、寬度及其他格式設定選項。您可以在表格上方的中間新增選擇性的**說明文字**和選擇性的**摘要**，以提供輔助技術 (例如自動化螢幕助讀程式) 額外相關內容。
4. 完成後按一下**確定**以關閉視窗。稍後若要變更這些表格設定值，請在表格按一下滑鼠右鍵，然後選擇**表格特性**。若要刪除此表格，請在表格按一下滑鼠右鍵，然後選擇**刪除表格**。

### 備註：

如果針對不同顯示大小 (例如行動裝置螢幕) 進行開發，您可以使用 **StarterTheme design.css** 檔案中的程式碼來**建立響應式表格**，在行動裝置上顯示時可啟用資料列堆疊處理。

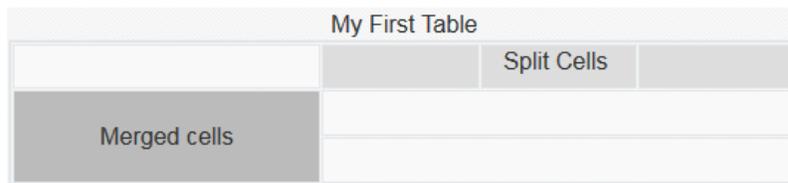
5. 若要新增、移除或修改特定資料列、資料欄或儲存格，請在資料列、資料欄或儲存格按一下滑鼠右鍵，然後從選項功能表中選擇適當的選項。例如：



 **備註：**

您一次只能變更一個儲存格的儲存格特性。

- 您可以合併及分割儲存格，以建立複雜的表格版面配置。例如：



- 若要將一個儲存格分割成兩個儲存格，請在該儲存格按一下滑鼠右鍵並選擇**儲存格**，接著再選擇**水平分割儲存格**或**垂直分割儲存格**。
- 若要合併兩個水平儲存格，請在左邊的儲存格按一下滑鼠右鍵並選擇**儲存格**，接著再選擇**向右合併**。
- 若要合併兩個垂直儲存格，請在頂端儲存格按一下滑鼠右鍵並選擇**儲存格**，接著再選擇**向下合併**。

- 使用一般格式功能表選項，可變更儲存格內文字的格式和對齊方式。

## 上傳網站檔案

您隨時都可以使用 Oracle Content Management 介面來上傳圖像和文件，以便與網站搭配使用。使用背景影像，以及使用影像、圖庫和文件元件時，也可以從「網站產生器」內上傳檔案。

網站所要使用的影像和文件檔案，可以是儲存在網站或您可存取的其他位置中的影像。您也可以使用與您共用的影像，或是您從本機或網路檔案位置上傳的影像。

### 上傳檔案

從本機或網路位置上傳一或多個檔案：

1. 按一下**文件**，然後前往要儲存檔案的位置。按一下**建立**，在目前的位置中建立新的資料夾。
2. 按一下**上傳**。
3. 尋找並選取一或多個檔案，然後按一下**開啟**。

### 在編輯器內上傳檔案

使用背景影像或像文件或影像之類的元件時，若要使用的檔案不在 Oracle Content Management 位置中，您可以直接上傳檔案。

例如，從本機或網路位置上傳一或多個檔案，以便與圖庫元件搭配使用：

1. 若要將影像新增至圖庫，請按一下圖庫的功能表圖示 ，選擇**設定值**，然後按一下**一般**頁籤上的**影像**。
2. 按一下**新增影像**。
3. 前往要儲存檔案的位置，或是按一下**建立**，在目前的位置中新增資料夾。
4. 按一下**上傳**。
5. 尋找並選取一或多個檔案，然後按一下**開啟**。
6. 選取儲存區域中的一或多個影像，然後按一下**確定**。

#### 備註：

視窗會顯示所有可用的檔案。選擇適用於相關資訊環境的檔案類型。例如，假設您要選擇影像檔，請選取有效的影像格式 (GIF、JPG、JPEG、PNG 或 SVG)。若要連結至檔案，請選取**使用原始檔案參照**，而不是**將檔案複製到網站**。如果您未選取此選項，系統就會在網站儲存檔案的複本，然後從網站參照。

# 6

## 在網站中使用樣板和主題

樣板含有開始建置網站所需的全部項目，包括網站程式碼架構、內含樣本頁面和內容的預設網站、包含樣式、資源 (例如影像)、甚至自訂元件的主題。主題定義網站的一般外觀與感覺 (整體網站)，包括色彩配置、字型大小、字型類型和頁面背景。

### 樣板

- [瞭解樣板](#)
- [從網站建立樣板](#)
- [變更樣板詳細資訊](#)
- [變更樣板規定](#)
- [變更樣板狀態或對象](#)
- [管理樣板](#)
- [匯出及匯入樣板](#)

### 主題

- [瞭解主題](#)
- [管理主題](#)
- [發布主題](#)

## 瞭解樣板

樣板含有您開始建置網站所需的一切項目，包括網站程式碼架構、含有範例頁面和內容的預設網站、含有樣式設定的主題、資源 (例如影像)，甚至是自訂元件。

### 預設樣板

Oracle Content Management 提供一些可供您用來建立網站的樣板。只要選取一個樣板並命名網站，您就可以立即開始新增內容。這些樣板通常會在服務起始後由服務管理員安裝。

樣板	描述
空白樣板	空白樣板是包含標頭、主體及頁尾槽位的單一頁面，可讓您充分自由地運用自己的設計。
並行展示櫥窗	「並行展示櫥窗」樣板具有靠右對齊的水平功能表樣式、簡單的頁面版面配置與最上層頁面，這些頁面會列出相關的視訊資產，後面接著巢狀的詳細資訊頁面以各自展示特定的視訊。它主要針對標準功能表元件提供，用於顯示目前登入的使用者。
新產品發表	「新產品發表」樣板具有一個靠右對齊的水平功能表和多個版面配置，可供您填入案例研究、功能詳細資訊、價格、貴公司資訊和聯絡資訊。首頁中有一個輪播式橫幅影像加文字。

樣板	描述
產品與服務總覽	「產品與服務總覽」樣板具有多種版面配置，可供您填入案例研究、產品方案詳細資訊、貴公司資訊、隱私權原則等資訊。首頁中有一個輪播式橫幅影像加文字。此樣板可完全自動響應。
入門樣板	您可以使用「入門」樣板來建立自己的現成網站解決方案。入門樣板提供簡單但功能完整的範例，讓您可以探索與擴充元件和互動。此樣板包括網站程式碼架構、包含範例頁面與內容的預設網站、具有樣式的主題、資源 (例如影像) 以及具備觸發程式和動作功能的自訂元件。範例頁面包括建立含有資源連結 (以提供更詳細的資訊) 樣板的相關資訊。
學習展示顯示	這些樣板彼此色彩各異。這些是響應式樣板，針對群組內的自訂 JavaScript 功能表元件提供。它們的適用對象為具有一些標準元件的自訂元件群組。它們也會顯示登入的使用者。
存取知識相關	這些樣板彼此色彩各異。這些是響應式樣板，針對群組內的自訂 JavaScript 功能表元件提供。它們的適用對象為具有一些標準元件的自訂元件群組。
共用	此樣板是一個以單一長頁面為特色的網站。它是響應式樣板，針對瀏覽至頁面上的位置而不是瀏覽至個別頁面的自訂 JavaScript 功能表提供。
JET 入門樣板	Oracle JET (JavaScript Extension Toolkit) 是一種使用一組開源 JavaScript 程式庫的模組化開源工具程式。請參閱 <a href="#">Oracle Jet</a> 以取得完整詳細資訊。

您也可以從現有網站建立樣板；或是匯出現有樣板、離線進行修改，然後將其匯入成新樣板。您的組織可能已具備可供您使用的樣板。

### 樣板如何運作

建立網站時，會依照下列方式來使用樣板：

- 如果已啟用網站治理，網站建立之前，可能需要先經過核准。請參閱[瞭解網站治理](#)。
- 樣板中的預設網站會複製至新網站，以作為您頁面的起點。
- 所有的必要支援檔案都會複製到新網站。
- 如果 themes 資料夾中沒有樣板主題，便會將主題複製至 themes 資料夾。網站會從主題在 themes 資料夾中的位置來參照主題。如果主題存在，則新網站只會參照現有的主題。
- 如果有一些自訂元件不存在於 components 資料夾中，便會將它們複製至 components 資料夾。網站會從元件在 components 資料夾中的位置來參照元件。如果有任何元件存在，則新網站會參照現有元件。

### 建立自訂樣板

如果您是 Web 開發人員，則樣板會將您建構網站所需的一切項目 (包括網站、版面配置、導覽、範例內容等) 收集在套件中。您可以將元件和互動新增到網站中，以提供符合業務需求的現成網站解決方案。

樣板以資料夾結構表示，您可以像使用其他資料夾般使用此資料夾結構。有些樣板元素 (例如主題和自訂元件) 是從它們在 Oracle Content Management 中的關聯位置參照。例如，樣板從可用的主題清單參照關聯的主題，其方式與網站參照主題的方式相同。

如果您從現有網站建立樣板，則新樣板會使用網站的複本作為其預設網站。樣板會參照網站所使用的主題，以及網站頁面中所使用的任何自訂元件。主題和自訂元件並不會複製到樣板中，而是以被網站參照的相同方式進行參照。

 **備註：**

樣板會反映建立樣板時所使用的網站。對用於建立樣板的網站所做的進一步變更，不會反映在使用此樣板而儲存的網站。

您可以建立可供用於共用內容模型的內容樣板，其中包括支援內容模型 (例如範例內容) 所需的內容版面配置、內容項目以及數位資產。內容樣板應該從含有已發布內容項目和數位資產的網站建立。注意，若從含有內容的套件匯入網站樣板，並不會建立內容類型。若從內容樣板建立網站，才會建立內容類型。因此，內容類型並不是由匯入網站樣板的使用者擁有，而是由從該樣板建立第一個網站的使用者所擁有。該使用者可視需要與其他使用者共用該樣板。根據您服務的 **Content Management Cloud** 訂閱類型和開始日期而定，有可能不提供這項功能。

匯入樣板時，所有樣板元素 (包括主題複本和任何元件) 都會收集在一個您可以下載並離線使用的樣板套件中。

匯入離線修改過的樣板時，如果樣板、主題或者自訂元件名稱或 ID 已經存在，系統就會提示您解決衝突。系統會提供您建立新樣板、主題或自訂元件的選項，或在某些情況下，您也可以將匯入的版本覆寫現有的樣板、主題或自訂元件。請參閱[開發樣板](#)。

您也可以將樣板匯入特定儲存區域。如果這麼做，會讓您選擇是要更新或複製現有資產。請參閱[將樣板匯入特定儲存區域](#)。

### 共用樣板

不論您是透過匯入、複製或從網站建立樣板，建立的樣板都要等到您共用後，才可供其他人使用。

 **備註：**

Oracle Content Management 所提供且由管理員安裝的樣板也是如此。如果您未見到任何樣板，請洽詢您的管理員。它們可能尚未與您共用。

若您是第一次與使用者共用樣板，系統會自動將關聯的主題和所有關聯的自訂元件與該使用者共用，而且授予使用者下載者角色，讓使用者在從該樣板建立網站時，可以使用這些關聯的主題和自訂元件。對樣板中該使用者之角色所做的後續變更，並不會更新關聯之主題或自訂元件的共用資訊。

如果已啟用網站治理，您可以透過樣板規定提供樣板。請參閱[瞭解網站治理](#)及[變更樣板規定](#)。

## 從網站建立樣板

如果您的網站要當成其他網站的起始點，則可以從該網站建立樣板。

### 備註：

若要從網站建立樣板，您必須擁有該網站的下載者、提供者或管理者角色。此外，管理員必須啟用**建立**功能表中的選項。如果您未在樣板頁面中見到**建立**功能表，請洽詢您的管理員。

以下步驟說明如何從網站建立樣板。您也可以匯入離線時所建立或修改的樣板套件。請參閱[匯出及匯入樣板](#)。

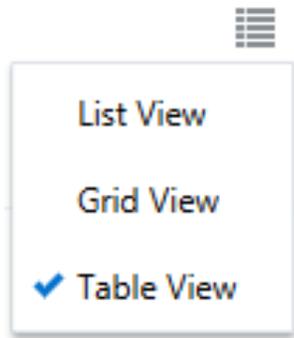
### 備註：

如果您用來建立樣板的網站使用來自多個儲存區域的內容，則需要使用 **OCE** 工具程式進行此作業。請參閱[使用 OCE 工具程式進行開發](#)和[使用 cec 命令行公用程式](#)。

使用 Oracle Content Management Web 介面從網站建立樣板：

1. 按一下**網站**並選取要使用的網站。
2. 從滑鼠右鍵功能表中選擇**建立樣板**，或按一下動作列中的 。
3. 輸入樣板的名稱。名稱中可以包含字母、數字、底線 ( \_ ) 以及連字號 (-)。如果您輸入空格，系統會自動以連字號取代。  
請勿使用下列作為樣板、主題、元件、網站、或網站頁面的名稱：authsite、content、pages、scstemplate\_\*、\_comps、\_components、\_compsdelivery、\_idcservice、\_sitescloud、\_sitesclouddelivery、\_themes、\_themesdelivery。您可以使用下列名稱作為網站頁面的名稱，但不可以作為樣板、主題、元件或網站的名稱：文件、網站。
4. 輸入樣板的選擇性描述。
5. 選擇是否將在樣板中包含未發布的內容項目和數位資產。
6. 完成時，請按一下**建立樣板**。

樣板建立之後，其名稱會顯示在樣板清單中。若要檢視所有樣板，請依序按一下**開發人員**和**檢視所有樣板**。您可以按一下檢視圖示並從清單中選取選項，來控制樣板的顯示方式。



若要檢視與樣板相關聯的資料夾和檔案，請從滑鼠右鍵功能表中選擇**開啟**，或按一下動作列中的 。若要檢視或變更樣板的名稱、描述和其他詳細資訊，請按一下樣板名稱、從滑鼠右鍵功能表中選擇**詳細資訊**，或按一下動作列中的 。

新樣板使用網站的複本作為其預設網站。樣板會參照網站所使用的主題，以及網站頁面中所使用的任何自訂元件。主題和自訂元件並不會複製到樣板中，而是以被網站參照的相同方式進行參照。

樣板若是從企業網站建立，預設將會為從此樣板建立的所有新網站，選取原始網站所使用的本地化規定和預設語言。

樣板會反映建立樣板時所使用的網站。未來對用於建立樣板的網站所做的任何變更，不會反映在使用此樣板而儲存的網站。

## 變更樣板詳細資訊

「樣板詳細資訊」頁籤會顯示樣板名稱、作者、描述、樣板主題、樣板中包含的自訂元件，以及樣板預覽影像。如果已啟用網站治理，您也會看到樣板的狀態 (建立網站時是否可使用樣板)。

如果您有適當的權限，便可以變更或更新樣板特性，例如名稱和描述。您也可以新增或移除樣板的預覽影像。

如果您是樣板的建立者 (即樣板擁有者)、網站管理員，或有人與您共用樣板且授予您「提供者」或「管理者」角色，您就能夠更新樣板詳細資訊。

如果您新增預覽影像，這些檔案會儲存在樣板的資產資料夾中。這些檔案不會出現在任何關聯的網站中，但樣板擁有者可以提供樣板本身的相關資訊。

檢視或變更樣板詳細資訊：

1. 依序按一下**開發人員**和**檢視所有樣板**。
2. 選取樣板並從滑鼠右鍵功能表中選擇**詳細資訊**，或按一下動作列中的 。
3. 依需求編輯資訊。如果無法編輯資訊，表示您沒有「提供者」或「管理者」角色。
4. 若要新增預覽影像，請按一下 。找出想要使用的影像，或者按一下**上傳**來上傳儲存在本機的影像。影像至少應為 **180x135px** (外觀比例為 **4:3**)。較小的影像可能無法顯示出最佳效果，過大的影像則可能會影響效能。

如果所有的預覽影像位置都已填滿，您必須先刪除影像後，才能夠加入新的影像。若要刪除預覽影像，請按一下影像的 。

5. 按一下**儲存**。

## 變更樣板規定

「樣板規定」頁籤會顯示樣板類型 (標準或企業)、需要的核准以及從樣板建立之網站的最低安全性。您也可以查看樣板的狀態 (建立網站時是否可使用樣板)。

只有在已啟用網站治理，並且您是網站管理員時，才會看到「樣板規定」頁籤。請參閱[瞭解網站治理](#)。

檢視或變更樣板規定：

1. 依序按一下**開發人員**和**檢視所有樣板**。
2. 選取樣板並從滑鼠右鍵功能表中選擇**詳細資訊**，或按一下動作列中的 。
3. 選取**規定**頁籤。
4. 選取從此樣板建立的網站要求是需要網站管理員的核准、可自動核准，還是需要特定人員的核准。若要限定由特定人員進行核准，請開始輸入要新增為核准者之人員或群組的名稱或電子郵件，然後從搜尋結果中選取人員或群組。若要移除核准者，請按一下其名稱旁邊的 **X**。
5. 選取從此樣板建立的網站所需的最低安全性。網站建立者可以視需要為其網站選取較高層級的安全性。
  - **特定服務使用者** - 只有可登入此 Oracle Content Management 執行處理的所選使用者，才能存取此網站。網站建立者可在網站建立後，選取 Oracle Content Management 使用者。請參閱[變更網站安全性](#)。
  - **特定雲端使用者** - 只有可登入您網域的選定使用者，才能存取此網站。網站建立者可在網站建立後，選取雲端使用者。請參閱[變更網站安全性](#)。
  - **服務使用者** - 所有可登入此 Oracle Content Management 執行處理的使用者，都能存取此網站。
  - **雲端使用者** - 所有可登入您網域的使用者，都能存取此網站。
  - **所有人** - 所有人都無須登入即可存取此網站。
6. 選取一個決定使用樣板之網站的到期規定。網站管理員可以指定是要讓到期的網站離線還是將它刪除。網站擁有者和管理者會在網站到期之前收到電子郵件通知，讓他們有機會可以延長到期時間。如果他們未延長到期期限，則會在網站離線或刪除後收到電子郵件通知。此時，他們可以延長到期期限，並視需要讓網站再上線或從資源回收將其回復。到期規定選項包括：
  - **永不**
  - **1 年**
  - **2 年**
  - **自訂**  
若選取**自訂**，即可將到期期限設為不到 1 年或超過 2 年。

 **備註：**

網站建立之後，網站管理員可以在**網站特性**對話方塊中變更網站的到期規定。

7. 若是編輯企業樣板，可以選取易記 URL 網站前置碼值的建立方式。此前置碼將附加至內容項目網址代稱值 (特定頁面或資產 URL 的部分)。您可以選擇讓系統根據網站名稱自動產生前置碼，或者讓使用者輸入前置碼。
8. 若您在編輯企業樣板，請選擇用來決定網站所需語言的本地化規定。
9. 完成之後，請按一下**儲存**。

## 變更樣板狀態或對象

樣板的狀態 (建立網站時是否可使用樣板) 和可使用此樣板建立網站的使用者會在「樣板對象」頁籤中顯示。

只有在已啟用網站治理，而且您為網站管理員時，才會見到「樣板對象」頁籤。請參閱[瞭解網站治理](#)。

檢視或變更樣板對象：

1. 依序按一下**開發人員**和**檢視所有樣板**。
2. 選取樣板並從滑鼠右鍵功能表中選擇**詳細資訊**，或按一下動作列中的 。
3. 選取**對象**頁籤。
4. 若要變更樣板狀態，請按一下狀態開關。您必須讓樣板成為作用中樣板，使用者才能使用它們建立網站。依照預設，啟用樣板後，所有網站建立者均可使用該樣板。若要變更可使用樣板的使用者，請在此頁籤中變更設定。
5. 選取可使用此樣板建立網站的使用者。
  - **特定人員** — 只有選取的人員可使用此樣板來建立網站。開始輸入要允許其使用此樣板之人員或群組的名稱或電子郵件，然後從搜尋結果中選取人員或群組。若要移除人員或群組，請按一下其名稱旁邊的 **X**。
  - **所有人** — 任何人都可以使用此樣板建立網站。
6. 完成之後，請按一下**儲存**。

## 管理樣板

您可以複製、重新命名及刪除樣板資料夾，方法就跟管理任何其他資料夾一樣；但您在匯入或共用樣板時，必須考慮一些特殊事項。

依序選取**開發人員**和**檢視所有樣板**，以開啟「樣板」頁面。使用「樣板」頁面的動作列或滑鼠右鍵功能表選項執行下列任一工作。

工作	描述
建立樣板	如果您的網站要當成其他網站的起始點，則可以從該網站建立樣板。
	<div data-bbox="803 373 938 415" data-label="Section-Header"><b>備註：</b></div> <div data-bbox="849 432 1343 636" data-label="Text"> <p>您的網站角色必須是下載者、提供者或管理者，且您的管理員必須啟用該選項。如果您未在樣板頁面中見到<b>建立</b>功能表，請洽詢您的管理員。此外，如果您用來建立樣板的網站使用來自多個儲存區域的內容，則需要使用 <b>OCE 工具程式</b> 進行此作業。請參閱 <a href="#">使用 OCE 工具程式進行開發</a> 和 <a href="#">使用 cec 命令行公用程式</a>。</p> </div>
	<p>在樣板頁面中，按一下 <b>建立</b>，然後選取 <b>從現有網站建立</b>。選取一個網站、為您的樣板命名，然後按一下 <b>建立</b>。請參閱 <a href="#">從網站建立樣板</a>。</p>
檢視和編輯樣板檔案	<p>若要探索組成樣板的資料夾和檔案，請選取樣板，然後從滑鼠右鍵功能表中選擇 <b>開啟</b>，或按一下動作列中的 。</p> <p>如果您是 <b>Web</b> 開發人員，可以下載及修改個別樣板檔案，也可以使用桌面應用程式來同步您在本機系統上所做的變更。您還可以匯出樣板套件（當中包含關聯的主題與任何自訂元件），並在您偏好的開發環境中進行離線處理。樣板以資料夾結構表示，您可以像使用其他資料夾般使用此資料夾結構。</p> <p>匯出樣板時，所有樣板元素（包括主題複本和任何元件）都會收集在一個您可以下載並離線使用的樣板套件中。請參閱 <a href="#">開發樣板</a>。</p>
建立網站	<p>若要使用樣板建立網站，請選取樣板，然後從滑鼠右鍵功能表中選擇 <b>建立網站</b>，或按一下動作列中的 。請參閱 <a href="#">建立網站</a>。</p>
重新命名樣板	<p>選取想要重新命名的樣板，然後從滑鼠右鍵功能表中選擇 <b>重新命名</b>，或按一下動作列中的 。輸入樣板的名稱，此名稱必須與相同伺服器上的任何其他樣板名稱不同。如需命名準則，請參閱 <a href="#">從網站建立樣板</a>。</p>
複製樣板	<p>您可以藉由複製現有樣板並對複本進行變更，以建立樣板。</p>

**備註：**

若是複製樣板，並不會複製該樣板的共用資訊。

選取想要複製的樣板，然後從滑鼠右鍵功能表中選擇 **複製**，或按一下動作列中的 。輸入樣板的名稱，此名稱必須與相同伺服器上的任何其他樣板名稱不同。如需其他的命名準則，請參閱 [從網站建立樣板](#)。進度列就會顯示新樣板名稱和複製狀態。

工作	描述
刪除或回復樣板	<p>如果您具有適當的權限，可以刪除樣板資料夾及其內容。當您刪除樣板時，樣板資料夾與其所有關聯的資料夾和檔案都會移動至資源回收。</p> <p>您若是樣板的建立者 (即為樣板擁有者)，或有人與您共用樣板且授予您提供者或管理者角色，您就能夠刪除或回復樣板。</p> <div style="border: 1px solid #0070c0; padding: 10px; margin: 10px 0;"> <p> <b>備註：</b></p> <p>刪除樣板並不會刪除關聯的主題和自訂元件。</p> </div> <p>若要刪除樣板，選取想要刪除的樣板，然後從滑鼠右鍵功能表中選擇<b>刪除</b>，或按一下動作列中的 。系統會提示您將樣板移動至資源回收。在發生以下情況之前，已刪除的樣板都會保留在資源回收中：</p> <ul style="list-style-type: none"> <li>· 您回復樣板。</li> <li>· 您永久刪除樣板。</li> <li>· 達到資源回收配額。</li> <li>· 系統根據服務管理員所設定的間隔自動清空資源回收。預設值為 90 天。</li> </ul> <p>若要回復樣板，請從「樣板」下拉式清單中選取<b>資源回收</b>，然後選取該樣板並從功能表列或滑鼠右鍵功能表中選擇<b>回復</b>。</p>
匯出或匯入樣板	<p>您可以匯出樣板並離線進行修改，然後再將它匯入成新樣板或取代現有樣板。您也可以匯出樣板，並將它移到另一個執行處理，然後在該處加以匯入。</p> <p>匯出樣板時，您是將樣板以單一 .zip 檔案的形式複製到資料夾。您可以直接從該資料夾下載樣板套件，將其解壓縮並處理個別的檔案。完成處理檔案之後，請建立一個包含該樣板套件的 .zip 檔案並將其匯入，然後選擇覆寫原始樣板或建立新樣板。</p> <p>請參閱<a href="#">匯出及匯入樣板</a>。</p>
共用樣板及管理成員角色	<p>若您是樣板的擁有者或管理者，您可以與其他使用者共用樣板，並指定一個定義使用者可對樣板執行哪些動作的角色。</p> <p>若未啟用治理，指定給使用者的任何角色，都會給予使用者從樣板建立網站的權限。若已啟用治理，您必須啟用樣板，並確保使用者包含在樣板對象內，才能給予他們從樣板建立網站的權限。請參閱<a href="#">變更樣板狀態或對象</a>。</p> <p>樣板的主題和自訂元件都會儲存為個別物件，而且只會在樣板中參照。因此，主題和自訂元件成員是個別管理的。當您與使用者共用樣板時，無論您提供給他們的樣板角色為何，使用者都會自動成為所有關聯主題的「下載者」角色，讓使用者從該樣板建立網站時可以使用那些物件。</p> <p>選取想要共用的樣板，然後從滑鼠右鍵功能表或動作列中選擇<b>成員</b>。</p> <p>輸入一或多個使用者名稱或電子郵件地址，並且指定以下其中一個角色：</p> <ul style="list-style-type: none"> <li>· <b>檢視者</b>：檢視者可以檢視樣板的資料夾和檔案，但是不能變更內容。</li> <li>· <b>下載者</b>：下載者可同時下載檔案，並且將下載的檔案儲存到自己的電腦。</li> <li>· <b>提供者</b>：提供者還可以編輯樣板詳細資訊和樣板檔案、上傳新檔案、刪除檔案以及刪除樣板本身。</li> <li>· <b>管理者</b>：管理者可同時新增使用者及指定其角色。樣板擁有者 (建立者) 會自動被指定管理者角色。</li> </ul>
檢視或編輯樣板詳細資訊	<p>若要檢視樣板詳細資訊，請選取樣板，然後從滑鼠右鍵功能表中選擇<b>詳細資訊</b>，或按一下動作列中的 。樣板詳細資訊包括樣板名稱、作者、支援檔案、描述、主題、自訂元件和縮圖。若您有適當的樣板角色，您就可以編輯詳細資訊。請參閱<a href="#">變更樣板詳細資訊</a>。</p>

工作	描述
檢視或編輯樣板規定 (需啟用治理)	<p>若已啟用治理且您為網站管理員，您就可以檢視樣板規定，例如必要的核准、最低安全性、網站到期日和本地化。選取樣板並從滑鼠右鍵功能表中選擇 <b>詳細資訊</b>，或按一下動作列中的 ，然後選取 <b>規定</b> 頁籤。請參閱 <a href="#">變更樣板規定</a>。</p>
將樣板設為可用於建立網站，並管理可使用樣板建立網站的使用者 (需啟用治理)	<p>若已啟用治理且您是網站管理員，您就可以將樣板設為可用於建立網站，並管理可使用樣板建立網站的使用者。作用中樣板 (可用於建立網站) 的旁邊會顯示 。</p> <p>若要變更狀態或對象，請選取樣板，然後從滑鼠右鍵功能表中選擇 <b>詳細資訊</b>，或按一下動作列中的 。</p> <ul style="list-style-type: none"> <li>· 若要將樣板設為可用於建立網站，請按一下狀態開關加以啟用。</li> <li>· 若要變更可使用樣板建立網站的使用者，請按一下 <b>對象</b> 頁籤。請參閱 <a href="#">變更樣板狀態或對象</a>。</li> </ul>

## 匯出及匯入樣板

您可以匯出樣板並離線進行修改，然後再將它匯入成新樣板或取代現有樣板。您也可以匯出樣板，並將它移到另一個 Oracle Content Management 執行處理，然後在該處加以匯入。

### 匯出樣板

匯出樣板時，基本上將樣板以單一 .zip 檔案的形式複製到 Oracle Content Management 資料夾。您可以直接從該資料夾下載樣板套件，將其解壓縮並處理個別的檔案。完成處理樣板檔案之後，請建立一個包含該樣板套件的 .zip 檔案，然後將它匯入您的網站中，並覆寫原始樣板或建立新樣板。



#### 備註：

匯出樣板時，不會包括該樣板的共用資訊。

匯出樣板：

1. 依序按一下 **開發人員** 和 **檢視所有樣板**。
2. 選取一個樣板，然後從功能表列或滑鼠右鍵功能表中選擇 **匯出**。
3. 瀏覽至現有資料夾，或者按一下 **建立** 並提供名稱和選擇性描述，以建立新資料夾。
4. 勾選資料夾的核取方塊並按一下 **確定**，以選擇該資料夾。

## 匯入樣板

### 備註：

如果您將樣板匯入到不同的伺服器，預設網站中的某些連結在新伺服器相關資訊環境中可能會無效。如果網站使用影像或其他內容的參照連結，而不是將內容直接複製到網站中，新伺服器上將無法使用該內容。即使您將該內容複製到新伺服器，內容也會有不同的內部 ID，因此連結也不會有效。您在匯入樣板時，系統會告知您包含無效參照連結的頁面。

匯入樣板套件：

1. 依序按一下**開發人員**和**檢視所有樣板**。
2. 按一下**建立**，然後選擇**匯入樣板套件**。
3. 如果您已上傳樣板套件，請瀏覽至包含該樣板套件的資料夾並將其開啟。如果您尚未上傳樣板套件，請前往您要上傳樣板套件的所在資料夾，或建立新資料夾。按一下**上傳**，然後找出並選取樣板套件，接著按一下**開啟**。
4. 若要使用樣板，請按一下樣板檔案名稱旁的核取方塊，並按一下**確定**。系統會為樣板、其關聯主題與任何自訂元件建立新資料夾。如果樣板、主題或者自訂元件名稱或 ID 已經存在，系統會提示您解決衝突。屆時您可能需要建立新的樣板、主題或元件，或以匯入的版本覆寫現有項目。

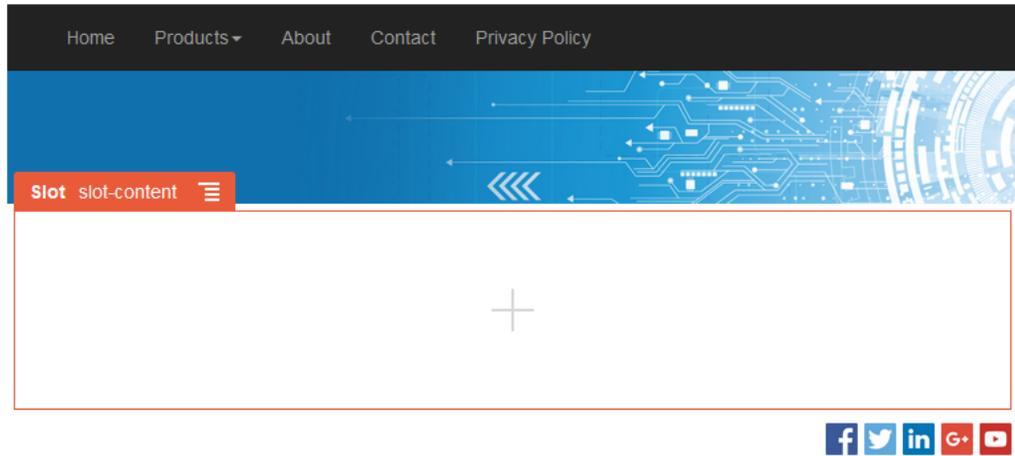
### 將樣板匯入特定儲存區域

您可以選取儲存區域，然後將一個樣板套件專門匯入至該儲存區域。

1. 按一下**內容**，然後從功能表中選取**儲存區域**。
2. 選取您要匯入樣板套件的儲存區域，然後按一下**匯入內容**。
3. 如果您已上傳樣板套件，請瀏覽至包含該樣板套件的資料夾並將其開啟。如果您尚未上傳樣板套件，請前往您要上傳樣板套件的所在資料夾，或建立新資料夾。按一下**上傳**，接著尋找並選取樣板套件，然後按一下**確定**。
4. 選擇是要以新修訂版本更新已經存在於儲存區域中的資產，還是要複製資產，然後按一下**匯入**。
5. 按一下**詳細資訊**，以查看匯入的資產和內容類型清單。完成時，按一下**確定**。

## 瞭解主題

**主題**定義網站的一般外觀與感覺 (整體網站)，包括色彩配置、字型大小、字型類型和頁面背景。主題提供網站各頁面間的視覺一致性。您可以調整設計，以及新增內容來建立可銷售樣式、商標和願景的網站。



主題包括：

- 頁面版面配置
- 階層式樣式表 (CSS)
- 樣式表的變化
- 組態檔
- 定義網站導覽的背景程式碼

每個網站都會使用主題。若從樣板建立網站，會繼承樣板的主題。您可以隨時變更網站的主題。

Oracle Content Management 提供一些樣板，這些樣板包含可供您用來建立網站的主題。若要建立新主題，請複製某個現有主題。您可以下載並修改主題檔案，或是可以使用桌面應用程式並同步您在本機系統所做的變更。如需其他主題建立方式的相關資訊，請參閱[開發主題](#)。

開發人員也可以建立使用特定元件子集的主題。建立使用這類主題的網站時，只會向您顯示該主題中允許使用的元件。因此不需要擔心某個元件在您選擇的主題中是否有效。您仍然可以使用任何隨主題元件內建的元件。

如果網站使用新的未發布主題，則會在第一次讓網站上線時自動發布網站的主題。如果您變更主題，並且想要更新線上網站來顯示變更，則必須明確地發布主題。只有主題擁有者或具有「管理者」權限的使用者才能明確地發布主題。

#### 備註：

如果您發布主題變更，則所有使用該主題的線上網站都會反映此變更。例如，如果您變更主題中的預設字型，並發布主題，則所有使用該主題的網站都將使用新的預設字型。

## 管理主題

與其他資料夾相同，您可以對主題資料夾進行複製、重新命名及刪除，但應特別考量主題，因其可由一個以上的網站所共用。

若要查看所有主題，請依序按一下**開發人員**和**檢視所有主題**。您可以使用「主題」頁面中的功能表列或滑鼠右鍵功能表執行這些工作。

工作	描述
建立新主題	若要建立新主題，請複製某個現有主題。如需其他主題建立方式的相關資訊，請參閱 <b>開發主題</b> 。
複製主題	您可以複製現有主題，然後變更複本來建立新的主題。

 **備註：**

當您複製主題時，不會複製與主題關聯的共用資訊。

選取想要複製的主題，然後選擇功能表列或滑鼠右鍵功能表中的**複製**。輸入未用於其他主題的名稱。名稱中可以包含字母、數字、底線 ( \_ ) 以及連字號 ( - )。如果您輸入空格，系統會自動以連字號取代。按一下**複製**。

請勿使用下列作為樣板、主題、元件、網站、或網站頁面的名稱：**authsite**、**content**、**pages**、**scstemplate\_\***、**\_comps**、**\_components**、**\_compsdelivery**、**\_idcservice**、**\_sitescloud**、**\_sitesclouddelivery**、**\_themes**、**\_themesdelivery**。您可以使用下列名稱作為網站頁面的名稱，但不可以作為樣板、主題、元件或網站的名稱：文件、網站。

共用主題	若您的管理員啟用共用功能，且您是主題的擁有者或管理者，您就可以將您的主題與其他使用者共用。共用主題時，您需指定角色以定義使用者對於該主題的使用權限。
------	--

 **備註：**

當有人與使用者共用樣板時，無論他們的樣板角色為何，使用者都會自動成為所有關聯主題的「下載者」角色，讓使用者從該樣板建立網站時可以使用主題。

在想要共用的主題上按一下滑鼠右鍵，選擇**共用**，然後按一下**新增成員**。輸入一或多個使用者名稱或電子郵件地址，並且指定以下其中一個角色：

- **檢視者**：檢視者可以檢視主題的資料夾和檔案，但無法變更內容。
- **下載者**：下載者可同時下載檔案，並且將下載的檔案儲存到自己的電腦。
- **提供者**：提供者還可以編輯主題、上傳新檔案、刪除檔案以及刪除主題本身。
- **管理者**：管理者可同時新增使用者及指定其角色。主題的建立者 (擁有者) 會自動具備管理者角色。

編輯主題	如果您是 <b>Web</b> 開發人員，可以下載並修改個別主題檔案，或者可以使用桌面應用程式來同步您在本機系統上所做的變更。 您也可以匯出樣板套件 (包括關聯的主題和任何自訂元件)，在您偏好的開發環境中離線使用。請參閱 <b>開發主題</b> 。 如果您變更主題，則必須發布該主題，而使用該主題的網站才會反映這些變更。
發布主題	如果網站使用新的未發布主題，則會在第一次讓網站上線時自動發布網站的主題。如果您變更主題，並且想要更新線上網站來顯示變更，則必須明確地發布主題。如果您是主題擁有者或被授予管理者角色，就可以發布主題。 從清單中選取主題，然後按一下功能表列中的 <b>發布</b> (若為先前已發布的主題，則為 <b>重新發布</b> )。清單中該主題的旁邊會新增已發布圖示  。請參閱 <b>發布主題</b> 。

工作	描述
刪除或回復主題	如果您具有適當的權限，則可以刪除主題資料夾及其內容。當您刪除主題時，主題資料夾及其所有相關聯資料夾與檔案都會移動至資源回收。 如果您建立了某主題 (您是主題擁有者)，或者有人與您共用主題並授予您提供者或管理者角色，您便能刪除或回復該主題。

 **備註：**

您無法刪除已由任一網站使用的主題。

若要刪除主題，在想要刪除的主題上按一下滑鼠右鍵，然後選擇**刪除**。系統會提示您將主題移動至資源回收。在發生以下情況之前，已刪除的主題都會保留在資源回收中：

- 您回復主題。
- 您永久刪除主題。
- 達到資源回收配額。
- 系統根據服務管理員所設定的間隔自動清空資源回收。預設值為 90 天。

若要回復主題，請按一下「主題」頁面功能表列中的**資源回收**，然後對清單中的主題上按一下滑鼠右鍵並選擇**回復**。

## 發布主題

主題將定義網站的一般外觀與感覺。您可以更新主題，以變更使用該主題的任何網站外觀。

如果網站使用新的未發布主題，則會在第一次讓網站上線時自動發布網站的主題。如果您變更主題，並且想要更新線上網站來顯示變更，則必須明確地發布主題。若要發布主題變更，您必須是主題擁有者或具備「管理者」角色。

 **備註：**

如果您發布主題變更，則所有使用該主題的線上網站都將反映此變更。請先確定您已離線測試變更，並且瞭解對相關聯網站的影響，再發布主題的更新項目。

發布主題：

1. 依序按一下**開發人員**和**檢視所有主題**。
2. 從主題清單中選取一個現有主題。
3. 從功能表列或滑鼠右鍵功能表中按一下**發布**。
4. 按一下**確認繼續進行**，然後按一下**確定**。清單中該主題的旁邊便會新增已發布圖示 。

# 7

## 管理自訂元件與版面配置

若您是開發人員，您可以建立和管理自訂元件與版面配置。若您是網站提供者，您可以註冊第三方元件 (應用程式) 和元件群組。

自訂元件包含您在編輯器中所建立的元件群組，以及您使用本節中描述的選項所建立的本機和遠端元件。自訂版面配置包括可在頁面槽位中排列元件的 [區段版面配置](#)，以及可排列內容項目中欄位的 [內容版面配置](#)。

- [瞭解自訂元件](#)
- [瞭解版面配置](#)
- [使用自訂元件和版面配置](#)
- [註冊遠端元件](#)
- [建立本機元件、版面配置、內容欄位編輯器或內容表單](#)
- [匯出/匯入元件或版面配置](#)

若要瞭解如何在您的網站使用元件，請參閱[排列頁面內容](#)。

如需有關使用個別元素的資訊，請參閱[使用內建元件](#)。

如需有關如何自行建立元件的詳細資訊，請參閱[開發元件](#)。

## 瞭解自訂元件

元件是網頁的個別部分。當您看著網頁時，您看到了什麼？您可能會看到一些標題、文字段落，以及網站中其他頁面的連結。您可能也會看到影像、按鈕、分隔線、地圖及圖庫。這些全都是元件。

若要將元件新增至頁面，請確定  已設為**編輯**，按一下 ，然後選擇想要使用的元件類型。

將面板中的元件拖放至頁面上的槽位中。就是這麼簡單。將標題、段落、影像與其他元件拖放到頁面上您所需的位置。

您可以使用元件管理程式中的選項，輕鬆地註冊及併入遠端元件 (應用程式)，甚至建立自己的元件。依序按一下**開發人員**和**檢視所有元件**。

按一下**建立**並選取關聯的選項，以建立新的本機元件或註冊遠端元件。您以此方式建立和共用的元件會列在編輯器的自訂元件面板中。

若要瞭解如何建立自己的元件，請參閱[開發元件](#)。

### 遠端元件

如果您有想要使用的第三方元件 (應用程式)，只要註冊該元件，即可在您的網站中加以使用。就是這麼簡單。

您在註冊遠端元件時，需要指定遠端元件本身的 URL，以及 Web 作者可針對遠端元件指定之任何設定值的第二個 URL。

註冊遠端元件之後，您可以將它與其他使用者共用。「網站產生器」的「自訂元件」面板中會列出您所擁有或與您共用的所有已註冊遠端元件。

您可以在編輯器內變更遠端元件的特性，操作方式與您變更任何其他元件的特性相同。

只要按一下功能表圖示 ，然後選擇**設定值**即可。**自訂**按鈕可開啟您在註冊遠端元件時所指定的設定值 URL。

#### 備註：

由於代管遠端元件的伺服器並非代管 Oracle Content Management 的伺服器，因此基於安全考量，會將這些元件包含在內嵌框架中 (使用 `iframe` 元素)。並非所有遠端元件都能夠被包含在內嵌框架中。請與提供者確認，瞭解是否可以將元件包含在內。

### 本機元件

開發人員可以建立能夠存取與 Oracle Content Management 提供之相同特色與功能的元件。您可以直接將元件插入頁面中，或是使用 `iframe` 元素將它包含在內嵌框架中。

當您按一下**建立**來建立本機元件後，系統就會將一個功能完整的範例元件以您指定的名稱和唯一的 ID 加到元件清單中。開發人員即可修改此範例來建立自己的解決方案。

若要瞭解如何建立自己的元件，請參閱[開發元件](#)。

## 瞭解版面配置

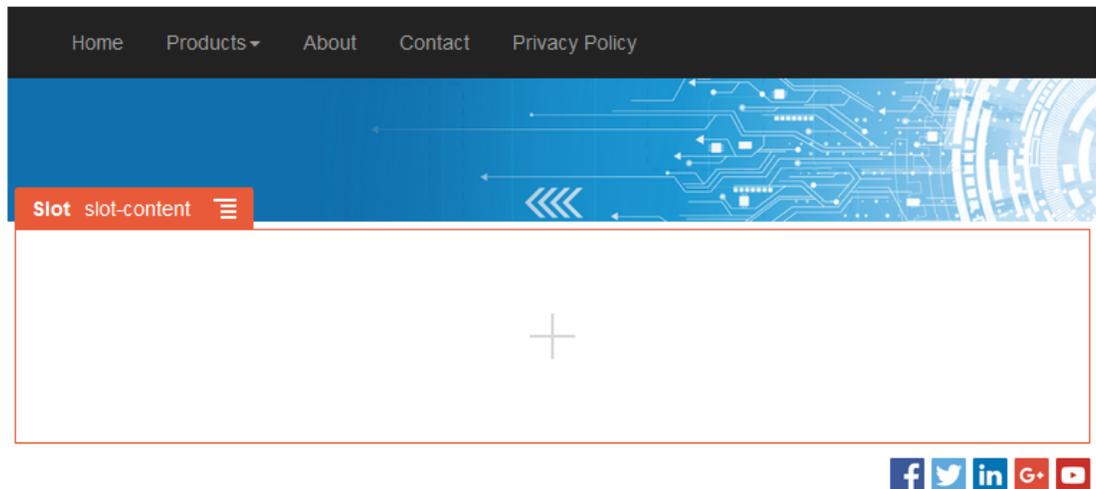
頁面版面配置可排列頁面上的槽位與內容。區段版面配置可排列槽位中的內容。內容版面配置可排列內容項目中的欄位。

一般而言，版面配置是指定內容的呈現方式，而非內容本身。將內容與呈現方式做區隔，就能更輕鬆地以不同的方式呈現相同內容，或可變更呈現方式而無須更動內容。

### 頁面版面配置

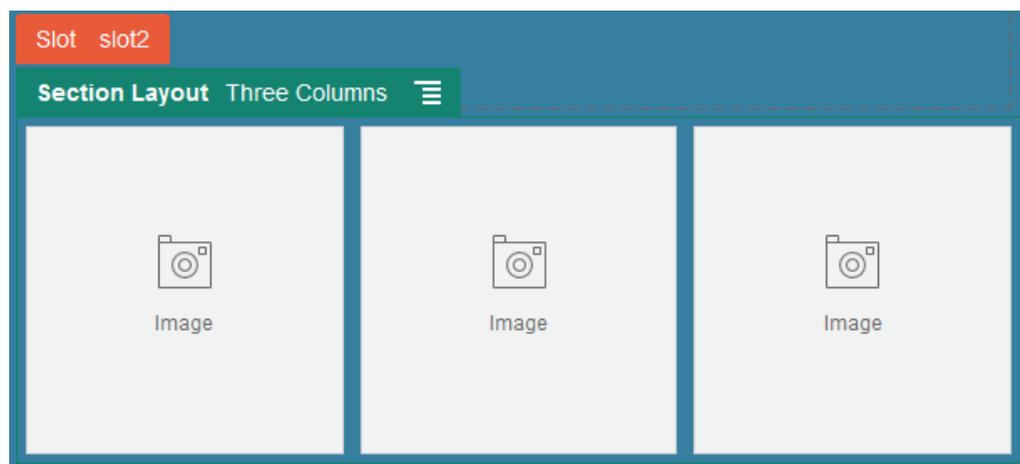
當您將頁面新增至網站時，會選取該頁面要使用的版面配置。每種版面配置在頁面上均包含多個區域 (稱為槽位)，提供者可拖放內容至其中。頁面版面配置定義頁面上的槽位數目和位置。版面配置也可包含預先定義和放置在頁面上的內容。此內容可以是靜態且無法編輯的內容 (如公司標誌)，或是可小幅度編輯的內容 (如可供提供者變更的標題文字)，但其位置或外觀則無法變更。

頁面版面配置儲存在主題中。主題可包含一或多個頁面版面配置。身為開發人員，您可以複製和修改現有的主題，即可建立新主題。請參閱[開發版面配置](#)。



### 區段版面配置

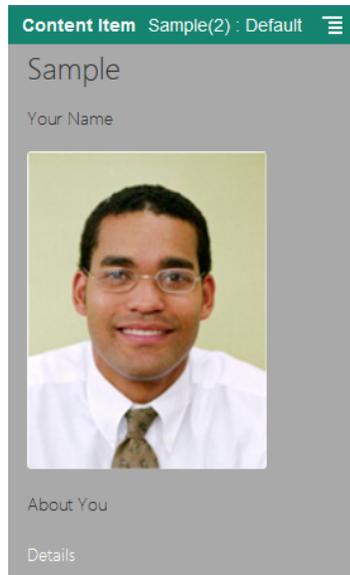
區段版面配置會自動組織新增至該位置的內容，讓提供者可以輕鬆新增內容，不需再另外格式化頁面上的內容。例如，區段版面配置會自動將內容組織為多個資料欄或垂直清單。網站提供者可以新增一或多個區段版面配置到槽位以組織內容。



除了提供的版面配置外，開發人員還可建立額外的區段版面配置，以解決特定的版面配置問題或簡化提供者的編寫工作。主題設計者甚至可以在頁面版面配置中的槽位建置區段版面配置。請參閱[開發版面配置](#)。

### 內容版面配置

企業使用者可以依據提供的內容類型和版面配置，建立及使用內容項目。提供者可以在「網站產生器」外組合以此種方式建構的內容項目內容。與內容類型關聯的多種內容版面配置讓網站設計者可以在不同的相關資訊環境中顯示內容項目，而不需更動已組合的內容。若您是開發人員，您可以從提供的預設版面配置建立新的版面配置。請參閱[開發版面配置](#)。



## 使用自訂元件和版面配置

若您是開發人員，您可以建立和管理自訂元件與版面配置。若您是網站提供者，您可以註冊第三方元件 (應用程式) 和元件群組。

自訂元件包含您在編輯器中所建立的元件群組，以及您使用如下所述的選項建立的本機和遠端元件。如果您有想要使用的第三方應用程式 (遠端元件)，只要註冊該元件，即可在您的網站中使用它。開發人員也可以建立能夠存取與 **Oracle Content Management** 提供之相同特色和功能的本機元件。

自訂版面配置包括可在頁面槽位中排列元件的 *區段版面配置*，以及可排列內容項目中欄位的 *內容版面配置*。

使用「元件」頁面中的功能表列或滑鼠右鍵功能表執行下列工作。

工作	描述
註冊遠端元件	<p>您在註冊遠端元件時，需要指定遠端元件本身的 URL，以及 Web 作者可針對遠端元件指定之任何設定值的第二個 URL。</p> <p>管理員必須啟用<b>建立</b>功能表中的選項，您才能夠建立元件。如果您未在元件頁面中見到<b>建立</b>功能表，請洽詢您的管理員。</p> <p>按一下<b>建立</b>並選取關聯的選項，以註冊遠端元件。請參閱<a href="#">註冊遠端元件</a>。</p>
建立本機元件或版面配置	<p>建立本機元件或版面配置時，系統會提供您一個功能完整的範例元件或版面配置，作為建立自己元件或版面配置的基礎。</p> <p>管理員必須啟用<b>建立</b>功能表中的選項，您才能夠建立元件或版面配置。如果您未在元件頁面中見到<b>建立</b>功能表，請洽詢您的管理員。</p> <p>按一下<b>建立</b>並選取關聯的選項，以建立新的本機元件或版面配置。請參閱<a href="#">建立本機元件</a>、<a href="#">版面配置</a>、<a href="#">內容欄位編輯器</a>或<a href="#">內容表單</a>。</p> <p>如需其他元件建立方式的相關資訊，請參閱<a href="#">開發元件</a>。</p>

工作	描述
複製元件或版面配置	<p>您可以藉由複製現有元件或版面配置並對複本進行變更，以建立新的元件或版面配置。</p> <div style="border: 1px solid #0070c0; padding: 10px; margin-top: 10px;"> <p> <b>備註：</b></p> <p>建立或註冊元件或版面配置之後，您便無法變更其名稱。但是您可以複製元件或版面配置，然後為其複本指定其他名稱。所有其他的註冊資訊 (包括遠端元件的金鑰值) 都會保留下來。共用資訊與註冊資訊是分開的，並不會被複製。</p> </div> <p>在想要複製的元件或版面配置上按一下滑鼠右鍵，然後選擇<b>複製</b>。輸入名稱並按一下<b>複製</b>。名稱中可以包含字母、數字、底線 ( _ ) 以及連字號 ( - )。如果您輸入空格，系統會自動以連字號取代。</p>
建立內容欄位編輯器	<p>您可以<b>建立</b>可升級的內容欄位編輯器，並於建立內容類型時可用來控制資料欄位外觀。支援<b>媒體</b>和<b>參照</b>以外的所有資料類型。元件必須先經過升級，才能夠在建立內容類型時使用。</p>
共用元件或版面配置	<p>您可以將您的元件或版面配置與其他 <b>Oracle Content Management</b> 使用者共用。若您是擁有者或被授予管理者角色，就可以共用元件或版面配置。若您共用元件或版面配置，就必須指定一個定義使用者可對元件或版面配置執行哪些動作的角色。</p> <p>在想要共用的元件或版面配置上按一下滑鼠右鍵，選擇<b>共用</b>，然後按一下<b>新增成員</b>。</p> <p>輸入一或多個使用者名稱或電子郵件地址，並且指定以下其中一個角色：</p> <ul style="list-style-type: none"> <li>· <b>檢視者</b>：檢視者可以見到並使用編輯器中的自訂元件和版面配置清單中的元件或版面配置。也可以檢視主題的資料夾和檔案，但是不能變更內容。</li> <li>· <b>下載者</b>：下載者可以下載檔案，同時將下載的檔案儲存到自己的電腦。</li> <li>· <b>提供者</b>：提供者還可以編輯元件或版面配置、上傳新檔案、刪除檔案，以及刪除元件或版面配置本身 (前提為網站中並未使用該元件或版面配置)。</li> <li>· <b>管理者</b>：管理者可同時新增使用者及指定其角色。元件或版面配置的建立者 (擁有者) 會自動被指定管理者角色。</li> </ul>
編輯元件或版面配置	<p>如果您是 <b>Web</b> 開發人員，可以下載並修改個別元件或版面配置檔案，或者可以使用桌面應用程式並同步您在本機系統上所做的變更。</p> <p>您也可以將元件或版面配置個別匯出，或匯出為樣板套件的一部分 (包括任何自訂元件與區段版面配置)，並在您偏好的開發環境中離線使用。</p> <p>如果您變更元件或版面配置，則必須發布該元件或版面配置，而使用該元件或版面配置的網站才會反映這些變更。</p>
匯出或匯入元件或版面配置	<p>您可以匯出元件或版面配置並離線進行修改，然後再將它匯入成新元件或新版面配置，或取代現有的元件或版面配置。您也可以匯出元件或版面配置，並將其移到另一個執行處理，然後在該處加以匯入。您可以將元件或版面配置個別匯出，或匯出為樣板套件的一部分 (包括任何自訂的元件和版面配置)。</p> <p>匯出元件或版面配置時，您是將元件或版面配置以單一 <b>.zip</b> 檔案的形式複製到某個資料夾。您可以直接從該資料夾下載元件或版面配置套件，將其解壓縮並處理個別的檔案。完成處理元件或版面配置檔案之後，請建立一個包含該元件或版面配置套件的 <b>.zip</b> 檔案並將其匯入，然後選擇覆寫或建立新的元件或版面配置。</p> <p>請參閱<a href="#">匯出/匯入元件或版面配置</a>。</p>

工作	描述
發布元件或版面配置	<p>如果網站使用新的未發布元件或版面配置，則會在第一次讓網站上線時自動發布網站的元件或版面配置。如果您變更元件或版面配置，並且想要更新線上網站來顯示變更，則必須明確地發布元件或版面配置。若您是擁有者或被授予管理者角色，就可以發布元件或版面配置。</p> <p>選取清單中的元件或版面配置，並按一下功能表列中的<b>發布</b> (或針對先前已發布的元件或版面配置，按一下<b>重新發布</b>)。清單中該元件或版面配置的旁邊便會顯示已發布圖示 。</p>
刪除或回復元件或版面配置	<p>如果您具有適當的權限，則可以刪除元件或版面配置資料夾及其內容。當您刪除元件或版面配置時，元件或版面配置資料夾與其所有關聯的資料夾和檔案都會移動至資源回收。</p> <p>您若是元件或版面配置的建立者 (即為元件或版面配置擁有者)，或有人與您共用元件或版面配置且授予您提供者或管理者角色，您就能夠刪除或回復元件或版面配置。</p>

#### 備註：

您不能刪除任何網站或更新項目 (包括在資源回收中的網站或更新項目) 仍在使用的元件或版面配置。

若要刪除元件或版面配置，請在想要刪除的元件或版面配置上按一下滑鼠右鍵，然後選擇**刪除**。系統會提示您將元件或版面配置移動至資源回收。在發生以下情況之前，已刪除的元件或版面配置都會保留在資源回收中：

- 您回復元件或版面配置。
- 您永久刪除元件或版面配置。
- 達到資源回收配額。
- 系統根據服務管理員所設定的間隔自動清空資源回收。預設值為 **90** 天。

若要回復元件或版面配置，請按一下**資源回收**，然後在清單中的元件或版面配置上按一下滑鼠右鍵並選擇**回復**。

## 註冊遠端元件

若要在網站中使用遠端元件，您必須先在 **Oracle Content Management** 中註冊該元件。您可以註冊第三方遠端元件和您開發的遠端元件。

#### 備註：

管理員必須啟用**建立**功能表中的選項，您才能夠註冊遠端元件。如果您未在元件頁面中見到**建立**功能表，請洽詢您的管理員。

註冊遠端元件以供 **Oracle Content Management** 使用：

1. 依序按一下**開發人員**和**檢視所有元件**。
2. 按一下**建立**，然後選擇**註冊遠端元件**。
3. 在「註冊遠端元件」視窗中，輸入或選取資訊，包括：

- **名稱**：使用者將會見到之元件的名稱。
- **描述**：使用者將會見到之元件的描述。
- **元件 URL**：iframe 中所使用的端點，用於呈現頁面中的元件內容。它必須是 HTTPS。
- **設定值 URL**：iframe 中所使用的端點，用於呈現新增到頁面之遠端元件的設定值。它必須是 HTTPS。
- **設定值寬度**：設定元件設定值面板的預設寬度 (像素)。
- **設定值高度**：設定元件設定值面板的預設高度 (像素)。
- **金鑰**：遠端元件的 192 位元 AES 金鑰，元件佈建之後，會使用此金鑰建立已簽署的雜湊權杖。此金鑰主要用來加密元件設定值，並確保可以安全地讀取及寫入這些設定值。

#### 4. 按一下**註冊**。

遠端元件建立之後，其名稱會顯示在元件清單中。只要按一下元件清單中的元件名稱，您即可瀏覽註冊該元件所使用的檔案。

元件註冊資訊會儲存在網站 (這些網站都建立於相同的 Oracle Content Management 執行處理中) 所使用的目錄中，但是元件則繼續維持為遠端服務。

若您是元件擁有者，編輯器的「自訂元件」面板中會有一個元件圖示，其中包含您指定給元件的名稱。您可以與其他使用者共用此元件，他們將會在編輯器的「自訂元件」面板中見到此元件。

## 建立本機元件、版面配置、內容欄位編輯器或內容表單

身為開發人員，當您建立本機元件或版面配置時，系統會提供您一個功能完整的範例，您可用於修改您自己的元件或版面配置。

管理員必須啟用**建立**功能表中的選項，您才能夠建立元件或版面配置。如果您未在元件頁面中見到**建立**功能表，請洽詢您的管理員。

開發人員可以建立能夠存取與 Oracle Content Management 提供之相同特色與功能的元件。

若要建立範例本機元件或版面配置：

1. 依序按一下**開發人員**和**檢視所有元件**。
2. 按一下**建立**，然後選取您想要建立之元件或版面配置的類型。
3. 輸入元件或版面配置的名稱。您不能使用其他元件或版面配置已經使用的名稱。  
名稱中可以包含字母、數字、底線 ( \_ ) 以及連字號 ( - )。如果您輸入空格，系統會自動以底線取代。  
請勿使用下列作為樣板、主題、元件、網站、或網站頁面的名稱：authsite、content、pages、scstemplate\_\*、\_comps、\_components、\_compsdelivery、\_idcservice、\_sitescloud、\_sitesclouddelivery、\_themes、\_themesdelivery。您可以使用下列名稱作為網站頁面的名稱，但不可以作為樣板、主題、元件或網站的名稱：文件、網站。
4. (選擇性) 輸入元件或版面配置的描述。
5. 對於元件類型，選取**預設**即可直接將元件插入頁面中、選取**封閉測試環境**即可將元件包含在內嵌框架中 (使用 iframe 元素)，或是選取**樣板**，只透過樣板來呈現。

 **備註：**

如果選取**樣板**，本機元件就不是 **Knockout** 元件，因此當置放於頁面上時，便無法使用巢狀元件。舉例來說，這表示編輯標題或主體文字都必須透過元件設定值面板中的自訂設定值進行，而不是選取頁面上顯示的元件文字。

6. 按一下**建立**。

進度列就會顯示建立狀態。元件、版面配置或表單建立之後，其名稱會顯示在元件清單中。按一下元件清單中的元件或版面配置名稱，即可瀏覽構成該元件或版面配置的資料夾和檔案。

7. 若要選取指定給元件或版面配置的預設圖示以外的圖示：

- a. 從清單中選取元件或版面配置。
- b. 按一下**特性**。
- c. 按一下**元件標誌**頁籤。
- d. 按一下標誌圖庫中的某個標誌，然後按一下**完成**。

如需有關如何建立您自己元件的詳細資訊，請參閱[開發元件](#)。

自訂元件、版面配置、內容欄位編輯器或內容表單之後，即可與他人共用或進行升級，讓他們能夠以下列方式使用這些項目：

- 自訂元件：如果使用「網站產生器」，元件圖示會加到編輯器中的「自訂元件」面板 (使用您指定的元件名稱)。
- 區段版面配置：如果使用「網站產生器」，區段版面配置圖示會加到編輯器中的「區段版面配置」面板 (使用您指定的版面配置名稱)。
- 內容版面配置：內容管理員可將版面配置指定給一或多個內容類型，以作為預設檢視或新增至版面配置清單中，讓網站設計者可以在「網站產生器」中選取，以指定該類型的內容項目要如何在頁面上顯示。
- 內容欄位編輯器：若要讓內容欄位編輯器可在建立內容類型時使用，請選取該編輯器並按一下**升級**，然後確認並按一下**確定**。升級之後，內容欄位編輯器即可在建立內容類型時，用來控制**媒體**和**參照**以外的所有資料欄位外觀。
- 內容表單：若要讓內容表單可在建立內容類型時使用，請選取該表單並按一下**升級**，然後確認並按一下**確定**。升級之後，內容表單便可搭配所有內容類型和資產類型使用。

## 匯出/匯入元件或版面配置

您可以匯出元件並離線進行修改，然後再將它匯入成新元件或取代現有元件。您也可以匯出元件，並將它移到另一個 **Oracle Content Management** 執行處理，然後在該處加以匯入。

### 匯出

當您匯出元件時，基本上將元件以單一 **.zip** 檔案的形式複製到 **Oracle Content Management** 中的某個資料夾。您可以直接從該資料夾下載元件，將其解壓縮並處理個別的檔案。完成處理元件檔案之後，請建立一個包含元件資料夾和檔案的 **.zip** 檔案並將其匯入，然後選擇覆寫原始元件或建立新元件。

 **備註：**

匯出元件時，不會包括該元件的共用資訊。

匯出元件：

1. 依序按一下**開發人員**和**檢視所有元件**。
2. 選取元件或版面配置，然後從功能表列或滑鼠右鍵功能表中選擇**匯出**。
3. 瀏覽至某個資料夾，或者按一下**新建**、提供名稱和選擇性的描述，然後按一下**建立**來建立新資料夾。若要開啟資料夾，請按一下資料夾圖示或資料夾名稱。
4. 按一下關聯資料夾的核取方塊以選取資料夾，然後按一下**確定**。

就會在選取的資料夾中，以元件名稱或版面配置名稱加 .zip 副檔名，建立一個元件或版面配置套件檔案。

### 匯入

管理員必須啟用**建立**功能表中的選項，您才能夠匯入或建立元件、版面配置或內容欄位編輯器。如果您未在元件頁面中見到**建立**功能表，請洽詢您的管理員。

匯入元件、版面配置或內容欄位編輯器：

1. 依序按一下**開發人員**和**檢視所有元件**。
2. 按一下**建立**，然後選擇**匯入元件**。選擇此選項以匯入元件、區段版面配置或內容版面配置。
3. 如果您已上傳元件或版面配置套件，請瀏覽至包含該元件或版面配置的資料夾，並開啟該資料夾。如果您尚未上傳套件，請前往您要上傳元件或版面配置的資料夾，或建立新的資料夾。按一下**上傳**，找到該元件或版面配置套件後按一下**開啟**。
4. 按一下元件或版面配置檔案名稱旁的核取方塊後，按一下**確定**。將建立新元件或版面配置，並將其新增至元件清單。如果該元件或版面配置名稱或 ID 已經存在，系統會提示您解決衝突。您可能需要建立新元件或版面配置，或者您也可以匯入的版本覆寫現有的元件或版面配置。

# 8

## 使用網站頁面

讓我們建立頁面、選擇版面配置，並決定頁面在網站中的位置。

- [瀏覽至頁面](#)
- [檢視頁面](#)
- [新增頁面](#)
- [移動頁面](#)
- [刪除頁面](#)
- [變更頁面設定值](#)
- [變更頁面版面配置](#)
- [變更背景或主題](#)

### 瀏覽至頁面

檢視和瀏覽頁面階層：

1. 開啟網站進行編輯。確定  已設為「編輯」。
2. 按一下  以顯示可管理與編輯頁面和頁面內容的選項。
3. 按一下  以列出第一層的頁面。
4. 按一下頁面即可檢視該頁面。  
含箭號 (>) 的頁面具有巢狀頁面層。按一下頁面即可顯示巢狀頁面。  
您也可以使用頁面清單頂端的搜尋方塊來找出想要的頁面。
5. 若要檢視或變更特定頁面的設定值，請按一下該頁面的 。請參閱[變更頁面設定值](#)。

### 檢視頁面

若要檢視網站中的頁面，請在編輯器中開啟該網站，然後使用不同的預覽、大小以及方向選項。

變更使用編輯器時檢視頁面的方式：

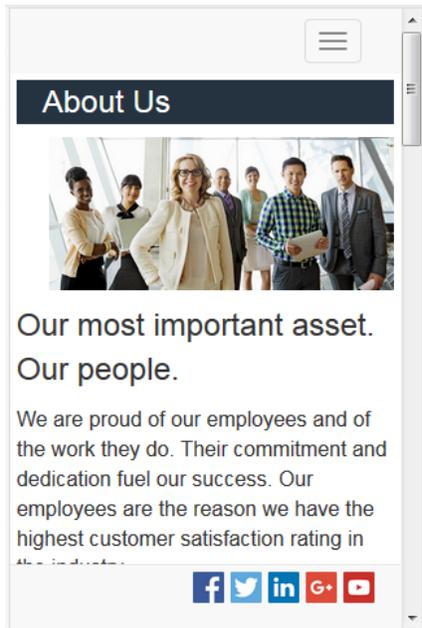
1. 開啟網站進行編輯。
2. 瀏覽至頁面。
3. 若要在編輯器中預覽頁面，請確定將  設為「預覽」。這樣一來，畫面上就會顯示網站訪客所看到的頁面，但不會顯示編輯時所使用的其他視覺化輔助。

4. 若要在個別的瀏覽器視窗中預覽目前套用至基本網站的更新，請按一下 。

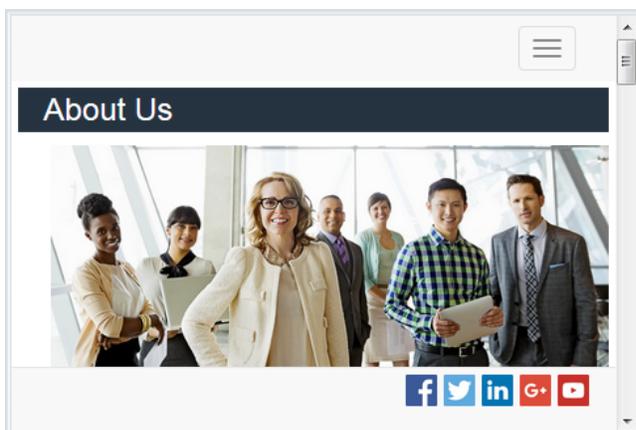
 **備註：**

網站頁面的連結在預覽模式中沒有作用。

5. 若要以頁面在裝置的特定畫面大小顯示的樣子進行預覽，請按一下頂端功能表列中的 **符合畫面大小**，然後從清單中選擇一組畫面尺寸。具有響應式設計的主題會自動根據所選螢幕大小排列頁面內容，讓使用者享有最佳的閱讀體驗。



6. 若要建立自訂大小，請按一下 **調整成視窗大小**，然後選擇 **建立裝置預先設定值...**。在必要欄位中提供值。若要刪除自訂大小，請按一下該大小旁的 。您也可以啟用尺規 ，然後從尺規上選取任意間隔以快速查看不同大小。
7. 若要將頁面方向切換為直向或橫向，請按一下 。



## 新增頁面

網站中的頁面建構方式類似資料夾的階層，或稱為**網站樹狀結構**。您可以新增頁面、定義頁面的設定值，以及決定頁面在網站中的位置。

您可以選擇新增頁面，或新增外部頁面的連結。

- **網頁**：頁面和內容位於網站中。您可以指定頁面並定義其內容、指定頁面在網站中的位置，以及指定頁面在目前網站相關資訊環境中的行為。
- **外部連結**：使用 URL 指定，參照其他位置的頁面。您可以指定頁面、指定頁面在網站中的位置，以及指定頁面在目前網站相關資訊環境中的行為。因為您是使用即時網站中的頁面，所以無法變更連結的頁面內容。

在網站樹狀結構中，連結至外部 URL 的頁面在頁面名稱前面會有 。

### 新增網頁

新增網頁至網站：

1. 開啟網站進行編輯。
2. 選取要新增頁面的層級或分支，然後按一下**新增頁面**；或者，若要新增子項頁面，請按一下 ，再按一下 。「新頁面」會新增至網站樹狀結構底端，而且系統會提示您命名頁面以及指定其他設定值。
3. 選取**網頁**作為頁面類型。如果您要管理頁面上的內容，而不是重複使用另一個網站中的頁面，請使用此選項。
4. 提供頁面的名稱。名稱中可以包含字母、數字、底線 ( \_ ) 以及連字號 (-)。請勿使用下列作為樣板、主題、元件、網站、或網站頁面的名稱：`authsite`、`content`、`pages`、`scstemplate_*`、`_comps`、`_components`、`_compsdelivery`、`_idcservice`、`_sitescloud`、`_sitesclouddelivery`、`_themes`、`_themesdelivery`。您可以使用下列名稱作為網站頁面的名稱，但不可以作為樣板、主題、元件或網站的名稱：文件、網站。
5. 指定頁面 URL。預設會使用頁面名稱作為頁面 URL。空格會自動以連字號取代。若要使用預設值以外的 URL，請按一下**覆寫**並新增該 URL 使用的檔案名稱。名稱中可以包含字母、數字、底線 ( \_ ) 以及連字號 (-)。請務必包括副檔名。預設副檔名為 `.html`。
6. 選擇頁面版面配置。這樣會定義頁面的一般結構，而不是內容。版面配置包含一或多個可放置內容的具名**槽位**。頁面版面配置的數目和類型取決於網站所關聯的主題。

7. 選擇行動頁面版面配置。這定義了在行動裝置上檢視時的頁面一般結構。
8. 指定頁面標題。此標題不會顯示在頁面本身，而是顯示於頁面的瀏覽器標題列或瀏覽器頁籤上。若要在頁面上新增標題，請使用編輯器來新增標題元件。
9. 新增選擇性描述。此描述不會顯示在頁面上，但您可以新增頁面資訊，方便其他提供者或您自己使用。
10. 指定以逗點區隔的選擇性關鍵字，協助搜尋引擎識別頁面內容。某些術語或概念不會顯示在頁面文字上或只會出現在影像中，指定關鍵字有助於識別這些術語或概念。關鍵字不會顯示在頁面上，但搜尋引擎可藉由關鍵字找出及識別您的網站。有意義的描述、關鍵字和同義字有助於提升網站流量。
11. 新增用於分析或追蹤的選擇性頁面標頭命令檔或標記。驗證標頭或頁尾中所使用的全程式碼，確定程式碼可以正常運作，且不會為網站帶來任何安全性風險。
12. 新增用於分析或追蹤的選擇性頁面頁尾命令檔或標記。根據預設，頁尾會包含所要顯示之即現式視窗中的文字，當中說明該網站的 **Cookie** 使用方式和隱私權原則連結。如果您是開發人員，可以編輯當中顯示的文字。驗證標頭或頁尾中所使用的全程式碼，確定程式碼可以正常運作，且不會為網站帶來任何安全性風險。
13. 選擇性：您可以使用任何可用的頁面選項：
  - **錯誤頁面**：當要求的頁面因錯誤而無法顯示時，讓系統顯示此頁面而不是預設的錯誤頁面。如果您選取此選項，預設會一併選取**隱藏頁面使其無法瀏覽**，以及所有搜尋引擎選項。若要在導覽或搜尋引擎結果中包含錯誤頁面，可以取消選取此選項。
  - **隱藏頁面使其無法瀏覽**：在自動產生的網站導覽中，不要包括頁面。
  - **詳細資訊頁面**：您可以在此頁面中顯示在另一個頁面中選取之內容項目的詳細資訊。結構化內容僅供企業使用者使用。如果您選取此選項，預設會一併選取**隱藏頁面使其無法瀏覽**。若要在導覽中包含詳細資訊頁面，可以取消選取此選項。  
結構化內容項目若設定成使用詳細資訊頁面，當使用者按一下結構化內容項目的詳細資訊連結時，就會顯示含有該內容項目之詳細資訊的詳細資訊頁面。
  - **搜尋頁面**：可使用此頁面顯示搜尋結果。您可以在「內容搜尋」元件的「連結」設定值中選取此頁面。請參閱[內容搜尋](#)。如果您選取此選項，預設會一併選取**隱藏頁面使其無法瀏覽**。若要在導覽中包含搜尋頁面，可以取消選取此選項。
14. 選擇性：選取搜尋引擎最佳化 (SEO) 選項：
  - **在搜尋引擎中隱藏頁面**：告知搜尋引擎不要編製頁面內容的索引，讓頁面內容不會在搜尋結果中顯示。
  - **在搜尋引擎中隱藏頁面連結**：通知搜尋引擎不要追蹤跟隨頁面上的連結，因此不對連結目的地編製索引。
  - **停用搜尋引擎頁面快取功能**：通知搜尋引擎不要快取此頁面。
  - **在搜尋引擎中隱藏頁面描述**：告知搜尋引擎不要在搜尋結果中的頁面之後包含描述 (前面指定的描述)。
15. 選擇性：選取此頁面是否將提供搭配瀏覽階段作業使用的 **Cobrowse** 按鈕。  
**Cobrowse** 階段作業使用 **Oracle Cobrowse Cloud Service** 管理與網站訪客之間的畫面分享體驗。請參閱[啟用 Cobrowse 整合](#)及[在頁面上使用 Cobrowse 功能](#)。
16. 選擇性：覆寫網站的分析追蹤程式碼片段，並將特定程式碼片段新增至此頁面。新增 **JavaScript** 追蹤程式碼片段以進行 **Web** 分析追蹤，讓其更易於與外部分析提供者 (例如 **Google**、**Adobe** 或 **Oracle Infinity**) 做整合。請參閱[新增分析追蹤功能](#)。

17. 若要儲存目前更新項目中的所有擱置中變更，請按一下**儲存**。

### 新增連結的頁面

將外部連結頁面新增至網站：

1. 開啟網站進行編輯。
2. 在網站樹狀結構中，前往您要新增頁面的層級。頁面會新增至目前的層級，您也可以輕鬆地將其移動到網站樹狀結構中的另一個位置。
3. 按一下**新增頁面**。「新頁面」會新增至網站樹狀結構底端，而且系統會提示您命名頁面以及指定其他設定值。
4. 選取**外部連結**作為頁面類型。
5. 提供頁面的名稱。名稱中可以包含字母、數字、底線 ( \_ ) 以及連字號 ( - )。在網站樹狀結構中，名稱是用來識別頁面。空格會自動取代之為連字號。
6. 指定頁面的完整 URL。例如：`https://www.example.com/sharedpage.htm`
7. 如果您想要在新的瀏覽器視窗或頁籤中開啟頁面，請選取**在新視窗中開啟連結**。如果您未選取此選項，就會在目前視窗中開啟頁面，並取代網站頁面和導覽。
8. (選擇性) 選取任何可用的選項：
  - **錯誤頁面**：當要求的頁面因錯誤而無法顯示時，讓系統顯示此頁面而不是預設的錯誤頁面。如果您選取此選項，預設會一併選取**隱藏頁面使其無法瀏覽**。若要在導覽中包含錯誤頁面，可以取消選取此選項。
  - **隱藏頁面使其無法瀏覽**：在自動產生的網站導覽中，不要包括頁面。
  - **詳細資訊頁面**：您可以在此頁面中顯示在另一個頁面中選取之內容項目的詳細資訊。結構化內容僅供企業使用者使用。如果您選取此選項，預設會一併選取**隱藏頁面使其無法瀏覽**。若要在導覽中包含詳細資訊頁面，可以取消選取此選項。結構化內容項目若設定成使用詳細資訊頁面，當使用者按一下結構化內容項目的詳細資訊連結時，就會顯示含有該內容項目之詳細資訊的詳細資訊頁面。
  - **搜尋頁面**：可使用此頁面顯示搜尋結果。您可以在「內容搜尋」元件的「連結」設定值中選取此頁面。請參閱**內容搜尋**。如果您選取此選項，預設會一併選取**隱藏頁面使其無法瀏覽**。若要在導覽中包含搜尋頁面，可以取消選取此選項。
9. 若要儲存目前更新項目中的所有擱置中變更，請按一下**儲存**。

## 移動頁面

您可以將頁面拖曳到網站樹狀結構目前層級的其他位置，或將頁面剪下並貼到另一個層級。

頁面階層中永遠都會有一個最上層頁面 (即您網站的「首頁」)，而所有其他頁面都會列在該頁面底下。您可以拖放指定層級的頁面，以變更其順序。您也可以剪下頁面，並將頁面貼入階層中的另一個位置。

頁面可以有巢狀頁面。如果您移動具有巢狀頁面的頁面，則也會移動巢狀頁面，並會在移動的頁面下方保持巢狀。如果您使用的主題包括自動化導覽，則變更頁面順序也會變更它們在導覽功能表中的列出順序。

將頁面移動至網站樹狀結構的其他位置：

1. 開啟網站進行編輯。
2. 前往您要移動的頁面。

3. 若要將頁面移動至網站樹狀結構目前層級的其他位置，請按一下頁面名稱並拖曳到頁面清單中的其他位置，然後放置頁面。
4. 若要將頁面移動至網站樹狀結構的其他層級，請依序按一下您要移動的頁面和該頁

面的 。按一下  將頁面從目前的位置剪下，或按一下  在目前的位置

留下頁面複本。前往您要移動頁面的目標位置所在層級，然後按一下 。如果您選取了該層級的某個頁面，您要貼上的頁面將以巢狀方式包含在該頁面底下。按一下  將頁面貼到目前的位置。若要重新排序頁面，請將頁面拖放到網站樹狀結構目前層級的其他位置。

## 刪除頁面

您可以在更新項目時刪除頁面。

當您按一下**儲存**時，會將已刪除的頁面記錄為更新項目的一部分。除非您發布更新項目，否則頁面不會從基本網站中刪除。所有具有提供者或管理者角色的人員都可以刪除頁面。

### 備註：

當您發布會刪除頁面的更新項目時，便會永久刪除頁面，且無法進行回復。如果自多語言網站中刪除某個頁面，發布網站時也會刪除頁面的翻譯版本。如果管理頁面的新增或刪除十分重要，在網站中新增或刪除頁面時，您可以使用專用的更新項目來隔離那些動作。

刪除網站中的頁面：

1. 開啟網站進行編輯。
2. 瀏覽至您要刪除的頁面，然後按一下 。
3. 按一下**確定**以確認刪除。按一下**儲存**以儲存目前更新項目中的所有擱置中變更。

## 變更頁面設定值

您可以變更頁面設定值，例如名稱、標題、URL、標頭、頁尾以及其他選項。

1. 開啟網站進行編輯。
2. 按一下想要變更設定值的頁面，然後按一下 。
3. 根據需要變更設定值：
  - **頁面類型：**
    - **網頁：**頁面和內容位於網站中。您可以指定頁面並定義其內容、指定頁面在網站中的位置，以及指定頁面在目前網站相關資訊環境中的行為。
    - **外部連結：**使用 URL 指定，參照其他位置的頁面。您可以指定頁面、指定頁面在網站中的位置，以及指定頁面在目前網站相關資訊環境中的行為。因為您是使用即時網站中的頁面，所以無法變更連結的頁面內容。

在網站樹狀結構中，連結至外部 URL 的頁面在頁面名稱前面會有 。

若選取網頁，您會見到下列設定值：

- **頁面名稱：**您可以使用字母、數字、底線 ( `_` ) 以及連字號 ( `-` )。  
請勿使用下列作為樣板、主題、元件、網站、或網站頁面的名稱：`authsite`、`content`、`pages`、`scstemplate_*`、`_comps`、`_components`、`_compsdelivery`、`_idcservice`、`_sitescloud`、`_sitesclouddelivery`、`_themes`、`_themesdelivery`。您可以使用下列名稱作為網站頁面的名稱，但不可以作為樣板、主題、元件或網站的名稱：文件、網站。
- **頁面 URL：**預設會使用頁面名稱作為頁面 URL。空格會自動取代為連字號。若要使用預設值以外的 URL，請按一下 **覆寫** 並新增該 URL 使用的檔案名稱。名稱中可以包含字母、數字、底線 ( `_` ) 以及連字號 ( `-` )。請務必包括副檔名。預設副檔名為 `.html`。
- **頁面版面配置：**頁面版面配置主要定義頁面的一般結構，而非內容。版面配置包含一或多個可放置內容的具名 *槽位*。頁面版面配置的數目和類型取決於網站所關聯的主題。
- **行動頁面版面配置：**行動頁面版面配置主要定義在行動裝置上檢視時，頁面的一般結構。
- **頁面標題：**頁面標題並不會顯示在頁面本身，而是在瀏覽器中顯示頁面時，顯示於瀏覽器標題列或瀏覽器頁籤中。若要在頁面上新增標題，請使用編輯器來新增標題元件。
- **頁面描述：**描述並不會顯示在頁面上，而是您可以新增頁面資訊，方便其他提供者或您自己使用。
- **關鍵字：**指定以逗點區隔的選擇性關鍵字，協助搜尋引擎識別頁面內容。某些術語或概念不會顯示在頁面文字上或只會出現在影像中，指定關鍵字有助於識別這些術語或概念。關鍵字不會顯示在頁面上，但搜尋引擎可藉由關鍵字找出及識別您的網站。有意義的描述、關鍵字和同義字有助於提升網站流量。
- **頁面標頭：**新增用於分析或追蹤的選擇性頁面標頭命令檔或標記。驗證標頭或頁尾中所使用的全部程式碼，確定程式碼可以正常運作，且不會為網站帶來任何安全性風險。
- **頁面頁尾：**新增用於分析或追蹤的選擇性頁面頁尾命令檔或標記。根據預設，頁尾會包含所要顯示之即現式視窗中的文字，當中說明該網站的 **Cookie** 使用方式和隱私權原則連結。如果您是開發人員，可以編輯當中顯示的文字。驗證標頭或頁尾中所使用的全部程式碼，確定程式碼可以正常運作，且不會為網站帶來任何安全性風險。
- **頁面選項：**
  - **錯誤頁面：**當要求的頁面因錯誤而無法顯示時，讓系統顯示此頁面而不是預設的錯誤頁面。如果您選取此選項，預設會一併選取 **隱藏頁面使其無法瀏覽**，以及所有搜尋引擎選項。若要在導覽或搜尋引擎結果中包含錯誤頁面，可以取消選取此選項。
  - **隱藏頁面使其無法瀏覽：**在自動產生的網站導覽中，不要包括頁面。
  - **詳細資訊頁面：**您可以在此頁面中顯示在另一個頁面中選取之內容項目的詳細資訊。結構化內容僅供企業使用者使用。如果您選取此選項，預設會一併選取 **隱藏頁面使其無法瀏覽**。若要在導覽中包含詳細資訊頁面，可以取消選取此選項。結構化內容項目若設定成使用詳細資訊頁面，當使用者按一下結構化內容項目的詳細資訊連結時，就會顯示含有該內容項目之詳細資訊的詳細資訊頁面。
  - **搜尋頁面：**可使用此頁面顯示搜尋結果。您可以在「內容搜尋」元件的「連結」設定值中選取此頁面。請參閱 [內容搜尋](#)。如果您選取此選項，預設會一併選取 **隱藏頁面使其無法瀏覽**。若要在導覽中包含搜尋頁面，可以取消選取此選項。
- **SEO 選項：**

- **在搜尋引擎中隱藏頁面**：告知搜尋引擎不要編製頁面內容的索引，讓頁面內容不會在搜尋結果中顯示。
- **在搜尋引擎中隱藏頁面連結**：通知搜尋引擎不要追蹤跟隨頁面上的連結，因此不對連結目的地編製索引。
- **停用搜尋引擎頁面快取功能**：通知搜尋引擎不要快取此頁面。
- **在搜尋引擎中隱藏頁面描述**：告知搜尋引擎不要在搜尋結果中的頁面之後包含描述 (前面指定的描述)。
- **Cobrowse**：若您的管理員已為系統啟用共同瀏覽，且此網站也已啟用共同瀏覽的話，您就會看到一個選項，可選取此頁面是否將提供搭配瀏覽階段作業使用的 **Cobrowse** 按鈕。**Cobrowse** 階段作業使用 **Oracle Cobrowse Cloud Service** 管理與網站訪客之間的畫面分享體驗。請參閱[啟用 Cobrowse 整合](#)和[在頁面上使用 Cobrowse 功能](#)。
- **分析選項**：此網站若包含分析追蹤程式碼片段，您可以覆寫此頁面的程式碼片段。按一下覆寫開關，然後編輯命令檔。如需詳細資訊，請參閱[新增分析追蹤功能](#)。如果您編輯網站程式碼片段，系統將會顯示該命令檔已遭修改訊息。若要移除自訂項目：**回復為最新的網站命令檔**。

若選取外部連結，您會見到下列設定值：

- **頁面名稱**：您可以使用字母、數字、底線 ( \_ ) 以及連字號 ( - )。在網站樹狀結構中，名稱是用來識別頁面。空格會自動取代為連字號。
  - **連結 URL**：指定頁面的完整 URL。例如：`https://www.example.com/sharedpage.htm`。
  - **在新視窗中開啟連結**：若您想要在新的瀏覽器視窗或頁籤中開啟頁面，請選取此選項。如果您未選取此選項，就會在目前視窗中開啟頁面，並取代網站頁面和導覽。
  - **頁面選項**：
    - **錯誤頁面**：當要求的頁面因錯誤而無法顯示時，讓系統顯示此頁面而不是預設的錯誤頁面。如果您選取此選項，預設會一併選取**隱藏頁面使其無法瀏覽**。若要在導覽中包含錯誤頁面，可以取消選取此選項。
    - **隱藏頁面使其無法瀏覽**：在自動產生的網站導覽中，不要包括頁面。
    - **詳細資訊頁面**：您可以在此頁面中顯示在另一個頁面中選取之內容項目的詳細資訊。結構化內容僅供企業使用者使用。如果您選取此選項，預設會一併選取**隱藏頁面使其無法瀏覽**。若要在導覽中包含詳細資訊頁面，可以取消選取此選項。  
結構化內容項目若設定成使用詳細資訊頁面，當使用者按一下結構化內容項目的詳細資訊連結時，就會顯示含有該內容項目之詳細資訊的詳細資訊頁面。
    - **搜尋頁面**：可使用此頁面顯示搜尋結果。您可以在「內容搜尋」元件的「連結」設定值中選取此頁面。請參閱[內容搜尋](#)。如果您選取此選項，預設會一併選取**隱藏頁面使其無法瀏覽**。若要在導覽中包含搜尋頁面，可以取消選取此選項。
4. 若要儲存目前更新項目中的所有擱置中變更，請按一下**儲存**。

## 變更頁面版面配置

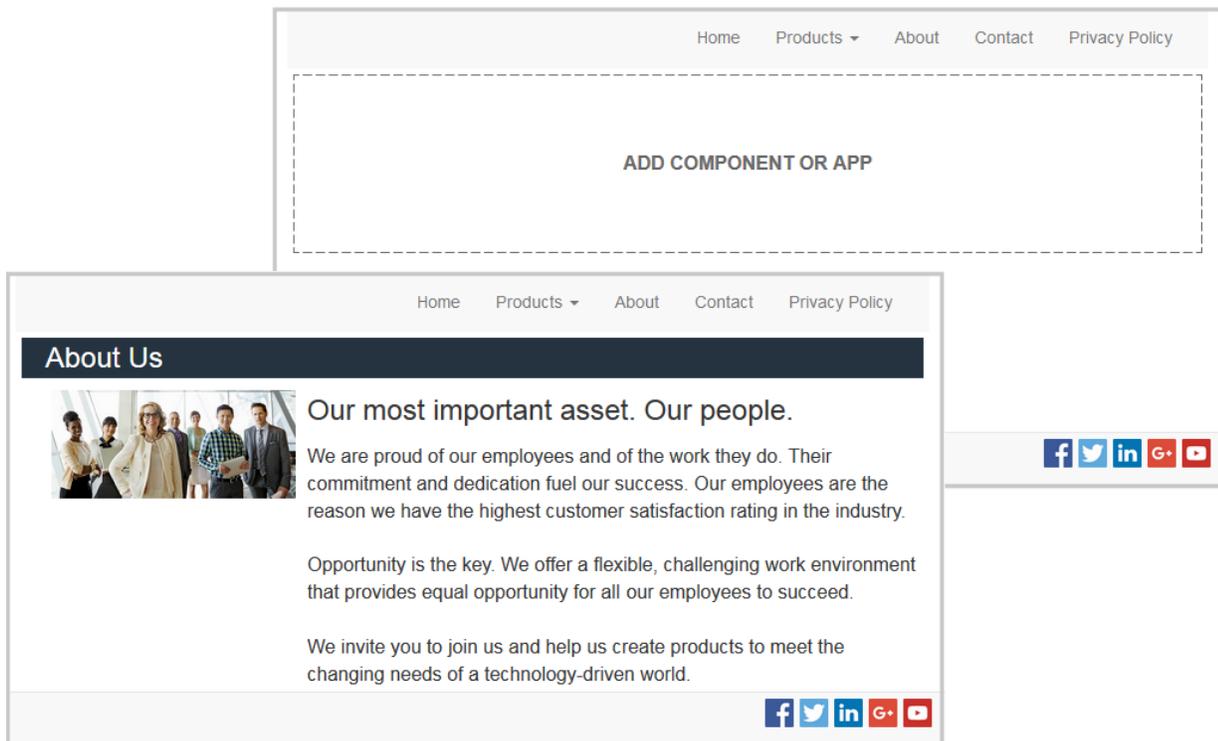
版面配置定義內容在頁面中的排列方式。不同的版面配置會有不同數目的具名槽位；槽位是指橫跨頁面寬度的區域。一個槽位可以有一或多種類型的內容。

每個主題都有數個頁面版面配置。當您將頁面新增至網站時，會選取該頁面要使用的版面配置。每個版面配置會有多個槽位供您拖放內容。這些槽位的內容由您所決定。它可以是任何項目，從標題、文字和分隔線到多媒體、圖庫和社群媒體等。您可以排列槽位中的內容，但無法變更頁面上槽位的數目或排列方式。若要這麼做，您必須使用新的頁面版面配置。

### 備註：

您可以將版面配置切換為其他版面配置。但切換時，請謹慎處理。如果所選擇之版面配置的具名槽位較少或不同，則其他槽位中的現有內容將不會顯示在新的版面配置中。除非您選擇的版面配置具有同名的槽位，否則內容不會被刪除，只是無法顯示而已。

下列圖解顯示頁面的範例版面配置。您可以在頁面版面配置中看到空槽位，並於完成的頁面中看到已新增標題、影像和文字。



選取頁面的其他版面配置：

1. 開啟網站進行編輯。
2. 瀏覽至頁面並按一下 。

3. 前往「頁面版面配置」欄位，然後從功能表中選取不同的版面配置。頁面版面配置的數目和類型取決於網站所關聯的主題。
4. 若要儲存目前更新項目中的所有擱置中變更，請按一下**儲存**。

## 變更背景或主題

您可以為頁面、頁面上的個別槽位以及槽位內的區段版面配置，指定背景色彩和影像。您也可以變更網站的整個主題。

槽位是橫跨頁面寬度的區域，可包含一或多個類型的內容。您為槽位指定的背景會套用到整個槽位及槽位中所有的元件。

### 變更背景

背景會彼此層層相疊。如果您指定槽位的背景，此背景會疊在為該頁面所指定背景的「上面」。對於大多數的影像與色彩而言，上層會覆蓋下層。如果您在背景色彩或影像中使用透明度，則下層色彩會穿透或融合於上層色彩。

有些元件 (例如段落與標題) 可提供背景色彩，作為預先定義樣式的一部分或作為自訂樣式。如果您為元件指定背景色彩，它將會層疊在任何區段版面配置、槽位或頁面背景上。請參閱[使用樣式和格式](#)。

頁面、槽位以及區段版面配置的背景選項雖然相似，但它們的存取方式卻略有不同：

1. 開啟網站進行編輯。
2. 瀏覽至您要編輯的頁面，並確定  已設為**編輯**。
3. 變更頁面或槽位的背景：
  - 按一下提要欄位中的 ，然後按一下 。
  - 若要指定整個頁面的背景，請按一下**完成頁面**，然後按一下 。若要指定頁面上槽位的背景，請選取槽位，然後按一下 。
4. 變更區段版面配置的背景：
  - 按一下區段版面配置的功能表圖示 ，然後選擇**設定值**。
  - 按一下**背景**。頁面、槽位以及區段版面配置的背景選項都相同。
5. 若要使用影像作為背景，請按一下**選取影像**。您可以使用您能夠存取的任何位置中的影像。您也可以使用與您共用的影像，或是您從本機或網路檔案位置上傳的影像。
6. 請選取影像，然後按一下**確定**。視窗會顯示所有可用的檔案。選擇適用於相關資訊環境的檔案類型。例如，假設您要選擇影像檔，請選取有效的影像格式 (GIF、JPG、JPEG、PNG 或 SVG)。
7. 若要調整影像設定值：
  - 使用**位置**將影像放置在頁面或槽位中。
  - 選取**比例**選項調整影像的呈現方式：
    - **最適大小**：將影像調整到整個影像裝入可用空間，但影像不扭曲。

- **延伸**：將兩個尺寸 (寬度或高度) 中較大尺寸調整到能剛好裝入可用空間，較小尺寸則延伸以填滿可用空間。
  - **無**：以完整的像素解析度使用影像，如果影像無法裝入可用空間，則會剪裁成統一的大小。
  - 選取**重複**選項，讓太小而無法填滿可用空間的影像並列顯示。您選取的**比例**選項為**延伸**時，則不適用此選項。
  - 選取**不隨頁面捲動**，讓使用者捲動頁面時影像保持不動。
8. 若要選取背景色彩，請從**色彩**功能表中選擇色彩，或按一下**其他**從完整的色彩選項中選取色彩。
- 按一下右側的色譜列可選擇色彩，並在色彩範圍顯示中顯示色彩的變化。此畫面的右上角顯示所選色彩，以及所選色彩的濃淡；向左移動會增加白色的量，向下移動會增加黑色的量。
  - 若要選取色彩的變化，請在色彩範圍顯示內按一下，或按一下選取點並拖曳到新的色彩位置。目前的色彩方塊和 6 位數色碼會顯示您選取的色彩。
  - 若要調整透明度 (Alpha 色頻)，請按一下透明度滑桿並向左拖曳，以增加透明度。Alpha 值為 0%，則色彩為透明，讓下層的色彩與內容能完全穿透顯示。Alpha 值為 100% 可讓色彩完全不透明，遮住下層所有的色彩與內容。其他的值可讓下層的影像與色彩穿透，並使其色彩與上層的色彩以不同的程度融合。
  - 若要套用目前選取的色彩，請按一下**選擇**。

### 變更主題

當您按一下**儲存**時，系統會將您對主題所做的變更記錄為更新。除非您發布更新項目，否則主題變更不會套用至網站。

#### 備註：

並非所有主題都彼此相容。不同的主題會包含不同的導覽、樣式名稱或頁面版面配置資訊。例如，假設您選擇的主題含有不同的版面配置名稱，則現有頁面的版面配置可能無法在該主題中使用。屆時相關頁面會一片空白。頁面其實並未刪除，只是無法顯示。您選擇的主題必須具有相同名稱的版面配置和槽位，頁面才會正常顯示。如果您選擇了不相容的主題，只要將主題變更回原始主題，即可回復頁面內容。切換時，務必使用相容的主題。如果您無法確定，請與主題開發人員確認。

變更網站使用的主題：

1. 開啟網站進行編輯。
2. 按一下提要欄位中的 ，然後按一下 。
3. 從可用的主題清單中選取主題，然後按一下**確定**。主題會套用至目前的更新項目中。若要儲存目前更新項目中的所有擱置中變更，請按一下**儲存**。

# 9

## 排列頁面內容

元件和內容項目提供您的使用者所需的功能和內容，而區段版面配置則可讓您自動排列這些內容。

- [新增元件和區段版面配置](#)
- [使用資產和內容項目](#)
- [使用觸發程式和動作](#)
- [使用水平區段版面配置](#)
- [使用垂直區段版面配置](#)
- [使用兩欄和三欄版面配置](#)
- [使用頁籤式區段版面配置](#)
- [使用滑桿區段版面配置](#)

如需使用個別元件的相關資訊，請參閱[使用內建元件](#)。

若要瞭解如何管理元件和版面配置，請參閱[管理自訂元件與版面配置](#)。

如需有關如何建立自己專屬元件的詳細資訊，請參閱[開發元件](#)。

## 新增元件和區段版面配置

元件是網頁的個別部分。[區段版面配置](#)會自動安排新增的內容，讓提供者可以輕鬆新增內容，不需再另外格式化頁面中的內容。

### 新增元件

新增元件到頁面：

1. 瀏覽至您要編輯的頁面，並確定  已設為**編輯**。
2. 按一下 ，然後按一下下列其中一種元件類型：
  - 按一下**主題**可顯示已選擇在網站相關聯主題中使用的元件清單。
  - 按一下**自訂**可顯示已與您共用的自訂元件清單。
  - 按一下**內建**可顯示服務提供的預設元件清單。
  - 按一下**全部**可顯示已與您共用的所有元件。
3. 按住特定圖示並將其拖曳到頁面上的所需位置。將項目拖曳到頁面上時，會顯示可用槽位、區段版面配置以及任何現有項目的界限。位置圖示  或垂直位置列  (位於左方或右方) 代表新內容的位置。區段版面配置或內容項目周圍的實心框線代表您可以將項目置於該處，系統將會自動安置該項目。一個槽位可以有多个項目，而且只要將頁面上的項目拖曳至新位置即可移動項目。

4. 若您覺得位置合適，可將項目放置在頁面上。
5. 若要調整項目的特性，請按一下項目的功能表圖示 ，然後選擇**設定**。視項目而定，您將需要新增文字、指定影像連結、指定其他網站或地圖的 URL、調整項目大小，或採取其他動作。

每個元件都包含大小、對齊方式、間距、色彩和框線等設定值。這些設定值會定義元件的外觀和功能。例如，段落設定值包含字型類型、字型大小和決定文字呈現方式的其他功能。

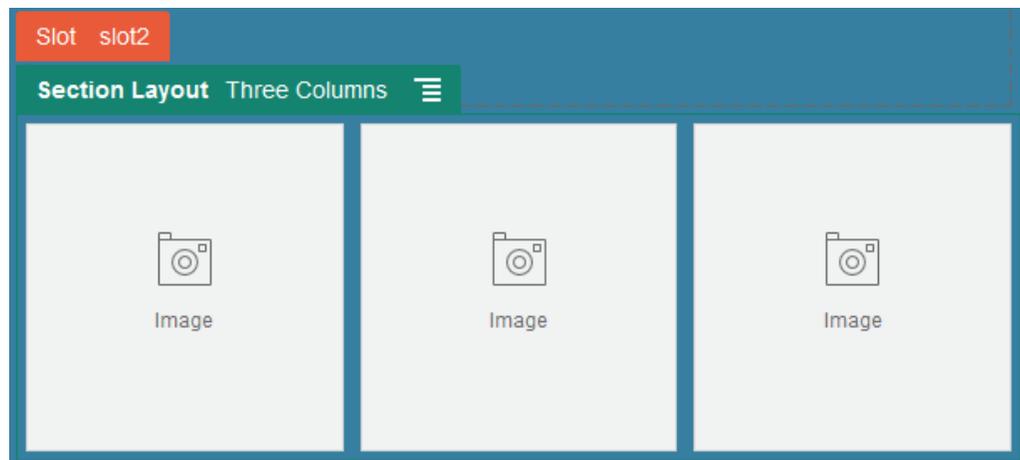
若要調整某個元件的特性，請選取該元件，然後按一下  **功能表** 並選擇**設定值**。設定值類型會因元件而有所不同：

設定值	描述
一般	一般設定值包括間距、對齊方式以及元件之間通用的設定值。
樣式	樣式是一組用於控制外觀的具名預設值。樣式在主題中加以定義。不同的主題可有不同的樣式，主題內不同的元件也可有不同的樣式。您也可以手動為特定的元件執行處理指定樣式設定值。
連結	連結設定值包括元件所使用之影像、文件或其他資源的位置。對於能夠執行動作的按鈕和其他元件，連結設定值也包括元件所支援的觸發程式和動作。
元件	包括其他標準元件的元件 (例如文章元件) 會有一份個別元件清單，而且您可以存取每個這些元件的設定值。
自訂	自訂設定值是元件的獨特設定值，會與標準設定值群組分開顯示。例如，遠端元件可以在獨特的 URL 儲存偏好設定，然後將它們顯示成自訂設定值。

### 使用區段版面配置

區段版面配置會自動將內容安排到多個資料欄、垂直清單或一組頁籤式區域。網站提供者可以新增一或多個區段版面配置到槽位以組織內容。若要新增版面配置至頁面：

1. 瀏覽至您要編輯的頁面，並確定  已設為**編輯**。
2. 按一下  可在頁面中插入區段版面配置。選擇您想要使用的版面配置類型，並依照和放置元件相同的準則來放置項目。



位置圖示  代表區段版面配置的位置 (上、下、左、右)。區段版面配置或內容項目周圍的實心框線代表您可以將項目置於該處，系統將會自動安置該項目：



您若是使用結構化內容項目的企業使用者，就可以在將內容項目清單加到頁面時指定區段版面配置，此區段版面配置將會自動為頁面上的項目製作格式。

除了提供的版面配置外，開發人員還可建立額外的區段版面配置，以解決特定的版面配置問題或簡化提供者的編寫工作。主題設計者甚至可以在頁面版面配置中的槽位建置區段版面配置。請參閱[開發版面配置](#)。

## 使用資產和內容項目

企業使用者可以在其網站中包含 *網站集合* 或關聯的儲存區域中所儲存的數位資產和內容項目。集合是儲存區域中可在網站上使用的資產子集。

若要為頁面新增數位資產或內容項目，請確定  已設為 **編輯**，然後按一下 。如果您的網站使用多個儲存區域，請選取要使用的儲存區域。如需在網站使用多個儲存區域的相關資訊，請參閱 *運用 Oracle Content Management 管理資產* 中的讓網站存取多個儲存區域。

您可以篩選資產以找出真正需要的資產。按一下  並選擇如何縮小選擇範圍。如果您未見到任何資產，代表資產可能未包含在網站集合中。請變更篩選條件以顯示儲存區域中的所有集合，而不要只顯示網站集合，看看這樣是否有所幫助。請參閱 *搜尋、篩選及排序資產* 以取得完整詳細資訊。

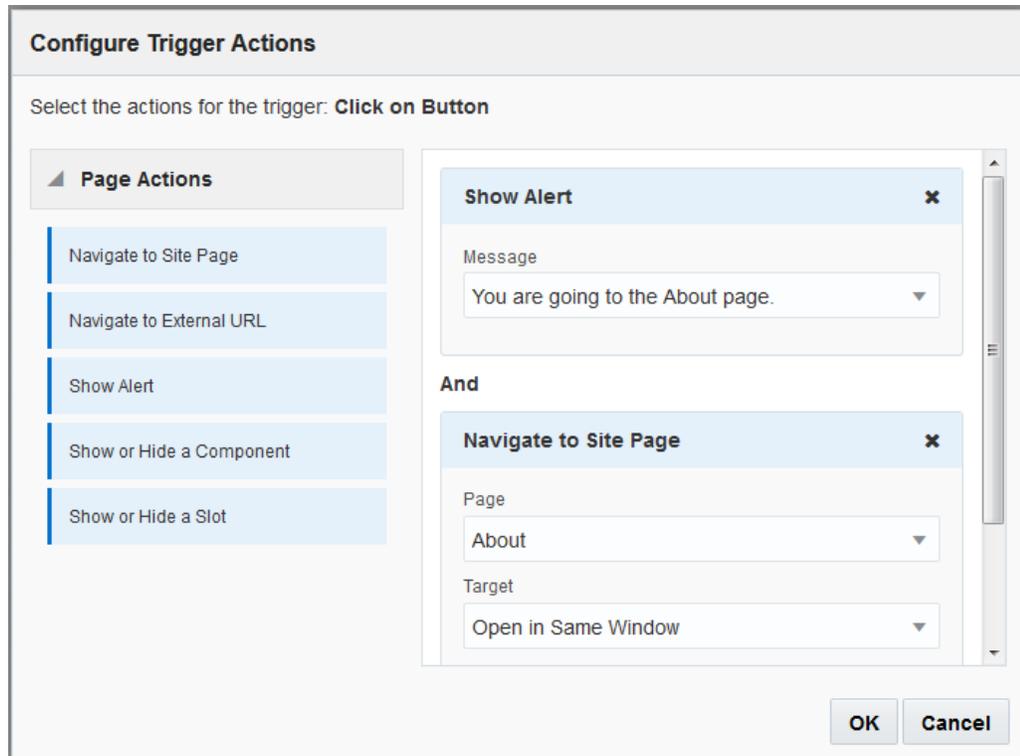
將數位資產或內容項目從面板拖放到頁面上的槽位。您可以將影像直接內嵌到「段落」元件中游標所在的位置，並使用相關選項輸入替代文字、設定影像高度與寬度，以及設定對齊方式。

若您是將網站集合中的項目加到頁面，該項目會自動放在適當類型的元件中。例如，若是新增影像數位資產，它會自動放在影像元件中。若是新增內容項目，它會自動放在內容項目元件中。您也可以先新增影像或內容項目元件，之後再將內容項目從「內容」面板拖放到此元件上。抑或是您可以選取該元件的 **設定值**，然後按一下 **選取**，從您的資產或文件清單中選擇一個影像。

## 使用觸發程式和動作

按鈕元件可起始一或多項動作 (例如，顯示或隱藏頁面元件，以及顯示訊息)。某些元件 (例如資料夾和檔案清單) 會依據使用者所做的選擇，在伴隨的元件中起始動作。

例如，您可以設定一個按鈕，當按下按鈕時 (觸發程式)，會將使用者導向另一個頁面或外部 URL 並顯示通知使用者發生變更的警示 (動作)。您看到的顯示內容將會根據使用的元件種類而改變。



如果您使用多個動作，請考慮作業順序，並依想要的執行順序來放置動作。在上述範例中，請優先列出警示動作。這會讓使用者在被重新導向至頁面之前有時間讀取和關閉訊息。若優先列出重新導向動作，使用者可能還來不及讀取訊息，訊息就已被新頁面取代。

### 頁面動作

支援觸發程式和動作的所有元件都支援頁面動作：

- **瀏覽至網站頁面**：選取目前網站上的頁面。
- **瀏覽至外部 URL**：指定外部頁面或網站的完整 URL。
- **顯示警示**：在視窗中顯示指定的訊息。
- **顯示或隱藏元件**：從元件清單中，選取要在目前頁面上顯示、隱藏或切換的元件。
- **顯示或隱藏槽位**：從槽位清單中，選取要在目前頁面上顯示、隱藏或切換的槽位。

### 元件特定動作

除了頁面動作之外，元件也可定義自己的動作。這些動作可讓元件與其他元件進行通訊，並且在伴隨的元件中起始動作。例如，當使用者從資料夾清單元件中選取某個資料夾後，檔案清單元件即可顯示所選資料夾中的檔案。在此案例中，資料夾清單元件支援**已選取資料夾**觸發程式，而檔案清單元件則支援**顯示檔案**動作。

如果您新增支援動作的元件到頁面，元件和其支援的所有動作都會新增到可用的動作清單中。有些元件只支援動作或特定動作。有些元件則提供觸發程式，但本身不支援任何動作。

若要瞭解如何建立自己的元件，請參閱[開發元件](#)。

例如，若要為按鈕元件指定一或多個動作：

1. 按一下按鈕的功能表圖示 ，然後選擇**設定值**。
2. 在「設定值」面板中，按一下**連結**。
3. 按一下**選取連結類型**，然後選擇**觸發程式動作**。
4. 在「可用的觸發程式」下方，按一下**按一下按鈕**。
5. 在「設定觸發程式動作」視窗中，按一下動作，並將動作從左邊的資料欄拖放至標籤為**執行第一個動作**的槽位。

## 使用水平區段版面配置

您可以使用區段版面配置自動決定您新增至版面配置的元件間距與排列方式。

水平版面配置可將新增至其中的項目以水平方式一個接著一個排列。版面配置會隨著頁面寬度的增加或減少，按照比例變更。項目預設會調整為一條水平線的位置，並且配置相等的水平空間。

若要新增版面配置至頁面：

1. 瀏覽至您要編輯的頁面，並確定  已設為**編輯**。
2. 將版面配置新增至頁面。
3. 若要將其他內容放置在區段版面配置中，請將內容拖放至版面配置上。

此版面配置會標示實心框線和橫幅，橫幅中顯示「新增項目」。



您可以繼續將項目新增至版面配置，版面配置將隨之設定項目的格式。您甚至可以新增其他區段版面配置，以建立精美的版面配置。

以下是一個水平版面配置，其中含有顯示版面配置中項目順序的已編號文字元件：



4. 若要編輯元件與其外觀，請按一下其功能表圖示 ，然後選擇**設定值**。若不要使用依比例調整大小的預設功能，您可以設定個別區域的寬度。您也可以設定對齊方式 (靠左、置中或靠右)。
5. 使用「一般」頁籤修改版面配置中個別元件的設定值。  
按一下元件名稱可查看該元件的設定值。
6. 使用「背景」頁籤修改版面配置的背景設定值。  
請參閱[變更背景或主題](#)。

## 使用兩欄和三欄版面配置

您可以使用多欄區段版面配置，自動決定新增至版面配置之元件的間距和排列方式。

若要新增版面配置至頁面：

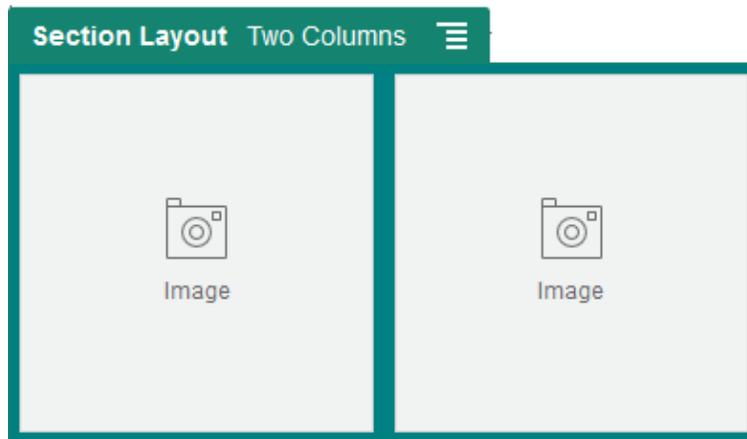
1. 瀏覽至您要編輯的頁面，並確定  已設為**編輯**。
2. 將版面配置新增至頁面。
3. 若要將其他內容放置在區段版面配置中，請將內容拖放至版面配置上。

此版面配置會標示實心框線和橫幅，橫幅中顯示「新增項目」。



您可以繼續將項目新增至版面配置，版面配置將隨之設定項目的格式。您甚至可以新增其他區段版面配置來建立精美的版面配置，或是新增元件群組。

以下是一個兩欄的版面配置，其中以影像元件作為預留位置：



如果您新增更多項目，它們會顯示在額外的資料列中，且每列最多兩個項目。

4. 若要編輯元件與其外觀，請按一下其功能表圖示 ，然後選擇**設定值**。
5. 使用「一般」頁籤修改版面配置中個別元件的設定值。  
按一下元件名稱可查看該元件的設定值。
6. 使用「背景」頁籤修改版面配置的背景設定值。
7. 選擇**自訂設定值**設定所顯示內容的其他預設值。
  - **第一個資料欄寬度 (%)**：以版面配置可用間距百分比為單位指定資料欄寬度。
  - **第二個資料欄寬度 (%)**：以版面配置可用間距百分比為單位指定資料欄寬度。
  - **第三個資料欄寬度 (%)**：以版面配置可用間距百分比為單位指定資料欄寬度。

- **響應式中斷點 (像素)**：對於當可用的顯示大小改變時會自動重新設定內容格式的響應式頁面設計，請以像素為單位，指定區段版面配置在標準兩欄版面配置，與您在下方所指定之**響應式行為**選項之間切換時的寬度。
- **響應式行為**：選取當可用顯示大小比**響應式中斷點**值小時，版面配置變更的方式。
  - **不動作**：不調整版面配置行為。
  - **堆疊資料欄**：將項目從上到下在單一資料欄中進行排列，其中先顯示第一個資料欄的所有項目，再顯示第二個資料欄的項目，依此類推。
  - **隱藏第一個資料欄**：隱藏第一個資料欄的內容，為其餘的資料欄提供更多空間。
  - **隱藏第二個資料欄**：隱藏第二個資料欄的內容，為其餘的資料欄提供更多空間。
  - **隱藏第三個資料欄**：隱藏第三個資料欄的內容，為其餘的資料欄提供更多空間。
  - **隱藏兩個資料欄**：隱藏版面配置中所有的內容。
  - **將第二個資料欄移動至第一個資料欄底下**：在單一資料欄中先排列第一個資料欄中的所有項目，後面再接著第二個資料欄中的所有項目。
  - **將第一個資料欄移動至第二個資料欄底下**：在單一資料欄中先排列第二個資料欄中的所有項目，後面再接著第一個資料欄中的所有項目。

## 使用垂直區段版面配置

您可以使用區段版面配置自動決定您新增至版面配置的元件間距與排列方式。

垂直版面配置可將新增至其中的項目以垂直方式一個接著一個排列。

若要新增版面配置至頁面：

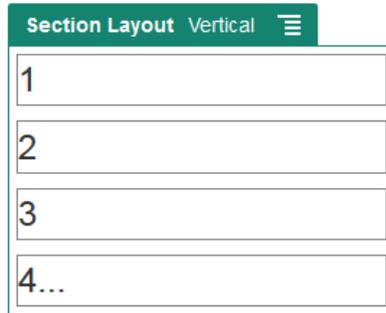
1. 瀏覽至您要編輯的頁面，並確定  已設為**編輯**。
2. 將版面配置新增至頁面。
3. 若要將其他內容放置在區段版面配置中，請將內容拖放至版面配置上。

此版面配置會標示實心框線和橫幅，橫幅中顯示「新增項目」。



您可以繼續將項目新增至版面配置，版面配置將隨之設定項目的格式。您甚至可以新增其他區段版面配置，以建立精美的版面配置。

以下是一個垂直版面配置，其中含有顯示版面配置中項目順序的已編號文字元件：



4. 若要編輯元件與其外觀，請按一下其功能表圖示 ，然後選擇**設定值**。
5. 使用「一般」頁籤修改版面配置中個別元件的設定值。  
按一下元件名稱可查看該元件的設定值。
6. 使用「背景」頁籤修改版面配置的背景設定值。

## 使用頁籤式區段版面配置

您可以使用頁籤式區段版面配置，為新增至版面配置中的元件建立間距與排列方式。

若要新增版面配置至頁面：

1. 瀏覽至您要編輯的頁面，並確定  已設為**編輯**。
2. 將版面配置新增至頁面。版面配置預設只有單一頁籤。按一下**新頁籤**以新增其他頁籤。
3. 若要在頁籤中放入其他內容，請將內容拖放到頁籤中。  
您可以繼續將項目新增至版面配置，版面配置將隨之設定項目的格式。您甚至可以新增其他區段版面配置來建立精美的版面配置，或是新增元件群組。
4. 若要編輯頁籤及其外觀，請按一下其功能表圖示 ，然後選擇**設定值**。
5. 使用「背景」區段可修改頁籤的背景設定值。  
您可以針對頁籤使用影像、變更其位置等等。請參閱[變更背景或主題](#)。
6. 使用「樣式」區段可修改頁籤的其他外觀設定值。選擇樣式 (例如極細、框架等等)，或自訂頁籤框線和角的樣式。

## 使用滑桿區段版面配置

使用滑桿區段版面配置建立的內容，在「滑動」至某一端以騰出空間給新內容使用之前，會維持在原本位置。使用者可按一下投影片區段底下的導覽點來瀏覽投影片。

將滑桿版面配置新增至頁面：

1. 瀏覽至您要編輯的頁面，並確定  已設為**編輯**。
2. 將版面配置新增至頁面。版面配置預設只有單一投影片。按一下**新投影片**可新增其他投影片。

3. 若要在特定投影片上放置內容，請在滑桿中顯示該投影片，然後將元件或內容項目拖放到投影片上。  
您可以繼續新增項目至投影片，並且放置在所需的位置。您可以新增其他區段版面配置來建立精美的版面配置，或是新增元件群組，讓您將內容放置到想要的位置。
4. 若要編輯投影片版面配置和其外觀，請按一下投影片的功能表圖示 ，然後選擇**設定值**。
5. 使用「背景」區段可修改整個投影片區域的背景設定值。  
請參閱[變更背景或主題](#)。您可以使用影像、變更影像的位置等等。
6. 使用「樣式」區段可修改其他外觀設定值。選擇樣式 (例如極細、框架等等)，或自訂投影片框線和角的樣式。
7. 若要修改個別投影片的設定值，請按一下投影片的名稱以開啟其設定值。視需要變更每個投影片的背景、樣式等等項目。

# 10

## 使用內建元件

元件、區段版面配置、數位資產以及結構化內容可讓您彈性提供使用者所需的內容和功能。

本節提供有關使用個別元件的資訊。如需元件的其他資訊，請參閱下列主題：

- 若要瞭解如何在您的網站中使用元件，請參閱[排列頁面內容](#)。
- 若要瞭解如何管理元件，請參閱[管理自訂元件與版面配置](#)。
- 若要瞭解如何建立自己的元件，請參閱[開發元件](#)。

### 基本元件

- [標題](#)
- [段落](#)
- [純文字](#)
- [按鈕](#)

### 結構元件

- [分隔線](#)
- [空白間隔](#)

### 媒體元件

- [影像](#)
- [圖庫](#)
- [圖庫格線](#)
- [YouTube 視訊](#)
- [視訊](#)

### 文件元件

- [文件](#)
- [資料夾清單](#)
- [檔案清單](#)
- [文件管理程式](#)
- [專案程式庫](#)

### 社群元件

- [社群列](#)
- [Facebook 按讚與推薦](#)

- [Twitter 分享與跟隨](#)
- [對話元件](#)
- [對話清單](#)

#### 處理作業元件

- [處理作業開始表單](#)
- [處理作業工作清單](#)
- [工作詳細資訊表單](#)

#### 內容項目

- [內容項目元件](#)
- [內容預留位置](#)
- [內容清單](#)
- [內容搜尋](#)
- [建議](#)

#### 其他元件

- [地圖](#)
- [大標題](#)
- [文章](#)
- [影像 \(含文字\)](#)
- [元件群組](#)
- [在頁面上使用 Cobrowse 功能](#)
- [Oracle Intelligent Advisor](#)
- [Oracle Visual Builder](#)

## 基本元件

基本元件提供文字內容的建置區塊。

- [標題](#)
- [段落](#)
- [純文字](#)
- [按鈕](#)

## 標題

新增和格式化標題元件：

1. 瀏覽至您要編輯的頁面，並確定  已設為**編輯**。

2. 新增元件至頁面。
3. 按一下標題元件，以輸入標題文字。文字會採用元件預設樣式的格式。
4. 在標題文字內新增連結：
  - a. 輸入並選取作為連結文字使用的文字，然後按一下 。
  - b. 按一下**選取連結類型**，然後選擇下列其中一個選項：
    - **網頁**：指定外部頁面或網站的完整 URL，並選取要在何處開啟連結。
    - **網站頁面**：使用頁面選擇器選取目前網站上的頁面，並選取要在何處開啟連結。您可以使用 `key1=value1&key2=value2` 格式指定其他 URL 參數。支援使用空的值；例如 `key1=&key2=value2`。您也可以指定 URL 錨點，但必須在目標網站頁面上要解析此錨點連結的位置新增特殊「錨點」區段版面配置，並在版面配置設定值中指定定義觸發程式動作時所使用的相同錨點名稱。

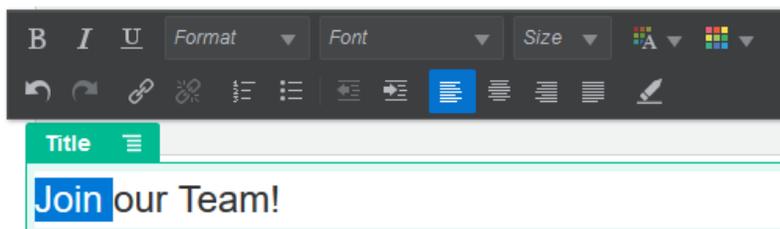
 **備註：**

需要使用 URL 錨點的「錨點」區段版面配置是由 OCE 工具程式提供。如需有關如何取得此工具程式的資訊，請參閱使用 OCE 工具程式進行開發。

- **檔案下載**：從儲存區域下載選取的檔案。選取檔案。  
如果選取數位資產，您可以選取特定的轉譯。若未選取轉譯，則會使用原始大小。若想要在網站發布時一同發布最新版本的資產，請選取**使用最新版本的資產**。若未選取**使用最新版本的資產**，則會使用最新的**已發布**版本，而不會使用更新的草稿版本 (如果有的話)。
- **檔案預覽**：在覆蓋於變暗且非作用中版本頁面上方的浮動層預覽選取的檔案，也可選擇加以下載。
- **內容項目**：從關聯的資產儲存區域中選取一個內容項目，選擇您要顯示的詳細資訊頁面，並將此頁面定位在同一個視窗或新視窗中開啟。
- **電子郵件**：指定有效的電子郵件地址，並視需要指定主旨。產生的訊息會在預設電子郵件從屬端中開啟，並透過預設電子郵件從屬端傳送。
- **地圖**：輸入有效的地址或座標，並選取是要在桌面還是行動裝置瀏覽器中開啟地圖。
- **電話**：輸入有效的電話號碼。

如果要移除連結，請在連結文字的任意處按一下 。

5. 如果您要變更預設格式，請選取您要格式化的文字，然後在格式設定工具列 (例如字型、色彩或對齊方式) 中選取任何選項。依據主題的不同，您不一定能見到所有這些選項。



系統會立即套用您所做的變更。這些格式變更會覆蓋基本樣式。如果您變更基本樣式，覆寫的樣式依舊會保留。

6. 若要移除使用這些選項所套用的格式設定，請選取文字，然後按一下 。

## 段落

新增和格式化段落元件：

1. 瀏覽至您要編輯的頁面，並確定  已設為**編輯**。
2. 新增元件至頁面。
3. 按一下段落元件來輸入文字。文字會採用元件預設樣式的格式。
4. 若要在段落中加入視訊 ()、影像 () 或內容項目 ()，請按一下適當的圖示。選取要新增的資產，並視需要調整設定值。例如，插入一個內容項目，然後選取不同的檢視來控制段落內的複雜版面配置。
5. 若要在段落元件內新增其他段落，只需要按 **Enter** 鍵。
6. 若要在段落內新增連結：
  - a. 輸入並選取作為連結文字使用的文字，然後按一下 。
  - b. 按一下**選取連結類型**，然後選擇下列其中一個選項：
    - **網頁**：指定外部頁面或網站的完整 URL，並選取要在何處開啟連結。
    - **網站頁面**：使用頁面選擇器選取目前網站上的頁面，並選取要在何處開啟連結。您可以使用 `key1=value1&key2=value2` 格式指定其他 URL 參數。支援使用空的值；例如 `key1=&key2=value2`。您也可以指定 URL 錨點，但必須在目標網站頁面上要解析此錨點連結的位置新增特殊「錨點」區段版面配置，並在版面配置設定值中指定定義觸發程式動作時所使用的相同錨點名稱。

### 備註：

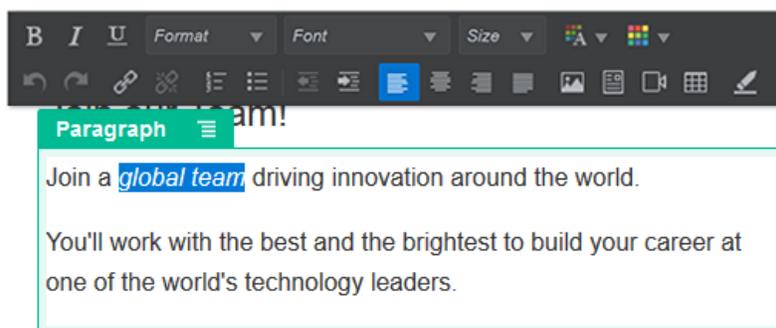
需要使用 URL 錨點的「錨點」區段版面配置是由 OCE 工具程式提供。如需有關如何取得此工具程式的資訊，請參閱使用 OCE 工具程式進行開發。

- **檔案下載**：從儲存區域下載選取的檔案。選取檔案。  
如果選取數位資產，您可以選取特定的轉譯。若未選取轉譯，則會使用原始大小。若想要在網站發布時一同發布最新版本的資產，請選取**使用最新版本的資產**。若未選取**使用最新版本的資產**，則會使用最新的**已發布**版本，而不會使用更新的草稿版本 (如果有的話)。
- **檔案預覽**：在覆蓋於變暗且非作用中版本頁面上方的浮動層預覽選取的檔案，也可選擇加以下載。
- **內容項目**：從關聯的資產儲存區域中選取一個內容項目，選擇您要顯示的詳細資訊頁面，並將此頁面定位在同一個視窗或新視窗中開啟。

- **電子郵件**：指定有效的電子郵件地址，並視需要指定主旨。產生的訊息會在預設電子郵件從屬端中開啟，並透過預設電子郵件從屬端傳送。
- **地圖**：輸入有效的地址或座標，並選取是要在桌面還是行動裝置瀏覽器中開啟地圖。
- **電話**：輸入有效的電話號碼。

如果要移除連結，請在連結文字的任意處按一下，然後按一下 。

7. 如果您要變更文字任何部分的預設格式，請選取您要格式化的文字，然後在格式設定工具列 (例如字型、色彩或對齊方式) 中選取任何選項。依據主題的不同，您不一定能見到所有這些選項。



系統會立即套用您所做的變更。這些格式變更會覆蓋基本樣式。如果您變更基本樣式，覆寫的樣式依舊會保留。

8. 若要移除已套用這些選項的格式，請選取文字，然後按一下 。

## 純文字

您可以透過文字元件將文字加到頁面，並且使用主題中定義的樣式設定文字格式。編輯內容時無法使用格式設定工具列，因此您無法以自訂格式覆寫選取的樣式。

如果您針對元素 (例如標題) 使用標準化格式，這會非常有用。您可以更新主題中定義的樣式，並自動更新網站的所有關聯文字，不會有格式覆寫的問題。

新增文字元件：

1. 瀏覽至您要編輯的頁面，並確定  已設為**編輯**。
2. 新增元件至頁面。
3. 按一下段落元件來輸入文字。文字會採用元件 (<p> 或段落元素) 之預設樣式的格式。若要新增其他段落，只要按 **Enter** 鍵。

### 備註：

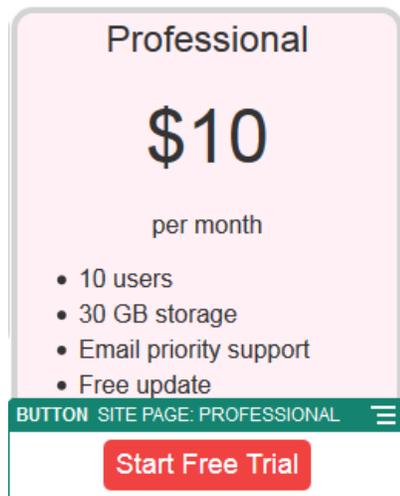
您可以複製其他來源的內容並貼到文字元件中，但是會移除影像和相關的 HTML 標記。

4. 變更文字元件的基本樣式：

- a. 按一下元件的功能表圖示 ，然後選擇**設定值**。
- b. 按一下**選擇樣式**，然後從功能表中選擇樣式。  
此清單包括段落和標題的標準 HTML 標記。您所選的樣式會指定關聯的標記給內容。  
元件會反映所選的基本樣式。

## 按鈕

按鈕可讓連結或其他功能在頁面上更為明顯。



在頁面中新增按鈕：

1. 瀏覽至您要編輯的頁面，並確定  已設為**編輯**。
2. 新增元件至頁面。
3. 若要調整按鈕的特性，請按一下其功能表圖示 ，然後選擇**設定值**。  
您可以指定按鈕上的文字 (標籤)、大小、對齊方式，以及按鈕的其他顯示選項。

### 備註：

如果您將**寬度**和**高度**欄位設為 0 (零)，則會自動調整按鈕大小，以符合您為標籤指定的文字長度。

若要指定背景色彩、字型、框線和其他設定值，請按一下**樣式**頁籤。您可以從目前主題中的預先定義樣式做選擇。依主題的不同，您可能也可以按**自訂**來指定自己的值。

4. 建立連結或其他動作與按鈕的關聯：
  - a. 在「設定值」面板中，按一下**連結**。
  - b. 按一下**選取連結類型**，然後選擇下列其中一個選項：

- **沒有連結**：使用者按一下按鈕時，按鈕不會執行任何動作。
- **網頁**：指定外部頁面或網站的完整 URL，並選取要在何處開啟連結。
- **網站頁面**：使用頁面選擇器選取目前網站上的頁面，並選取要在何處開啟連結。您可以使用 `key1=value1&key2=value2` 格式指定其他 URL 參數。支援使用空的值；例如 `key1=&key2=value2`。您也可以指定 URL 錨點，但必須在目標網站頁面上要解析此錨點連結的位置新增特殊「錨點」區段版面配置，並在版面配置設定值中指定定義觸發程式動作時所使用的相同錨點名稱。

 **備註：**

需要使用 URL 錨點的「錨點」區段版面配置是由 OCE 工具程式提供。如需有關如何取得此工具程式的資訊，請參閱使用 OCE 工具程式進行開發。

- **檔案下載**：從儲存區域下載選取的檔案。選取檔案。  
如果選取數位資產，您可以選取特定的轉譯。若未選取轉譯，則會使用原始大小。若想要在網站發布時一同發布最新版本的資產，請選取**使用最新版本的資產**。若未選取**使用最新版本的資產**，則會使用最新的已發布版本，而不會使用更新的草稿版本 (如果有的話)。
- **檔案預覽**：在覆蓋於變暗且非作用中版本頁面上方的浮動層預覽選取的檔案，也可選擇加以下載。
- **內容項目**：從關聯的資產儲存區域中選取一個內容項目，選擇您要顯示的詳細資訊頁面，並將此頁面定位在同一個視窗或新視窗中開啟。
- **電子郵件**：指定有效的電子郵件地址，並視需要指定主旨。產生的訊息會在預設電子郵件從屬端中開啟，並透過預設電子郵件從屬端傳送。
- **觸發程式動作**：選取要在按下按鈕時執行的一或多個頁面動作。頁面動作包括：
  - **瀏覽至網站頁面**：選取目前網站上的頁面。
  - **瀏覽至外部 URL**：指定外部頁面或網站的完整 URL。
  - **顯示警示**：在視窗中顯示指定的訊息。
  - **顯示或隱藏元件**：選取要在目前頁面上顯示、隱藏或切換的元件。
  - **顯示或隱藏槽位**：選取要在目前頁面上顯示、隱藏或切換的槽位。
- **地圖**：輸入有效的地址或座標，並選取是要在桌面還是行動裝置瀏覽器中開啟地圖。
- **電話**：輸入有效的電話號碼。

請參閱[使用觸發程式和動作](#)。

## 結構元件

結構元件可協助分隔頁面中的內容。

- [分隔線](#)
- [空白間隔](#)

## 分隔線

您可以使用分隔線 (水平線)，在資料欄中或跨頁面建立視覺中斷效果。

新增分隔線至頁面：

1. 瀏覽至您要編輯的頁面，並確定  已設為**編輯**。
2. 新增元件至頁面。



3. 若要調整分隔線的特性，請按一下其功能表圖示 ，然後選擇**設定值**。

您可以調整左右間距選項來指定分隔線的水平位置和長度。

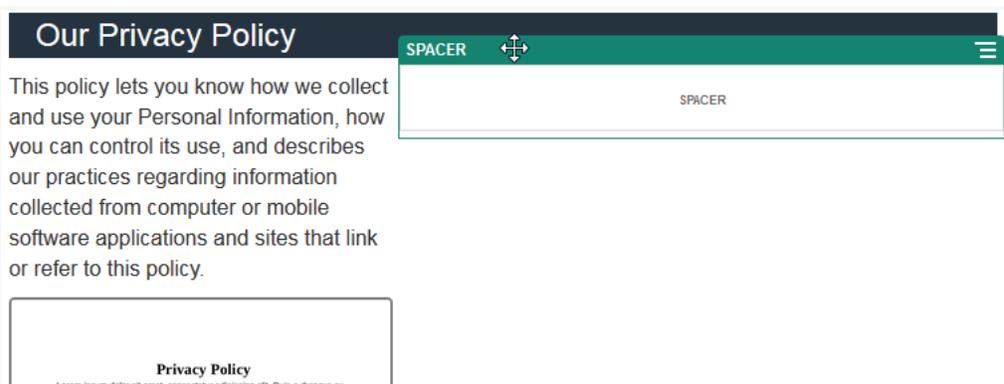
線條色彩、粗細和其他設定值是透過元件樣式所定義。您可以從目前主題的預先定義樣式中進行選擇，或按一下**自訂**以指定您自己的值。

## 空白間隔

管理頁面上的留空可以讓頁面在視覺上更加吸引人，而且更容易閱讀。使用空白間隔元件來新增空格，而不需要覆寫樣式或其他元件中所定義的間距。

在頁面中的元件之間新增垂直空白間隔：

1. 瀏覽至您要編輯的頁面，並確定  已設為**編輯**。
2. 新增元件至頁面。



3. 若要調整空白間隔的高度，請按一下其功能表圖示 ，然後選擇**設定值**。

# 媒體元件

媒體元件可將影像和視訊導入您的頁面。

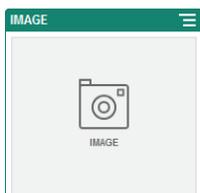
- [影像](#)
- [圖庫](#)
- [圖庫格線](#)
- [YouTube 視訊](#)
- [視訊](#)

## 影像

影像可以吸引檢視者的注意力，驅使檢視者探索頁面上的不同內容區域。

新增影像至頁面：

1. 瀏覽至您要編輯的頁面，並確定  已設為**編輯**。
2. 新增元件至頁面。影像元件會顯示預留位置影像，直到您選取您要使用的影像為止。



3. 若要選取影像，請按一下功能表圖示 ，選擇**設定值**，然後按一下「影像」欄位旁的**選取**。
4. 從網站儲存區域中選取影像 (此影像來自與您共用的文件資料夾中)，或上傳影像到文件資料夾。

### 備註：

視窗會顯示所有可用的檔案。您必須選擇適用於相關資訊環境的檔案類型。例如，如果您要選擇影像檔，則必須選取有效的影像格式 (GIF、JPG、JPEG、PNG 或 SVG) 檔案。

- a. 找到並選取您要使用的圖像。

若未見到任何數位資產，請按一下 ，然後將集合篩選條件變更為**全部**。

- b. 若從文件資料夾中選取影像，您可以連結到檔案，而不是將檔案複製到網站。若要連結至檔案，請選取**使用原始檔案參照**，而不是**將檔案複製到網站**。若未選取此選項，系統就會在網站儲存檔案的複本，然後從網站參照。若連結至原始的檔案，就不需要複製內容。即使檔案的權限變更或其他因素而限制了檢視，網站訪客仍能透過連結查看內容。

- c. 按一下**確定**。
  - d. 如果選取數位資產，您可以選取特定的轉譯。若未選取轉譯，則會使用原始大小。若想要在網站發布時一同發布最新版本的資產，請選取**使用最新版本的資產**。
5. 若要編輯影像，請按一下 ，然後使用以下任一動作編輯影像：
- 若要剪裁影像，請按一下  **剪裁**。從剪裁工具列中選取其中一個預先定義的影像比例，或視需要拖曳影像上的剪裁控點。完成之後，請按一下剪裁工具列中的**剪裁**。
  - 若要旋轉或翻轉影像，請按一下  **旋轉**。在旋轉工具列中，輸入自訂的旋轉角度、使用按鈕將影像向左或向右旋轉，或是選取要將影像水平翻轉還是垂直翻轉。
  - 若要為影像加上浮水印，請按一下  **浮水印**。使用浮水印工具，視需要新增影像的文字、變更文字大小、樣式、色彩以及不透明度。
  - 若要變更影像格式，請按一下  **選項**，然後從**格式**下拉式清單中選取一個新的格式。
  - 若要變更背景色彩，請按一下  **選項**，然後從**背景色彩**下拉式功能表中選取一個選項。
  - 如果是編輯 .jpg 或 .webp (可在 Google Chrome 瀏覽器中使用)，可以變更影像品質以製作較小的影像檔。按一下  **選項**，然後在**品質**方塊中輸入新的百分比。
  - 若要還原或重做變更，請按一下  或 。若要移除所做的全部變更，請按一下**重設**。
  - 若要變更影像的放大倍率，請使用縮放控制項 ( )。
6. 在**設定值**功能表  的**一般**面板中新增說明文字，或是修改間距、對齊方式、樣式以及其他選項。
7. 您可以為影像設定連結或其他動作：
- a. 在「設定值」面板中，按一下**連結**。
  - b. 請選取下列其中一個選項：
    - **沒有連結**：使用者按一下影像時，影像不會執行任何動作。
    - **網頁**：指定外部頁面或網站的完整 URL，並選取要在何處開啟連結。
    - **網站頁面**：使用頁面選擇器選取目前網站上的頁面，並選取要在何處開啟連結。您可以使用 key1=value1&key2=value2 格式指定其他 URL 參數。支援使用空的值；例如 key1=&key2=value2。您也可以指定 URL 錨點，但必須在目標網站頁面上要解析此錨點連結的位置新增特殊「錨點」區段版面配置，並在版面配置設定值中指定定義觸發程式動作時所使用的相同錨點名稱。

 **備註：**

需要使用 URL 錨點的「錨點」區段版面配置是由 OCE 工具程式提供。如需有關如何取得此工具程式的資訊，請參閱使用 OCE 工具程式進行開發。

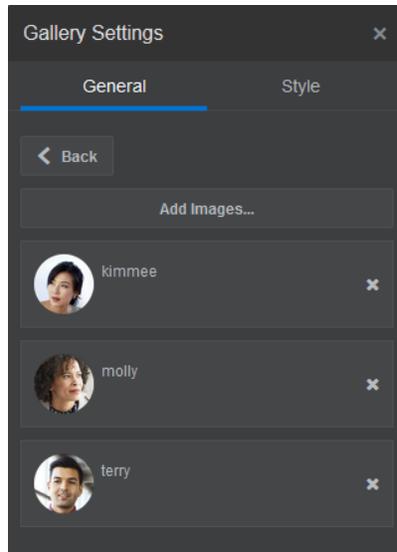
- **檔案下載：**從儲存區域下載選取的檔案。選取檔案。  
如果選取數位資產，您可以選取特定的轉譯。若未選取轉譯，則會使用原始大小。若想要在網站發布時一同發布最新版本的資產，請選取**使用最新版本的資產**。若未選取**使用最新版本的資產**，則會使用最新的**已發布**版本，而不會使用更新的草稿版本 (如果有的話)。
- **檔案預覽：**選取的檔案將會顯示成頁面上的浮動層。
- **內容項目：**從關聯的資產儲存區域中選取一個內容項目，選擇您要顯示的詳細資訊頁面，並將此頁面定位在同一個視窗或新視窗中開啟。
- **電子郵件：**指定有效的電子郵件地址，並視需要指定主旨。產生的訊息會在預設電子郵件從屬端中開啟，並透過預設電子郵件從屬端傳送。
- **影像預覽：**選取的影像將會顯示成頁面上的浮動層。
- **地圖：**輸入有效的地址或座標，並選取是要在桌面還是行動裝置瀏覽器中開啟地圖。
- **電話：**輸入有效的電話號碼。

## 圖庫

您可以使用影像圖庫呈現一系列的影像。您可以選擇讓影像自動循環顯示，或讓使用者手動翻看影像。

新增影像圖庫至頁面：

1. 瀏覽至您要編輯的頁面，並確定  已設為**編輯**。
2. 新增元件至頁面。圖庫元件會顯示預留位置影像，直到您選取您要使用的影像為止。
3. 若要將一或多個影像新增至圖庫，請按一下其功能表圖示 ，並選擇**設定值**，然後按一**一般**頁籤上的**影像**。
4. 按一下**新增影像**。



5. 選取一或多個影像。

#### 備註：

視窗會顯示所有可用的檔案。您必須選擇適用於相關資訊環境的檔案類型。例如，如果您要選擇影像檔，則必須選取有效的影像格式 (GIF、JPG、JPEG、PNG 或 SVG) 檔案。

- a. 找到並選取想要使用的影像。

若未見到任何數位資產，請按一下 ，然後將集合篩選條件變更為**全部**。

- b. 若從文件資料夾中選取影像，您可以連結到檔案，而不是將檔案複製到網站。若要連結至檔案，請選取**使用原始檔案參照，而不是將檔案複製到網站**。若未選取此選項，系統就會在網站儲存檔案的複本，然後從網站參照。若連結至原始的檔案，就不需要複製內容。即使檔案的權限變更或其他因素而限制了檢視，網站訪客仍能透過連結查看內容。
- c. 按一下**確定**。
- d. 如果選取數位資產，您可以選取特定的轉譯。若未選取轉譯，則會使用原始大小。若想要在網站發布時一同發布最新版本的資產，請選取**使用最新版本的資產**。

選取的影像會新增至影像清單中。拖放影像即可重新排列影像在清單中的順序。每個影像的預設標題為不含副檔名的檔案名稱。

6. 若要變更特定影像的標題、描述或其他選項，請按一下清單中的影像，然後進行變更。

您也可以將連結或其他動作與圖庫中的影像建立關聯：

- a. 在特定影像的「設定值」面板，請按一下「連結」欄位。
- b. 請選取下列其中一個選項：
  - **沒有連結**：使用者按一下影像時，影像不會執行任何動作。
  - **網頁**：指定外部頁面或網站的完整 URL，並選取要在何處開啟連結。

- **網站頁面**：使用頁面選擇器選取目前網站上的頁面，並選取要在何處開啟連結。您可以使用 `key1=value1&key2=value2` 格式指定其他 URL 參數。支援使用空的值；例如 `key1=&key2=value2`。您也可以指定 URL 錨點，但必須在目標網站頁面上要解析此錨點連結的位置新增特殊「錨點」區段版面配置，並在版面配置設定值中指定定義觸發程式動作時所使用的相同錨點名稱。

 **備註：**

需要使用 URL 錨點的「錨點」區段版面配置是由 OCE 工具程式提供。如需有關如何取得此工具程式的資訊，請參閱使用 OCE 工具程式進行開發。

- **檔案下載**：從儲存區域下載選取的檔案。選取檔案。  
如果選取數位資產，您可以選取特定的轉譯。若未選取轉譯，則會使用原始大小。若想要在網站發布時一同發布最新版本的資產，請選取**使用最新版本的資產**。若未選取**使用最新版本的資產**，則會使用最新的**已發布**版本，而不會使用更新的草稿版本 (如果有的話)。
  - **內容項目**：從關聯的資產儲存區域中選取一個內容項目，選擇您要顯示的詳細資訊頁面，並將此頁面定位在同一個視窗或新視窗中開啟。
  - **電子郵件**：指定有效的電子郵件地址，並視需要指定主旨。產生的訊息會在預設電子郵件從屬端中開啟，並透過預設電子郵件從屬端傳送。
  - **地圖**：輸入有效的地址或座標，並選取是要在桌面還是行動裝置瀏覽器中開啟地圖。
  - **電話**：輸入有效的電話號碼。
- c. 按一下**上一步**返回影像設定值面板。再按一下**上一步**返回影像清單，選取另一個要更新的影像。
  - d. 完成個別影像的更新之後，請按**上一步**指定圖庫選項。
7. 若要調整圖庫中影像的大小：
    - a. 使用**寬度**指定槽位中圖庫的寬度 (單位為像素)。按一下**填滿**以外的對齊方式選項以指定寬度。在您設定寬度之後，可以使用**填滿**將影像延長至指定的寬度。
    - b. 選取**縮放**選項調整圖庫中影像的呈現方式：
      - **剪裁**：將寬度或高度中尺寸較小的一邊調整到能剛好裝入可用空間，並剪裁尺寸較大的另一邊，以防止影像延伸。
      - **最適大小**：調整每個影像，使整個影像與可用空間的大小相符，而不會扭曲變形。
      - **延伸**：將兩個尺寸 (寬度或高度) 中較大尺寸調整到能剛好裝入可用空間，較小尺寸則延伸以填滿可用空間。
      - **無**：以完整的像素解析度使用影像，如果影像無法裝入可用空間，則會剪裁成統一的大小。
  8. 若要協助使用者在圖庫中移動：
    - a. 選取**瀏覽**方法：
      - **縮圖**：在圖庫下方依序顯示圖庫中影像的清單。使用者按一下縮圖影像即可前往圖庫中關聯的影像。
      - **索引子**：在圖庫下方顯示一系列按鈕 ，以代表圖庫中的每一個影像。使用者按一下按鈕即可前往圖庫中關聯的影像。

- **無**：不提供視覺瀏覽工具。使用者可以向右或向左滑動，以顯示相鄰的影像。此手動方法隨時都可使用。
  - b. 按一下**顯示上一個/下一個**即可在每一個影像包含箭號圖示，以前往圖庫中的上一個或下一個影像。
9. 您可以選取**顯示說明文字**方塊，選擇是要顯示還是隱藏影像的說明文字。
  10. 若要讓圖庫自動循環播放影像，請選取**自動播放**，然後指定**轉換時間**和**顯示時間**。使用者仍然可以使用您所設定的所有手動瀏覽功能。

## 圖庫格線

使用圖庫格線，就可以一次在資料列和資料欄中呈現一組影像。

新增影像圖庫至頁面：

1. 瀏覽至您要編輯的頁面，並確定  已設為**編輯**。
2. 新增元件至頁面。除非您選取想要使用的影像，否則圖庫格線元件會顯示預留位置影像。
3. 若要將一或多個影像新增至圖庫，請按一下其功能表圖示 ，並選擇**設定值**，然後按一下**一般**頁籤上的**影像**。
4. 按一下**新增影像**。
5. 選取一或多個影像。

### 備註：

視窗會顯示所有可用的檔案。您必須選擇適用於相關資訊環境的檔案類型。例如，如果您要選擇影像檔，則必須選取有效的影像格式 (GIF、JPG、JPEG、PNG 或 SVG) 檔案。

- a. 找到並選取想要使用的影像。
 

若未見到任何數位資產，請按一下 ，然後將集合篩選條件變更為**全部**。
- b. 若從文件資料夾中選取影像，您可以連結到檔案，而不是將檔案複製到網站。若要連結至檔案，請選取**使用原始檔案參照**，**而不是將檔案複製到網站**。若未選取此選項，系統就會在網站儲存檔案的複本，然後從網站參照。若連結至原始的檔案，就不需要複製內容。即使檔案的權限變更或其他因素而限制了檢視，網站訪客仍能透過連結查看內容。
- c. 按一下**確定**。
- d. 如果選取數位資產，您可以選取特定的轉譯。若未選取轉譯，則會使用原始大小。若想要在網站發布時一同發布最新版本的資產，請選取**使用最新版本的資產**。

選取的影像會新增至影像清單中。拖放影像即可重新排序清單 (和格線) 中的影像。每個影像的預設標題為不含副檔名的檔案名稱。

6. 若要變更特定影像的標題、描述或其他選項，請按一下清單中的影像，然後進行變更。

您也可以將連結或其他動作與圖庫中的影像建立關聯：

- a. 在特定影像的「設定值」面板，請按一下「連結」欄位。
- b. 請選取下列其中一個選項：
  - **沒有連結**：使用者按一下影像時，影像不會執行任何動作。
  - **網頁**：指定外部頁面或網站的完整 URL，並選取要在何處開啟連結。
  - **網站頁面**：使用頁面選擇器選取目前網站上的頁面，並選取要在何處開啟連結。您可以使用 `key1=value1&key2=value2` 格式指定其他 URL 參數。支援使用空的值；例如 `key1=&key2=value2`。您也可以指定 URL 錨點，但必須在目標網站頁面上要解析此錨點連結的位置新增特殊「錨點」區段版面配置，並在版面配置設定值中指定定義觸發程式動作時所使用的相同錨點名稱。

 **備註：**

需要使用 URL 錨點的「錨點」區段版面配置是由 OCE 工具程式提供。如需有關如何取得此工具程式的資訊，請參閱使用 OCE 工具程式進行開發。

- **檔案下載**：從儲存區域下載選取的檔案。選取檔案。

如果選取數位資產，您可以選取特定的轉譯。若未選取轉譯，則會使用原始大小。若想要在網站發布時一同發布最新版本的資產，請選取**使用最新版本的資產**。若未選取**使用最新版本的資產**，則會使用最新的已發布版本，而不會使用更新的草稿版本 (如果有的話)。
  - **內容項目**：從關聯的資產儲存區域中選取一個內容項目，選擇您要顯示的詳細資訊頁面，並將此頁面定位在同一個視窗或新視窗中開啟。
  - **電子郵件**：指定有效的電子郵件地址，並視需要指定主旨。產生的訊息會在預設電子郵件從屬端中開啟，並透過預設電子郵件從屬端傳送。
  - **影像預覽**：選取的影像將會顯示成頁面上的浮動層。
  - **地圖**：輸入有效的地址或座標，並選取是要在桌面還是行動裝置瀏覽器中開啟地圖。
  - **電話**：輸入有效的電話號碼。
- c. 按一下**上一步**返回影像設定值面板。再按一下**上一步**返回影像清單，選取另一個要更新的影像。
  - d. 完成個別影像的更新之後，請按**上一步**指定圖庫選項。
7. 使用**版面配置**選項排列格線中的影像。
- 瀑布流
  - 資料欄
  - 自訂

下面步驟將說明所有這些選項。

8. 選擇**瀑布流**版面配置，自動排列可用空間之資料列中的影像。產生的資料列具有統一高度，但沒有定義資料欄。



- a. 指定**高度**，按比例將所有影像調整為指定的高度 (像素)。
  - b. 指定**影像間距**，增加或減少資料列中影像之間的空格。
9. 選擇**資料欄**版面配置，排列資料列和資料欄中的影像。
- a. 選取**縮放**選項，調整影像在格線中的呈現方式：
    - **剪裁**：將寬度或高度中尺寸較小的一邊調整到能剛好裝入可用空間，並剪裁尺寸較大的另一邊，以防止影像延伸。
    - **最適大小**：調整每個影像，使整個影像與可用空間的大小相符，而不會扭曲變形。

例如，下列格線使用 4 個資料欄，並將 6 個影像調整為最適大小：



下面是相同的格線，但使用裁剪過的影像：



- b. 指定**外觀比例**，決定格線中儲存格的外觀。
    - **方形**：外觀比例 1:1。
    - **橫向**：外觀比例 16:9。
    - **縱向**：外觀比例 9:16。
    - **自訂**：指定您自己的外觀比例數值。
  - c. 指定**資料欄**數目。  
格線會自動調整，以建立等寬的資料欄。
  - d. 指定**影像間距**，增加或減少資料列和資料欄中影像之間的空格。
10. 選擇**自訂**版面配置，根據指定的影像大小和寬度來排列資料列和資料欄中的影像。
- a. 選取**縮放**選項，調整影像在格線中的呈現方式：
    - **剪裁**：將寬度或高度中尺寸較小的一邊調整到能剛好裝入可用空間，並剪裁尺寸較大的另一邊，以防止影像延伸。
    - **最適大小**：調整每個影像，使整個影像與可用空間的大小相符，而不會扭曲變形。
  - b. 指定**影像高度**和**影像寬度**，決定格線中儲存格的外觀。  
格線會自動調整，以建立具有所指定維度的儲存格。
  - c. 指定**影像間距**，增加或減少資料列和資料欄中影像之間的空格。
11. 指定**對齊方式**、**寬度**和**間距**選項，以放置槽位內的格線。  
使用**寬度**指定槽位中圖庫的寬度 (單位為像素)。按一下**填滿**以外的對齊方式選項以指定寬度。在您設定寬度之後，可以使用**填滿**將影像延長至指定的寬度。

## YouTube 視訊

在頁面中包含 YouTube 視訊串流，可增加動態內容和視覺吸引力。

若要使用不是由 YouTube 所代管的視訊，請參閱[視訊](#)。

新增 YouTube 視訊至頁面：

1. 瀏覽至您要編輯的頁面，並確定  已設為**編輯**。
2. 新增元件至頁面。在您選取想要使用的視訊之前，元件會顯示一個預留位置影像。
3. 若要指定要使用的 YouTube 視訊並調整其顯示特性，請按一下其功能表圖示 ，然後選擇**設定值**。
4. 尋找您想要使用的視訊，然後複製 URL 並貼到 **YouTube URL** 欄位中。
5. 指定任何顯示選項：
  - **顯示控制項**：啟用畫面和裝置選項以允許使用者手動控制視訊播放。
  - **顯示資訊**：暫時在視訊的左上角包含視訊描述。
  - **自動播放**：自動開始視訊播放。
  - **循環播放**：在視訊播放完畢後自動重複播放。
6. 使用已為視訊定義的**外觀比例 (自動)** 或選擇另一個外觀比例，以決定視訊顯示外觀。

7. 指定任何對齊方式或間距選項來放置視訊。

## 視訊

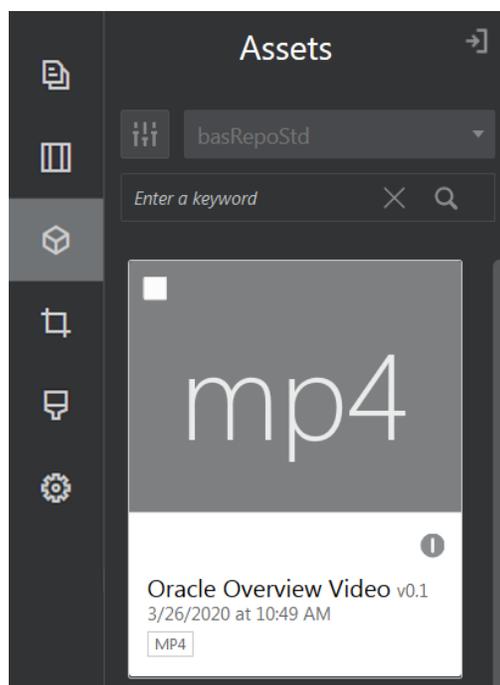
在頁面中包含來自 Oracle Content Management 的視訊，可以增加動態內容和視覺吸引力。

Oracle Content Management 提供數個可在頁面中新增視訊的選項。您可以內嵌 YouTube 網站上代管的 YouTube 視訊，或者您可以新增由 Oracle Content Management 儲存、管理以及傳遞的視訊，充分運用 Video Plus 提供的自動轉碼和最佳化串流處理。當網站在多個不同功能的裝置上提供服務時，這會非常有用。如果您的系統管理員尚未啟用 Video Plus，您可以選擇標準視訊。

若要使用由 YouTube 所代管的視訊，請參閱 YouTube 視訊。

使用儲存在 Oracle Content Management 文件或資產儲存區域中的視訊：

1. 瀏覽至您要編輯的頁面，並確定「編輯」開關 (  ) 已設為編輯。
  - a. 若要使用儲存為數位資產的視訊，請開啟資產面板，並使用它來搜尋您要使用的視訊，然後選取視訊並拖曳至頁面上。



- b. 或者，將「視訊」元件從元件清單中新增至頁面。在您選取想要使用的視訊之前，元件會顯示一個預留位置影像。
2. 如果您已從「資產」面板中拖曳資產，或從元件清單中拖曳視訊元件，則請使用功能表圖示 (  ) 調整視訊顯示特性，或瀏覽至視訊元件的視訊。按一下功能表，然後選擇設定值。
  3. 若要選擇視訊元件的視訊，請按一下選取以開啟文件和數位資產頁籤，然後瀏覽至您要從 Oracle Content Management 新增的視訊。選取此項目並按一下確定。
    - a. 若已啟用 Video Plus，則您可以選取不同的儲存區域、搜尋通道、集合與關鍵字，以及篩選資產來協助您在瀏覽至視訊時縮小搜尋範圍。

- b. 如果使用標準視訊，請選擇 MP4 類型的視訊檔案。所有支援的瀏覽器都能使用 MP4 視訊格式。此外，某些標準視訊包含預先定義的預覽影像。這些視訊不會顯示空白畫面作為預覽影像。若想要使用圖形來取代空白畫面，請在視訊設定值對話方塊中按一下**選取**，然後從 Oracle Content Management 中選擇預覽影像。
4. 從資產面板中拖曳視訊或從設定值對話方塊中選取視訊來選取視訊之後，請使用設定值對話方塊來指定任何顯示選項：
  - **顯示控制項**：啟用畫面和裝置選項以允許使用者手動控制視訊播放。
  - **自動播放**：自動開始視訊播放。
  - **循環播放**：視訊播放完畢後再自動重複播放。
  - **靜音**：頁面載入時自動將視訊設為靜音。
5. 指定任意的寬度、對齊方式或間距選項，以決定視訊的位置。
6. 按一下**樣式**以選擇或定義視訊框線的自訂樣式。

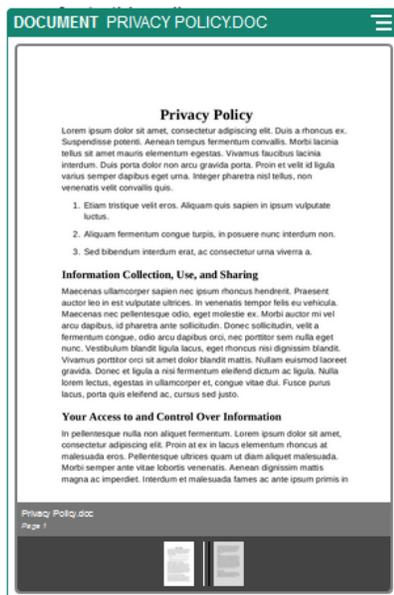
## 文件元件

讓我們來瞭解一些可讓您存取及顯示檔案與資料夾的元件。

- [文件](#)
- [資料夾清單](#)
- [檔案清單](#)
- [文件管理程式](#)
- [專案程式庫](#)

## 文件

您可以直接從頁面檢視多頁文件和投影片簡報。



新增文件至頁面：

1. 瀏覽至您要編輯的頁面，並確定  已設為**編輯**。
2. 新增元件至頁面。文件元件會顯示預留位置影像，直到您指定要顯示的文件為止。
3. 若要選取文件，請按一下其功能表圖示 ，選擇**設定值**，然後按一下「文件」欄位旁的**選取**。

文件必須儲存在網站儲存區域中，或者您可以存取的其他儲存區域。您也可以使用與您共用的文件，或是您從本機或網路檔案位置上傳的文件。

4. 選取文件，然後按一下**確定**。

#### 備註：

視窗會顯示所有可用的檔案。您必須選擇適用於相關資訊環境的檔案類型。例如，如果您選擇文件檔案，您就必須選取有效文件格式的檔案，例如 TXT 或 DOC。

5. 在「設定值」面板中新增標題，或是修改間距、對齊方式、樣式以及其他呈現選項。
6. 協助使用者瀏覽文件：
  - a. 按一下**顯示頁碼**可在每頁下方顯示頁碼。
  - b. 選取**瀏覽方法**：
    - **縮圖**：在文件下方依序顯示文件中的頁面清單。使用者只要按一下某個縮圖影像即可前往關聯的頁面。
    - **索引子**：在文件下方顯示一系列按鈕 ，代表文件中的每個頁面。使用者只要按一下某個按鈕即可前往關聯的頁面。
    - **無**：不提供視覺瀏覽工具。使用者可以向右或向左滑動，以顯示相鄰的頁面。此手動方法隨時都可使用。
  - c. 按一下**顯示上一個/下一個**以在每一頁包含箭號圖示，以前往文件的上一頁或下一頁。

## 資料夾清單

您可以使用資料夾清單列出 Oracle Content Management 帳戶中某個指定資料夾內的資料夾。

若您使用此元件搭配頁面中的一或多個檔案清單或文件管理程式元件，這些元件會自動顯示資料夾清單中所選資料夾的內容。

將資料夾清單元件新增至頁面：

1. 瀏覽至您要編輯的頁面，並確定  已設為**編輯**。
2. 新增元件至頁面。
3. 若要編輯元件及其外觀，請按一下其功能表圖示 ，然後選擇**設定值**。

4. 選擇**自訂設定值**，以便設定顯示之內容的預設詳細資訊。
5. 按一下**選擇資料夾**旁的**選取**以變更顯示的資料夾。您必須選取文件儲存區域首頁以外的資料夾。完成之後請按一下**返回**。

 **備註：**

資料夾清單的所有使用者都具備下載者存取權。使用者不管具備什麼角色，都能夠檢視及下載檔案。如果網站訪客的權限高於針對元件指定的權限，訪客的個別權限將會覆寫針對元件設定的權限。

6. 請從下列選項中選擇，設定顯示內容的其他預設值。
  - **選擇預設選項**：若您已選取 **Oracle Documents 資料夾**，請選取要在清單中顯示為已選取的資料夾 (若有的話)。
  - **顯示資料夾名稱標頭**：選取此選項即可在內嵌元件的標題中顯示資料夾名稱。
  - **資料夾排序**：選擇項目最初的顯示方式，可依名稱的字母順序或依項目上次更新時間排序。
  - **顯示子資料夾**：使用滑動軸可限制顯示的子資料夾數目。
7. 在「一般」頁籤中修改間距、對齊方式以及其他呈現選項。
8. 在「樣式」頁籤中使用預先定義樣式或您自訂的選項，設定包含元件之框架的格式。

## 檔案清單

您可以使用檔案清單提供 Oracle Content Management 帳戶中指定資料夾內的檔案檢視。

若您將此元件與頁面中的一或多個資料夾清單元件搭配使用，檔案清單元件便會自動顯示資料夾清單中所選資料夾的內容。在使用者按一下清單中的檔案時，您也可以設定元件以執行一或多個動作。例如，您可以在個別的瀏覽器視窗、「燈箱」浮動層，或者頁面上的「文件管理程式」元件中，預覽選取的檔案。

將檔案清單元件新增至頁面：

1. 瀏覽至您要編輯的頁面，並確定  已設為**編輯**。
2. 將元件群組新增至頁面。
3. 若要編輯元件及其外觀，請按一下其功能表圖示 ，然後選擇**設定值**。
4. 選擇**自訂設定值**，以便設定顯示之內容的預設詳細資訊。
5. 按一下**選擇資料夾**旁的**選取**以變更顯示的資料夾。您必須選取文件首頁以外的資料夾。完成之後請按一下**返回**。

 **備註：**

檔案清單的所有使用者都具備下載者存取權。使用者不管具備什麼角色，都能夠檢視及下載檔案。如果網站訪客的權限高於針對元件指定的權限，訪客的個別權限將會覆寫針對元件設定的權限。

6. 請從下列選項中選擇，設定顯示內容的其他預設值。

- **顯示選項**：選擇將隨列出之檔案顯示的詳細資訊。
    - **資料夾名稱標頭**：在標題中顯示資料夾名稱。
    - **檔案描述**：顯示檔案描述 (若有的話)。
    - **檔案區隔符號**：每個檔案以一行區隔。
    - **下載圖示**：會包含一個下載圖示，供使用者在需要時下載檔案。
    - **上次更新時間**：顯示上次更新檔案的日期。
    - **檔案大小**：顯示檔案的大小。
    - **影像**：顯示檔案內容的縮圖影像。
  - **觸發程式和動作**：選擇當「資料夾清單」元件中的資料夾內容是已選取狀態時，是否要自動重新整理檔案清單。當已選取某個檔案時，您也可以選擇啟用「已選取檔案」觸發程式。使用「連結」頁籤以設定動作與**已選取檔案**觸發程式的關聯。
  - **檔案排序**：選擇項目最初的顯示方式，可依名稱的字母順序或依項目上次更新時間排序。
  - **顯示檔案**：使用滑桿列選擇是否顯示所有檔案，或是要限制顯示的檔案數目。
7. 在「一般」頁籤中修改間距、對齊方式以及其他呈現選項。
  8. 在「樣式」頁籤中使用預先定義樣式或您自訂的選項，設定包含元件之框架的格式。
  9. 使用「連結」頁籤以設定動作與**已選取檔案**觸發程式的關聯。例如，若要設定使用檔案清單元件以在燈箱浮動層中預覽選取的檔案：
    - a. 確定在「檔案清單設定值」視窗中選取**檔案選取後啟用觸發程式**。
    - b. 在檔案清單設定值的「連結」頁籤中，按一下**已選取檔案**觸發程式。
    - c. 按一下**燈箱預覽**頁面動作，並將其拖曳至動作清單。
    - d. 在檔案 ID 或 URL 欄位中，選擇**檔案連結**。

**檔案連結**使用具下載者權限的參照連結，因此所有的網站訪客都能預覽及視需要下載該檔案。如果您使用**檔案 ID**，就會使用成員連結。成員連結就只能讓註冊的使用者預覽及視需要下載選取的檔案。

若使用者按一下檔案清單中的檔案，會在變暗且非作用中版本的頁面上開啟檔案預覽。

## 文件管理程式

您可以使用文件管理程式提供您的 Oracle Content Management 首頁或檔案檢視。

在頁面新增文件管理程式元件：

1. 瀏覽至您要編輯的頁面，並確定  已設為**編輯**。
2. 新增元件至頁面。
3. 若要編輯元件及其外觀，請按一下其功能表圖示 ，然後選擇**設定值**。
4. 選擇**自訂設定值**，以便設定顯示之內容的預設詳細資訊。
5. 按一下**選擇資料夾**旁的**選取**以變更顯示的資料夾。

若要選取資料夾，請勾選資料夾名稱旁的方塊。若要開啟資料夾，請按一下資料夾名稱。按一下路徑中的資料夾名稱即可返回該資料夾，或是按一下**瀏覽首頁**即可返回首頁資料夾。完成之後請按一下**返回**。

6. 按一下**選取資料夾存取權**，然後選擇要授予訪客的存取角色。

訪客檢視及使用資料夾內容的能力，將依據其角色和在下一步啟用的**瀏覽選項**而定。

- **成員存取權**：訪客將能夠使用資料夾成員可用的所有功能，例如檢視對話、附註或項目的自訂特性。
- **檢視者**：檢視者可以查看檔案和資料夾，但不能變更內容。
- **下載者**：下載者可同時下載檔案，並且將下載的檔案儲存到自己的電腦。
- **提供者**：提供者還可以修改檔案、更新檔案、上傳新檔案以及刪除檔案。

設定資料夾存取權時，請考量下列事項：

- 網站作者不能授予超出本身所具備之存取權的資料夾存取權。例如，若作者具備資料夾的下載者存取權，他就不能授予網站訪客提供者權限。
- 對元件中資料夾所設定的權限可以提升訪客的權限。例如，若訪客具備資料夾的檢視者權限 (或無權限)，元件可以根據選取的角色授予更高的權限。這些提高的權限只對元件本身有效。
- 如果網站訪客的權限高於針對元件指定的權限，訪客的個別權限將會覆寫針對元件設定的權限。
- 資料夾的權限會套用至該資料夾內的資料夾與檔案。

7. 請從下列選項中選擇，設定顯示內容的其他預設值。

- **版面配置**：選取資料夾和檔案的最初格線、清單或精簡清單版面配置。使用者可以在檢視完成的內嵌元件時變更此版面配置。
- **色彩配置**：為內嵌資料夾清單選擇其中一個可用的色彩配置。
- **排序順序**：選擇項目最初的顯示方式 (依名稱的字母順序或依項目的上次更新時間)。
- **瀏覽選項**：選擇將可供使用者在選取項目時使用的選項。例如，您可以選擇讓使用者能夠檢視檔案、下載檔案、共用或複製檔案和資料夾，或是刪除檔案。如果您想要限制使用者能夠對您的檔案和資料夾執行的動作，請取消選取此清單中的某個選項。如果您在上一個步驟中選擇「成員存取權」，您可以選擇顯示側邊窗格，其中將顯示所有對話、附註或自訂特性。
- **檢視器選項**：選擇使用者檢視檔案的方式。您可以在內嵌框架中或在另一個頁籤 (或視窗，依瀏覽器設定值而定) 中檢視檔案。您也可以隱藏或顯示縮圖，以及自訂視訊的檢視方式。
- **顯示縮放控制項**：選擇是否在資料夾的內嵌檢視中顯示滑桿列或縮放控制項。
- **檢視器最適大小模式**：您可以選擇以填滿頁面、填滿頁面寬度或以原始大小的方式顯示檔案。
- **觸發程式和動作**：選擇當檔案清單也與資料夾清單元件搭配使用時，是否要重新整理檔案清單。

8. 在「一般」頁籤中修改間距、對齊方式以及其他呈現選項。

9. 在「樣式」頁籤中使用預先定義樣式或您自訂的選項，設定包含元件之框架的格式。

新增文件管理程式之後，檢視者將會在您的網站的框架中見到內嵌的所選資料夾檢視。使用者可以使用提供的顯示選項，變更資料夾和檔案的列出方式。若使用者選取某個檔案或資料夾，他們可選擇功能表列提供的選項，或使用滑鼠右鍵功能表，依據其角色和您指定的**瀏覽選項**執行任何允許的動作。

如果對話與某個項目關聯，您將會在該項目看到 。按一下該圖示即可開啟對話窗格，供您檢視附註和註解。獨立對話 (沒有關聯資料夾的項目) 必須使用對話元件新增。

項目若含有自訂特性，您可以在窗格中檢視特性。按一下 **其他**，然後選取 **自訂特性**，即可開啟特性窗格。

## 專案程式庫

您可以使用專案程式庫將不同位置的資料夾組合在儲存區域中，完全不需要變更原始資料夾或其位置。您也可以使用專案程式庫元件為每個資料夾指定不同的權限層級，以因應不同的專案團隊角色。

例如，若將其中一個訪客角色 (檢視者、下載者或提供者) 指定給專案程式庫中的一個資料夾，訪客就能使用該角色的權限來檢視資料夾內容並與其互動。如果您指定成員存取權，就只有成員能夠在專案程式庫中見到該資料夾。成員使用各自的原始資料夾權限與資料夾內容互動。

如果您使用此元件搭配頁面中的一或多個檔案清單或文件管理程式元件，這些元件會自動顯示專案程式庫中所選資料夾的內容。

在頁面新增專案程式庫元件：

1. 瀏覽至您要編輯的頁面，並確定  已設為 **編輯**。
2. 新增元件至頁面。
3. 若要編輯元件及其外觀，請按一下其功能表圖示 ，然後選擇 **設定值**。
4. 選擇 **自訂設定值** 以選取對話並選取顯示選項。
5. 指定一個標題。預設標題為「專案程式庫」。您可以選取或取消選取下方的 **標題** 顯示選項，以選擇是要顯示還是隱藏標題。
6. 選取一個 **色彩配置**。

### 備註：

如果您使用觸發程式和動作來建立專案程式庫與文件管理程式元件的關聯，您為專案程式庫選取的色彩配置也會套用到要在文件管理程式元件中顯示的資料夾。若您建立的是專案程式庫與資料夾清單或檔案清單元件的關聯，就不適用以上的情況，因為這些元件不支援色彩配置。

7. 按一下資料夾清單旁的 **新增**，以新增現有資料夾或建立新資料夾。
  - a. 瀏覽並選取一或多個資料夾，或者按一下 **建立** 來建立資料夾。您在此視窗中選取的每個資料夾都會新增至專案程式庫。

您可以使用工具列中的所有選項，包括檢視和排序選項。
  - b. 完成之後請按一下 **返回**。
8. 若要組織清單，請選取一個資料夾並按一下 **上移** 或 **下移** 以變更其在清單中的順序，或者按一下 **移除** 以將資料夾自清單中移除。
9. 若要根據使用者的角色指定某個資料夾的存取權，請從資料夾清單中選取該資料夾，按一下 **選取資料夾權限**，然後選擇存取角色。
  - **成員存取權**：唯有具備資料夾權限的已註冊使用者才會在專案程式庫中見到該資料夾。成員使用各自的原始資料夾權限與資料夾內容互動。

- **檢視者**：檢視者可以查看檔案和資料夾，但不能變更內容。
- **下載者**：下載者可同時下載檔案，並且將下載的檔案儲存到自己的電腦。
- **提供者**：提供者還可以修改檔案、更新檔案、上傳新檔案以及刪除檔案。

設定資料夾存取權時，請考量下列事項：

- 網站作者不能授予超出本身所具備之存取權的資料夾存取權。例如，若作者具備資料夾的下載者存取權，他就不能授予網站訪客提供者權限。
- 對元件中資料夾所設定的權限可以提升訪客的權限。例如，若訪客具備資料夾的檢視者權限 (或無權限)，元件可以根據選取的角色授予更高的權限。這些提高的權限只對元件本身有效。
- 如果網站訪客的權限高於針對元件指定的權限，訪客的個別權限將會覆寫針對元件設定的權限。
- 資料夾的權限會套用至該資料夾內的資料夾與檔案。

10. 對於具有**成員存取權**權限的資料夾，您可以按一下**在文件管理程式中顯示對話窗格**，以顯示與資料夾或內容關聯的對話。

如果您選取此選項，並且在頁面中設定文件管理程式元件以顯示選取的資料夾，使用者就可以按一下對話圖示以顯示與該資料夾關聯的所有對話。如果您未選取此選項，就不會顯示對話圖示。

11. 當您完成之後，請關閉視窗。
12. 在「一般」頁籤中修改間距、對齊方式以及其他呈現選項。
13. 在「樣式」頁籤中使用預先定義樣式或您自訂的選項，設定包含元件之框架的格式。
14. 在「連結」頁籤中設定動作與**選取的資料夾**觸發程式的關聯。例如，如果您也在頁面上增加文件管理程式元件，您可以使用文件管理程式元件來顯示專案資料夾中所選資料夾的內容：
  - a. 在專案程式庫設定值的「連結」頁籤中，按一下**選取的資料夾**觸發程式。
  - b. 在「設定觸發程式動作」視窗中，按一下「文件管理程式」。
  - c. 按一下並拖曳**顯示文件**動作到動作清單。
  - d. 在「資料夾 ID」或「URL 欄位」中，選擇**選取的資料夾**。

當使用者在專案程式庫中按一下某個資料夾時，該資料夾內容就會顯示在頁面的文件管理程式元件中。

## 社群元件

社群元件可協助您的使用者保持聯繫與溝通。

- [社群列](#)
- [Facebook 按讚與推薦](#)
- [Twitter 分享與跟隨](#)
- [對話元件](#)
- [對話清單](#)

## 社群列

輕鬆地將圖示與連結新增至熱門的社群媒體，例如 Facebook 與 Twitter。社群列預設包含一些目標，但您可以新增與移除社群列中的項目。



若要新增與修改社群列：

1. 瀏覽至您要編輯的頁面，並確定  已設為**編輯**。
2. 新增元件至頁面。社群列會顯示預設包含的圖示。
3. 若要新增或變更社群列中的圖示，請按一下其功能表圖示 ，選擇**設定**，然後按一下**一般**頁籤頂端的**圖示**。  
若要移除社群圖示，請按一下名稱旁邊的 X。要新增的圖示必須儲存在網站儲存區域中，或是另一個您可以存取的儲存區域中。您也可以使用與您共用的影像，或是您從本機或網路檔案位置上傳的影像。
4. 從本機或網路位置上傳一或多個圖示：
  - a. 按一下面板頂端的**新增圖示**。
  - b. 瀏覽至儲存區域中您要儲存影像的位置，或按一下  在目前位置中新增新的資料夾。
  - c. 按一下 。
  - d. 找到並選取一或多個影像檔，然後按一下**開啟**。  
影像檔便會上傳至儲存區域的目前位置。
5. 選取儲存區域中的一或多個影像，然後按一下**確定**。
  - a. 找到並按一下想要使用的一或多個影像。
  - b. 若要連結至儲存區域中的檔案，請選取**使用原始檔案參照，而不是將檔案複製到網站**。如果您未選取此選項，就會在網站儲存檔案的複本，然後從網站參照。若連結至原始的檔案，就不需要複製內容。即使檔案的權限變更或其他因素而限制了檢視，網站訪客仍能透過連結查看內容。
  - c. 按一下**確定**。  
選取的影像會新增至影像清單中。拖放影像即可重新排列影像在清單中的順序。每個影像的預設標題為不含副檔名的檔案名稱。
6. 若要變更特定影像的目標 URL、標題、描述或其他選項，請按一下清單中的影像，然後進行變更。  
更新完個別圖示後，請按一下**上一頁**以指定一般選項。
7. 指定社群列中所有圖示的大小、間距、方向及對齊方式。

## Facebook 按讚與推薦

您可以在頁面中新增一個「Facebook 按讚」按鈕，讓檢視者可以在 Facebook 上輕鬆給您的網站按讚。

1. 瀏覽至您要編輯的頁面，並確定  已設為**編輯**。
2. 新增元件至頁面。
3. 若要編輯應用程式與其外觀，請按一下其功能表圖示 ，然後選擇**設定值**。
4. 選擇 **Facebook 按讚設定值**或 **Facebook 推薦設定值**以設定 URL，同時選擇是否新增一個**共用**按鈕，以便能夠輕鬆地在 Facebook 頁面中張貼您網站的連結。
5. 使用「一般」頁籤修改間距、對齊方式以及其他呈現方式選項。
6. 在「樣式」頁籤中使用預先定義樣式或您自訂的選項，設定包含應用程式之框架的格式。

## Twitter 分享與跟隨

您可以在頁面中增加一個「Twitter 分享」按鈕，讓檢視者可以快速地在 Twitter 帳戶上分享連結。

1. 瀏覽至您要編輯的頁面，並確定  已設為**編輯**。
2. 新增元件至頁面。
3. 若要編輯應用程式與其外觀，請按一下其功能表圖示 ，然後選擇**設定值**。
4. 選擇 **Twitter 跟隨設定值**以設定 Twitter 使用者名稱，並且選擇是否要顯示使用者名稱，以及選擇**跟隨**按鈕的大小。選擇 **Twitter 分享設定值**以設定下列預設值。使用者可以在使用此按鈕時變更這些值。
  - **分享 URL**：使用者可以分享之網站的 URL。
  - **推文**：與頁面有關的推文。
  - **透過 @**：推文時所使用的 Twitter 使用者帳戶。
  - **推薦 @**：用於 Twitter 推薦的 Twitter 使用者帳戶。
  - **主題標籤 #**：想要使用於張貼內容的主題標籤。
  - **次數**：顯示分享次數 (垂直或水平)。
  - **大型按鈕**：為應用程式選擇大型或小型按鈕。
5. 在「一般」頁籤中修改間距、對齊方式以及其他呈現選項。
6. 在「樣式」頁籤中使用預先定義樣式或您自訂的選項，設定包含應用程式之框架的格式。

## 對話元件

您可以直接從您的網站使用對話，開始某個主題的討論。



### 備註：

若要讓對話能夠在網站上運作，網站必須是只有指定的使用者或具備「Oracle Content Management 使用者」角色的使用者能夠存取的安全網站。請參閱[變更網站安全性](#)。

新增對話元件至頁面：

1. 瀏覽至您要編輯的頁面，並確定  已設為**編輯**。
2. 新增元件至頁面。
3. 若要編輯元件及其外觀，請按一下其功能表圖示 ，然後選擇**設定值**。
4. 選擇**自訂設定值**，以便選取對話和顯示的色彩配置。
5. 按一下**選擇對話**旁的**選取**以選取現有對話，或者建立新對話。
  - a. 從可用對話清單中選取一個對話，或按一下**建立**來建立新對話，並加以命名。

清單中包含您可以存取的所有對話。  
您可以篩選此清單：

    - **全部**：顯示標示為靜音以外的所有對話。
    - **我的最愛**：顯示標示為我的最愛的對話。
    - **靜音**：顯示已設為靜音的對話，這些對話不會顯示在對話清單中。
    - **已關閉**：顯示標示為已關閉的所有對話。

您也可以排序此清單：

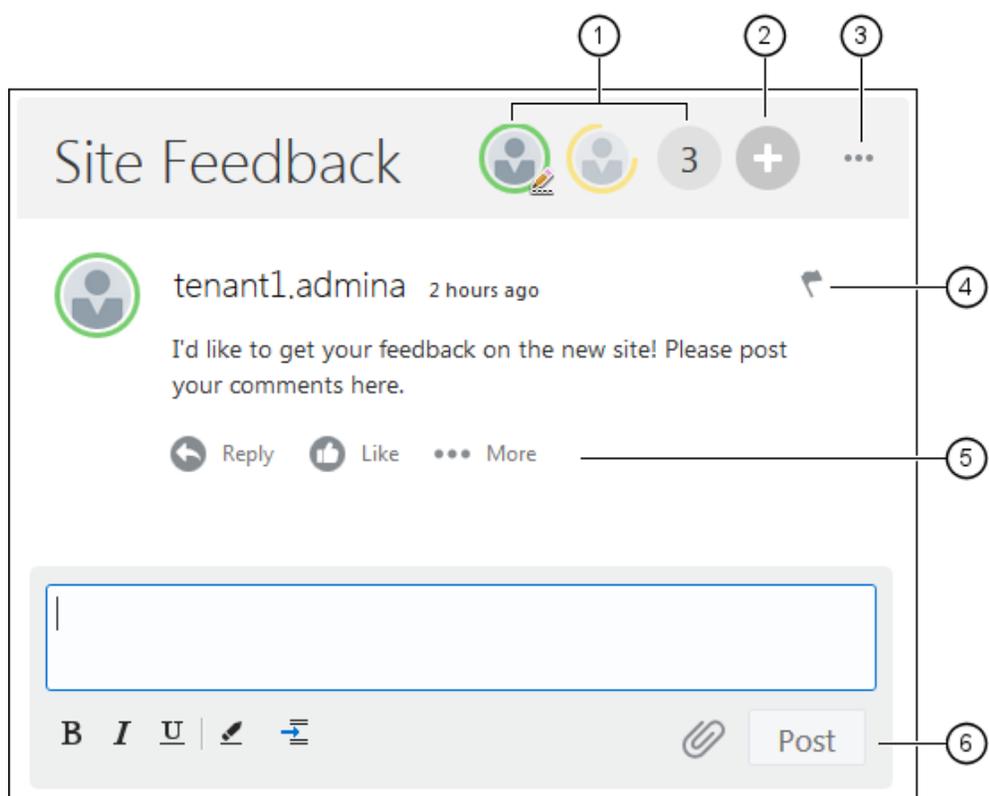
    - **上次更新時間**：依最近更新時間到最早更新時間順序列出對話。
    - **名稱**：依最低到最高的文數字順序列出對話。
    - **未讀取**：先列出未讀取的對話，且依未讀取註解數目最多到最少的順序排列。
  - b. 完成時，請按一下**返回**。
  - c. 如果選取獨立對話 (不是以檔案或資料夾為基礎的對話)，就可以針對檢視對話的人員設定存取權限。
  - d. 選取一個**色彩配置**。
  - e. 若要將此對話元件與頁面中的對話清單元件搭配使用，請選取**根據對話清單元件中的選擇項目自動重新整理對話**。

如果使用者從清單中選取某個對話，此對話就會顯示在對話元件中。
  - f. 當您完成之後，請關閉視窗。
6. 在「一般」頁籤中修改間距、對齊方式以及其他呈現選項。

7. 使用「樣式」頁籤，可以透過預先定義的樣式或您自訂的選項來格式化包含元件的框架。在已發布的網站上，訪客將在您網站的框架中見到對話。使用者可以瀏覽對話內容，以讀取及回覆註解。

 **備註：**

訪客若還不是對話的成員，他們仍然可以讀取及回覆註解，但是不會見到下圖中 1 到 4 的項目。



- **功能表列 (1)** 含有對話中個別參與者的資訊。按一下使用者圖示即可取得狀態資訊，以及查看可與使用者互動的選項。數字圖示代表參與者總數。按一下此圖示即可查看完整清單。
- 按一下**新增使用者 (2)** 即可新增使用者至對話。
- 按一下**其他選項 (3)** 即可查看可使用的對話選項清單。此選項清單會依使用者角色的不同而有所不同。若為與資料夾和檔案關聯的對話，此選項清單也會依據資料夾或檔案的權限而定。
- 按一下**旗標 (4)** 即可將通知旗標指定給特定使用者，以對其發出警示。使用者就會收到電子郵件通知 (使用者偏好設定中指定的選項)。
- 使用**註解選項 (5)** 對「對話」中的特定註解執行動作，例如回覆註解、對註解按讚、編輯或刪除註解。

- 如果您新增或編輯註解，請使用**編輯選項 (6)** 來新增或移除基本格式設定 (例如粗體或加上底線)、新增附件，以及張貼註解至對話。網站訪客若不是對話成員，就不能夠新增附件。

## 對話清單

您可以直接從您的網站使用對話，開始某個主題的討論。

若您將此元件與頁面中的一或多個對話清單元件搭配使用，對話清單元件便會自動顯示對話清單中所選對話的內容。

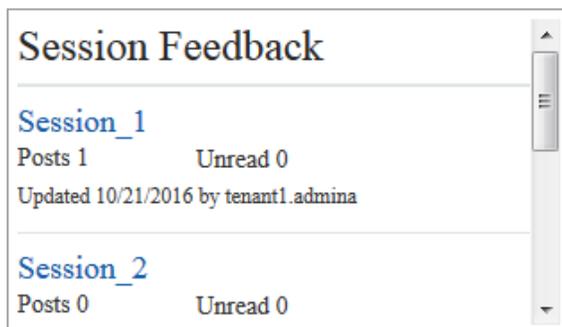
### 備註：

若要讓對話或對話清單能夠在網站上運作，網站必須是只有指定的使用者或具備「Oracle Content Management 使用者」角色的使用者能夠存取的安全網站。請參閱[變更網站安全性](#)。

新增對話清單元件至頁面：

1. 瀏覽至您要編輯的頁面，並確定  已設為**編輯**。
2. 新增元件至頁面。
3. 若要編輯元件及其外觀，請按一下其功能表圖示 ，然後選擇**設定值**。
4. 選擇**自訂設定值**以選取對話並選取顯示選項。
5. 指定一個標題。預設標題為「對話清單」。您可以選取或取消選取下方的**標題**顯示選項，以選擇是要顯示還是隱藏標題。
6. 按一下對話清單旁的**新增**，即可新增現有對話或建立新對話。
  - a. 從可用對話清單中選取一或多個對話，或按一下**建立**來建立新對話並加以命名。
  - b. 完成時，請按一下**返回**。
  - c. 選擇檢視對話清單的使用者能夠具備的權限。
7. 若要組織對話清單，請選取對話並按一下**上移**或**下移**以變更其在清單中的順序，或按一下**移除**以從清單中移除對話。
8. 選取一個**色彩配置**，接著選擇其他**顯示選項**。您選取的顯示選項會顯示在清單中的對話名稱下方。
9. 當您完成之後，請關閉視窗。
10. 使用「一般」頁籤修改間距、對齊方式以及其他呈現方式選項。
11. 使用「樣式」頁籤，可以透過預先定義的樣式或您自訂的選項來格式化包含元件的框架。

下列影像顯示標題為 *Session Feedback* 且已選取所有顯示選項的對話清單。



## 處理作業元件

處理作業元件可讓網站使用者起始及管理預先定義之處理作業的工作。

### 備註：

若要使用處理作業元件，您必須使用 Oracle Process Cloud Service 版本 17.1.3 或更新版本。

- [處理作業開始表單](#)
- [處理作業工作清單](#)
- [工作詳細資訊表單](#)

## 處理作業開始表單

您可以使用處理作業開始表單來起始在 Oracle Process Cloud Service 定義的處理作業。

在網站上使用處理作業開始表單的先決條件如下：

- 您必須先在 Oracle Process Cloud Service 中定義關聯的處理作業和處理作業開始表單，才能夠使用此元件顯示它們。請參閱 *Using Processes in Oracle Integration* 中的 *Developing Structured Processes*。
- 必須將與包含開始表單之處理作業泳道關聯的角色指定給使用者，使用者才能使用處理作業開始表單。
- 管理員必須設定 Oracle Process Cloud Service 與 Oracle Content Management 之間的整合。請參閱 *Administering Oracle Content Management* 中的與 Oracle Process Cloud Service 整合。整合這兩項服務時需要 SSO 登入，所以這兩項服務必須位於相同的識別網域內。

將處理作業開始表單元件新增至頁面：

1. 瀏覽至您要編輯的頁面，並確定  已設為**編輯**。
2. 新增元件至頁面。
3. 若要編輯元件及其外觀，請按一下其功能表圖示 ，然後選擇**設定值**。

4. 選擇**自訂設定值**，以便選取處理作業開始表單及設定表單預設值。
  - a. 網站若將會是公用網站，請選取代理主機服務。
  - b. 選取開始表單的分割區。您可以使用「測試」分割區驗證處理作業是否依照計畫運作，或使用「生產」分割區部署處理作業以用於一般用途。
  - c. 若要一律使用選取的預設處理作業版本，請選取**使用預設處理作業版本**。若未選取此選項，就必須選取一個特定版本，而當處理作業更新之後，您將需要更新這些設定值的選項。
  - d. 選取一個處理作業。
  - e. 選取開始表單。表單的語法如下：`process type:version:processname:start`。例如，`Basic Approval:1.0:Process:Start Basic Approval`。

若視窗上方的訊息顯示「沒有 Process Cloud Service 連線」，表示 Oracle Process Cloud Service 與 Oracle Content Management 之間未設定整合。請諮詢您的管理員。

處理作業作者必須將您新增為處理作業的起始者，您才能在清單中看見處理作業。處理作業作者必須將所有網站訪客新增為處理作業的起始者，否則訪客將只能填寫表單，但無法起始處理作業。
  - f. 從下列選項中選擇，以設定表單的其他預設值。
    - **表單標題**：(選擇性) 將預設的表單標題取代為自己的標題。
    - **「送出」按鈕名稱**：(選擇性) 以您指定的值重新命名**送出**按鈕。
    - **顯示「送出」按鈕**：(選擇性) 顯示或隱藏元件的**送出**按鈕 (預設為顯示此按鈕)。如果表單本身已提供類似的按鈕，您可以選擇將元件的**送出**按鈕隱藏起來。
    - **送出確認**：(選擇性) 以您自己的訊息取代預設確認訊息。
    - **顯示送出確認**：(選擇性) 按一下**送出**按鈕時，顯示或隱藏確認訊息。預設會顯示此訊息，而且顯示在處理作業開始表單元件內。
    - **顯示「儲存」按鈕**：(選擇性) 顯示或隱藏元件的**儲存**按鈕 (預設為隱藏此按鈕)。如果關聯的處理作業允許您儲存工作並於稍後返回工作，您可以選擇顯示元件的**儲存**按鈕。
    - **顯示「捨棄」按鈕**：(選擇性) 顯示或隱藏元件的**捨棄**按鈕 (預設為隱藏此按鈕)。如果您想要讓使用者能夠捨棄表單內容並從頭開始，您可以選擇顯示**捨棄**按鈕。
    - **顯示附件**：(選擇性) 顯示或隱藏表單中可上傳一或多個檔案的**附件**區域 (預設為隱藏此區域)。上傳的檔案會儲存為 Oracle Process Cloud Service 中處理作業的一部分。
    - **自訂預設值**：您可以設定要在開始表單中預先填入的名稱與值組。按一下**新增欄位**，然後輸入欄位名稱和值。名稱是其中一個表單欄位，並非表單中所使用的標籤，值則是該欄位所允許的內容。若要插入目前登入的使用者資料，請使用 `%%username%%` 和 `%%userid%%` 特殊值。
5. 在「一般」頁籤中修改間距、對齊方式以及其他呈現選項。
6. 在「樣式」頁籤中使用預先定義樣式或您自訂的選項，設定包含元件之框架的格式。
7. 使用「連結」頁籤指定動作給元件提供的觸發程式：

- **開始表單已送出**：使用者按一下**送出**按鈕時，會出現此觸發程式。
- **開始表單已儲存**：使用者按一下**儲存**按鈕時，會出現此觸發程式。
- **開始表單已捨棄**：使用者按一下**捨棄**按鈕時，會出現此觸發程式。

按一下觸發程式即可指定動作。如需指定觸發程式和動作的詳細資訊，請參閱 [使用觸發程式和動作](#)。

## 處理作業工作清單

您可以使用處理作業工作清單，視需要列出使用 **Oracle Process Cloud Service** 定義的處理作業。您可以顯示工作清單元件中工作的詳細資訊，或者將處理作業工作清單元件與工作詳細資訊元件搭配使用，以簡化清單並僅顯示所選工作的詳細資訊。

在網站上使用處理作業工作清單的先決條件如下：

- 您必須先在 **Oracle Process Cloud Service** 中定義關聯的處理作業，才能夠使用此元件顯示它們。請參閱 *Using Processes in Oracle Integration* 中的 **Developing Structured Processes**。
- 網站必須是只有指定的使用者或具備「**Oracle Content Management 使用者**」角色的使用者能夠存取的安全網站。請參閱 [變更網站安全性](#)。
- 管理員必須設定 **Oracle Process Cloud Service** 與 **Oracle Content Management** 之間的整合。請參閱 *Administering Oracle Content Management* 中的與 **Oracle Process Cloud Service** 整合。整合這兩項服務時需要 **SSO** 登入，所以這兩項服務必須位於相同的識別網域內。

若要將處理作業工作清單元件新增至頁面：

1. 瀏覽至您要編輯的頁面，並確定  已設為**編輯**。
2. 新增元件至頁面。
3. 若要編輯元件與其外觀，請按一下其功能表圖示 ，然後選擇**設定值**。
4. 選擇**自訂設定值**以設定顯示選項並篩選可用的工作。
5. 從下列選項中選擇，以設定工作清單的顯示選項。

### 備註：

您可以為元件設定顯示選項，但工作本身的顯示是由 **Oracle Process Cloud Service** 中的設計決定。

- **顯示詳細資訊**：選取以將工作詳細資訊含括在工作清單中。將處理作業工作清單元件與工作詳細資訊元件搭配使用，以簡化清單並僅顯示所選工作的詳細資訊。
- **顯示搜尋**：選取以將搜尋列含括在工作清單的頂端。如果您選取**顯示搜尋**，您會具有選取**顯示篩選條件**的額外選項，以將篩選條件選項含括在搜尋列中。使用下方的篩選選項來設定預設篩選值。
- **顯示全選**：選取以允許使用者選取所有顯示的等待處理工作。

- **頁面大小**：調整以指定要顯示的工作數目上限。如果工作少於指定的數目上限，元件會自動調整為較小的數目。如果工作多於指定的數目上限，元件會顯示數目上限並新增額外頁面的連結。
6. 從下列選項中選擇，以篩選清單中的可用工作。
- 您套用的篩選會決定初始的工作清單。如果您選擇上方的**顯示搜尋**和**顯示篩選條件**選項，使用者可以調整設定值以修改除了**來源使用者**之外所有設定值的篩選。
- **搜尋關鍵字**：視需要指定一或多個搜尋詞。搜尋結果僅會包括標題含有所有指定關鍵字的工作。如果您選取**顯示搜尋**，這些關鍵字會顯示在搜尋列中，且可由使用者移除或修改。
  - **狀態**：選取其中一個可用的「狀態」值。預設為**已指定**。
  - **指定對象**：選取其中一個可用的「指定對象」值。預設為**我和我的群組全部**，這包括使用者和其群組的所有可用工作 (包括已宣告但無法進行的工作)。
  - **來源使用者**：視需要選取一或多個發起工作的使用者。開始輸入使用者名稱以在關聯的 **Oracle Process Cloud Service** 上起始可用使用者的搜尋。您所指定的值無法對一般使用者顯示，且無法移除。使用者可以新增其他使用者名稱以進一步展開使用者清單。
  - **到期日**：視需要選取到期日是在從行事曆中選取的日期當天，或早於、晚於或者介於之間。
  - **應用程式**：視需要選取特定應用程式的名稱。您可以從使用者的可用應用程式清單中選取一或多個應用程式。
7. 在「一般」頁籤中修改間距、對齊方式以及其他呈現選項。
8. 在「樣式」頁籤中使用預先定義樣式或您自訂的選項，設定包含元件之框架的格式。

 **備註：**

如果您具有頁面上的處理作業工作清單和工作詳細資訊表單，工作詳細資訊元件會自動顯示從清單中所選工作的詳細資訊。如需觸發程式和動作的相關資訊，請參閱[使用觸發程式和動作](#)。

## 工作詳細資訊表單

您可以使用工作詳細資訊表單搭配處理作業工作清單，顯示所選工作的詳細資訊。如果您的頁面上有處理作業工作清單和處理作業詳細資訊表單，處理作業工作詳細資訊元件會自動顯示從清單中選取的工作的詳細資訊。

在網站上使用工作詳細資訊表單的先決條件如下：

- 您必須先在 **Oracle Process Cloud Service** 中定義關聯的處理作業，才能夠使用此元件顯示它們。請參閱 *Using Processes in Oracle Integration* 中的 **Developing Structured Processes**。
- 網站必須是只有指定的使用者或具備「**Oracle Content Management 使用者**」角色的使用者能夠存取的安全網站。請參閱[變更網站安全性](#)。
- 管理員必須設定 **Oracle Process Cloud Service** 與 **Oracle Content Management** 之間的整合。請參閱 *Administering Oracle Content Management* 中的與 **Oracle**

Process Cloud Service 整合。整合這兩項服務時需要 SSO 登入，所以這兩項服務必須位於相同的識別網域內。

將工作詳細資訊元件新增至頁面：

1. 瀏覽至您要編輯的頁面，並確定  已設為**編輯**。
2. 新增元件至頁面。
3. 若要編輯元件與其外觀，請按一下其功能表圖示 ，然後選擇**設定值**。
4. 選擇**自訂設定值**以設定顯示選項。
5. 從下列項目中選擇，以設定工作詳細資訊表單的顯示選項。

#### 備註：

您可以為元件設定顯示選項，但工作詳細資訊的顯示是由 Oracle Process Cloud Service 中的設計決定。

- **顯示動作**：選取時會顯示使用者可用的動作，例如核准、拒絕等。
  - **顯示儲存**：選取時會顯示**儲存**按鈕。
  - **顯示關閉**：選取時會顯示**關閉**按鈕。
  - **顯示附件**：選取時會在詳細資訊表單中顯示**附件**區段。
  - **顯示註解**：選取時會在詳細資訊表單中顯示**註解**區段。
  - **顯示歷史記錄**：選取時會在詳細資訊表單中顯示**歷史記錄**區段。
  - **顯示其他資訊**：選取時會在詳細資訊表單中顯示**其他資訊**區段。
  - **顯示連結**：選取時會在詳細資訊表單中顯示**連結**區段。
6. 在「一般」頁籤中修改間距、對齊方式以及其他呈現選項。  
依照預設，工作詳細資訊元件會展開，顯示所有指定的詳細資訊。按一下**設定高度**並調整高度值以設定特定高度。
  7. 在「樣式」頁籤中使用預先定義樣式或您自訂的選項，設定包含元件之框架的格式。
  8. 使用「連結」頁籤指定動作給元件提供的觸發程式：
    - **工作詳細資訊已送出**：使用者按一下**送出**按鈕時，會出現此觸發程式。
    - **工作已核准**：使用者按一下**核准**按鈕時，會出現此觸發程式。
    - **工作已拒絕**：使用者按一下**拒絕**按鈕時，會出現此觸發程式。
    - **工作已關閉**：使用者按一下**關閉**按鈕時，會出現此觸發程式。
    - **工作已儲存**：使用者按一下**儲存**按鈕時，會出現此觸發程式。
    - **工作註解已新增**：使用者新增註解文字並按一下**張貼註解**按鈕時，會出現此觸發程式。

按一下觸發程式即可指定動作。

 **備註：**

處理作業工作清單元件不支援手動的觸發程式或動作，不過，如果您在頁面新增工作詳細資訊元件，它會自動顯示從清單中選取的工作的詳細資訊。如需觸發程式和動作的相關資訊，請參閱[使用觸發程式和動作](#)。

如需使用 Oracle Process Cloud Service 的相關資訊，請參閱處理入門。

## 內容項目

對於企業使用者而言，每個網站都會有一個集合，其中包含與網站關聯的數位資產和內容項目。內容項目元件可讓您輕鬆將項目新增到網站。

您可以直接從編輯器的「內容」面板拖放數位資產和內容項目，系統就會自動使用適當的元件，無論是數位資產的影像或結構化內容項目都沒問題。

從「元件」面板選取的其他元件，可讓您依據內容類型動態顯示內容項目。

- [內容項目元件](#)
- [內容預留位置](#)
- [內容清單](#)
- [內容搜尋](#)
- [建議](#)

## 內容項目元件

企業使用者可以使用內容項目元件，協助設定頁面的版面配置及設定頁面互動，直到準備好新增內容項目本身為止。

當您將內容項目從「內容」面板拖放到頁面，除非該內容項目為自訂數位資產類型，否則會自動插入內容項目元件來存放項目。將自訂數位資產拖曳到頁面，會以影像或視訊元件的形式插入該資產。您必須移除該資產並在新增內容項目元件後，將自訂數位資產或標準內容項目從「內容」面板拖曳到該元件，才能正確顯示。

在頁面新增內容項目元件：

1. 瀏覽至您要編輯的頁面，並確定  已設為**編輯**。
2. 新增元件至頁面。
3. 若要編輯內容項目元件內容，請按一下其功能表圖示 ，然後選擇**編輯**，或選取**設定值**以開啟設定值面板，然後按一下編輯圖示 。

 **備註：**

如果內容項目元件中所用的內容類型為自訂數位資產而不是標準內容類型，您將無法編輯自訂數位資產類型。元件功能表將不會顯示**編輯**選項，設定值面板也不會顯示編輯圖示。

4. 變更內容項目，然後按一下**儲存**。內容項目會儲存為內容儲存區域中的新修訂版本。
5. 若要編輯元件外觀，請按一下其功能表圖示 ，然後選擇**設定值**。
6. 在「一般」頁籤中修改間距、對齊方式以及其他呈現選項。  
如果元件已有指定的內容項目，就會顯示內容項目的縮圖檢視。如果尚未指定任何內容項目，則會顯示預留位置影像和訊息。
7. 請從下列選項中選擇，設定顯示內容的其他預設值。
  - **要使用的版本**：如果您選取**使用最新版本的資產**，而且有較新、未發布的內容項目版本，那麼在發布目前的網站更新時，將會自動發布此較新的版本。除非特別要求，否則所有項目都將是草稿或最新版本。
  - **項目檢視**：選取用來顯示內容項目的版面配置。提供的**預設**版面配置會顯示內容項目中的所有欄位。如果內容項目有其他為其設計的自訂版面配置，您可以選擇任何可用的版面配置。如果選取的自訂版面配置已啟用**在網站中使用時新增對自訂設定值的支援**，您可以在該版面配置新增自訂資料，方法為按一下 ，在**自訂資料**欄位中輸入資料，然後按一下**上一步**以返回「一般」頁籤。
  - **要顯示個別項目的頁面**：如果您指定了一或多個頁面作為詳細資訊頁面，那些頁面會列在此處。請選擇當使用者按內容項目的連結以檢視詳細資訊時，要顯示詳細資訊的頁面。

#### 備註：

如果您未建立詳細資訊頁面，預設版面配置中就不會顯示內容項目的詳細資訊連結。

8. 在「樣式」頁籤中使用預先定義樣式或您自訂的選項，設定包含元件之框架的格式。

## 內容預留位置

企業使用者可以使用內容預留位置元件，動態顯示一或多種類型的內容項目。

例如，可以在指定的詳細資訊頁面中使用內容項目預留位置，當使用者按一下連結來取得特定內容項目更詳細的資訊時，它就會自動載入關聯之內容項目的詳細檢視。例如，如果有一個含有多篇文章的頁面，其中的每篇文章都有大標題和影像，當使用者按一下特定文章時，不論選擇的是哪一篇文章，詳細資訊頁面都會顯示完整文章。

在頁面新增內容預留位置元件：

1. 瀏覽至您要編輯的頁面，並確定  已設為**編輯**。
2. 新增元件至頁面。
3. 若要編輯元件及其外觀，請按一下其功能表圖示 ，然後選擇**設定值**。
4. 在「一般」頁籤中修改間距、對齊方式以及其他呈現選項。
5. 請從下列選項中選擇，設定顯示內容的其他預設值。
  - **內容類型**：選取一或多個可用的內容類型。內容類型是指網站集中的內容項目，以及包含自訂數位資產類型。

- **項目檢視**：選取用來顯示內容項目的版面配置。提供的**預設**版面配置會顯示內容項目中的所有欄位。如果內容項目有其他為其設計的自訂版面配置，您可以選擇任何可用的版面配置。
- **要顯示個別項目的頁面**：如果您指定了一或多個頁面作為詳細資訊頁面，那些頁面會列在此處。請選擇當使用者按內容項目的連結以檢視詳細資訊時，要顯示詳細資訊的頁面。

 **備註：**

如果您未建立詳細資訊頁面，預設版面配置中就不會顯示內容項目的詳細資訊連結。

6. 在「樣式」頁籤中使用預先定義樣式或您自訂的選項，設定包含元件之框架的格式。

## 內容清單

企業使用者可以使用內容清單元件動態顯示特定類型的內容項目。

例如，可以在指定的詳細資訊頁面中使用內容項目清單，當使用者按一下連結來取得特定內容項目的更詳細資訊時，它就會自動載入關聯之內容項目的詳細檢視。例如，如果有一個含有多篇文章的頁面，其中的每篇文章都有大標題和影像，當使用者按一下特定文章時，不論選擇的是哪一篇文章，詳細資訊頁面都會顯示完整文章。

將內容清單元件新增至頁面：

1. 瀏覽至您要編輯的頁面，並確定  已設為**編輯**。
2. 新增元件至頁面。
3. 若要編輯元件及其外觀，請按一下其功能表圖示 ，然後選擇**設定值**。
4. 在「一般」頁籤中修改間距、對齊方式以及其他呈現選項。
5. 請從下列選項中選擇，設定顯示內容的其他預設值。
  - **內容類型**：選取其中一種可用的內容類型。內容類型是指網站集中的內容項目，以及包含自訂數位資產類型。
  - **項目數上限和開始項目**：指定要顯示的項目數上限，以及開始顯示的項目。額外的項目將不會顯示。
  - **分頁**：指定是否要在清單中包括分頁選項。如果選取此選項，就可以視需要搭配使用按鈕或頁碼及不同的標籤，建立分頁的顯示方式。
  - **日期**：使用提供的選項，根據在所選日期或日期範圍之前、之後、之間或之內的項目建立日期，選取要顯示的內容項目。
  - **類別**：選取類別來篩選內容項目清單，以只顯示特定類別中的內容項目。按一下**選取類別**以開啟滑出式面板，您可用它來選取網站儲存區域中的類別。如果選取相同分類中的多個類別，顯示的項目只需要具有被指定的其中一個選定類別即可。如果從不同的分類中選取多個類別，顯示的項目則必須具有被指定的所有選定類別。例如，如果從**目的地**分類中選取**城市**和**公園**等類別，顯示的項目會是城市或公園。如果從**目的地**分類中選取**城市**類別，從**地區**分類中選取**歐洲**類別，顯示的項目則是位於歐洲的城市。

依照預設，所選類別的所有子項都可以使用。如果想要將清單限制為某個特定類別節點中的項目，請選取該節點並取消選取**包含子項類別**。

- **語言**：從與儲存區域關聯的語言選擇一個語言。
- **其他查詢字串 (選擇性)**：使用類似 `field.dept eq "Finance"` 的語法指定其他查詢參數，以進一步縮小所顯示項目清單的範圍。如需可用的運算子清單，請參閱下表。
- **排序依據**：依名稱或日期以遞增或遞減順序排序項目。若內容類型包含其他日期、編號或十進位數欄位，您也可以依那些欄位排序。

您也可以選取**自訂**，然後輸入自訂表示式。例如，若要依部門編號以遞增順序排序，您可以輸入 `fields.deptno:asc`。您也可以定義 **URL** 參數來進行排序。例如，您可以輸入 `{{URLParams.sortBy}}`，然後再將下列參數新增到您網站 **URL** 的結尾：  
`sortBy=fields.deptno:asc`。

- **項目檢視**：選取用來顯示內容項目的版面配置。提供的**預設**版面配置會顯示內容項目中的所有欄位。如果內容項目有其他為其設計的自訂版面配置，您可以選擇任何可用的版面配置。如果選取的自訂版面配置已啟用**在網站中使用時新增對自訂設定值的支**

**援**，您可以在該版面配置新增自訂資料，方法為按一下 ，在**自訂資料**欄位中輸入資料，然後按一下**上一步**以返回「一般」頁籤。

- **要顯示個別項目的頁面**：如果您指定了一或多個頁面作為詳細資訊頁面，那些頁面會列在此處。請選擇當使用者按內容項目的連結以檢視詳細資訊時，要顯示詳細資訊的頁面。

#### 備註：

如果您未建立詳細資訊頁面，預設版面配置中就不會顯示內容項目的詳細資訊連結。

- **清單檢視**：選取一個區段版面配置來排列項目。此清單可以包含自訂版面配置。以下是提供的版面配置：
  - **水平**：以水平方式將項目一個接著一個排列。
  - **垂直**：以垂直方式將項目一個接著一個排列。若未選取版面配置，這會是預設的版面配置。
  - **2 欄**：以兩個為一組的方式在多列中排列項目。
  - **3 欄**：以三個為一組的方式在多列中排列項目。

按一下所選版面配置旁邊的向右箭號，即可修改該版面配置的設定值。

- **空白的清單檢視**：選取顯示空白清單的方式。
- **選項**：選擇在「內容搜尋」元件中指定的搜尋查詢上是否應有一個自動重新整理選項。
- **間距**：選擇顯示項目的方式。

6. 在「樣式」頁籤中使用預先定義樣式或您自訂的選項，設定包含元件之框架的格式。

表格 10-1 查詢運算子

運算子	範例	支援的資料類型	描述
eq	<pre>?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"</pre>	文字、參照、數字、小數、布林值、日期時間	等於運算子 (eq) 會比對與查詢中所提供完全相符的值。此運算子不適用於多重值資料類型。以此運算子提供的值不區分大小寫，但標準欄位除外。此運算子甚至會考量值中的特殊字元。
co	<pre>?q=(type eq "Employee" AND name co "john alex") ?q=(type eq "Car" AND fields.features co "manual")</pre>	文字、參照、數字、小數、日期時間、大量文字	包含運算子 (co) 會比對條件中提供的每個單字。單字會藉由以特殊字元分割值的方式構成。它會產生至少包含其中一個單字的結果 (在此範例中為 <b>john</b> 或 <b>alex</b> 或兩者兼具)。進行搜尋時，此運算子不會考量值中的特殊字元。此運算子不會搜尋忽略字。若要進一步瞭解忽略字，請參考 <b>Apache Lucene</b> 文件。就單一值屬性而言，此運算子適用於文字、大量文字，就多重值屬性而言，則適用於文字、參照、數字、小數、日期時間及大量文字。若要瞭解可能的日期時間格式，請參考下面的支援的日期/日期時間格式 (24 小時制) 表格。以此運算子提供的值不區分大小寫。
sw	<pre>?q=type eq "Employee" AND name sw "Joh" ?q=type eq "Employee" AND fields.city sw "Los"</pre>	文字	開頭為運算子 (sw) 只會比對欄位條件中所提供的初始字元值。此運算子不適用於多重值資料類型。以此運算子提供的值不區分大小寫。
ge	<pre>?q=(type eq "Employee" AND fields.age ge "40") ?q=type eq "DigitalAsset" AND updatedAt ge "20171026"</pre>	數字、小數、日期時間	大於或等於運算子 (ge) 只會比對數字和日期時間值。若要瞭解可能的日期時間格式，請參考下面的支援的日期/日期時間格式 (24 小時制) 表格。此運算子不適用於多重值資料類型。
le	<pre>?q=(type eq "Employee" AND fields.weight le "60.6")</pre>	數字、小數、日期時間	小於或等於運算子 (le) 只會比對數字和日期時間值。若要瞭解可能的日期時間格式，請參考下面的支援的日期/日期時間格式 (24 小時制) 表格。此運算子不適用於多重值資料類型。

表格 10-1 (續) 查詢運算子

運算子	範例	支援的資料類型	描述
gt	?q=(type eq "Employee" AND fields.age gt "20")	數字、小數、日期時間	大於運算子 (gt) 只會比對數字和日期時間值。若要瞭解可能的日期時間格式，請參考下面的支援的日期/日期時間格式 (24 小時制) 表格。若要瞭解可能的日期時間格式，請參考下面的支援的日期/日期時間格式 (24 小時制) 表格。此運算子不適用於多重值資料類型。
lt	?q=(type eq "Employee" AND fields.age lt "20") ?q=type eq "Employee" AND createdAt lt "1994/09/26T16:23:45.208"	數字、小數、日期時間	小於運算子 (lt) 只會比對數字和日期時間值。若要瞭解可能的日期時間格式，請參考這一節： <b>支援的日期/日期時間格式 (24 小時制)</b> 。此運算子不適用於多重值資料類型。
mt	?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*")	文字、大量文字	詞組查詢或鄰近搜尋 (比對) 運算子 (mt) 可讓您尋找彼此位於特定距離範圍內的單字。結果會依最佳相符項目排序。當 "petrol 20kmpl" 條件中提供的值需要尋找可能包含「在高速公路中行駛的汽油油耗量為 20KMPL」的實際內容時，就很適合運用此運算子搜尋內容項目。 比對運算子也可以在指定的值內使用萬用字元，並且在單一值內同時支援單一和多個萬用字元搜尋。使用 ? 代表搜尋單一萬用字元，使用 * 代表搜尋多個萬用字元。用於單一萬用字元搜尋的 "Jo?n" 和用於多個萬用字元搜尋的 "Jo*"，都可以搜尋出 "John" 和 "Joan"。 此運算子同時適用於單一值和多重值資料類型。此運算子不會搜尋忽略字。若要進一步瞭解忽略字，請參考 Apache Lucene 文件。以此運算子提供的值不區分大小寫。
sm	?q=(type eq "Employee" And fields.city sm "Rome")	文字、大量文字	相似性查詢運算子 (sm)。此運算子允許搜尋與所指定條件發音相似的值 (也稱為模糊搜尋)，預設最多使用兩次編輯來比對結果。"Rome" 與 "Dome" 相似。此運算子同時適用於單一值和多重值資料類型。以此運算子提供的值不區分大小寫。

表格 10-1 (續) 查詢運算子

運算子	範例	支援的資料類型	描述
AND	?q=(type eq "Employee" AND name eq "John" AND fields.age ge "40")	N/A	AND 運算子可在多個查詢條件之間放入「且」條件。此運算子的優先順序高於 OR。
OR	type eq "Employee" AND name eq "John" OR fields.age ge "40"	N/A	OR 運算子可在多個查詢條件之間放入「或」條件。
()	?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	用來在搜尋條件中建立條件組合的括號(括住)運算子。此運算子具備最高優先順序，其次是 AND，最後是 OR。

## 內容搜尋

企業使用者可以使用內容搜尋元件，以及指定搜尋所傳回的動作。

可以插入自訂搜尋列來變更或重新整理頁面上顯示的內容，或是選擇其他動作，例如開啟搜尋結果頁面或顯示警示。

在頁面新增內容搜尋元件：

1. 瀏覽至您要編輯的頁面，並確定  已設為**編輯**。
2. 新增內容搜尋元件至頁面。
3. 若要編輯元件及其外觀，請按一下其功能表圖示 ，然後選擇**設定值**。
4. 在「一般」頁籤中修改預留位置文字、間距、對齊方式以及其他呈現選項。
5. 在「樣式」頁籤中使用預先定義樣式或您自訂的字型、框線、背景色彩等等選項，設定包含元件之框架的格式。
6. 若要使用搜尋元件重新整理目前頁面上的資料，請新增「內容清單」元件至頁面。前往**設定值**，然後選擇內容類型和任何其他查詢資訊，例如要顯示多少項目、是否將結果分頁或是否要啟用捲動時遲緩載入功能(在捲動頁面時載入內容)。您現在就可以前往「檢視」模式並試著搜尋。
7. 若要使用搜尋元件重新整理目前頁面上的資料，但頁面上若有多個「內容清單」元件的話，就必須將顯示結果之內容清單以外的其他內容清單的自動查詢關閉。前往「內容清單」設定值，然後取消勾選在**「內容搜尋」元件中搜尋查詢時自動重新整理**。
8. 若要在頁面上使用多個內容搜尋元件(每個搜尋元件分別驅動一個特定的結果內容清單)，則必須取消勾選「內容清單」元件中的所有自動重新整理選項，改用觸發程式/動作將內容搜尋元件與其個別的內容清單關聯。

- 對於搜尋元件，請在「連結」頁籤中設定動作與元件的關聯。從「選取連結類型」下拉式清單中選擇一個選項。選取**觸發程式動作**，然後按一下**搜尋查詢時**，即可查看可用的觸發程式或建立新的觸發程式。找到想要搜尋的內容清單，展開清單，然後將**搜尋內容**拖曳至動作清單。從**搜尋**底下選取**搜尋字串**。您選擇的內容清單會在**執行此動作於**清單中被選取起來。
- 您也可以使用內容搜尋元件，將查詢傳送到搜尋結果頁面。在搜尋元件的「連結」頁籤中，選取**搜尋頁面**以顯示搜尋結果頁面。您可以使用預設頁面，或是連結至您所建立的搜尋結果頁面。從「頁面」下拉式清單中選取頁面，然後選擇該頁面的顯示動作。您可以針對特定內容類型設計結果、在新視窗開啟結果等等。

下列要點可協助您設計有效的搜尋結果頁面：

- 建立一個頁面，然後將它指定為搜尋頁面。此頁面預設會標示為隱藏，但您可以在頁面特性中變更設定。
- 編輯頁面並新增內容清單至頁面。您可以編輯內容清單的設定值，以指定內容類型。也可以在上述的搜尋元件連結設定值中指定內容類型。如果使用連結設定值，就可以使用能夠顯示不同內容類型結果的搜尋結果頁面 (取決於起始搜尋的元件)。
- 若要編輯顯示，請變更設定值，像是選擇遲緩載入分頁 (因為頁面可能只用於搜尋結果)。
- 您也可以將內容搜尋元件置於搜尋結果頁面中。此元件將會呼應用於啟動頁面的搜尋字串，讓使用者視需要縮小搜尋範圍。

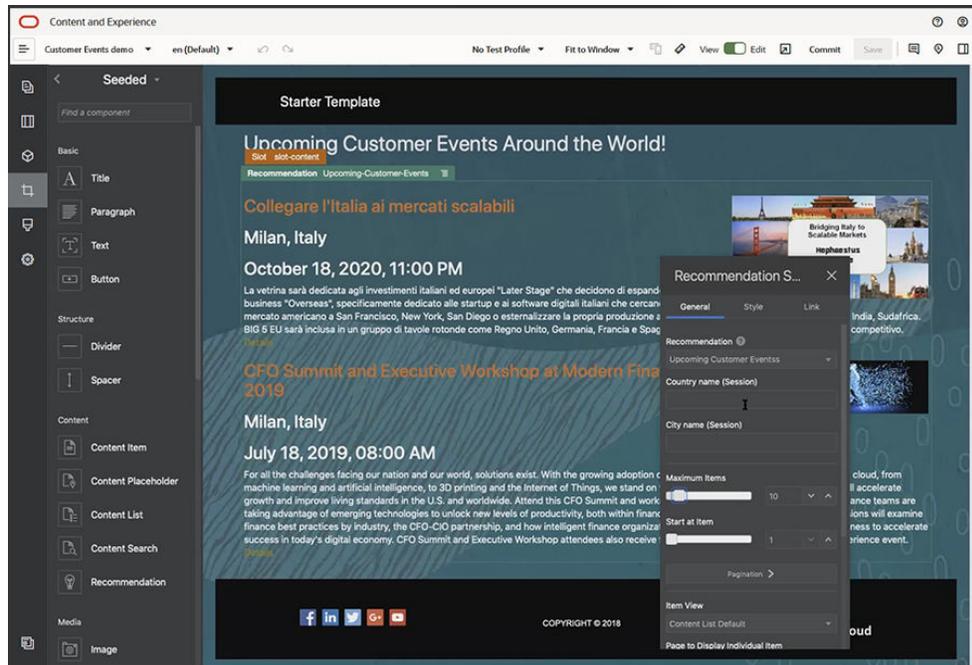
## 建議

企業使用者可以使用**建議**元件，根據網站訪客所在位置或感興趣項目來顯示網站資產，提供訪客個人化體驗。儲存區域提供者會在建立建議時定義一組規則，用來尋找符合受眾屬性 (例如網站訪客的地理位置資訊) 的資產。例如，使用歐洲 IP 位址的網站訪客就會優先在網站首頁上看到歐洲的事件公告，而北美洲的網站訪客將會看到美國和加拿大的事件。

建議就像數位資產一樣與儲存區域相關聯，而且可以透過複查工作流程進行編輯和移動，並由內容提供者發布至儲存區域。不過即使您沒有提供者權限，還是可以先檢視和測試建議來查看其運作方式，再將它們運用於網站或無頭 (headless) 體驗。

在頁面新增建議：

- 按一下側邊功能表中的**網站**，選取要新增建議的網站，然後從滑鼠右鍵功能表中選擇**開啟**，或按一下動作列中的 。
- 將網站切換至**編輯**，然後選取要使用的更新或建立新的更新。
- 從側邊功能表中選取**元件**。
- 開啟**內建**。
- 按一下並拖曳「內容」區段底下的**建議**，將它放在頁面上。
- 從「建議」功能表 () 中選取**設定值**。



7. 在「建議」設定值的一般頁籤中，選取要使用的建議。頁面將重新整理並顯示建議的資產。
8. (選擇性) 按一下所選建議旁邊的箭號，輸入此建議所使用之受眾屬性的預設值。完成新增值之後，請按一下**上一步**。
9. 變更所有想要的其他特性，例如要使用哪一個版本、要顯示的項目數上限或預設的內容版面配置。完成後，請關閉建議設定值，然後按一下**儲存**。

### 備註：

如果您已建立含有預先定義受眾屬性值的測試設定檔，請從功能表列的測試設定檔功能表中選取一個設定檔，預覽建議如何回應這些輸入值。測試設定檔值會覆寫您在元件設定值中設定的預設值。

10. 如果您的建議在其規則中使用目前日期 (系統)，而您想要將此建議當成是不同的日期來進行測試，請從測試設定檔功能表中選取**系統日期**，然後選取要用來測試的日期。

## 其他元件

讓我們來瞭解一些可結合不同內容類型的元件。

- [地圖](#)
- [大標題](#)
- [文章](#)
- [影像 \(含文字\)](#)
- [元件群組](#)
- [在頁面上使用 Cobrowse 功能](#)

- Oracle Intelligent Advisor
- Oracle Visual Builder

## 地圖

在網站中新增地圖，讓使用者以互動方式探索位置周圍的區域。



在頁面中新增地圖：

1. 瀏覽至您要編輯的頁面，並確定  已設為**編輯**。
2. 新增元件至頁面。Oracle 地圖元件會顯示預設位置。
3. 若要變更地圖位置，以及調整其顯示特性，請按一下其功能表圖示 ，然後選擇**設定**。
4. 輸入地圖的起始**位置**。您可以使用地址、郵遞區號，或以逗點區隔的緯度和經度 (例如，40.5,-57.6)。

如果您使用地址，請確定提供足夠的資訊來符合單一位置。如果地址符合多個位置，則地圖會空白。

5. 指定地圖的起始**縮放**比例。您可以選擇性地允許使用者使用滑鼠、觸控板或畫面控制項來調整縮放比例。
6. 指定任何樣式、對齊方式和顯示選項：
  - **縮放**：啟用畫面和裝置選項，允許使用者調整地圖的縮放比例。
  - **平移**：啟用畫面和裝置選項，允許使用者移動地圖的焦點區域。
  - **顯示標記**：使用圖釘圖示來標示起始位置。
  - **比例**：包括顯示目前縮放比例的比例指標。
  - **總覽**：包括在較大相關資訊環境中顯示目前地圖的嵌入地圖。

## 大標題

您可以新增含有標題文字和支援之影像與段落的大標題，以吸引使用者注意網站上的某個特定點。

1. 瀏覽至您要編輯的頁面，並確定  已設為**編輯**。
2. 新增元件至頁面。

3. 按一下元件的不同區域以新增文字。您需將內容新增至主要大標題及主要大標題底下的段落。文字會採用元件預設樣式的格式。按 **Enter** 鍵以新增額外的文字行。
4. 如果您要變更文字任何部分的預設格式，請選取您要格式化的文字，然後在格式設定工具列 (例如字型、色彩或對齊方式) 中選取任何選項。
5. 若要移除使用這些選項所套用的格式設定，請選取文字，然後按一下 。
6. 若要編輯元件及其外觀，請按一下其功能表圖示 ，然後選擇**設定值**。
7. 選擇**元件**以設定與顯示之內容有關的詳細資訊。

- **影像：**

- 按一下**選取**以在大標題中使用影像。從網站儲存區域中選取影像 (此影像來自與您共用的文件資料夾中)，或上傳影像到文件資料夾。找到並選取您要使用的圖像。若未見到任何數位資產，請按一下 ，然後將集合篩選條件變更為**全部**。若從文件資料夾中選取影像，您可以連結到檔案，而不是將檔案複製到網站。若要連結至檔案，請選取**使用原始檔案參照，而不是將檔案複製到網站**。若未選取此選項，系統就會在網站儲存檔案的複本，然後從網站參照。若連結至原始的檔案，就不需要複製內容。即使檔案的權限變更或其他因素而限制了檢視，網站訪客仍能透過連結查看內容。按一下**確定**。如果選取數位資產，您可以選取特定的轉譯。若未選取轉譯，則會使用原始大小。若想要在網站發布時一同發布最新版本的資產，請選取**使用最新版本的資產**。

若要編輯影像，請按一下 ，然後使用以下任一動作編輯影像：

- \* 若要剪裁影像，請按一下  **剪裁**。從剪裁工具列中選取其中一個預先定義的影像比例，或視需要拖曳影像上的剪裁控點。完成之後，請按一下剪裁工具列中的**剪裁**。
- \* 若要旋轉或翻轉影像，請按一下  **旋轉**。在旋轉工具列中，輸入自訂的旋轉角度、使用按鈕將影像向左或向右旋轉，或是選取要將影像水平翻轉還是垂直翻轉。
- \* 若要為影像加上浮水印，請按一下  **浮水印**。使用浮水印工具，視需要新增影像的文字、變更文字大小、樣式、色彩以及不透明度。
- \* 若要變更影像格式，請按一下  **選項**，然後從**格式**下拉式清單中選取一個新的格式。
- \* 若要變更背景色彩，請按一下  **選項**，然後從**背景色彩**下拉式功能表中選取一個選項。
- \* 如果是編輯 .jpg 或 .webp (可在 Google Chrome 瀏覽器中使用)，可以變更影像品質以製作較小的影像檔。按一下  **選項**，然後在**品質**方塊中輸入新的百分比。
- \* 若要還原或重做變更，請按一下  或 。若要移除所做的全部變更，請按一下**重設**。
- \* 若要變更影像的放大倍率，請使用縮放控制項 ()。

- **標題**：輸入工具提示中所要顯示的文字。
  - **替代文字**：輸入使用輔助功能時將顯示的替代文字。
  - **說明文字**：輸入顯示在影像下的說明文字。
  - **對齊方式、寬度與間距**：視需要變更影像的版面配置。
  - **標題**：變更大標題的間距。
  - **段落**：變更大標題下方所顯示文字的間距。
8. 在「一般」頁籤中修改間距、對齊方式以及其他呈現選項。
  9. 在「樣式」頁籤中新增圍繞文字的格式，同時自訂背景色彩、字型以及框線。

## 文章

您可以將文章新增至您的網站，文章可將大標題、段落及影像的元件結合成一個容易使用的元件。

1. 瀏覽至您要編輯的頁面，並確定  已設為**編輯**。
  2. 新增元件至頁面。
  3. 按一下元件的各個不同區域來新增文字。您需將內容新增至文章大標題及其底下的子標題。然後可以在子標題底下新增文章的文字。所有文字都會採用元件預設樣式的格式。按 **Enter** 鍵即可新增其他文字行。
  4. 如果您要變更文字任何部分的預設格式，請選取您要格式化的文字，然後在格式設定工具列 (例如字型、色彩或對齊方式) 中選取任何選項。
  5. 若要移除使用這些選項所套用的格式設定，請選取文字，然後按一下 。
  6. 若要編輯元件及其外觀，請按一下其功能表圖示 ，然後選擇**設定值**。
  7. 選擇**元件**以設定與顯示之內容有關的詳細資訊。
    - **影像**：
      - 按一下**選取**以在文章中使用影像。從網站儲存區域中選取影像 (此影像來自與您共用的文件資料夾中)，或上傳影像到文件資料夾。找到並選取您要使用的圖像。若未見到任何數位資產，請按一下 ，然後將集合篩選條件變更為**全部**。若從文件資料夾中選取影像，您可以連結到檔案，而不是將檔案複製到網站。若要連結至檔案，請選取**使用原始檔案參照，而不是將檔案複製到網站**。若未選取此選項，系統就會在網站儲存檔案的複本，然後從網站參照。若連結至原始的檔案，就不需要複製內容。即使檔案的權限變更或其他因素而限制了檢視，網站訪客仍能透過連結查看內容。按一下**確定**。如果選取數位資產，您可以選取特定的轉譯。若未選取轉譯，則會使用原始大小。若想要在網站發布時一同發布最新版本的資產，請選取**使用最新版本的資產**。
- 若要編輯影像，請按一下 ，然後使用以下任一動作編輯影像：
- \* 若要剪裁影像，請按一下  **剪裁**。從剪裁工具列中選取其中一個預先定義的影像比例，或視需要拖曳影像上的剪裁控點。完成之後，請按一下剪裁工具列中的**剪裁**。

- \* 若要旋轉或翻轉影像，請按一下  **旋轉**。在旋轉工具列中，輸入自訂的旋轉角度、使用按鈕將影像向左或向右旋轉，或是選取要將影像水平翻轉還是垂直翻轉。
  - \* 若要為影像加上浮水印，請按一下  **浮水印**。使用浮水印工具，視需要新增影像的文字、變更文字大小、樣式、色彩以及不透明度。
  - \* 若要變更影像格式，請按一下  **選項**，然後從**格式**下拉式清單中選取一個新的格式。
  - \* 若要變更背景色彩，請按一下  **選項**，然後從**背景色彩**下拉式功能表中選取一個選項。
  - \* 如果是編輯 .jpg 或 .webp (可在 Google Chrome 瀏覽器中使用)，可以變更影像品質以製作較小的影像檔。按一下  **選項**，然後在**品質**方塊中輸入新的百分比。
  - \* 若要還原或重做變更，請按一下  或 。若要移除所做的全部變更，請按一下**重設**。
  - \* 若要變更影像的放大倍率，請使用縮放控制項 ( )。
    - **標題**：輸入工具提示中所要顯示的文字。
    - **替代文字**：輸入使用輔助功能時將顯示的替代文字。
    - **說明文字**：輸入顯示在影像下的說明文字。
    - **對齊方式、寬度與間距**：視需要變更影像的版面配置。
      - **標題**：變更文章標題的間距。
      - **段落 (子標題) 和段落**：變更文章標題下方所顯示文字的間距。
8. 在「一般」頁籤中修改元件的間距、對齊方式以及其他呈現選項。
  9. 在「樣式」頁籤中新增圍繞文字的格式，同時自訂背景色彩、字型以及框線。

## 影像 (含文字)

您可以使用影像與文字元件，此元件可將段落和影像的元件結合成一個容易使用的元件。

1. 瀏覽至您要編輯的頁面，並確定  已設為**編輯**。
2. 新增元件至頁面。
3. 按一下元件的段落區域以新增文字。所有文字均採用元件的預設樣式格式。按 **Enter** 鍵即可新增其他文字行。
4. 如果您要變更文字任何部分的預設格式，請選取您要格式化的文字，然後在格式設定工具列 (例如字型、色彩或對齊方式) 中選取任何選項。
5. 若要移除使用這些選項所套用的格式設定，請選取文字，然後按一下 。
6. 若要編輯元件及其外觀，請按一下其功能表圖示 ，然後選擇**設定值**。

7. 選擇**元件**，以設定與顯示內容相關的詳細資訊。· **影像：**

- 按一下**選取**以使用影像。從網站儲存區域中選取影像 (此影像來自與您共用的文件資料夾中)，或上傳影像到文件資料夾。找到並選取您要使用的圖像。若未見到任何數位資產，請按一下 ，然後將集合篩選條件變更為**全部**。若從文件資料夾中選取影像，您可以連結到檔案，而不是將檔案複製到網站。若要連結至檔案，請選取**使用原始檔案參照，而不是將檔案複製到網站**。若未選取此選項，系統就會在網站儲存檔案的複本，然後從網站參照。若連結至原始的檔案，就不需要複製內容。即使檔案的權限變更或其他因素而限制了檢視，網站訪客仍能透過連結查看內容。按一下**確定**。如果選取數位資產，您可以選取特定的轉譯。若未選取轉譯，則會使用原始大小。若想要在網站發布時一同發布最新版本的資產，請選取**使用最新版本的資產**。

若要編輯影像，請按一下 ，然後使用以下任一動作編輯影像：

- \* 若要剪裁影像，請按一下  **剪裁**。從剪裁工具列中選取其中一個預先定義的影像比例，或視需要拖曳影像上的剪裁控點。完成之後，請按一下剪裁工具列中的**剪裁**。
- \* 若要旋轉或翻轉影像，請按一下  **旋轉**。在旋轉工具列中，輸入自訂的旋轉角度、使用按鈕將影像向左或向右旋轉，或是選取要將影像水平翻轉還是垂直翻轉。
- \* 若要為影像加上浮水印，請按一下  **浮水印**。使用浮水印工具，視需要新增影像的文字、變更文字大小、樣式、色彩以及不透明度。
- \* 若要變更影像格式，請按一下  **選項**，然後從**格式**下拉式清單中選取一個新的格式。
- \* 若要變更背景色彩，請按一下  **選項**，然後從**背景色彩**下拉式功能表中選取一個選項。
- \* 如果是編輯 .jpg 或 .webp (可在 Google Chrome 瀏覽器中使用)，可以變更影像品質以製作較小的影像檔。按一下  **選項**，然後在**品質**方塊中輸入新的百分比。
- \* 若要還原或重做變更，請按一下  或 。若要移除所做的全部變更，請按一下**重設**。
- \* 若要變更影像的放大倍率，請使用縮放控制項 ( )。
- **標題**：輸入要顯示在工具提示中的文字。
- **替代文字**：輸入使用輔助功能時顯示的替代文字。
- **說明文字**：輸入顯示在影像下的說明文字。
- **對齊方式、寬度及間距**：視需要變更影像的版面配置。

· **段落**：變更文字の間距。

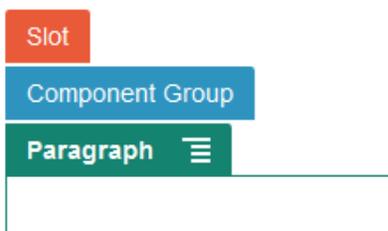
8. 使用「一般」頁籤修改元件の間距、對齊方式及其他呈現方式選項。
9. 使用「樣式」頁籤新增圍繞文字的格式設定，以及自訂背景色彩、字型和框線。

## 元件群組

您可以結合一或多個元件，建立可命名和重複使用的元件群組。

當您儲存元件群組時，元件群組會以您所指定名稱儲存成為自訂元件，然後會出現在編輯器的自訂元件清單中。

1. 瀏覽至您要編輯的頁面，並確定  已設為**編輯**。
2. 將元件群組新增至頁面。元件群組是透過  來識別。
3. 將一或多個元件拖放至自訂元件的框架。
4. 在元件群組內設定元件位置和大小，就和槽位中元件的設定方式一樣。
5. 若要編輯元件與其外觀，請按一下其功能表圖示 ，然後選擇**設定值**。如果您按一下元件名稱，而非功能表圖示，則可以看到和選取元件群組 (或槽位) 的功能表圖示：



元件群組的「設定值」頁籤可讓您指定元件群組的位置、背景影像，以及套用至整個元件群組的其他設定值。

6. 當您準備好儲存對元件群組的變更時，請按一下元件群組、按一下其功能表圖示 ，然後選擇**儲存**。
  - a. 在對話方塊中，輸入元件群組的名稱。名稱中可以包含字母、數字、底線 ( \_ ) 以及連字號 (-)。如果您輸入空格，系統會自動以連字號取代。

如果這是新的元件群組，則您無法使用現有自訂元件的名稱。如果您已將現有元件群組新增至頁面，並修改元件群組，然後嘗試儲存變更，便可以提供名稱來建立新的元件群組，或是選取**覆寫現有元件群組**以使用變更來更新現有元件群組。
  - b. 按一下**儲存**。

元件群組會以指定的名稱儲存為自訂元件。如果您是擁有人或有人將元件與您共用，則它會出現在元件管理程式內專屬的資料夾中以及編輯器的自訂元件清單中。您可以將元件群組共用成為任何自訂元件。

## 在頁面上使用 Cobrowse 功能

Oracle Cobrowse Cloud Service 是一種協作工具，可讓您與他人共用畫面或起始 Cobrowse 階段作業。例如，您可以在訂購單包含此功能，讓業務代表可在客戶訂購的同時檢視客戶的畫面。

若要使用此功能，您必須先對網站啟用此功能。您可以啟用兩種類型的啟動器命令檔：一種使用自訂按鈕 (Launch Point 2)，另一種使用預設的 Cobrowse 按鈕 (Launch Point 1)。您需要在對網站啟用此功能時決定將使用哪一種啟動器，然後新增必要的命令檔。請參閱[啟用 Cobrowse 整合](#)瞭解詳細資訊。

對網站啟用 Cobrowse 後，就可以將所有頁面設定成允許使用 Cobrowse 功能。

1. 瀏覽至您要編輯的頁面，並確定  已設為**編輯**。
2. 按一下 。選取 Cobrowse 區段中的核取方塊。
3. 按一下**關閉**。

如果啟用 Launch Point 1 命令檔，下次您檢視該頁面或按快速鍵 (若有設定) 後，將會見到預設的 Cobrowse 按鈕。

如果啟用 Launch Point 2 命令檔，您需要在頁面上新增自訂按鈕。

1. 確定  設為**編輯**。
2. 新增元件至頁面。「Cobrowse 啟動器」元件會列在元件的「整合」區段中。
3. 若要編輯元件及其外觀，請按一下其功能表圖示 ，然後選擇**設定值**。您可以調整按鈕的標籤、它的外觀、大小以及對齊方式。在「樣式」頁籤中新增圍繞文字的格式，同時自訂背景色彩、字型以及框線。如需使用更多進階樣式，請在目前網站樣板之主題設計資料夾的 `design.json` 和 `design.css` 檔案中，編輯或新增樣式類別。樣式類別前置碼為 `scs-cobrowse`。

網站發布之後，網站訪客便可使用 Cobrowse 按鈕，與您組織的業務代表啟始一個階段作業。訪客按一下 Cobrowse 按鈕後，系統便會給予一個安全階段作業 ID。訪客接著透過電話將此 ID 傳達給您組織可以存取 Cobrowse 代理程式主控台的業務代表。代理程式會使用此主控台啟始一個階段作業，此階段作業會一直繼續到訪客或代理程式將其終止為止。如果使用 Launch Point 2 設定，啟動器元件會有一個固定的 ID：`cec-start-cobrowse`。請在 Cobrowse 主控台中使用此 ID。

### 使用注意事項

若使用 Cobrowse 即時模式 (ICB)，除非相同的 Cobrowse 網站 ID 啟用 iFrame 內容，否則不能在頁面上檢視視訊或內嵌的 iFrame。因此，部分 Oracle Content Management 元件不會在 ICB 模式的 Cobrowse 代理程式主控台顯示。請使用 Cobrowse 進階模式 (ACB) 來顯示下列元件：

- 視訊
- Youtube
- 文件管理程式
- Facebook 按讚
- Twitter 跟隨
- Twitter 分享
- Facebook 推薦
- 對話

使用 iFrame 取得內容的自訂元件也不會在即時模式顯示。

請參閱 *Cobrowse Deployment and Use Guide* 中的 [Cobrowse Overview](#)，瞭解關於 Oracle Cobrowse Cloud Service 的詳細資訊。請參閱[啟用 Cobrowse 整合](#)，瞭解在安全網站和開發中網站使用 Cobrowse 的詳細資訊。

## Oracle Intelligent Advisor

Oracle Intelligent Advisor (之前稱為 Oracle Policy Automation) 可用來實行線上互動式提問案例，例如疑難排解意見或服務的資格評估。它可以藉由擷取 Microsoft Word 和 Excel 文件中使用的自然語言的規則，然後依據規則建置稱為互動式提問的互動式客戶服務體驗，進而提供跨通道的建議。

您必須先設定並啟用 Intelligent Advisor 功能之後，才能開始使用。您的服務管理員需啟用服務的功能，包括新增使用中之「Intelligent Advisor 中心」的主機名稱、URL、使用者名稱與密碼。整合這兩項服務時需要 SSO 登入，所以這兩項服務必須位於相同的識別網域內。請參閱 *Integrating and Extending Oracle Content Management* 中的 [Integrate with Intelligent Advisor](#)。

在 Intelligent Advisor 端，則必須在主機網站上建立並儲存互動式提問。此外，您的 Oracle Content Management 服務必須經過授權，以供 Intelligent Advisor 主機使用。

Intelligent Advisor 設定並啟用之後，您就可以新增 Intelligent Advisor 元件到您的網站頁面。

1. 瀏覽至您要編輯的頁面，並確定  已設為**編輯**。
2. 新增元件至頁面。元件會顯示在元件清單的「整合」區段中。
3. 若要編輯元件與其外觀，請按一下其功能表圖示 ，然後選擇**設定值**。您可以調整元件的標籤、外觀、大小及對齊方式。若要使用 Intelligent Advisor 主機之互動式提問的預設樣式，請使用「樣式」頁籤。如需使用更多進階樣式，請在目前網站樣板之主題設計資料夾的 `design.css` 檔案中，編輯或新增樣式類別。樣式類別前置碼為 `scs-opainterview-`。

網站發布之後，網站訪客就會見到在 Intelligent Advisor 元件中選擇的互動式提問。如需有關 Intelligent Advisor 的詳細資訊，請參閱 [Intelligent Advisor 文件庫](#)。

## Oracle Visual Builder

Oracle Visual Builder 是一種代管應用程式開發基礎架構的環境。它提供運用開放原始碼標準的解決方案，能夠在 Oracle Cloud 內開發、協作及部署應用程式。

### 起始步驟

您必須先啟用並設定 Oracle Visual Builder，才能使用此服務。您的服務管理員會啟用服務的功能，包括新增應用程式建立及儲存所在的主機名稱。請參閱 *Integrating and Extending Oracle Content Management* 中的 [Integrate with Oracle Visual Builder](#)。整合這兩項服務時需要 SSO，所以這兩項服務必須位於相同的識別網域內。

在 Oracle Visual Builder 中，您必須先完成下列動作，才能搭配 Oracle Content Management 使用此功能：

- 必須在 Oracle Visual Cloud Service 端啟用跨源資源共享 (CORS)。
- 應用程式必須已經建立、設為可用於內嵌，而且必須加以設定才能與 Oracle Content Management 搭配使用。

- Web 應用程式必須已經建立，並且設為可內嵌在 `iframe` 之中。Sites SDK 必須已經匯入，並且在 Web 應用程式中參照。名為 “id” 的頁面 URL 參數必須已經新增至 Web 應用程式。

### 建立 Oracle Visual Builder 元件

整合啟用及應用程式和 Web 應用程式建立完成並可開始使用之後，必須為要新增至網站頁面的每個應用程式建立新元件。

- 在 Oracle Visual Builder 中，取得所發布 Web 應用程式的 URL。按一下包含該 Web 應用程式的即時專案，再按一下該 Web 應用程式。從位址列中複製 URL。
- 在 Oracle Content Management 中，按一下**開發人員**，然後按一下**檢視所有元件**。就會顯示已註冊的遠端元件與版面配置。
- 按一下**建立**並選擇**建立視覺化產生器元件**。
- 輸入元件的名稱。您不能使用其他元件或版面配置已經使用的名稱。

名稱中可以包含字母、數字、底線 (`_`) 以及連字號 (`-`)。如果您輸入空格，系統會自動以底線取代。

請勿使用下列作為樣板、主題、元件、網站、或網站頁面的名稱：`authsite`、`content`、`pages`、`scstemplate_*`、`_comps`、`_components`、`_compsdelivery`、`_idcservice`、`_sitescloud`、`_sitesclouddelivery`、`_themes`、`_themesdelivery`。您可以使用下列名稱作為網站頁面的名稱，但不可以作為樣板、主題、元件或網站的名稱：文件、網站。

- (選擇性) 輸入元件的描述。
- 貼上即時 Web 應用程式或即時傳統應用程式的 URL。
- 按一下**建立**。元件建立之後，其名稱會顯示在元件清單中。按一下清單中的元件名稱，即可瀏覽構成該元件或版面配置的資料夾和檔案。
- 若要選取一個圖示，不是指定給元件的預設圖示：
  - 從清單中選取元件。
  - 按一下**特性**。
  - 按一下**元件標誌**頁籤。
  - 按一下標誌圖庫中的某個標誌，然後按一下**完成**。

### 將「元件」新增至網站頁面

現在可以將應用程式的元件新增至您的其中一個網站頁面。您可以將元件新增至公用或安全頁面。

- 瀏覽至您要編輯的頁面，並確定  已設為**編輯**。
- 新增元件至頁面。元件會顯示在「自訂」區段中。
- 若要編輯元件及其外觀，請按一下其功能表圖示 ，然後選擇**設定值**。您可以調整元件的標籤、外觀、大小及對齊方式。請透過「樣式」頁籤使用與 Oracle Visual Builder 主機的元件相關聯的預設樣式。如需使用更多進階樣式，請在目前網站樣板之主題設計資料夾的 `design.css` 檔案中，編輯或新增樣式類別。樣式類別前置碼為 `scs-component`。

您可以在編輯網站時，以預覽模式檢視元件。網站發布之後，網站訪客就會看到針對與該元件搭配使用所選擇的 Oracle Visual Builder 應用程式，正在頁面上的 `iFrame` 中執行。

# 第 III 部分

## 發布和管理網站

此部分詳細說明如何使用、保護、改善以及發布網站。其中包含下列章節：

- [管理網站](#)
- [發布網站](#)
- [保護網站](#)
- [使用多語言網站](#)
- [使用網站重新導向或 URL 對應](#)
- [改善網站效能](#)

# 11

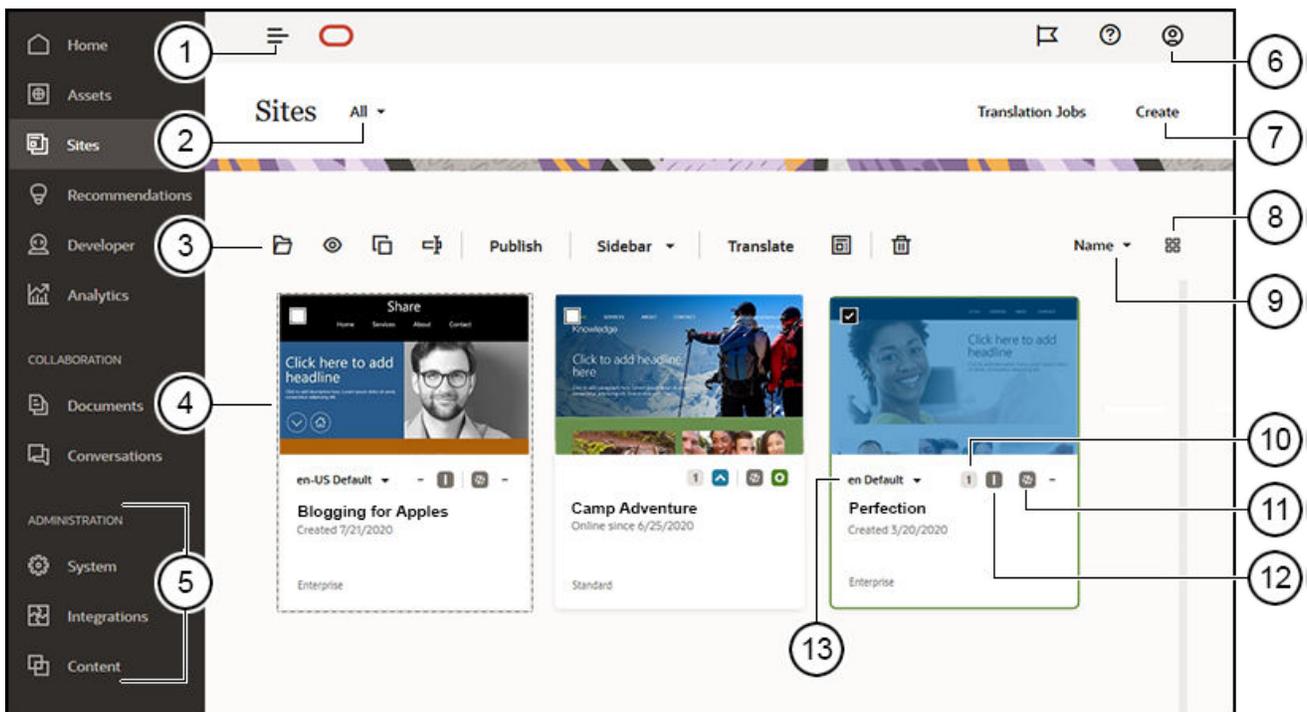
## 管理網站

若要變更網站版面配置或內容，請在編輯器中建立並開啟更新。若要建立及管理網站本身和網站的特性，請使用網站管理程式中的選項。

- [瞭解網站頁面](#)
- [管理網站和網站設定值](#)
- [讓網站上線或離線](#)
- [變更網站描述、標誌或內嵌特性](#)
- [設定搜尋引擎特性](#)
- [自訂網站設定值](#)
- [啟用 Cobrowse 整合](#)
- [新增分析追蹤功能](#)

## 瞭解網站頁面

「網站」頁面是您開始使用網站的入口。



註標	描述
1	<b>面板切換</b> 會隱藏及顯示導覽功能表。
2	<b>篩選功能表</b> 可讓您篩選網站清單，以查看您可存取的所有網站或一部分網站。如果已啟用網站治理，您便可以檢視擱置中的網站要求。
3	<p>從清單中選取網站之後，即可使用<b>動作列</b>。您可以使用動作列上的選項來開啟網站、檢視網站、變更網站狀態或重新命名網站，以及執行其他工作。</p> <p>顯示的選項取決於您對選取的網站所具備的角色。例如，如果是您建立的網站，則會具有該網站的「管理者」角色，因此可以執行所有列出的工作。但如果其他人與您共用網站並將「檢視者」角色指定給您，則您可以檢視網站特性，但無法進行任何變更。</p> <p>動作列選項也取決於網站目前的狀態。只有在網站離線時，才能使用<b>重新命名</b>和<b>刪除</b>選項。</p> <p><b>提示：</b>是否在尋找捷徑？在網站上按一下滑鼠右鍵以開啟相關內容功能表，然後選擇選項。</p>
4	<b>網站清單</b> 會顯示您所擁有或已與您共用的所有網站。此清單包括網站名稱、更新次數、網站是否安全，以及網站的狀態（線上、離線或等待更新）。
5	<b>管理功能表</b> 會在您以內容管理員身分登入時顯示。此處的選項可讓您變更服務設定值、設定整合，以及設定儲存區域和其他資產相關功能。
6	<b>使用者功能表</b> 中的選項可設定偏好設定、提供意見、存取說明，以及登出。按一下使用者圖片，即可顯示功能表選項。
7	按一下 <b>建立</b> 即可開始建立網站的程序。管理員必須先啟用網站建立功能並提供一或多個樣板，您才能建立網站。如果您在網站頁面上或樣板頁面的樣板上看不到 <b>建立</b> 選項，請洽詢服務管理員。
8	按一下 <b>檢視圖示</b> 即可選取是要以方格、清單還是表格檢視顯示網站。
9	使用 <b>排序選項</b> 可變更網站的顯示順序。
10	按一下 <b>更新數字</b> 即可編輯網站的現有更新，或為網站建立新的更新。網站若沒有任何更新，您可以開啟網站並將「網站產生器」變更為編輯模式，即可建立新的更新。
11	按一下 <b>離線</b> 、 <b>線上</b> 或 <b>重新發布</b> 圖示，即可變更網站的狀態，或發布已確認但尚未發布至即時網站的變更。
12	<b>安全性</b> 圖示會顯示網站是否要求必須登入。
13	使用 <b>語言清單</b> 即可查看網站相關本地化規定中所指定的語言。若要預覽本地化的版本，請選取語言，然後開啟網站。

## 管理網站和網站設定值

網站管理包括建立及編輯網站，以及管理網站中所使用的內容。您可以使用**網站產生器**來建立、複製及刪除網站頁面和頁面內容。若要建立、複製、共用及刪除整個網站，請使用**資料夾與檔案管理程式**。

您可以使用「網站」頁面上和「網站產生器」中之功能表列或滑鼠右鍵功能表中的選項，執行以下工作。

工作	描述
篩選網站檢視	<p>若要篩選網站清單，請在篩選功能表中選取下列其中一個選項：</p> <ul style="list-style-type: none"> <li>· <b>全部</b> — 顯示您可以存取的所有網站。如果您是網站管理員且已啟用網站治理，您便可以存取自己環境中的所有網站。</li> <li>· <b>您所擁有</b> — 顯示您擁有的所有網站。</li> <li>· <b>與您共用</b> — 顯示已與您共用的所有網站。</li> <li>· <b>線上</b> — 顯示在線上 (運作中) 的網站。</li> <li>· <b>離線</b> — 顯示離線的網站。</li> <li>· <b>要求</b> (啟用網站治理後才能使用) — 顯示擱置中的網站要求。如果您是網站管理員，將能看到系統中的所有網站要求，否則只會看到自己送出的網站要求。</li> <li>· <b>資源回收</b> — 顯示您所有擁有但已被刪除的網站。如果您是網站管理員且已啟用治理，則此時也會顯示所有已刪除的網站。</li> </ul>
建立網站	<p>若要建立網站，請按一下 <b>建立</b>。建立網站時，是利用樣板來開始進行。樣板含有您開始建置網站所需的一切項目，包括網站程式碼架構、含有範例頁面和內容的預設網站、含有樣式設定的主題、資源 (例如影像)，甚至是自訂元件。請參閱 <a href="#">建立網站</a>。</p> <p>您可以查看 <b>網站</b> 頁面上的篩選功能表，瞭解是否已啟用網站治理。如果您看到 <b>要求</b> 選項，即表示已啟用網站治理。請參閱 <a href="#">瞭解網站治理</a>。</p>
檢視和管理網站要求	<p>若要檢視擱置中的網站要求 (只有在啟用網站治理的情況下才可使用)，請在篩選功能表中選擇 <b>要求</b>。如果您是網站管理員，將能看到系統中的所有網站要求，否則只會看到自己送出的網站要求。請參閱 <a href="#">管理網站要求</a>。</p>
編輯網站	<p>若要編輯網站，請從滑鼠右鍵功能表中選擇 <b>開啟</b>，或按一下動作列中的 。請參閱 <a href="#">編輯網站</a>。</p>
討論或加註網站	<p>您可以在「網站產生器」中討論及加註網站。選取網站並從滑鼠右鍵功能表中選擇 <b>開啟</b>，或按一下動作列中的 。若要討論某個網站，請按一下  以開啟對話面板。若要加註某個網站，請按一下 。請參閱 <a href="#">瞭解網站產生器頁面</a>。</p>
檢視網站	<p>若要查看網站上線後的外觀，請選取網站並從滑鼠右鍵功能表中選擇 <b>檢視</b>，或按一下動作列中的 。</p>
發布網站	<p>若要發布網站，請選取網站，然後從滑鼠右鍵功能表或動作列中選擇 <b>發布</b>。請參閱 <a href="#">發布網站變更</a>。</p>
重新發布	<p>如果先前已發布網站，但關聯的項目或規定已經改變 (例如關聯的本地化規定已經更新以包含更多語言)，可以 <b>重新發布</b> 網站以重新整理變更。選取網站並從滑鼠右鍵功能表或動作列中選擇 <b>重新發布</b>。</p>
取消發布	<p>如果先前發布了網站，但現在想要將網站檔案自 Oracle Cloud 中的代管位置移除的話，請選取網站並從滑鼠右鍵功能表或動作列中選擇 <b>取消發布</b>。網站必須離線才能取消發布。</p>
讓網站上線或離線	<p>網站必須已經發布，才能夠上線。若要讓網站上線或離線，請選取網站並從滑鼠右鍵功能表中選擇 <b>上線</b> 或 <b>離線</b>。您也可以按一下網站磚塊中的  讓網站上線，或按一下網站磚塊中的  讓網站離線。網站在 <b>線上</b> 時，使用者可以使用 Web 瀏覽器以指定位址 (URL) 來進行檢視。網站 <b>離線</b> 時，即無法公開檢視。只能在 Oracle Content Management 中檢視該網站。請參閱 <a href="#">讓網站上線或離線</a>。</p>
重新命名網站	<p>若要重新命名網站，請對網站按一下滑鼠右鍵，然後從滑鼠右鍵功能表中選擇 <b>重新命名</b>，或按一下動作列中的 。輸入網站名稱，此名稱必須與相同伺服器上的任何現有網站名稱不同。</p> <p>如果網站處於線上狀態，則無法重新命名。若要讓網站離線，您必須是網站的網站擁有者或具有管理者角色。</p>

工作	描述
複製網站	<p>以複製網站的方式建置網站，是不錯的開始。原始網站中的所有項目 (包括主題、所有未處理更新項目、頁面、頁面內容，以及所有其他資產 (例如影像)) 都會複製到您所提供的新名稱下的新網站。您的新網站會離線，並可供編輯。</p> <p><b>注意：</b>如果啟用網站治理功能，網站可能需要先經過核准後才能建立複本。</p> <p>如果您複製的網站使用來自多個儲存區域的內容，則需要使用 <b>OCE 工具程式</b> 進行此作業。請參閱 <a href="#">使用 OCE 工具程式進行開發</a> 和 <a href="#">使用 cec 命令行公用程式</a>。</p> <p>選取想要複製的網站，然後從滑鼠右鍵功能表中選擇 <b>複製</b>，或按一下動作列中的 。請參閱 <a href="#">複製網站</a>。</p>
從網站建立樣板	<p>如果您的網站要當成其他網站的起始點，則可以從該網站建立樣板。選取網站並從滑鼠右鍵功能表中選擇 <b>建立樣板</b>，或按一下動作列中的 。請參閱 <a href="#">從網站建立樣板</a>。</p>
	<div style="border: 1px solid #0070c0; padding: 10px; background-color: #e6f2ff;"> <p> <b>備註：</b></p> <p>如果您用來建立樣板的網站使用來自多個儲存區域的內容，則需要使用 <b>OCE 工具程式</b> 進行此作業。請參閱 <a href="#">使用 OCE 工具程式進行開發</a> 和 <a href="#">使用 cec 命令行公用程式</a>。</p> </div>
刪除或回復網站	<p>如果您具有適當的權限，則可以刪除網站及其內容。當您刪除網站時，網站資料夾中您已新增至頁面的所有項目 (包括所有未處理更新項目、網站頁面、頁面內容以及資產 (例如影像)) 都會放在資源回收中。</p> <p>如果您建立了某網站 (您是網站擁有者)，或者有人與您共用網站並授予您提供者或管理者角色，您便能刪除或回復該網站。</p> <p>如果網站是線上網站，則無法刪除網站。若要讓網站離線，您必須是網站的網站擁有者或具有管理者角色。</p> <p>您也不能刪除已發布的網站。您必須先取消發布內容。如果網站有來自多個儲存區域的內容，您必須取消發布每個儲存區域中與其關聯的內容。</p> <p>若要刪除網站，請選取網站並從滑鼠右鍵功能表中選擇 <b>刪除</b>，或按一下動作列中的 。系統會提示您將網站和所有網站更新移動至資源回收。已刪除的網站仍會保留在資源回收中，直到：</p> <ul style="list-style-type: none"> <li>· 您回復網站。</li> <li>· 您永久刪除網站。</li> <li>· 達到資源回收配額。</li> <li>· 系統根據服務管理員所設定的間隔自動清空資源回收。預設值為 90 天。</li> </ul> <p>若要回復某個網站，請在「網站」功能表中，選擇 <b>資源回收</b>。從清單中選取該網站，然後選擇 <b>回復</b>。</p>

工作	描述
為網站新增成員	<p>如果您的管理員已啟用共用，您可以將您的網站與其他的 <b>Oracle Content Management</b> 使用者共用，並根據您給予他們的權限來允許他們檢視、修改或管理 <b>Oracle Content Management</b> 網站。</p> <p>任何可存取服務的人都會被視為成員。當您共用網站時，您會指定一個角色來定義成員可以或不可在您的網站進行的事項。成員只能根據您指定給他們的角色來使用您的網站。</p> <p>選取想要共用的網站，然後從滑鼠右鍵功能表中選擇<b>成員</b>，或按一下動作列中的<b>提要欄位</b>並選擇<b>成員</b>以開啟成員提要欄位。成員提要欄位開啟後，請按一下<b>新增成員</b>。</p> <p>輸入一或多個使用者名稱或電子郵件地址，並且指定以下其中一個角色：</p> <ul style="list-style-type: none"> <li>· <b>檢視者</b>：檢視者可以在編輯器中檢視網站，但無法變更任何內容。</li> <li>· <b>下載者</b>：就網站而言，下載者角色的權限與檢視者角色的權限相同，使用者可從網站建立新的樣板。</li> <li>· <b>提供者</b>：與檢視者相同，還能夠在網站離線時編輯網站、刪除網站頁面及刪除網站。</li> <li>· <b>管理者</b>：與提供者相同，還能夠新增使用者與指定其角色、發布變更至線上網站，及將網站切換成線上與離線。網站的建立者(擁有者)會自動具備管理者角色。</li> </ul>
<div style="border: 1px solid #0070C0; padding: 10px; background-color: #E6F2FF;"> <p> <b>備註：</b></p> <p>如果您共用的網站可存取多個儲存區域，只會共用預設儲存區域。任何其他儲存區域都必須個別共用。</p> </div>	
變更網站特性 (描述、標誌等)	若要變更網站描述、新增標誌、允許內嵌網站、新增虛名 URL，以及檢視網站的相關重要資訊 (包括網站 URL、網站擁有者及其他詳細資訊)，請選取網站並從滑鼠右鍵功能表中選擇 <b>特性</b> ，或按一下動作列中的 <b>提要欄位</b> 並選擇 <b>特性</b> 。請參閱 <a href="#">變更網站描述、標誌或內嵌特性</a> 。
翻譯網站	如果企業網站的相關儲存區域允許，您便可以藉由匯出網站檔案、翻譯這些檔案，然後匯入已翻譯的檔案，將企業網站翻譯成多種語言。請參閱 <a href="#">翻譯網站</a> 。
設定搜尋引擎特性	您可以在「網站產生器」中提供關鍵字和文字，以協助搜尋引擎識別網站的內容。請參閱 <a href="#">設定搜尋引擎特性</a> 。
自訂網站設定 (我的最愛圖示、控制器檔案等)	您可以在「網站產生器」中指定與不同瀏覽器和平台搭配使用的網站圖示，或是新增控制器檔案來處理連結行為。請參閱 <a href="#">自訂網站設定值</a> 。
啟用 Cobrowse 功能	您可以在「網站產生器」中啟用 <b>Oracle Cobrowse Cloud Service</b> ，這是一種協作工具，可讓您與他人共用畫面或起始 <b>Cobrowse</b> 階段作業。例如，您可以在訂購單包含此功能，讓業務代表可在客戶訂購的同時檢視客戶的畫面。請參閱 <a href="#">啟用 Cobrowse 整合</a> 。
為網站新增分析追蹤	您可以在「網站產生器」中為網站增加 <b>JavaScript</b> 追蹤程式碼片段以進行 <b>Web</b> 分析追蹤，讓其更易於與外部分析提供者 (例如 <b>Google</b> 、 <b>Adobe</b> 或 <b>Oracle Infinity</b> ) 做整合。請參閱 <a href="#">新增分析追蹤功能</a> 。
保護即時網站	<p>當您發布網站並讓它上線後，便會公開給所有使用者使用。不過，如果您是網站擁有者或具備管理者角色，則可以限制只有已註冊的使用者或一部分使用者能夠使用線上網站。若要變更網站狀態，您必須是網站擁有者或具有管理者角色。</p> <p>選取想要保護的網站，然後從滑鼠右鍵功能表中選擇<b>特性</b>，或按一下動作列中的<b>提要欄位</b>並選擇<b>特性</b>，然後按一下<b>網站安全性</b>頁籤。請參閱<a href="#">變更網站安全性</a>。</p>
新增網站重新導向	網站 URL 如果變更，重新導向可將一個 URL (來源) 轉送至另一個 URL (目標)。這有助於保留使用者書籤和搜尋引擎排名。請參閱 <a href="#">新增網站重新導向</a> 。

## 變更網站描述、標誌或內嵌特性

大多數的網站特性是在您建立或更新網站時所設定的。特性可為您提供與網站相關的重要資訊，包括網站 URL、網站擁有人和其他詳細資訊。

變更網站描述、網站標誌、新增虛名網站，以及允許內嵌網站：

1. 選取網站並從滑鼠右鍵功能表中選擇**特性**，或按一下動作列中的**提要欄位**並選擇**特性**。
2. 若要變更網站的選擇性描述，請按一下**網站特性**頁籤，以輸入或變更描述。
3. 您無法直接變更網站 URL，但可以選取及複製網站 URL，然後貼到文件、簡報與電子郵件中，以供存取網站。

如果網站在線上，請按一下  直接前往線上網站。

4. 若要讓網站能夠作為內嵌網站使用，請切換**可內嵌網站**旁邊的**是**。
5. 若要變更網站影像，請依序按一下**網站標誌**頁籤和**變更**。找出並選取您要使用的影像。影像必須是 .png、.jpeg 或 .jpg 檔案，且外觀比例應為 4:3 (長方形)。由於較小的影像可能會扭曲，較大的影像可能會影響效能，因此建議您使用大小為 300x225 像素的影像。
6. 完成時，請按一下**關閉**。

## 設定搜尋引擎特性

您可以提供關鍵字和文字，協助搜尋引擎識別網站內容。

您可以定義網站層級和頁面層級的搜尋引擎最佳化 (SEO) 設定值。網站層級設定值會擴增或覆寫下表所述個別頁面的類似設定值。

選項	網站層級	頁面層級
描述或頁面描述	提供網站本身未包含的一般網站資訊。網站中的每個頁面都會包含網站描述。 若未使用 <b>頁面設定值</b> 設定首頁的描述，此描述也會是首頁的頁面層級描述。	提供頁面本身未包含的一般頁面資訊。網站中的每個頁面都會包含頁面描述和網站描述。
關鍵字	識別套用至網站上所有頁面的術語或概念。 這些值會新增 (附加) 至指定給個別頁面的關鍵字。	識別套用至個別頁面的術語或概念。 頁面關鍵字可能適用於識別未出現在頁面文字中或出現在影像中的術語或概念。
標頭或頁面標頭	將進行分析或追蹤的命令檔或標記新增至網站。網站中的每個頁面都會包含網站標頭內容。	將進行分析或追蹤的命令檔或標記新增至網站。每個頁面都包括頁面標頭內容和網站標頭內容。
頁尾或頁面頁尾	將進行分析或追蹤的命令檔或標記新增至網站。網站中的每個頁面都會包含網站頁尾內容。	將進行分析或追蹤的命令檔或標記新增至網站。每個頁面都會包含頁面頁尾內容和網站頁尾內容。

若您在網站層級選取下列搜尋不包括的項目選項，設定會套用至所有頁面，並且會覆寫個別頁面的設定。若不是在網站層級選取此選項，則只有個別指定此選項的頁面會使用此選項。

選項	網站層級	頁面層級
在搜尋引擎中隱藏	如果選取，請對每個頁面增加 <b>NOINDEX</b> 描述標記，搜尋引擎就不會編製網站上任何頁面內容的索引。在此情況下，網站和其所有頁面都不會顯示在 <b>Web</b> 搜尋結果中。	如果選取，請對目前的頁面增加 <b>NOINDEX</b> 描述標記，搜尋引擎就不會編製此頁面內容的索引。在此情況下，個別頁面將不會顯示在 <b>Web</b> 搜尋結果中。
在搜尋引擎中隱藏頁面連結	如果選取，請對每個頁面增加 <b>NOFOLLOW</b> 描述標記，搜尋引擎就不會追蹤網站上任何頁面中的連結 (及編製目的地索引)。	如果選取，請對目前的頁面增加 <b>NOFOLLOW</b> 描述標記，搜尋引擎就不會追蹤此頁面中的連結 (及編製目的地索引)。
在搜尋引擎中隱藏頁面描述	如果選取，請對每個頁面增加 <b>NOSNIPPET</b> 描述標記，搜尋引擎就不會在搜尋結果中的頁面之後包含描述 (前面指定的描述)。	如果選取，請對目前的頁面增加 <b>NOSNIPPET</b> 描述標記，搜尋引擎就不會在搜尋結果中的頁面之後包含頁面描述 (前面指定的描述)。

變更搜尋引擎最佳化 (SEO) 設定值：

1. 開啟網站進行編輯。
2. 先按一下提要欄位中的 ，然後按一下  **SEO**。
3. 提供網站的選擇性描述。網站中的每個頁面都會包含網站描述。  
若未使用**頁面設定值**設定首頁的描述，此描述也會是首頁的頁面層級描述。
4. (選擇性) 指定以逗點區隔的關鍵字，協助搜尋引擎識別網站內容。  
網站關鍵字會識別套用至網站上所有頁面的術語或概念。這些值會新增 (附加) 至使用**頁面設定值**指定給個別頁面的關鍵字。
5. (選擇性) 將進行分析或追蹤的標頭命令檔或標記新增至網站。網站的每個頁面上都會包含標頭內容。請驗證您在標頭中使用的所有程式碼，以確定其運作正常且不會為網站帶來任何安全性風險。
6. (選擇性) 將進行分析或追蹤的頁尾命令檔或標記新增至網站。網站中的每個頁面都會包含頁尾內容。請驗證您在頁尾中使用的所有程式碼，以確定其運作正常且不會為網站帶來任何安全性風險。
7. (選擇性) 選取一或多個選項 (如上表所述) 以排除不要在搜尋結果中顯示的資訊。
8. 若要儲存目前更新項目中的所有擱置中變更，請按一下**儲存**。

## 啟用搜尋引擎最佳化的預先呈現服務

您可以讓使用 Oracle Content Management 所建置的網站預先呈現靜態 HTML 頁面，以回應搜尋蒐集程式要求。

如果啟用預先呈現服務，而且要求來自搜尋蒐集程式，則會在快取中搜尋頁面。

- 如果在快取中找到頁面，且頁面小於 15 天，則會將快取頁面提供給要求。
- 如果快取中的頁面超過 15 天，或在快取中找不到頁面，則會從 Oracle Content Management 擷取新版本、更新快取以及提供新的預先呈現頁面。
- 如果在快取或 Oracle Content Management 中找不到頁面，則會傳回錯誤頁面。

公用網站或新網站上尚未快取的過期頁面一天會呈現一次。如果已快取頁面，則不會更新快取。啟用預先呈現服務時，針對已更新和重新發布的網站，會每小時更新預先呈現快取一次。如果您對網站頁面進行變更，且要測試這些變更對最佳化的影響，則可以在網站特性中手動更新預先呈現快取。

1. 按一下側邊導覽中的**網站**。
2. 選取網站並從滑鼠右鍵功能表中選擇**特性**，或按一下動作列中的**提要欄位**並選擇**特性**。
3. 按一下特性提要欄位中的 **SEO**。
4. 便會列出上次快取重新整理的日期與時間。若要更新快取，請按一下**立即重新整理**。進度列會顯示快取狀態。

 **備註：**

如果在網站 SEO 設定值中設定搜尋引擎找不到頁面，則不會預先呈現或快取該頁面。

在 Oracle Content Management 中啟用預先呈現服務：

1. 按一下側邊導覽「管理」區域中的**系統**。
2. 選取系統設定功能表中的**網站的 SEO**。
3. 按一下**啟用**。
4. 定義所需的其他使用者代理程式。

**表格 11-1 Oracle Content Management 中的硬式編碼使用者代理程式**

使用者代理程式	使用者代理程式	使用者代理程式	使用者代理程式
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		

**表格 11-2 config.cfg 檔案的 PrerenderUserAgents 特性中其他預先設定的使用者代理程式**

使用者代理程式	使用者代理程式	使用者代理程式	使用者代理程式
AddSearchBot	AdIdxBot	AdsBot-Google	AdsBot-Google-Mobile-Apps
AppEngine-Google	Baidu-YunGuanCe	Bingbot	BingPreview
DuckDuckBot	DuckDuckGo-Favicons-Bot	endeca webcrawler	Exabot
Facebot	Feedfetcher-Google	FeedValidator	Fetch

表格 11-2 (續) config.cfg 檔案的 PrerenderUserAgents 特性中其他預先設定的使用者代理程式

使用者代理程式	使用者代理程式	使用者代理程式	使用者代理程式
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

## 設定靜態網站傳遞選項

如果貴公司使用編譯網站，那麼您可以控制靜態網站的快取時間長度，以及網站編譯支援哪一個調適型行動版面配置的行動使用者代理程式呼叫。

- [啟用發布時自動編譯](#)
- [覆寫編譯網站的預設快取控制標頭](#)
- [指定行動使用者代理程式以支援經過編譯的調適型版面配置](#)

### 啟用發布時自動編譯

Oracle Content Management 包含內建的編譯服務，可讓您不需要進行任何額外組態即可輕鬆編譯網站。或者您的服務管理員可以為測試用途手動設定編譯服務，以驗證網站編譯或使用自訂程式庫。無論使用的編譯服務為何，您都可以選擇在發布或重新發布網站時編譯網站。

1. 登入 Oracle Content Management 後，請按一下側邊導覽功能表中的**網站**。
2. 選取想要修改的網站，然後按一下**特性**。
3. 按一下特性對話方塊中的**靜態傳遞**。
4. 在**啟用自動編譯**區段中，啟用**發布後編譯網站**。
5. 完成之後，請按一下**完成**。

當您發布或重新發布已啟用自動編譯的網站時，會在**網站**頁面的網站磚塊中追蹤並顯示發布狀態。發布完成之後，會在網站特性對話方塊的**靜態傳遞**區段中追蹤編譯處理作業。編譯完成之後，網站特性靜態傳遞區段便會列出上次編譯的日期和時間，並提供下載編譯日誌的連結。

### 覆寫編譯網站的預設快取控制標頭

預設會在使用者的瀏覽器快取編譯網站 300 秒 (5 分鐘)。您的服務管理員可以變更此預設值，但如果您是網站開發人員，您就可以在網站特性中覆寫特定網站的預設值。

1. 登入 Oracle Content Management 後，請按一下側邊導覽功能表中的**網站**。
2. 選取想要修改的網站，然後按一下**特性**。
3. 按一下特性對話方塊中的**靜態傳遞**。
4. 在**快取回應標頭**區段中先輸入 Cache-control: max-age=，然後輸入一個數值，代表您希望在使用者瀏覽器中快取頁面的秒數。例如，Cache-control: max-age=600 會快取頁面 10 分鐘。
5. 完成之後，請按一下**完成**。

假如執行處理使用 Akamai，為了繼續使用現有的 Akamai 設定值，請將**快取回應標頭**區段留白。若要覆寫現有 Akamai 設定值，請輸入 Edge-Control: !no-store,max-age=1800,downstream-ttl=1800，其中的粗體項目為預設值 (秒)。

- !no-store 指示此設定應覆寫特性的相對應 Akamai 快取組態。
- max-age 決定 Akamai 可快取此頁面的時間長度。預設值為 1800 秒 (30 分鐘)。在這一段時間，Akamai 將會完成快取此頁面的要求，但不會要求快取 Oracle Content Management 中的頁面。
- downstream-ttl 會通知 Akamai 傳送 "Cache-Control: max-age" 標頭及其回應給從屬端瀏覽器，以指示這些瀏覽器在配置的時間內快取頁面。預設值為 1800 秒 (30 分鐘)。

## 指定行動使用者代理程式以支援經過編譯的調適型版面配置

編譯網站時，系統可以建立行動頁面來支援調適型版面配置。您可以指定讓伺服器傳遞行動頁面的使用者代理程式，而非網站的標準經過編譯頁面。比對瀏覽器所傳送的使用者代理程式標頭時，會將此處輸入的值視為不區分大小寫的子字串。請注意，行動裝置上的瀏覽器通常會在使用者代理程式字串中使用關鍵字 Mobile。

1. 登入 Oracle Content Management 後，請按一下側邊導覽功能表中的**網站**。
2. 選取想要修改的網站，然後按一下**特性**。
3. 按一下特性對話方塊中的**靜態傳遞**。
4. 在**行動使用者代理程式**區段中，對您希望提供行動頁面的使用者代理程式輸入使用者代理程式子字串清單 (使用逗號區隔)。子字串中若有任何部分符合瀏覽器的使用者代理程式字串，系統就會提供行動頁面。

## 指定並設定虛名 URL

設定網站的虛名 URL，可讓使用者在指定網域上存取 Oracle Content Management 網站，而非 Oracle Content Management 執行處理網域。例如，執行處理 URL 可能是 `https://myinstance.cec.ocp.oraclecloud.com/site/MyCustomerSite/`，但 `https://www.example.com` 這種易記 URL 更容易記住、更有利於打造品牌，而且通常也比較容易使用。而且，也可以視需要使用自訂路徑來代管 Oracle Content Management 網站，例如 `https://www.example.com/store/`

若要使用虛名網域，需要執行幾個步驟。

- 使用內容傳遞網路 (CDN)。

- 使用網域名稱系統 (DNS) 管理虛名網域，以便將網域正規名稱 (CNAME) 記錄對應至 CDN。
- 在 CDN 上部署有效的憑證以保護虛名網域。
- 設定 Oracle Content Management 網站接受虛名網域上的要求。
- 設定 CDN 將要求適當地遞送至 Oracle Content Management。

### 使用內容傳遞網路

內容傳遞網路 (CDN) 是一種全球分散式伺服器平台，可改善網站效能和安全性。CDN 可將使用者與伺服器之間的距離縮到最短，同時最佳化內容要求效能。雖然 CDN 的主要目標是改善使用者體驗，但也可以用來更改傳輸中的要求，讓使用者看到的內容更為簡潔，即使幕後處理作業並非如此。

為了支援在虛名網域上代管 Oracle Content Management 網站，您需要使用 CDN 將它設定成處理所有來自所設定虛名網域的要求，並將它們適當地遞送回 Oracle Content Management，然後更改要求，讓 Oracle Content Management 適當地處理和保護它們。

### 使用網域名稱系統 (DNS) 管理網域

任何網域都可作為 Oracle Content Management 網站的虛名網域。您必須控制任何作為虛名網域的網域，再將其設定為與 Oracle Content Management 網站搭配使用。

基於 DNS 的限制，若要使用不含 *www* 或其他子網域 (例如 *store.example.com*) 的根網域 (例如 *example.com*) 幾乎是不可能的。請洽詢 DNS 和 CDN 提供者，以瞭解是否可以使用根網域。

由於 DNS 作用於網域層級而非路徑層級，所以為了讓 Oracle Content Management 代管您網域的一些路徑，並讓其他服務代管其他路徑，將需要透過 CDN 來處理遞送。DNS 只是用來區隔網域和子網域層級的流量。

### 部署憑證

憑證可用來保護需要由 CDN 建立和代管的虛名網域。憑證可以保護單一網域、多個網域和子網域，或 *\*.example.com* 這類萬用字元子網域。虛名網域可以接受任何組合。所有受保護網域都會顯示在憑證詳細資訊中，因此如果不想公開分享這些詳細資訊，則應該使用個別的憑證。

#### 備註：

憑證建立和代管作業通常是 CDN 特有的處理，而且需要指定最佳作法。

### 設定虛名網域的 Oracle Content Management 網站

若要在使用虛名網域時適當地載入 Oracle Content Management 網站，您必須先讓 Oracle Content Management 知道哪個網站 URL 正在使用虛名網域。此作業是在 Oracle Content Management 的網站特性頁籤中完成。

1. 在 Oracle Content Management 中，按一下側邊導覽中的**網站**。
2. 選取要與虛名網域搭配使用的網站，然後從滑鼠右鍵功能表中選擇**特性**，或選擇動作列中的**提要欄位**。
3. 在虛名網域欄位中輸入虛名網域，然後按一下**儲存**。

 **備註：**

Oracle Content Management 大約需要一個小時以上的時間才能準備就緒以接受網域要求。請隨時前往網站特性面板檢查進度。

### 設定內容傳遞網路來遞送要求

Oracle Content Management 適當地設定並準備好接受它們之後，會透過 DNS 將對虛名網域提出的要求遞送至 CDN，而且 CDN 必須將要求適當地轉遞至 Oracle Content Management。例如，如果 Oracle Content Management 網站的網站 URL 為 `https://myinstance.cec.ocp.oraclecloud.com/site/MyCustomerSite/`，且虛名網域設定為 `https://www.example.com/store`，則必須對 CDN 進行下列設定：

- 辨識虛名網域：`https://www.example.com/store`
- 使用下列虛名網域來識別原始 Oracle Content Management 執行處理：`https://myinstance.cec.ocp.oraclecloud.com/`
- 附加特定網站的網站路徑：`site/MyCustomerSite/`
- 並將完整網站 URL 傳送至原始 Oracle Content Management 執行處理：`https://myinstance.cec.ocp.oraclecloud.com/site/MyCustomerSite/`
- Oracle Content Management 會接收要求並回應至 CDN，以滿足對使用者瀏覽器的要求，並且只會向使用者顯示虛名網域：`https://www.example.com/store`

CDN 組態步驟通常僅適用於 CDN，因此請與 CDN 提供者合作適當地設定原始定義，以便讓 CDN 知道如何到達原始 Oracle Content Management 執行處理，以及附加正確網站路徑的網站路徑。

 **備註：**

要求不得包含下列任一路徑：

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

對這些路徑的要求不應包括網站路徑。它們應該解析為要適當處理之原始 Oracle Content Management 執行處理的根。

## 啟用 Cobrowse 整合

Cobrowse 功能是一種搭配 Oracle Cobrowse Cloud Service 使用的協作工具。

與 Cobrowse Cloud Service 整合時，必須先以服務管理員已接受的整合方式新增。請參閱 *Integrating and Extending Oracle Content Management* 中的 *Integrate with Oracle Cobrowse Cloud Service*。

啟用 Oracle Cobrowse Cloud Service 整合之後，就可以針對網站設定功能，然後新增到特定網站頁面以供使用。

對網站啟用 Cobrowse 功能：

1. 開啟網站進行編輯。
2. 按一下提要欄位中的 ，然後按一下 。
3. 在 Cobrowse 區段中，選取**在此網站啟用 Oracle Cobrowse**。
4. 輸入此網站的啟動器命令檔。Oracle Cobrowse Cloud Service 管理員可以存取 Cobrowse 管理主控台，以取得適合的啟動器 Javascript 程式碼片段。啟動器有兩種不同的類型。
  - Launch Point 1：會自動在頁面上增加一個 Cobrowse 按鈕。
  - Launch Point 2：可供您自訂在頁面上增加的按鈕和介面。
5. 按一下**關閉**。

對網站啟用 Cobrowse 之後，您就可以將它加到頁面或自訂其在頁面上的使用方式。請參閱在 [頁面上使用 Cobrowse 功能](#) 以瞭解詳細資訊。

### 在安全網站和網站產生器測試功能使用 Cobrowse

Cobrowse 有兩種模式：即時 (ICB) 和進階 (ACB)。在安全網站以 ICB 模式使用 Cobrowse 功能或使用 Cobrowse 功能預覽仍在開發中的網站，需要進行額外的組態設定。這項組態設定需要在 Cobrowse 管理主控台進行。請參閱 [Using Standalone Cobrowse](#) 中的 *Log in to the Agent Console*。

在「自訂函數」欄位中新增一個函數，讓代理程式能夠查看已發布之安全網站上的受密碼保護資源：

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

若要在已發布的安全網站使用此函數，以及使用此函數來檢視/預覽開發中的網站，請增加以下的額外程式碼：

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

```
}
}
```

設定受保護的資源是一項新的 Cobrowse 功能。它使用與 Cobrowse Page Masking 相同的萬用字元 URL 樣式。如需更多詳細資訊，請參閱 *Cobrowse Deployment and Use Guide* 中的 [Configure page masking](#)。

## 新增分析追蹤功能

您可以在「網站產生器」中為網站或頁面增加 JavaScript 追蹤程式碼片段以進行 Web 分析追蹤，讓其更易於與外部分析提供者 (例如 Google、Adobe 或 Oracle Infinity) 做整合。

新增網站的分析追蹤功能：

1. 開啟網站進行編輯。
2. 先按一下提要欄位中的 ，然後按一下  **分析**。
3. 按一下此開關以啟用分析追蹤功能。
4. 在 **JavaScript 追蹤程式碼片段** 方塊中，新增程式碼片段或編輯現有命令檔。您的管理員有可能已提供適合您環境的程式碼片段。若有的話，方塊中將會顯示該程式碼片段。您可以自訂此命令檔，或新增自己的命令檔。如果您編輯管理員提供的程式碼片段，系統將會顯示該命令檔已遭修改訊息。若要移除自訂項目：**回復為最新的租用戶命令檔**。

以下是 Google Analytics (分析) 追蹤程式碼片段範例：

```
<!-- 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>
```

您必須儲存並發布此變更，若有必要的話，請先讓網站上線後再收集該網站的分析資料。

### 檢視分析資料

發布網站並讓它上線之後，您便可以在廠商的網站 (例如 Google Analytics (分析)) 上檢視所追蹤的分析資料。如果您已使用程式碼片段來進行 Oracle Infinity 分析追蹤，請前往 Oracle Infinity 首頁，然後按一下 **分析** 來檢視資料及選取或建立報表。

# 12

## 發布網站

瞭解讓網站上線、關閉網站以及發布網站變更的相關資訊。

- [讓網站上線或離線](#)
- [發布網站變更](#)

### 讓網站上線或離線

當網站在**線上**時，使用者只要具備適當的存取權，便可以使用標準 Web 瀏覽器在指定的位址 (URL) 檢視網站。網站**離線**時，即無法公開檢視。您只能在 **Oracle Content Management** 中檢視網站。

網站必須已經發布，才能夠上線。若要發布網站，請參閱[發布網站變更](#)。

右側的狀態圖示主要顯示網站目前為上線或離線狀態：

- 如果網站之前從未發布過，您會見到破折號 (-)。
- 如果網站為線上狀態，您會見到 。
- 如果網站為離線狀態，您會見到 。

如需瞭解哪些使用者可以存取線上或離線網站，請參閱[瞭解網站安全性](#)。

若要變更網站的狀態，您必須是網站擁有人或具備管理者角色，或者，如果已啟用網站治理，不管是否為與其共用的網站，網站管理員皆可變更所有網站的狀態。

1. 在「網站」頁面上，從清單中選取網站。
2. 若要讓網站上線或離線，請從滑鼠右鍵功能表中選擇**上線**或**離線**。您也可以按一下網站磚塊中的  讓網站上線，或按一下網站磚塊中的  讓網站離線。  
系統會提示您確認選擇。

當您讓網站上線時，會建立網站的完整呈現 HTML 版本，並將其複製到 **Oracle Cloud** 中的代管位置。線上網站會在網站名稱下面顯示其 URL。預設 URL 的格式為：

```
https://service_name.identity_domain.sites.oraclecloud.com/site_name
```

將網站離線時，會移除 **Oracle Cloud** 中代管位置的網站和其資料夾及檔案。

### 發布網站變更

您必須是網站擁有人或具備管理者角色，才能夠發布網站變更。

發布網站變更時請記住以下事項：

- 將變更發布至已上線的網站時，所有可以存取網站的人員都會立即見到確認的變更。

- 發布網站變更時，可以選擇發布定位至該網站通道的所有資產，還是只發布網站頁面上使用的資產 (包括建議)。例如，如果發布定位至網站通道的所有資產，而該網站所包含清單參照不直接屬於網站的內容項目，參照的資產還是一併會被發布。
  - 將變更發布至多語言網站時，只會發布該網站相關聯本地化規定中所定義語言的翻譯內容。
  - 如果先前曾發布某個語言的翻譯內容，但現在網站相關聯本地化規定中已不再定義該語言，翻譯的內容將會自發布的網站中移除。
  - 如果自多語言網站中刪除某個頁面，發布網站時也會刪除翻譯的頁面。
1. 若要發布網站變更，請選取網站，然後從滑鼠右鍵功能表或動作列中選擇**發布**。
  2. 選取是要發布網站和定位至網站發布通道的所有資產，還是要發布網站和新增至網站頁面的所有資產。
  3. **Oracle Content Management** 會驗證所有網站字串和資產是否都已獲得必要的核准與翻譯。如果有下列情況，將無法發布網站：沒有網站相關聯本地化規定中所要求的翻譯內容、未將資產標示為已翻譯，或資產需要核准但未標示為已核准。展開項目可查看更多詳細資訊。如果有問題，請加以更正，然後嘗試再發布一次。如果所有項目都有效，請按一下**發布**。系統會提示您確認選擇。
  4. 如果網站還沒上線 ()，需要讓網站上線，使用者才能使用。

# 13

## 保護網站

發布網站並讓網站上線之後，您可以控制哪些人員可存取網站。

如果您想要保護網站，您需要根據指定的角色，指定可以存取已發布 (線上) 網站的使用者群組。這些角色都是服務執行處理的管理員所指定的全服務角色。

- [瞭解網站安全性](#)
- [變更網站安全性](#)

管理員可以採取與網站安全性相關的額外動作，例如啟用自訂登入、允許共用網站和主題、限制網站、樣板和元件建立、啟用治理，以及其他動作。請參閱 *管理 Oracle Content Management* 中的設定網站和資產設定值。

## 瞭解網站安全性

您可以套用安全性來控制哪些使用者能夠見到已發布 (線上) 網站、能夠見到網站上的安全內容並與其互動，以及能夠見到並編輯未發布 (離線) 網站。

### 網站安全性

發布網站並讓網站上線之後，您可以控制哪些人員可存取網站。視系統和網站管理員所做的環境設定而定，您可以將網站設為公開給所有人使用、限制為僅供註冊的使用者使用，或限制為僅供特定使用者使用。

您必須是網站擁有者或具備管理者角色，才能變更網站安全性或任何其他設定。若要變更登入要求，網站必須離線。不過，若要變更指定的使用者或使用者角色，網站可為線上狀態。將網站離線時，會移除 Oracle Cloud 中代管位置的網站和其資料夾及檔案。

可用的安全性選項可能受到樣板規定限制 (如果已啟用網站治理) 或受到租用戶規定限制 (如果已停用網站治理)。請參閱[瞭解網站治理](#)。

如果您想要保護網站，您需要根據指定的角色，指定可以存取已發布 (線上) 網站的使用者群組。這些角色都是服務執行處理的管理員所指定的全服務角色。

- **雲端使用者**：認證的使用者可透過使用者名稱和密碼登入服務執行處理。這包括具備或不具備**訪客**角色或**使用者**角色的所有已認證使用者。
- **訪客**：只有具備此角色的使用者才能存取網站。例如，可以將此角色授予能夠看見已發布網站，但無法存取此 Oracle Content Management 執行處理中資料夾和檔案的使用者。

### 備註：

這不包括具備**使用者**角色的使用者，除非他們是網站擁有者或網站已明確分享給他們共用。

- **服務使用者**：只有具備此角色的使用者才能存取此網站。例如，可以將此角色授予能夠見到已發布網站，並可存取此 Oracle Content Management 執行處理之資料夾和檔案的使用者。

- **特定使用者：**只有您新增為網站成員的使用者才能看見已發布的網站。

### 網站共用功能

您可以利用 *網站共用功能*，指定能夠存取您的未發布 (離線) 網站的個別使用者，並且依據您給予他們的權限，允許他們檢視、修改或管理網站。若您是網站擁有者，或是已與您共用網站且您具備「管理者」角色，您就能共用網站。

#### 備註：

您指定給使用者的任何共用角色會提升其安全性角色。例如，若使用者具備**訪客**角色，而您與他們共用網站並授予其提供者角色，他們就能夠修改離線網站，但其他具備**訪客**角色的使用者則只能檢視線上網站。

- **檢視者：**檢視者可以在編輯器中檢視網站，但無法變更任何內容。
- **下載者：**就網站而言，此角色的權限與檢視者角色的權限相同。
- **提供者：**與檢視者相同，還能夠在網站離線時編輯網站、刪除網站頁面及刪除網站。
- **管理者：**與提供者相同，還能夠新增使用者與指定其角色、發布變更至線上網站，及將網站切換成線上與離線。網站的建立者 (擁有者) 會自動具備管理者角色。

建立網站時，會以該網站名稱建立一個通道。若要與其他人共用通道，您必須共用此網站並至少將提供者角色授予某人，以便該人員能夠使用該通道發布資產。使用者必須具備「管理者」角色才能夠發布網站。

### 元件共用

某些元件可以存取共用資源，例如資料夾、檔案或對話。*元件共用*需顧及到網站安全性 (哪些使用者可以檢視已發布網站) 和資源共用 (哪些使用者可以檢視及使用資料夾、檔案和對話)。

例如，如果您將文件管理程式元件新增至您的網站，網站的所有訪客都將能夠見到資料夾的內容，而且依據其角色和其他權限的不同，他們可能還能夠新增、修改或刪除資料夾的內容。

一般準則：

- 網站作者不能授予超出本身所具備之存取權的資料夾存取權。例如，若作者具備資料夾的下載者存取權，他就不能授予網站訪客提供者權限。
- 對元件設定的權限可以提升訪客的權限。例如，若訪客具備資料夾的檢視者權限 (或無權限)，文件管理程式元件可以根據元件中選取的角色授予更高的權限。這些提高的權限只對元件本身有效。
- 如果網站訪客的權限高於針對元件指定的權限，訪客的個別權限將會覆寫針對元件設定的權限。
- 資料夾的權限會套用至該資料夾內的資料夾與檔案。

對於公用網站：

- 唯有安全網站支援對話元件。
- 文件管理程式元件預設會授予所有訪客關聯資料夾的下載者權限。您可以依上列準則變更角色，而且可以使用元件本身的設定值限制使用者所能見到的選項。

- 資料夾清單元件和檔案清單元件的所有使用者都具備下載者存取權。使用者不管具備什麼角色，都能夠檢視及下載檔案。

### 安全網站 URL

當您讓網站上線時，會建立網站的完整呈現 HTML 版本，並將其複製到 Oracle Cloud 中的代管位置。線上網站會在網站名稱下面顯示其 URL。

不安全網站的預設 URL 格式為：

```
https://service_name.identity_domain.sites.oraclecloud.com/site_name
```

安全網站的預設 URL 格式為：

```
https://service_name.identity_domain.sites.oraclecloud.com/authsite/site_name
```

請注意 URL 中新增了 authsite。

## 變更網站安全性

發布網站並讓網站上線之後，您可以控制哪些人員可存取網站。視系統和網站管理員所做的環境設定而定，您可以將網站設為公開給所有人使用、限制為僅供註冊的使用者使用，或限制為僅供特定使用者使用。

您必須是網站擁有者或具備管理者角色，才能變更網站安全性或任何其他設定。若要變更登入要求，網站必須離線。不過，若要變更指定的使用者或使用者角色，網站可為線上狀態。將網站離線時，會移除 Oracle Cloud 中代管位置的網站和其資料夾及檔案。若要讓網站離線，請從

「網站」頁面中選取網站，然後從滑鼠右鍵功能表中選擇**離線**，或按一下動作列中的 。系統會提示您確認選擇。

變更網站安全性：

1. 在「網站」頁面中選取網站，然後按一下滑鼠右鍵功能表或動作列中的**特性**。
2. 按一下**網站安全性**頁籤。

網站安全性頁籤上的可用選項取決於下列因素：系統管理員所做的網站安全性設定、是否已啟用網站治理，以及網站管理員在此網站的基礎樣板中所做的安全性設定。請參閱[開始建立網站](#)和[瞭解網站治理](#)。

3. 若已註冊的使用者需要登入才能見到線上網站，請按一下「需要登入」旁邊的**是**。若要移除登入要求，並讓網站上線後可公開使用，請按一下「需要登入」旁邊的**否**。
4. 選取哪些已註冊的使用者群組可以存取線上網站。若要選取個別群組，請先將**雲端使用者**取消選取。
  - **雲端使用者**：只有認證的使用者才能存取此網站。認證的使用者可透過使用者名稱和密碼登入您的網域。這包括擁有 **Oracle Content Management Cloud 訪客**角色或 **Oracle Content Management Cloud 使用者**角色的使用者。
  - **訪客**：唯有具備此角色的使用者可以存取此網站。這不包括擁有 **Oracle Content Management Cloud 使用者**角色的使用者。
  - **服務使用者**：只有可登入此 Oracle Content Management 執行處理的使用者才能存取此網站。

- **特定使用者**：指定可以存取此網站的個人。按一下**新增成員**。在搜尋欄位中輸入使用者名稱或部分的使用者名稱。從顯示的清單中選取使用者，並重複以上動作來新增其他使用者。完成後，請按一下**新增**。若要移除使用者，請在該使用者名稱下方的功能表中按一下**移除**。
5. 按一下**儲存**以儲存變更並關閉視窗。  
網站會顯示為離線，而且需要登入。
  6. 若要讓網站上線，請從滑鼠右鍵功能表中選擇**上線**，或按一下動作列中的 。按一下**確認繼續進行**，然後按一下**確定**。

當您讓網站上線時，會建立網站的完整呈現 HTML 版本，並將其複製到 Oracle Cloud 中的代管位置。線上網站會在網站名稱下面顯示其 URL。

不安全網站的預設 URL 格式為：

```
https://service_name-identity_domain.cec.ocp.oraclecloud.com/site/  
site_name
```

安全網站的預設 URL 格式為：

```
https://service_name-identity_domain.cec.ocp.oraclecloud.com/site/  
authsite/site_name
```

請注意 URL 中新增了 authsite。

您可以新增登出 URL，並將它實作為顯示在功能表上的連結、按鈕或頁面。請參閱[段落](#)、[按鈕](#)以及[新增頁面](#)。

登出 URL 的格式為：

```
https://service_name-identity_domain.cec.ocp.oraclecloud.com/cloudgate/  
logout.html?postlogouturl=%2Fsite%2Fauthsite%2Fsite_name
```

 **備註：**

postlogouturl 必須採用上面的編碼格式。

# 14

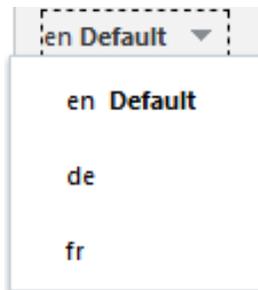
## 使用多語言網站

瞭解目標的多語言體驗網站翻譯。

- [多語言網站總覽](#)
- [翻譯網站](#)
- [管理網站翻譯工作](#)

### 多語言網站總覽

您可以將網站翻譯成本地化規定中指定的任何語言。若要查看網站相關本地化規定中指定的語言，請在網站標題中，按一下語言旁邊的向下箭號。若要預覽本地化的版本，請選取語言，然後開啟網站。



#### 備註：

您無法使用 Oracle Content Management 入門版建立多個網站。如需完整功能集和無限制網站，請升級至 Oracle Content Management 進階版。

當您選取網站以供翻譯時，系統會建立網站檔案的 .zip 檔案。

#### 備註：

如果網站包含多個儲存區域的資產，則 .zip 只會包含來自預設儲存區域的資產。

此 .zip 檔包含下列檔案和資料夾：

- assets 資料夾 — 您選取翻譯整個網站或只翻譯目標資產時，才会有此資料夾。
- root 資料夾

- \* 每個內容項目各有一個 `<contentItem_ID>.json` 檔案 — 其中包含內容項目的名稱、描述、內容項目中欄位的可翻譯字串，以及其他不應該編輯的相關資訊。
- `job.json` — 描述翻譯工作的檔案。請勿編輯此檔案。
- `site` 資料夾 — 您選取翻譯整個網站或只翻譯網站內容時，才有此資料夾。
  - `root` 資料夾
    - \* 每個網站頁面各有一個 `<page_ID>.json` 檔案 — 其中包含頁面的名稱、標題、描述、關鍵字、頁首、頁尾、頁面之元件的可翻譯字串，以及其他不應該編輯的相關資訊。如果您有自訂的元件，則它們可能已設定成使用可翻譯的字串。請參閱[開發多語言網站適用的可翻譯元件](#)。
    - \* `siteinfo.json` — 其中包含描述、關鍵字、頁首以及頁尾。
    - \* `structure.json` — 其中包含導覽和網站結構。
  - `job.json` — 描述翻譯工作的檔案。請勿編輯此檔案。

## 翻譯網站

您在已翻譯網站頁面上的資產 (不論是直接在該頁面上還是在內容清單中) 若已經翻譯的話，系統將會以與網站相同的語言顯示這些資產。您也可以將內容項目從網站獨立出來分開翻譯；請參閱本地化內容項目。

如果翻譯的網站包含來自多個儲存區域的資產，翻譯工作中只會包含來自預設儲存區域的資產

### 備註：

您無法使用 Oracle Content Management 入門版來處理翻譯。如需完整功能集和無限制網站，請升級至 Oracle Content Management 進階版。

### 建立翻譯工作

不論是匯出翻譯套件進行人工翻譯，還是使用翻譯連線器，您都需要建立翻譯工作。

1. 選取您想要翻譯的網站，然後按一下**翻譯**。您可能需要按一下**其他**，才能見到**翻譯**選項。
2. 輸入您要匯出之網站檔案的資訊 — 翻譯工作，然後按一下**建立**：
  - 輸入翻譯工作的名稱。
  - 選取網站的目標翻譯語言。
  - 選取是要匯出整個網站 (網站內容 (頁面、結構與網站資訊) 加目標資產)，還是只匯出網站內容或只匯出目標資產。
  - 選取翻譯連線器，或選擇匯出翻譯套件以進行人工翻譯。
  - 完成後，請按一下**建立**。

翻譯工作完成之後，經過翻譯的套件便會匯入 Oracle Content Management。

### 手動翻譯來源語言檔案

若是手動翻譯內容，當翻譯工作完成之後，您需要下載網站檔案的 .zip 檔案。

1. 按一下橫幅中的**翻譯工作**，選取翻譯工作，然後按一下**下載**。
2. 針對您選取作為目標語言的每個語言，在 .zip 檔案中與 root 資料夾相同的層級建立資料夾，例如 de、es 及 fr。您可以翻譯一部分選取的語言。例如，可以現在先翻譯成德文 (de)，稍後再翻譯成西班牙文 (es) 和法文 (fr)。
3. 將**所有** .json 檔案從 root 資料夾複製到每個語言資料夾。
4. 將所有 .json 檔案中的字串翻譯成適當的語言。請勿刪除 .json 檔案中的任何字串，也不要將檔案重新命名。
5. 壓縮 assets (如果翻譯工作包含資產) 和 site (如果翻譯工作包含網站內容) 資料夾，包括 job.json、root，和內有翻譯檔案的所有語言資料夾。

### 匯入手動翻譯的套件

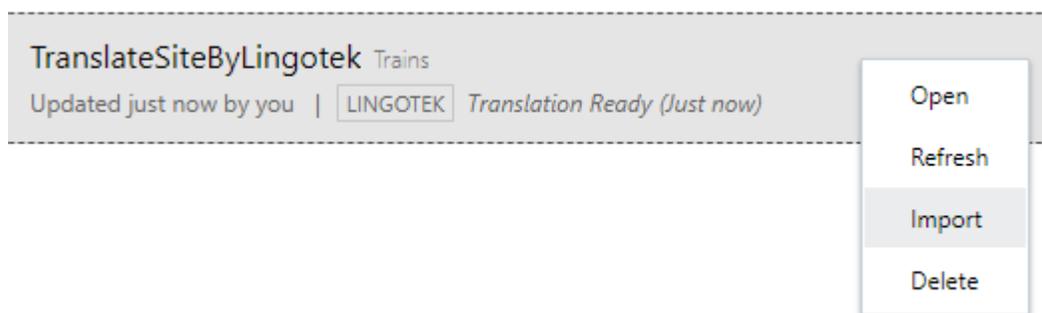
不論是手動翻譯還是自動翻譯，您都需要匯入完成的翻譯套件。

1. 在**翻譯工作**頁面中，按一下**匯入**。
2. 按一下**上傳**，選取所翻譯網站檔案的 .zip 檔案，然後按一下**開啟**。
3. 上傳完成後，請按一下**確定**。
4. Oracle Content Management 會驗證 .zip 檔案中是否有工作中所定義的翻譯。若要查看翻譯工作所包含的頁面和資產，請按一下對話方塊中的連結。
5. 準備好匯入翻譯之後，請按一下**匯入**。  
匯入的狀態會顯示在橫幅上方。您可以按一下**詳細資訊**來檢視工作的詳細資訊。

### 從翻譯連線器匯入翻譯的套件

不論是手動翻譯還是自動翻譯，您都需要匯入完成的翻譯套件。

1. 按一下 ●●●，然後選取**翻譯工作**。
2. 對完成的翻譯工作按一下滑鼠右鍵並選取**匯入**。



3. Oracle Content Management 會驗證工作中定義的翻譯。若要查看翻譯工作所包含的頁面和資產，請按一下對話方塊中的連結。
  4. 準備好匯入翻譯之後，請按一下**匯入**。  
匯入的狀態會顯示在橫幅上方。您可以按一下**詳細資訊**來檢視工作的詳細資訊。
- 匯入翻譯之後，您需要發布網站並確定網站已上線，然後才能在網站上提供翻譯。

匯入翻譯套件之後，若對翻譯服務的翻譯進行任何變更的話，可以按一下**重新整理**來更新翻譯。

如果您在翻譯後變更網站，則必須翻譯所有新的或已編輯的字串。在「網站產生器」中編輯網站時，所編輯的是網站的預設語言版本。您所做的任何網站結構變更 (例如新增元件或重新排列頁面)，都會複製到網站的本地化版本中。您可以接著建立新的翻譯工作來翻譯已更新的字串。

## 管理網站翻譯工作

當您選取網站以供翻譯時，系統會建立一個**翻譯工作**。接著，您便可以下載要翻譯的檔案、翻譯這些檔案，然後匯入已翻譯的檔案。

若要建立翻譯工作，請參閱**翻譯網站**。

「翻譯工作」頁面會列出所有翻譯工作與其狀態：

-  (就緒) - 網站檔案的 .zip 檔案已可供下載。
-  (進行中) - .zip 檔案已經下載。狀態會維持進行中，直到所有目標語言的全部翻譯都已順利匯入為止。
-  (完成) - 此工作之所有目標語言的翻譯都已順利匯入。
-  (失敗) - 翻譯工作失敗。您應該已在橫幅上方看到和工作失敗原因有關的失敗訊息。如果您需要再看一次訊息，可以**重新送出**翻譯工作。請更正問題，然後**重新送出**工作。

您可以執行下列動作：

- 開啟翻譯工作以檢視詳細資訊。詳細資訊包括來源語言和所有選取的目標語言，以及這些翻譯的狀態。
- 若要下載網站檔案的 .zip 檔案，請選取工作，然後按一下**下載**。
- 選取工作並按一下**刪除**，以刪除某个工作。
- 若要匯入翻譯，請依序按一下**匯入**和**上傳**，選取所翻譯網站檔案的 .zip 檔案，然後按一下**確定**。  
Oracle Content Management 會驗證 .zip 檔案中是否有工作中所定義的全部翻譯。若要查看翻譯工作所包含的網站頁面和資產，請按一下對話方塊中的連結。準備好匯入翻譯之後，請按一下**匯入**。

## 翻譯的地區設定

送出項目進行翻譯後，系統會透過代碼識別目標語言，讓語言服務提供者知道該項目要翻譯成哪一種語言以傳回。例如，**fr** 代表法文，**de** 代表德文。

這些代碼可擴充到其他地域性方言。例如，**de-LI** 是在列支敦斯登使用之德文的代碼，**de-LU** 是在盧森堡使用之德文的代碼。但如果是語言服務提供者不支援的地域性方言，則會將提供的代碼截斷成兩個字元的基本語言。例如，如果是 **de-LI** 和 **de-LU**，代碼會截斷成 **de**。

如果語言服務提供者支援其中一種而非全部的地域性方言，它可能會採取替代的方式。舉例來說，**ms-BN** 是在汶萊使用之馬來文的代碼，但語言服務提供者如果不支援該方言，就會切換成它支援的方言，例如 **ms-MY**，這是在馬來西亞使用之馬來文的代碼。

如果語言服務提供者未在各方言之間做區分 (例如 **en-BZ** 代表在貝里斯所使用的英文，**en-JM** 代表在牙買加所使用的英文)，則代碼將會截斷成基本語言，在此案例中，**en** 代表英文。

## 翻譯的自訂地區設定

開發人員可根據您組織的需求，建立自訂地區設定。自訂地區設定代碼包含基本語言、任何地域性方言代碼 (如果有的話)、一個用於指定其為自訂地區設定的 **x**，以及您組織所需的任何其他可識別自訂項。例如，英文的自訂地區設定看起來會像是 **en-JM-x-custom**。

由於自訂地區設定為您組織所專屬，因此當送出以翻譯成基本語言和地域性方言時 (假如語言服務提供者有支援的話)，自訂地區設定代碼會被截斷。在上述範例中，**en-JM-x-custom** 會截斷成 **en-JM**，排除了自訂項特定的代碼部分。或者，如果語言服務提供者不支援牙買加 (JM) 的地域性方言代碼，則該地區設定代碼可能會被截斷成只剩基本語言 **en**。

## 設定 URL 重新導向的地區設定別名

您可以輕鬆地為網站 URL 在程式實際執行時所使用的地區設定及程式實際執行預覽的地區設定設定別名。假如您組織使用的自訂地區設定是以較長的長度來增加 URL 複雜度，這就會特別有用。

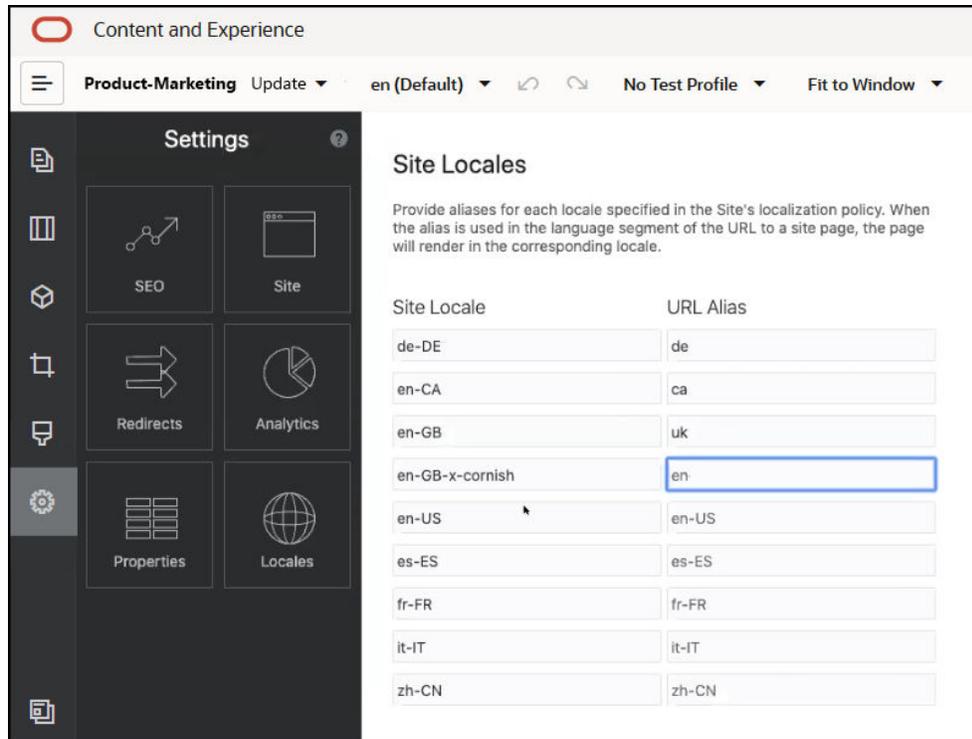
例如，您可以將自訂地區設定定義為 **en-GB-x-cornish**，在 URL 中看起來會像下面這樣：

```
https://example.com/site/BlogSite/en-GB-x-cornish/home.html
```

透過新增別名的方式，便能重新導向至更簡單的 URL：

```
https://example.com/site/mysite/en/home.html
```

1. 開啟網站進行編輯。
2. 按一下提要欄位中的 ，然後按一下**地區設定**。  
網站使用的所有地區設定清單便會顯示在對應的「URL 別名」欄位旁邊。



3. 在您希望設有別名的每個網站地區設定旁邊都各輸入一個別名，按一下**關閉**，然後按一下**儲存**。
4. 若要預覽程式實際執行時 URL 中所使用的別名，請按一下 .
5. 當您發布更新時，系統會發布變更並應用。

## 使用網站重新導向或 URL 對應

重組網站結構或移動網站時，您可以將使用者要求從舊的 URL 重新導向至目前的 URL。網站重新設計後若要繼續使用原來的書籤和已發布的連結，請為 URL 指定 30x 重新導向。

當您從其他基礎架構技術搬移至 Oracle Content Management 代管的網站時，可將搜尋引擎熱搜的頁面移到其他 URL。重新導向有助於重組網站的 URL 結構，同時維持搜尋引擎排名。

- [重新導向計畫](#)
- [新增網站重新導向](#)
- [在 JSON 檔案指定重新導向規則](#)
- [將重新導向規則檔上傳至網站](#)
- [對應網站 URL](#)

### 重新導向計畫

您可以針對傳送 HTTP 30x 回應的指定 URL 設定重新導向。要求若不符合任一指定的重新導向，系統就會進行一般的 URL 處理並以一般方式傳回頁面。

您可以建立指定重新導向的 JSON 檔案，然後將該檔案上傳到伺服器。伺服器將會在處理內送要求 URL 時使用此 JSON 檔案。

您可以利用以下的兩種重新導向規則，將連入 URL 重新導向至新的位置：

- [簡單的字串對字串比對](#)
- [簡化的萬用字元比對](#)

### 簡單的字串對字串比對

對於簡單的字串比對和取代，您可以指定明確的 URL，然後藉由將每個 URL 直接對應至目標 URL 來重新導向每個 URL。

下表顯示一些簡單的字串對字串比對。

來源 URL	目標位置 URL
/index.html	/home.htm
/products/widget	/items/knickknack
/index?page=widgets	/items/widgets

字串對字串對應相當易於瞭解和測試。規則會使用簡單字串比對來快速評估，然後對應查詢。

不過，在 URL 查詢參數方面不太有彈性。它們必須完全相符。額外的 URL 參數或參數順序不同會導致規則不符。

## 簡化的萬用字元比對

簡化的萬用字元比對既可讓規則比對許多 URL，又可限制取得結果所需的正規表示式回溯數量。

由於正規表示式可能過於複雜而難以撰寫，且評估結構不佳之正規表示式的所需時間不確定 (ReDoS)，因此第二種規則可允許使用簡化的比對機制。它使用萬用字元 ("\*") 比對內送 URL 中的零 (0) 或多個字元，並使用關鍵字 'wildcard' 搭配索引值，將 URL 的內送部分複製到重新導向的 URL。

下表顯示一些簡化萬用字元比對的範例。

來源 URL	目標位置 URL
/old/*	/new/<\${wildcard(1)}>
/dispatch.asp?page=*&facet=Lang*	/page<\${wildcard(1)}>/<\${wildcard(2)}>

在比對 URL 方面，相較於簡單的字串對字串比對，簡化的萬用字元比對更強大，且無須透過無限制的正規表示式處理即可達到此目的。語法相當簡單，且您可以在 URL 上使用樣式比對，以一個樣式比對大量 URL。

由於它們以正規表示式為基礎，因此萬用字元規則的評估速度會比簡單字串比對慢。大量規則可能對一般頁面傳遞效能帶來負面的效能影響。

## 新增網站重新導向

網站 URL 如果變更，重新導向可將一個 URL (來源) 轉送至另一個 URL (目標)。這有助於保留使用者書籤和搜尋引擎排名。

有兩種類型的重新導向可以使用：

- 永久重新導向，使用 301 HTTP 服務回應代碼
- 暫時重新導向，使用 302 HTTP 服務回應代碼

上傳 redirect.json 檔案：

1. 開啟網站進行編輯。
2. 先按一下提要欄位中的 ，然後按一下  重新導向。
3. 按一下**選取要上傳的檔案**，瀏覽至要使用的檔案並選取該檔案，然後按一下**確定**。
4. 當您發布更新時，系統會發布變更並應用。

## 在 JSON 檔案指定重新導向規則

您可以在 JSON 檔案中指定 URL 的重新導向規則。

請使用以下格式在 JSON 檔案中指定 URL 的重新導向規則。

```
{  
  "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 檔案中的外圍內容結構是一個陣列。此陣列包含規則執行處理。

系統會先評估 "string" 規則，再評估 "wildcard" 規則。其中若有一個規則符合，後續的規則評估便會終止，然後產生相對應的重新導向。

每個規則都具有下列特性：

- "comment" 特性：選擇性字串，對規則評估無任何影響。它包括附註或註記。
- "expression" 特性：必要字串，主要比對內送的網站相對 URL。在萬用字元規則中，星號 (\*) 記號可比對零個或多個字元。
- "location" 特性：必要字串，指示重新導向的位置或目的地。重新導向可以是完整 URL 或相對 URL。
- "code" 特性：選擇性整數，提供發出重新導向時所要使用的 HTTP 回應代碼。值必須是下列其中一個整數：
  - 301：代表資源永久移動。如果省略 "code" 特性，此為預設值。
  - 302：代表資源暫時移動。
- "type" 特性：選擇性字串，指示重新導向規則類型。值必須是下列其中一個字串：
  - "string" 規則較為快速，其表示式會確實地比對整個輸入 URL。
  - "wildcard" 指定可比對一些 URL 的萬用字元規則。如果省略此特性，此為預設值。

### 位置記號

您可以使用位置記號來協助製作重新導向位置。以下的每個位置記號均可協助指定重新導向：

- <\$urlPath\$>：相符 URL 的路徑部分。
- <\$urlQueryString\$>：相符 URL 中的整個 URL 查詢字串。
- <\$urlQueryStringExcept(name1,name2)\$>：相符 URL 中的整個 URL 查詢字串，但扣除具名的參數。
- <\$wildcard(N)\$>：相符 URL 中以一為基礎的相符萬用字元索引。(這類似於正規表示式中的 \1..\9。)
- <\$name\$>：具名查詢字串參數的值。舉例來說，如果輸入的查詢字串為 msmith: ?page=42，您可以在位置使用 <\$page\$>，將 '42' 放入位置中。

## 限制

下列限制適用於整個 `redirects.json` 檔案與其所含的規則：

- Oracle Content Management 接受的整體檔案大小上限為 250 KB。
- `redirects.json` 檔案中的規則數目上限為 1,000 個。
- 規則的 "expression" 長度上限為 1,000 個字元。
- 規則的 "location" 長度上限為 2,000 個字元。
- 萬用字元規則表示式中的 '\*' 記號數目上限為 10 個。

## 字串比對範例

規則：

```
{
  "type": "string",
  "expression": "/old/page.jsp?id=material&type=glass",
  "location": "/new/<$id$>.htm"
}
```

下列 URL 符合規則：

`/old/page.jsp?id=material&type=glass`

- 產生的位置會是：`/new/material.htm`
- 整個 URL 都符合，包括查詢字串。
- 雖然位置使用 `<$id$>`，但因為只有一個可能的查詢字串會相符，所以此範例並不需要。位置可寫為 `/new/material.htm`。

下列 URL 不符合規則：

- `/old/page.jsp`  
(規則表示式指定了一個必須相符的查詢字串。)
- `/old/page.jsp?id=material&type=glass&index=2`  
(候選 URL 中的額外 `&index=2` 未完全符合規則表示式。)
- `/old/page.jsp?type=glass&id=material`  
(查詢字串參數的順序必須與 "string" 規則指定的順序相同。)

## 萬用字元比對範例

規則：

```
{
  "type": "wildcard",
  "expression": "/old/*/pages/*?id=*item=sheet-*",
  "location": "/new/<$id$>/<$wildcard(4)$>.html"
}
```

下列 URL 符合規則：

- /old/phones/android/pages/info.asp?id=XT1045&item=sheet-specs
  - 產生的位置會是：/new/XT1045/specs.html
  - URL 的路徑部分符合，因此也會檢查查詢字串是否符合條件。
  - 此範例中的參數與規則表示式中的參數順序相同，但這不是必要條件。
- /old/phones/android/pages/info.asp?item=sheet-specs&id=XT1045
  - 產生的位置會是：/new/XT1045/specs.html
  - URL 的路徑部分與規則表示式中問號 (?) 之前的部分相符，因此另外檢查了參數是否相符。
  - 雖然規則表示式中參數的順序不相同，但個別參數都符合條件。
- /old/phones/android/pages/info.asp?id=XT1045&item=sheet-specs&unrelated=thing
  - 產生的位置會是：/new/XT1045/specs.html
  - URL 的路徑部分符合，因此也會檢查查詢字串是否符合條件。
  - 候選 URL 含有一個額外的 &unrelated=thing 參數，但因為規則表示式中的具名查詢參數相符，所以此規則將視為相符。
  - unrelated 參數將會成為位置記號 <\$unrelated\$>，其值為 thing (但與規則是否符合無關)。

下列 URL 不符合規則：

- /old/pages/info.jsp  
(URL 的路徑部分不符合規則表示式的路徑部分。)
- /old/phones/android/pages/info.asp  
(URL 的路徑部分符合規則表示式的路徑部分，但規則表示式中的查詢參數不符。)
- /old/phones/android/pages/info.asp?id=cellular  
(URL 的路徑部分符合規則表示式的路徑部分，但規則表示式中的部分查詢參數不符。)

### 定義權杖陣列

您也可以在 redirects.json 檔案內建立一個權杖定義陣列，以在設定支援多個虛名 URL 的重新導向時提供協助。這可讓您依據內送 URL 的特性適當地進行重新導向。

請使用以下格式在 redirects.json 檔案中定義要在重新導向規則 URL 中使用的權杖。

```
{
  "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 具有下列特性：

- "token"：要定義的權杖名稱。
- "type"：可為下列其中一項：
  - "hostmatch"：比對內送 URL 的主機值。
  - "pathmatch"：比對內送 URL 的路徑名稱值。
  - "querymatch"：比對內送 URL 的查詢值。
- "expression"：應用於比對的表示式。支援萬用字元。
- "value"：應用於權杖的值。
- "flags"：除非 flags 值設為 caseinsensitive，否則表示式比對預設會區分大小寫

計算權杖值時，會依序列舉 tokenDefinitions 陣列，並且使用第一個符合的定義。如果沒有任何權杖定義符合該權杖，則會改用空的字串。基於方便性與效能考量，應將常用的權杖放在 tokenDefinitions 清單最前面。

tokenDefinitions 的限制條件如下：

- 最多只能建立 250 個權杖定義。
- token 名稱必須在 100 個字元以內。
- expression 最多可含 10 個萬用字元。
- expression 必須在 1000 個字元以內。
- value 必須在 1000 個字元以內。

## 範例

舉例來說，您的 redirects.json 檔案可能如下：

```

{
  "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"
      },
    ]
  }
}
```

在本案例中，規則的 location 特性含有 <\$pathPrefix\$> 權杖。pathPrefix 權杖在 tokenDefinitions 區段中定義。內送 URL 如果符合 "vanity.com"，pathPrefix 值將會設為 /fashion。這將用於 location 回應中，並產生 /fashion/about/new-privacy-policy.html。

假設第一個虛名網域 URL 為 http://example.com/legacy-privacy-policy.html。這符合第一個也是唯一一個重新導向規則。

此規則的已宣告 location 為 <\$pathPrefix\$>/about/new-privacy-policy.html。在此情況下，必須評估 <\$pathPrefix\$> 權杖。若要這麼做，需要列舉 tokenDefinitions 陣列以尋找相符項目。

系統會先採用第一個權杖定義。它的 token 也為所需的權杖，因此將對其做進一步的評估。vanity.com 表示式與內送 URL 的 example.com 不符，因此此定義不符合需求，所以系統將繼續列舉。

這時已沒有其他權杖定義，所以 <\$pathPrefix\$> 權杖的值會使用空的字串。針對此重新導向所傳回的最終位置為 /about/new-privacy-policy.html。

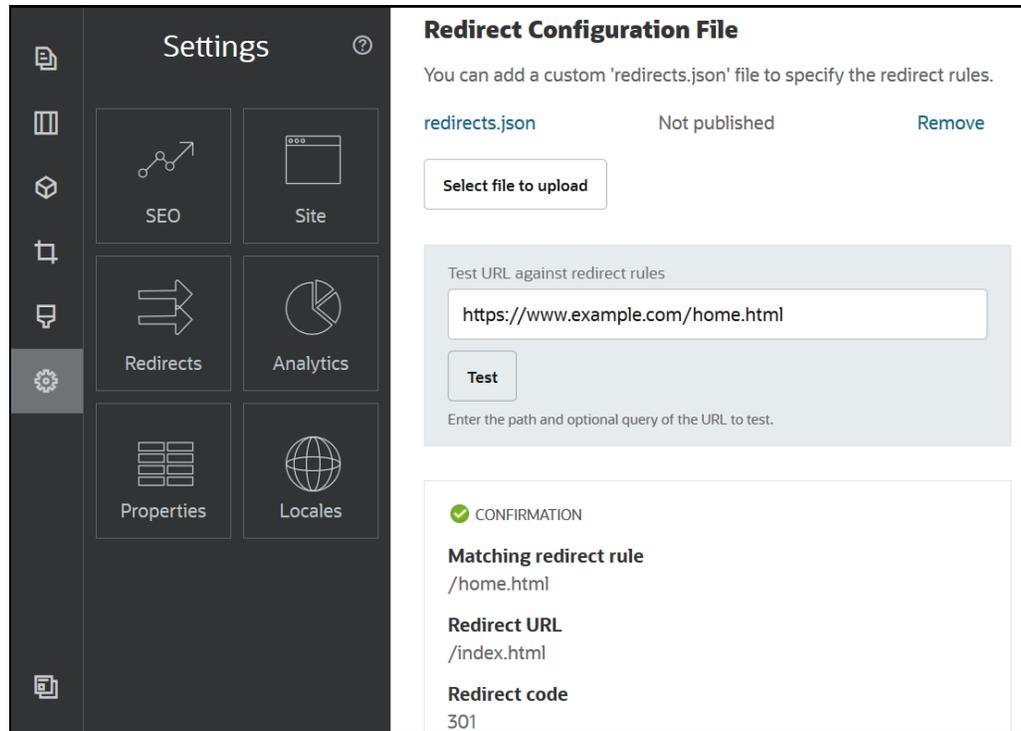
假設第二個虛名網域 URL 為 http://vanity.com/legacy-privacy-policy.html。與第一個 URL 一樣，此規則的已宣告 location 為 <\$pathPrefix\$>/about/new-privacy-policy.html。在此情況下，必須評估 <\$pathPrefix\$> 權杖。若要這麼做，需要列舉 tokenDefinitions 陣列以尋找相符項目。

系統會先採用第一個權杖定義。如上述情況，它的 token 也為所需的權杖，因此會對其做進一步的評估。vanity.com 表示式符合內送 URL 的 vanity.com，因此此定義符合需求，所以使用 /fashion 值作為權杖的值。

因為找到與權杖相符的項目，所以系統停止列舉權杖定義陣列，並且計算出的最終位置為 /fashion/about/new-privacy-policy.html。

### 測試網站重新導向

在編輯網站時，您可以開啟**設定值**面板並按一下**重新導向**，以測試網站重新導向。輸入要測試的 URL 並按一下**測試**。



## 將重新導向規則檔上傳至網站

您可以在 Oracle Content Management 中將重新導向規則上傳至網站。

將 `redirect.json` 檔案上傳至網站：

1. 開啟網站以進行編輯。
2. 按一下提要欄位中的 ，然後按一下 。
3. 按一下 **選取要上傳的檔案**，瀏覽至要使用的檔案並選取該檔案，然後按一下 **確定**。
4. 當您發布更新時，系統會發布變更並應用。

## 對應網站 URL

在使用 Oracle Content Management 建立與發布網站後，您可以設定網域名稱系統 (DNS)，以便透過註冊的網域名稱 (例如 `www.mysite.com`) 來存取此網站

網域名稱系統 (DNS) 可藉由將您的網域名稱對應到您網站的位置或正規名稱 (CNAME)，指定他人可在何處找到您的網頁。

若要對應您的網域名稱，您需要有下列各項：

- Oracle Content Management 執行處理的 URL。格式通常如下：  
`service-tenant.documents.datacenter.oraclecloud.com`
- 您在網域名稱登錄機構所登錄的網域名稱。  
例如 `www.example.com`。也可以是子網域，例如 `www.example.com/subdomain`。

- 一個具有內容傳遞網路 (CDN) 提供者的帳戶。Oracle Content Management 提供與 Akamai 的整合。若要針對執行處理設定 Akamai，請洽詢 Oracle 客戶服務部。

如果您要使用自己的 CDN (而不是 Oracle Content Management 提供的 Akamai)，請執行以下步驟。

不同的「網域名稱系統」提供者會有不同的 Web 介面，而其更新 CNAME 記錄的步驟也有所不同。下列步驟提供您需要的資訊及應當遵循的一般步驟。

若要將網站 URL 對應至網域名稱：

1. 向您的內容傳遞網路提供者要求提供網域的 Secure Sockets Layer (SSL) 憑證。例如，`https://www.example.com`。
2. 設定內容傳遞網路，以達成以下目標：
  - a. 內容傳遞網路接受網域的所有內送要求，並使用安全協定 (https) 轉送這些要求。
  - b. 來源指向 Oracle Content Management 的網域  
`service-tenant.documents.datacenter.oraclecloud.com`
3. 變更 DNS 伺服器區域檔案，將網域名稱對應至內容傳遞網路提供者所提供的邊緣處理伺服器：  
`domain CNAME CDN Server`
4. 等待更新傳輸。視您的 DNS 服務而定，可能需要 2 到 48 個小時。  
傳輸變更後，便可以使用您的網域名稱來存取網站。例如：

`https://www.mysite.com/site_name`

依照預設，如果您使用為網站提供的標準 URL，即可使用 Oracle Cloud REST API for Content Management 的端點。例如，資料夾清單元件和檔案清單元件都使用 REST API 執行資料夾和檔案作業。如果您使用自訂 URL，請確認您可以使用您的網域名稱存取此端點。例如：

`https://www.mysite.com/documents`

# 16

## 改善網站效能

您可以利用瀏覽器快取，改善內容傳遞和在瀏覽器中呈現的效能。網頁頭版區域 (ATF) 呈現也可以改善網站的呈現功能。

- [利用快取功能改善效能](#)
- [網頁頭版區域 \(ATF\) 呈現](#)

## 利用快取功能改善效能

內容項目、數位資產與網站的傳遞應充分利用訪客的瀏覽器快取，以提升瀏覽器之內容傳遞與呈現的效能。

網站、主題、內容項目和數位資產都會快取到訪客的瀏覽器快取中一段時間。網站、主題、內容項目或數位資產更新之後，會變更 URL 中的快取索引鍵，讓瀏覽器必須擷取不同的 URL 來取得新的項目。

快取索引鍵可透過只參照目前的資源，協助管理瀏覽器快取的使用。雖然快取索引鍵包含在 URL 中，但通常的情況下，它是一個邏輯元素，而非實體位置 (資料夾)。快取索引鍵變更並不會指向其他實體位置來尋找資源；它只是通知伺服器擷取資源的目前版本。

資源可以是靜態 (如 CSS、JS 和影像檔)，也可以是動態 (如頁面資料、網站資料和內容項目資料)。可供建置網站的資源有以下五種類別：

- 產品資源 – 屬於產品之一部分的資源，每當新版本的產品發行或修正時，資源也會隨之更新。
- 網站資源 – 屬於網站之一部分的資源，例如 structure.json、頁面資料以及影像。網站發布時，這些資源也會隨之更新。下文提供控制器描述。
- 主題資源 – 屬於主題之一部分的資源，例如版面配置、CSS 以及影像。主題發布時，這些資源也會隨之更新。
- 元件資源 – 屬於自訂元件之一部分的資源。這些包括 HTML、JS 和 CSS，以及構成元件的影像檔。元件發布時，這些資源也會隨之更新。如果一個元件變更並重新發布，所有元件的快取索引鍵也會跟著變更，因為此索引鍵是全部元件的單一索引鍵。
- CaaS 資源 – 提供給內容項目和數位項目的資源。當內容項目發布或重新發布時，或是集合目標變更時，這些資源也會隨之更新。

下列主題描述 Oracle Content Management 程式實際執行和「網站產生器」的快取功能：

- [程式實際執行快取功能](#)
- [網站產生器快取功能](#)

## 程式實際執行快取功能

就程式實際執行而言，Oracle Content Management 的 Cache-Control 標頭設為 15 天。此外對於所有資源，會在 URL 加上一個快取索引鍵。

只要 URL 沒變，瀏覽器就會從其本機快取提供資源服務 (若有的話)。如果資源更新，URL 中的快取索引鍵也會一併更新，這會強制瀏覽器對伺服器提出新的要求並更新本機快取。

包含快取索引鍵的控制器也會被快取 1 分鐘。因此，最長會有 1 分鐘的時間不會見到任何已更新的快取索引鍵。

程式實際執行時，伺服器會傳回 controller.html，其中含有最新的產品、網站、主題、元件及 CaaS 資源快取索引鍵。controller.html 會加上內含索引鍵的命令檔；例如：

```
<script type="text/javascript">
  var SCSCacheKeys = {
    product: '123',
    site: '456',
    theme: '789',
    component: '012',
    caas: '345'

  };
</script>
```

controller.js 將使用這些索引鍵建構 URL，如下表所示。

資源類型	範例
產品資源	<i>/sitePrefix/productCacheKey/_sitesclouddelivery/...</i>  <i>/mySite/_cache_947d/_sitesclouddelivery/</i>
主題資源	<i>/sitePrefix/themeCacheKey/_themesdelivery/themeName/...</i>
元件資源	<i>/sitePrefix/compCacheKey/_compdelivery/compName/...</i>
網站資源	<i>/sitePrefix/siteCacheKey/content/...</i> <i>/sitePrefix/siteCacheKey/structure.json</i> <i>/sitePrefix/siteCacheKey/pages/100.json</i>
CaaS 資源	<i>RegularCaaSUrl?cacheKey=caasCacheKey</i>

透過在 URL 中插入快取索引鍵這種方式，Oracle Content Management 可實際變更 URL，讓瀏覽器以為更新的資源是新的資源，因而強制瀏覽器載入更新的資源。



**備註：**

對於安全網站，系統只會快取產品、主題和元件資源，不會快取網站或 CaaS 內容。

## 網站產生器快取功能

在「網站產生器」中，靜態資源在快取中的存留時間為 15 天。

使用「網站產生器」時，系統會快取產品、主題以及元件資源。(不會快取網站和 CaaS 資源。)當「網站產生器」啟動或重新整理後，系統會重新產生主題和元件快取索引鍵。

對某個主題或元件進行變更後，若要在「網站產生器」中顯示該變更，您需要重新整理「網站產生器」(F5)。

## 網頁頭版區域 (ATF) 呈現

ATF 呈現方式讓網站外觀的載入非常快速。目標是要先呈現頁面的所有可見部分，然後在使用者向下捲動之前，再呈現頁面中一開始不可見的其餘部分。

槽位可以指定「網頁頭版區域」，這會在頁籤上顯示圖示。

若要以這種新方式呈現槽位，必須以 `scs-atf` 標示槽位，如下：

```
<div class="scs-slot scs-atf" id="headline"></div>
```

元件必須在完成呈現時通知呈現器。內建元件預設即會這麼做。自訂元件可進行額外的呼叫，且必須執行下列動作：

1. 通知呈現器它希望呈現器等到它完成呈現為止。
2. 在已完成時通知呈現器。

針對第 1 點，在自訂元件的 `appinfo.json` 檔案中，新增下列特性：

```
"initialData": {  
  . . .  
  "customRenderComplete": true,  
  . . .  
}
```

針對第 2 點，在元件的 `render.js` 檔案中，藉由進行下列呼叫，確定在完成時會通知呈現器：

```
SitesSDK.setProperty('renderComplete', true);
```

如果並非所有 ATF 槽位中的元件都及時回報它們已完成，則呈現器將先等待 2 秒，然後才繼續呈現頁面的其餘部分。如果您知道此時間將不夠長，您可以在頁面樣板中宣告下列全域變數來延長時間：

```
var SCSAtfPassTimeout = 3000;
```

 **備註：**

時間以毫秒為單位，因此這個範例是將逾時設為 3 秒。

API 會提供 ATF 處理作業的診斷資料。您可以在除錯主控台中呼叫下列方法，或視需要從頁面加以存取：

```
SCSRRenderAPI.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

## 網站開發

此部分可幫助您瞭解及開發樣板、主題、樣式、元件以及其他有用的項目，開始進行網站開發。其中包含下列章節：

- [自訂設計與樣式](#)
- [瞭解背景使用](#)
- [設定觸發程式和動作](#)
- [開發樣板](#)
- [開發主題](#)
- [開發版面配置](#)
- [開發元件](#)

## 自訂設計與樣式

每個 Oracle Content Management 主題都必須要有一項設計，以指定頁面的外觀與感覺及預設網站中用於該主題的元件樣式設定值。

- [關於設計](#)
- [設計檔案](#)
- [自訂對話清單樣式](#)
- [自訂資料夾清單和檔案清單樣式](#)
- [自訂社群列圖示](#)
- [設定 Oracle Intelligent Advisor 互動式提問樣式擴充功能](#)

### 關於設計

每個 Oracle Content Management 主題都必須要有一項設計，以指定頁面的外觀與感覺及該主題所用的元件樣式設定值。

使用者在建立網站時必須選取一個樣板，而此樣板預設即已包含主題。您可以另外為網站建立或選擇使用其他主題。

建立主題時，除了設計頁面版面配置外，您還必須針對使用者可使用的每一種元件類型，指定要在「設定值」面板顯示的可用樣式 (段落、標題、影像等等)。一項設計中可有多種樣式以指定不同元件的預設值。您可以選擇文字字型與字型大小、影像框架等，然後將它們另存為具名的設計。此設計中的項目將會影響頁面版面配置呈現時的外觀。

請參閱 *運用 Oracle Content Management 建置網站* 中的使用網站頁面。

### 設計檔案

design.json 和 design.css 是主題的兩個預設設計檔案。

- design.json：指定元件樣式
- design.css：提供類別值定義 (例如色彩和字型)

這兩個檔案位於主題結構的 /designs/default/ 目錄中。

#### design.json 檔案

design.json 檔案的結構如下：

```
{
  "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": []
}
}
}
```

每個 "styles":[] 項目都可包含該特定元件的樣式清單。例如，標題元件提供以下預設樣式：

```
"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"
}
]
```

 **備註：**

如果您為元件建立自訂樣式並將它對應至 `design.json` 中的樣式，您就不需要像系統提供的那樣使用完整名稱。只要指定想要使用的字串即可。例如，不使用 `"COMP_STYLE_BOX"` 作為名稱，只使用 `"Box"` 即可。這表示在該元件的「設定值」面板中，**樣式**頁籤清單只會顯示 `"Box"` 這個名稱，而非 `"COMP_STYLE_BOX"`。

`name` 值會對應至使用者介面中顯示的實際內容，如下所示：

```
"COMP_STYLE_FLAT": "Flat",  
"COMP_STYLE_HIGHLIGHT": "Highlight",  
"COMP_STYLE_DIVIDER": "Divider",
```

### design.css 檔案

`design.css` 檔案提供類別值的定義。以下是幾個範例。

```
.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; }
```

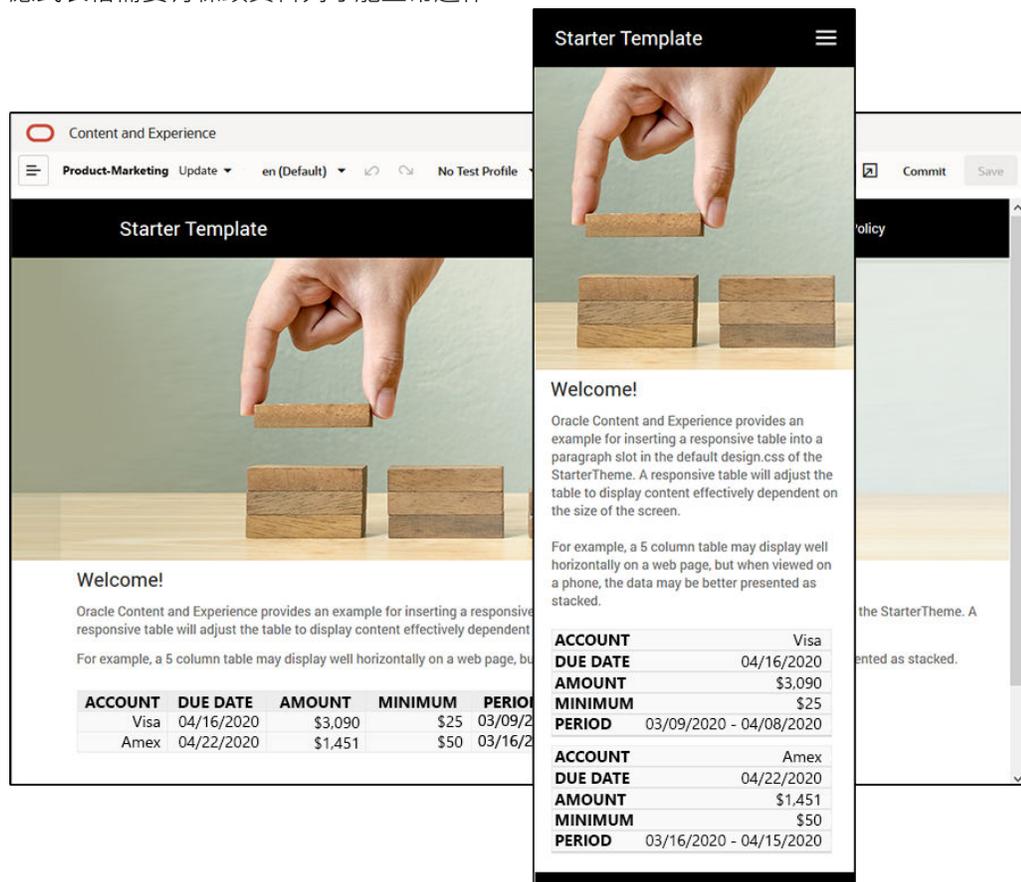
```
.scs-button-default-style .scs-button-button:hover {  
  background: #f7f8f9;  
  border: 1px solid #c4ced7;  
  color: #0572ce;  
  box-shadow: inset 0 1px 0 #f7f8f9;  
  text-shadow: 0 1px 0 #f7f8f9; }
```

```
.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; }
```

## 響應式表格設計

Oracle Content Management 提供段落元件內響應式表格的 CSS 範例，在行動裝置上顯示時可啟用資料列堆疊處理。

響應式表格可根據螢幕大小調整表格以有效地顯示內容。例如，5 欄表格可在網頁上以水平方式正確地顯示，但在電話上檢視時，資料可能最好以堆疊形式呈現。請注意，響應式表格需要有標頭資料列才能正常運作。



在下列產生的 HTML 中，請注意，已對具有符合資料欄標頭文字值的每個表格儲存格新增資料標籤屬性。

```
<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>
```

每個儲存格都有該屬性之後，TD 會在套用下面的 CSS 規則時彼此堆疊。

```
.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;
}
```

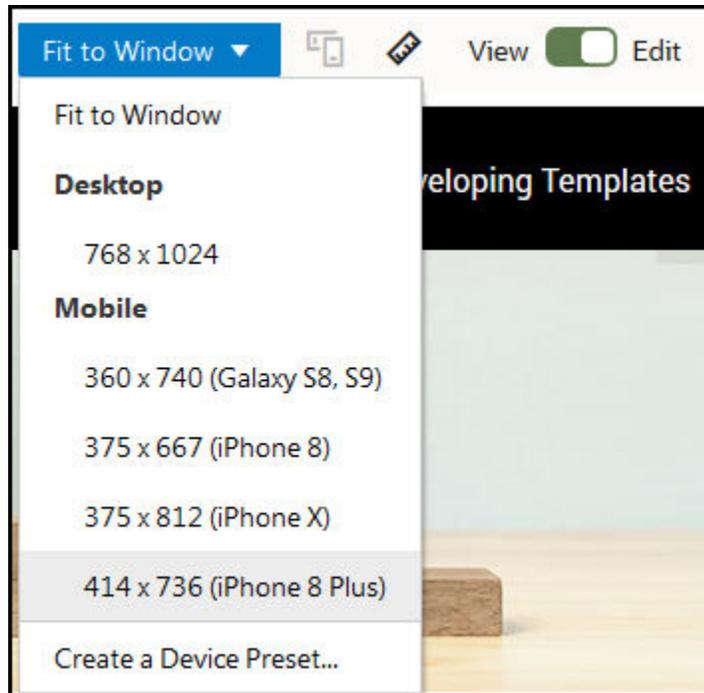
請注意，下方的媒體規則會查詢螢幕大小，且只有在螢幕大小小於 767 像素時才會生效：

```
@media screen and (max-width: 767px) {
```

使用 CSS 讓插入的表格成為響應式表格的程式碼範例位於所提供 **StarterTheme** 的預設 **design.css** 中。如果從 **StarterTheme** 建置網站，插入至段落槽位的表格預設為響應式表格。若要插入表格：

1. 在 Oracle Content Management 中，開啟網站，並切換至**編輯**模式。
2. 建立新的更新，或選擇要修改的現有更新。
3. 將新的段落元件拖曳至頁面，然後按一下想要插入表格的位置，或在想要插入表格的現有段落按一下。
4. 在 RTF 工具列上，按一下 ，並設定表格特性。請確定選取第一列作為標頭，並調整表格寬度，以在預期使用的最小螢幕有效地運作。例如，如果您預期在電話上檢視網站，則 767 像素的預設寬度可能會太寬，因此無法在電話螢幕上完整顯示，即使以堆疊處理也是一樣。建議您將表格寬度設為較小大小 (如 300 像素)，或將寬度設為 100%。

完成時，請切換回**檢視**模式，並選取螢幕寬度小於 767 像素的檢視選項來預覽結果。您必須處於檢視模式，因為表格在編輯模式時不能以響應式的方式運作。



如果您想要在從不同主題建置網站時使用響應式表格，則需要將 StarterTheme design.css 中的程式碼複製到您目前使用之主題的 design.css 檔案。

1. 若要複製 StarterTheme design.css 的程式碼，請按一下 Oracle Content Management 側邊導覽中的**開發人員**。
2. 按一下**檢視所有主題**。
3. 選取 **StarterTheme**，然後按一下**開啟**。
4. 按一下設計以開啟資料夾，然後按一下**預設**。
5. 選取 StarterTheme 的 design.css 檔案，然後按一下**下載**。
6. 在文字編輯器中開啟檔案，並找到開頭為註解 *An example CSS of how to render a table responsively* 的檔案區段。
7. 選取程式碼，直到下個註解為止，並進行複製。

```
/**
 * 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. 重複步驟來下載您想要修改之樣板的 `design.css` 檔案，並開啟樣板，然後將複製的程式碼貼到檔案中。
9. 儲存變更，並將修改過的 `design.css` 檔案以新版本方式上傳至您目前修改的主題。

## 自訂對話清單樣式

您可以在 `design.css` 檔案中新增選取器，以自訂「對話清單」元件的樣式。

您可以使用以下 **CSS** 選取器自訂「對話清單」元件的樣式。

選取器名稱	描述
<code>.scs-convo-list-cust .scs-convo-list-container</code>	元件的最外層 DIV
<code>.scs-convo-list-cust .scs-convo-list-title</code>	清單中所選對話的標題
<code>.scs-convo-list-cust .scs-convo-list-line-separator</code>	清單標題與清單之間的分隔線
<code>.scs-convo-list-cust .scs-convo-list-convo-title</code>	清單中的對話標題

選取器名稱	描述
<code>.scs-convo-list-cust .scs-convo-list-convo-line-separator</code>	每個對話之間的分隔線
<code>.scs-convo-list-cust .scs-convo-list-active</code>	清單中所選對話的標題
<code>.scs-convo-list-cust .scs-convo-list-convo-posts</code>	對話的張貼項目數
<code>.scs-convo-list-cust .scs-convo-list-convo-unread</code>	對話的未讀取訊息數
<code>.scs-convo-list-cust .scs-convo-list-convo-updated</code>	對話的上次更新日期
<code>.scs-convo-list-cust .scs-convo-list-no-convo-msg</code>	清單是空的時的訊息
<code>.scs-convo-list-cust .scs-convo-list-no-auth-msg</code>	對話清單在公用網站上呈現且無使用者授權時的訊息

請參閱 *運用 Oracle Content Management 建置網站* 中的使用樣式和格式。

### 範例

本範例解說使用自訂 CSS 變更「對話清單」的字型、樣式以及色彩。



下列程式碼顯示用於建立此範例的自訂 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;
    }

```

## 自訂資料夾清單和檔案清單樣式

您可以在 `design.css` 檔案中新增選取器，以自訂資料夾清單和檔案清單元件的樣式。

您可以使用「資料夾清單」元件列出 Oracle Content Management 帳戶中某個指定資料夾內的資料夾。資料夾清單會自動與頁面上的「檔案清單」元件和文件管理程式通訊，以顯示資料夾清單中所選資料夾內的檔案。

您可以使用「檔案清單」元件提供 Oracle Content Management 帳戶中指定資料夾內的檔案檢視。檔案清單會自動與頁面上的「資料夾清單」元件通訊，以顯示資料夾清單中所選資料夾內的檔案。

### 資料夾清單 CSS 選取器

您可以使用以下 CSS 選取器自訂「資料夾清單」元件的樣式。

選取器名稱	描述
<code>.scs-folder-list-cust .scs-folder-list-container</code>	元件的最外層 DIV
<code>.scs-folder-list-cust .scs-folder-list-folder-title</code>	資料夾名稱
<code>.scs-folder-list-cust .scs-folder-list-line-separator</code>	資料夾名稱與子資料夾清單之間的分隔線
<code>.scs-folder-list-cust .scs-folder-list-sub-folder-title</code>	子資料夾名稱
<code>.scs-folder-list-cust .scs-folder-list-sub-folder-title-active</code>	選取子資料夾名稱

選取器名稱	描述
<code>.scs-folder-list-cust .scs-folder-list-no-folder-msg</code>	沒有子資料夾可供顯示時的訊息

### 檔案清單 CSS 選取器

您可以使用以下 CSS 選取器自訂「檔案清單」元件的樣式。

選取器名稱	描述
<code>.scs-file-list-cust .scs-file-list-container</code>	元件的最外層 DIV
<code>.scs-file-list-cust .scs-file-list-folder-title</code>	資料夾名稱
<code>.scs-file-list-cust .scs-file-list-line-separator</code>	資料夾名稱與檔案清單之間的分隔線
<code>.scs-file-list-cust .scs-file-list-row</code>	包含檔案資訊的資料列
<code>.scs-file-list-cust .scs-file-list-left-col</code>	元件左側區段中的檔案縮圖
<code>.scs-file-list-cust .scs-file-list-mid-col</code>	元件的中間區段，其中包含檔案的名稱、描述、上次修改時間以及大小
<code>.scs-file-list-cust .scs-file-list-file-title</code>	應用程式中間區段中的檔案名稱
<code>.scs-file-list-cust .scs-file-list-file-desc</code>	應用程式中間區段中的檔案描述
<code>.scs-file-list-cust .scs-file-list-file-lastModified</code>	檔案的上次修改時間
<code>.scs-file-list-cust .scs-file-list-file-size</code>	檔案的大小，與上次修改時間之間有一條垂直分隔線
<code>.scs-file-list-cust .scs-file-list-file-size-no-sep</code>	檔案的大小，且無垂直分隔線(未顯示上次修改時間)
<code>.scs-file-list-cust .scs-file-list-right-col</code>	應用程式的右側區段
<code>.scs-file-list-cust .scs-file-list-file-download-icon</code>	應用程式右側區段中的下載圖示
<code>.scs-file-list-cust .scs-file-list-no-file-msg</code>	沒有檔案可供顯示時的訊息

請參閱 *運用 Oracle Content Management 建置網站* 中的檔案清單和資料夾清單。

## 自訂社群列圖示

您可以建立自訂社群圖示，用於主題預設網站的社群列。

網站社群列中顯示的社群圖示，取決於網站主題的設計。如果您變更網站的主題，社群圖示也會隨著主題變更。Oracle Content Management 主題隨附適用於 Facebook、Twitter、LinkedIn、Google+ 及 YouTube 的常見社群圖示。

您可以藉由編輯 `design.json` 和 `design.css` 檔案，將自訂社群圖示新增至社群圖示元件。

### design.json 檔案

在 design.json 檔案中，您可以使用名稱和類別結構來指定新圖示，如這個範例程式碼所示：

```
"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 檔案

在 design.css 檔案中，您可以使用名稱和 url 規格來新增圖示，如這個範例程式碼所示：

```
.scs-facebook-icon {
  background-image: url("facebook.png"); }
.scs-twitter-icon {
  background-image: url("twitter.png"); }
.scs-linkedin-icon {
  background-image: url("linkedin.png"); }
```

## 設定 Oracle Intelligent Advisor 互動式提問樣式擴充功能

您可以設定 Oracle Intelligent Advisor (OIA) (之前稱為 Oracle Policy Automation) 互動式提問的樣式，將其外觀與您公司的外觀與感覺統合。

您可以在 design.css 檔案中設定下列 CSS 類別選取器。這些類別選取器的前置碼皆為“scs-opainterview-”。

每個類別均定義指定元件的所有樣式。類別具有完全的掌控性，不會用到現有的 OIA 樣式。

選取器	適用於	描述
interview	interview	互動式提問區域包含整個互動式提問內容，包括標頭、頁尾及導覽區域。
interviewContent	互動式提問內容	互動式提問內容區域包含畫面標題和控制項，但不包括標頭、頁尾及導覽區域。
screenTitleBlock	畫面標題區塊	畫面標題區塊有一個區域，當中有畫面標題以及包含在該列中的任何其他小工具，例如畫面下拉式清單和 (或) 下一步與上一步按鈕。
screenTitle	畫面標題	畫面標題區域就是包含畫面標題的區域。
nextButton	下一步按鈕	下一步按鈕。
backButton	上一步按鈕	上一步按鈕。
restartButton	重新啟動按鈕	重新啟動按鈕。
exitButton	結束按鈕	結束按鈕。
header	header	標頭區域。
footer	footer	頁尾區域。
question	問題文字	問題文字樣式設定。
control	控制項容器	包含控制項之元素的樣式設定。
label	標籤控制項	標籤控制項樣式設定。
controlError	所有控制項	錯誤文字容器樣式設定。
controlErrorText	所有控制項	錯誤文字範圍樣式設定。
textInput	單行文字方塊、密碼和遮罩	文字輸入控制項樣式設定。
textAreaInput	多行文字方塊	多行文字輸入控制項樣式設定。
calendarInput	行事曆	行事曆輸入控制項樣式設定。支援可變更行事曆圖示色彩的 iconColor 欄位，以及指示是否應顯示行事曆圖示的 keepIcon 欄位。
dropDownInput	下拉式清單	下拉式清單輸入控制項樣式設定。
filterDropDownInput	經過篩選的下拉式清單	經過篩選下拉式清單輸入控制項樣式設定。支援可變更下拉箭頭色彩的 iconColor 欄位。
listInput	固定式清單	固定式清單輸入控制項樣式設定。
radioInput	選項按鈕	選項按鈕輸入控制項樣式設定。可使用 borderColor 和 fillColor 選項設定樣式。iconType 特性可以變更使用的圖示類型。目前只支援 'tick' 和 'fill' 替代選項。
checkboxInput	核取方塊	核取方塊輸入控制項樣式設定。可使用 borderColor 和 fillColor 選項設定樣式。iconType 特性可以變更使用的圖示類型。目前只支援 'square' 和 'fill' 替代選項。
autoCompleteInput	自訂搜尋	使用 customSearch 擴充功能時，autocomplete 欄位的樣式設定。
captchaInput	CAPTCHA 輸入欄位	使用者於其中輸入 CAPTCHA 的輸入欄位樣式設定。
signatureInput	簽章控制項	簽章控制項樣式設定。支援可變更用於簽章筆墨的額外 inkColor 欄位。
explanationHeader	說明控制項	說明控制項最上層可展開標頭的樣式設定。
explanationText	說明控制項	展開之說明控制項文字的樣式設定。

選取器	適用於	描述
signatureClearButton	簽章控制項	簽章控制項清除按鈕的樣式設定。
uploadAddButton	上傳控制項	上傳控制項的新增按鈕樣式設定。
entityRemoveButton	實體收集控制項	實體收集控制項。

### design.css 中定義的樣式擴充功能範例

```
.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;
}
```

## 瞭解背景使用

您可以指定網站頁面和頁面上個別槽位的背景色彩和影像。頁面、槽位與元件的背景會層層相疊。舉例來說，若指定槽位的背景，槽位背景會疊在指定的頁面背景上面。

- [關於背景和主題](#)
- [背景的實行方式](#)
- [設定值的儲存位置](#)

## 關於背景與主題

背景功能主要用於網站頁面與槽位。頁面與槽位的背景集應由使用者在主題層級設定，而非由開發人員設定。

使用者可以在主題槽位中設定背景效果，不過，這些效果會覆寫在編輯網站時為頁面設定的任何背景效果。

背景功能不會更改主題或成份主題檔案。頁面的背景設定值並無法更改主題，而網站中若有以相同版面配置為基礎的另一個頁面，該頁面也不會繼承來源頁面的任何背景設定值。透過主題新增或變更背景時，需要對主題進行更新。

使用者可以在網站的每個頁面上，以及使用相同版面配置的不同頁面上，設定不同的背景。主題則無法這麼做。雖然主題可以指定頁面的背景，但使用者的背景設定值會覆寫這項指定。(背景設定值只會針對特定頁面套用覆寫；不會以任何方式修改主題本身。)

請小心避免讓主題覆寫使用者將在網站頁面與槽位中設定的背景樣式。有數種方式可能導致發生此情況：

- 當您在 `<body>` 與槽位中使用元素型 `"style"` 屬性時，主題便可覆寫網站中的頁面背景設定值。
- 當您在主題階層式樣式表 (CSS) 檔案中將主題背景樣式標示為 `"!important"` 時，主題便可覆寫網站中的頁面背景設定值。

請參閱 [運用 Oracle Content Management 建置網站中的變更背景或主題](#)。

## 背景的實行方式

Oracle Content Management 以動態方式在頁面的 `<head>` 中建立 CSS 樣式表，實行頁面和槽位背景功能。

頁面背景設定值的樣式選取器是一種標記式選取器 (`body`)。至於槽位背景設定值和 ID 式選取器，則是使用槽位 ID。

例如，設定頁面的背景色彩時，可能會在頁面的 `<head>` 中產生下列 CSS 標記：

```
body
{
```

```
    background-color: #fa7c9d;
}
```

同樣地，設定槽位的背景影像時，可能會在頁面的 <head> 中產生下列 CSS 標記：

```
#PageFooter
{
    background-image: url("footer_image.png");
}
```

這項實行所代表的意義是，直接在 **body** 標記或槽位元素的 "style" 屬性中指定樣式，可覆寫樣式表的 <head> 程式碼中設定的設定值。

### ❗ 重要事項：

主題開發人員應注意不要讓背景設定值被元素型樣式所覆寫。

請參閱 *運用 Oracle Content Management 建置網站* 中的變更背景或主題。

## 設定值的儲存位置

頁面呈現後，會將背景設定值動態寫入該頁面 <head> 程式碼中的 "style" 標記。

背景設定值會保存於頁面模型檔案中 (例如 <pageid>.json)。具體來說，頁面背景設定值會儲存在 properties.styles 區段，槽位背景設定值則儲存在 slots[<slot\_id>].styles 區段。

背景設定值會儲存在頁面 JSON 檔案中的 "styles" 內，如以下的代表性範例所示。

```
{
  "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>
</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"
        }
    }
}
```

## 設定觸發程式和動作

您可以設定元件 (包括在內嵌框架中呈現的元件) 之間的通訊，讓某個元件內的觸發程式呼叫另一個元件上的動作。

- [關於觸發程式和動作](#)
- [設定觸發程式](#)
- [設定動作](#)

### 關於觸發程式和動作

您可以設定元件 (包括在內嵌框架中呈現的元件) 之間的通訊，讓某個元件內的**觸發程式**呼叫另一個元件上的**動作**。

觸發程式是 **Oracle Content Management** 元件間通訊的一部分。任何元件都可引發任意數目的觸發程式。元件可以為觸發程式提供有效負載，然後有效負載會被傳送給引發該觸發程式時所執行的任何動作。您可以為每個觸發程式選取將被執行的動作。元件若是建立成一起運作，便可自動引發觸發程式，對另一個元件執行動作，而無需使用者互動。

基本程序包括：

1. 註冊觸發程式
2. 引發觸發程式
3. 註冊動作
4. 執行動作以確認設定

例如，您可以使用按鈕元件執行一或多個動作，像是顯示或隱藏頁面元件及顯示訊息。您可以在一個元件中包含公司辦公室位置清單，而當使用者按一下清單中的某個位置時，就在另一個元件中顯示該位置的相關詳細資訊。

就您的自訂元件而言，觸發程式和動作為元件註冊資料的一部分，而不是元件實行的一部分。在註冊資料中，有 "triggers":[]，和 "actions":[]，項目，當中包含元件所支援的觸發程式和動作清單。本機與遠端元件的實際語法相同，只有位置和擷取方式不同。

請參閱 *運用 Oracle Content Management 建置網站* 中的使用觸發程式和動作。

### 設定觸發程式

元件可包含將在其他元件中執行動作的觸發程式。您必須註冊觸發程式，元件才能夠引發觸發程式。

元件提供有效負載給觸發程式，這會傳送到觸發程式引發時所執行的任何動作。您可以為每個觸發程式選取將被執行的動作。為共同運作而建立的元件，可以在無須使用者互動的情況下，自動引發觸發程式來執行動作。

## 註冊觸發程式

就自訂元件而言，觸發程式會註冊成為元件之註冊資料的一部分。若要新增觸發程式，請更新元件支援之各個觸發程式的 "triggers" 特性陣列。此外，還必須指定觸發程式支援的有效負載，如此才能建立使用者介面，讓使用者可以將有效負載內的值對應至動作所支援的特性。

1. 編輯 `appinfo.json` 檔案並複查 `"triggers":[]`，項目：

```
"triggers": [{
  "triggerName": "helloWorldWhoAreYou",
  "triggerDescription": "Show Who I Am",
  "triggerPayload": [{
    "name": "whoAreYou",
    "displayName": "Who I Am"
  }]
}],
```

2. 將檔案同步到網站伺服器。

在此範例觸發程式項目中，您定義了 `triggerName` ("helloWorldWhoAreYou")。此名稱值必須是唯一的。接著，提供觸發程式的描述 ("Show Who I Am")，供使用者介面對話方塊用來顯示您的觸發程式。最後，為觸發程式定義單一值有效負載；使用者將可以選取此有效負載中的項目，將項目對應至動作的欄位。

觸發程式註冊之後，當您前往元件「設定值」面板中的**連結**頁籤時，應該就能夠見到並選取該觸發程式。

## 引發觸發程式

元件可在任意點引發觸發程式。觸發程式通常由使用者互動引發，例如點選按鈕或選取表格中的資料列。元件可根據任意條件引發觸發程式，例如因 REST API 呼叫而造成資料變更時。您可以在觸發程式引發後執行任意數目的動作。

以下是如何引發觸發程式的範例：

1. 編輯 `render.js` 檔案並在 `viewModel` 物件中新增 JavaScript 函數，此函數將會呼叫 Sites SDK 來引發觸發程式。

```
self.raiseTrigger = function (triggerName) {
  SitesSDK.publish(SitesSDK.MESSAGE_TYPES.TRIGGER_ACTIONS, {
    'triggerName': 'helloWorldWhoAreYou',
    'triggerPayload': { "whoAreYou": "This is " + self.whoAreYou()
+ "!" }
  });
};
```

2. 在使用者介面中新增一個項目，用以呼叫函數來引發觸發程式 (`-edit template.html`)，並且在 `</div>` 之前新增一個按鈕。

```
<button data-bind="click raiseTrigger">Who Am I?</button>
```

3. 將 `render.js` 檔案同步或上傳至您的 Oracle Content Management 執行處理伺服器。

在 ViewModel 物件中，您建立了一個點選按鈕後所呼叫的 JavaScript 函數。此函數會呼叫 Sites SDK 並指示它觸發為此觸發程式 "helloWorldWhoAreYou" 定義的所有動作。此函數也透過包含單一欄位 ("whoAreYou") 的 triggerPayload 傳送。"helloWorldWhoAreYou" 和 "whoAreYou" 這兩個值會是您在上一個步驟中註冊觸發程式時輸入的值。

#### 備註：

沒有預先定義的動作執行順序。雖然動作會依所列表的順序呼叫，但並不會等待動作完成後才呼叫下一個動作。如果某個動作進行非同步呼叫，在下一個動作執行前它也許還沒完成。

## 設定動作

您可以設定讓元件利用動作註冊，以便將其拖放至要執行元件內動作的頁面上。

### 註冊動作

觸發程式引發之後，便會在元件上呼叫動作。元件可以註冊任意數目的動作，也可以定義動作所支援的有效負載。當使用者選取動作時，可以填入要傳送至動作的有效負載。

和註冊觸發程式一樣，您可以在主題的 appinfo.json 註冊資料中註冊元件所支援的動作。

以下是如何註冊動作的範例：

1. 編輯元件的 appinfo.json 檔案，並更新 "actions": [], 項目。

```
"actions": [{
  "actionName": "helloWorldChangeWhoIAm",
  "actionDescription": "Change Who I Am",
  "actionPayload": [{
    "name": "whoAreYou",
    "description": "Who are you?",
    "type": {
      "ojComponent": {
        "component": "ojInputText"
      }
    },
    "value": ""
  }]
}]
```

2. 註冊之後，便可在元件「設定值」面板的連結頁籤中，在按一下觸發程式後所呼叫的動作對話方塊中看到此動作。

### 執行動作

動作註冊之後，您便可以將元件拖放至要執行元件內動作的頁面上。要執行動作的元件必須監聽 EXECUTE\_ACTION 訊息。此訊息同時包含傳送至動作的有效負載 (必須從其中擷取預期值)。

例如，若要監聽 EXECUTE\_ACTION 訊息，請編輯 `render.js` 檔案並更新 `ViewModel` 物件的這些項目：

```
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);
  }
};
```

這會建立一個執行動作的 **JavaScript** 函數，然後在每次引發 EXECUTE\_ACTION 訊息時使用 **Sites SDK** 呼叫此函數。

每當引發 EXECUTE\_ACTION 訊息時便會呼叫此動作，並且由設計只能用來處理此動作的元件進行處理。為此，您必須檢查動作的名稱，以確保其為可處理的動作。

動作的有效負載是一個值陣列。在此範例中，假設值是陣列中的第一個項目。您通常必須從陣列中找尋在意的有效負載值。

 **備註：**

動作監聽器是一種回呼，因此您應使用 **JavaScript** 閉包或適當地連結函數，以確保在執行函數後能夠存取 `ViewModel`。

# 20

## 開發樣板

樣板套件包含網站的開發版本、具有頁面版面配置、樣式及導覽的主題，以及網站中使用的相關元件。**Oracle Content Management** 提供一組可用於建立網站的樣板，同時也可作為建立自訂樣板的起點。

- [關於樣板](#)
- [基本樣板結構](#)
- [建立樣板](#)
- [匯出樣板](#)
- [匯入樣板](#)
- [使用入門樣板](#)
- [從 \*\*Bootstrap\*\* 或網站設計樣板建立網站樣板](#)
- [使用 \*\*Developer Cloud Service\*\* 開發樣板](#)

## 關於樣板

樣板包含使用者開始建立網站所需的一切項目，包括含有範例頁面與內容的網站、含有樣式設定的主題、導覽、資產 (例如圖像) 以及關聯的元件。

**Oracle Content Management** 提供一些可用來建立網站的樣板。這些樣板通常會在服務起始後由管理員安裝。請參閱 *管理 Oracle Content Management* 中的 設定網站設定值。

每當您建立新網站時，都必須選取一個樣板。樣板將主題與網站及元件結合，以驅動某項功能或解決方案，例如合作夥伴入口網站或行銷促銷活動。

如需內建樣板清單，請參閱[瞭解樣板](#)。

開發人員的主要工作之一就是設計新樣板，不過開發人員和使用者都能建立新樣板及修改並取代現有樣板。此程序基本上包含以下這些步驟。

1. 複製現有樣板 (例如 **JET** 入門樣板) 來建立一個新樣板。這同時將提供與該樣板關聯的主題。

例如：

```
cec create-template My_JET_Template -f JETStarterTemplate
```

2. 將樣板以 `.zip` 檔形式匯至您的開發環境。
3. 開啟樣板套件中的檔案，然後進行變更。
4. 以 `.zip` 檔形式建立修訂過的樣板套件。
5. 使用 **Oracle Content Management** 介面將新樣板匯入您的執行處理。
6. 共用樣板，讓其他人也可以使用它。

或者，也可以使用 **Oracle Content Management** 介面修改樣板，方法為新增及修改頁面版面配置和資產、擴充網站結構、新增元件至網站頁面，以及新增內建內容 (以顯示在任何使用此樣板主題的網站)。

另請參閱[管理樣板](#)。

## 基本樣板結構

樣板的基本結構包括一個網站 (含有資產、版面配置、頁面與內容)、一個關聯的主題以及任何自訂元件。

建立網站時，您必須選擇一個提供網站結構和最初內容的樣板、一個含有設計和版面配置設定的主題，以及任何自訂元件。

樣板以特定結構安排，以下範例顯示基本的資料夾與檔案。

```
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
```

元件和主題資料夾與檔案在本指南的其他章節會有詳細說明。請參閱[關於開發元件](#)和[基本主題結構](#)。

#### 附註：

- 主題一般會在樣板之間共用，但「JET 入門樣板」或「入門樣板」使用的則是主題的複本。請參閱[使用入門樣板](#)。
- 主題已不再包含網站。網站資料夾和檔案都在 `/template` 資料夾中。

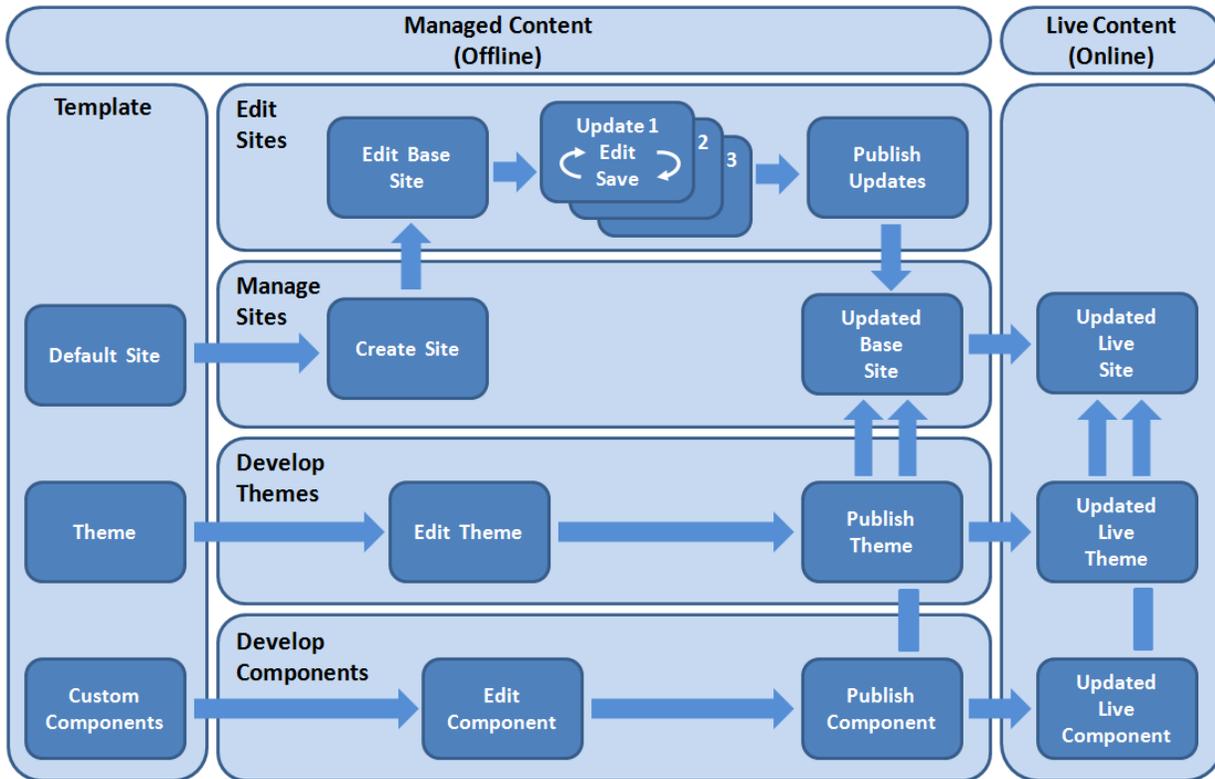
`template_name/template` 資料夾包含網站的資料夾和檔案。

- `assets`：包含使用者介面的樣板詳細資訊頁面中顯示的影像。
- `content`：包含網站中使用的受管理內容。
- `layouts`：目前未使用。
- `pages`：包含所有頁面 JSON 檔案與資料。使用 `nnn.json` 格式，其中 `nnn` 為頁面 ID。
- `variants`：包含網站的所有更新詳細資訊。
- `_folder.json`：包含樣板的描述資料，例如網站作者、網站名稱、項目 GUID、網站的簡要和完整描述。
- `_folder_icon.png`：代表使用者介面中的網站。
- `componentsused.json`：(已不再使用)。記錄網站內使用的自訂元件 (若有的話)。僅保留供回溯相容性之用。
- `controller.html`：包含在瀏覽器中顯示網站的主要程式碼。若要變更此檔案，Oracle 建議您透過介面從網站設定值進行變更。您可以離線修改此檔案。請參閱[自訂控制器檔案](#)。
- `siteinfo.json`：指定網站名稱和關聯主題的名稱，以及網站的其他描述資料。請勿修改此檔案。
- `structure.json`：定義網站的頁面階層 (父項和子項頁面)。建立網站的導覽時，可以使用呈現 API 來描繪樹狀結構。請參閱[網站導覽及呈現 API 參考](#)。

## 建立樣板

如果您的網站要當成其他網站的起始點，則可以從該網站建立樣板。您也可以藉由複製現有樣板並對複本進行變更，建立新的樣板。

如果您從現有網站建立樣板，則新樣板會使用網站的複本作為其預設網站。樣板會參照網站所使用的主題，以及網站頁面中所使用的任何自訂元件。主題和自訂元件並不會複製到樣板中，而是以被網站參照的相同方式進行參照。樣板會反映建立樣板時所使用的網站。隨樣板儲存的網站中並不會反映對建立樣板時所使用之網站所做的進一步變更。



如果您是藉由複製現有樣板並將複本重新命名來建立新樣板，即是對該複本進行變更。請注意，複製樣板時並不會複製該樣板的共用資訊。

請勿使用下列作為樣板、主題、元件、網站、或網站頁面的名稱：`authsite`、`content`、`pages`、`scstemplate_*`、`_comps`、`_components`、`_compsdelivery`、`_idcservice`、`_sitescloud`、`_sitesclouddelivery`、`_themes`、`_themesdelivery`。您可以使用下列名稱作為網站頁面的名稱，但不可以作為樣板、主題、元件或網站的名稱：文件、網站。

如果您想要建立自己的自訂樣板（含網站與主題），最好是使用 **JET 入門樣板** 或 **Oracle Content Management** 提供的入門樣板，當中會包含網站及相關主題的基本元素。入門樣板的網站頁面中包含了各式資訊和指示，可協助您探索如何在自訂樣板中為網站與主題進行版面配置和設計。

請參閱 [管理樣板](#) 和 [使用入門樣板](#)。

## 匯出樣板

您可以匯出樣板並離線進行修改，然後再將它匯入成新樣板或取代現有樣板。您也可以匯出樣板，並將它移到另一個 **Oracle Content Management** 執行處理，然後在該處加以匯入。

匯出樣板時，基本上將樣板以單一 `.zip` 檔案的形式複製到 **Oracle Content Management** 資料夾。您可以直接從該資料夾下載樣板套件，將其解壓縮並處理個別的

檔案。完成處理樣板檔案之後，請建立一個包含該樣板封裝項目的 .zip 檔案，並將其匯入 Oracle Content Management，以覆寫原始樣板或建立新樣板。

### 備註：

匯出樣板時，不會包括該樣板的共用資訊。

匯出樣板：

1. 按一下 Oracle Content Management 側邊導覽中的**開發人員**。  
便會顯示**開發人員**頁面。
2. 按一下**檢視所有樣板**。  
就會顯示現有樣板清單。
3. 選取樣板並從滑鼠右鍵功能表中選擇**匯出**，或按一下動作列中的 。
4. 瀏覽至某個資料夾，或者按一下**建立**、提供名稱和選擇性的描述，然後按一下**建立**來建立新資料夾。  
若要開啟資料夾，請按一下資料夾圖示或資料夾名稱。
5. 按一下關聯資料夾的核取方塊以選取資料夾，然後按一下**確定**。  
隨即會在選取的資料夾中以樣板名稱和 .zip 副檔名建立一個樣板套件檔案。

## 匯入樣板

您可以匯出樣板並離線進行修改，然後再將它匯入成新樣板或取代現有樣板。您也可以匯出樣板，並將其移到另一個 Oracle Content Management 執行處理，然後在該處加以匯入。

匯出樣板時，基本上將樣板以單一 .zip 檔案的形式複製到 Oracle Content Management 資料夾。您可以直接從該資料夾下載樣板套件，將其解壓縮並處理個別的檔案。完成處理樣板檔案之後，請建立一個包含該樣板封裝項目的 .zip 檔案，並將其匯入 Oracle Content Management，然後覆寫原始樣板或建立新樣板。

匯入樣板套件：

1. 按一下 Oracle Content Management 側邊導覽中的**開發人員**  
便會顯示**開發人員**頁面。
2. 按一下**檢視所有樣板**。  
就會顯示現有樣板清單。
3. 按一下**建立**，然後選擇**匯入樣板套件**。
4. 如果您已上傳樣板套件，請瀏覽至包含該樣板套件的資料夾。若要開啟資料夾，請按一下資料夾圖示或資料夾名稱。  
如果您還未上傳樣板套件：
  - a. 瀏覽至想要上傳樣板套件的資料夾，或者按一下**新建**、提供名稱和選擇性的描述，然後按一下**建立**來建立資料夾。
  - b. 按一下**上傳**。

- c. 找出並選取該樣板套件，然後按一下**開啟**。  
進度列就會顯示檔案名稱和上傳狀態。
5. 按一下檔案名稱旁邊的核取方塊來選取樣板套件，然後按一下**確定**。  
如果匯入之樣板的內容與任何現有的樣板、主題或自訂元件之間沒有任何衝突，系統就會為樣板、其關聯的主題以及任何自訂元件建立新的 **Oracle Content Management** 資料夾。
6. 如果樣板、主題或者自訂元件名稱或 ID 已經存在，系統就會提示您解決衝突。  
依據衝突之性質的不同，系統會提供您建立新樣板、主題或自訂元件的選項，或在某些情況下，您也可以用匯入的版本覆寫現有的樣板、主題或自訂元件。

## 使用入門樣板

您可以使用 **Oracle Content Management** 提供的入門樣板複本，建立具有主題和自訂元件的新網站。

建置自訂網站所需的所有項目，全都收集在入門樣板的一個套件中，這些項目包括：預設網站、版面配置、導覽、範例內容、主題、相關內容項目等等。入門樣板與 **Oracle Content Management** 隨付的其他樣板不同，提供您易於使用的基本架構來建立新網站，包括新主題。



### 備註：

主題通常會在樣板之間共用，而入門樣板使用的是主題複本。

**StarterTemplate** 和 **JETStarterTemplate** 入門樣板是管理員在安裝並設定 **Oracle Content Management** 時，於該服務中隨其他樣板一起提供的樣板。**Oracle JavaScript Extension Toolkit (JET)** 入門樣板包含最新的 **JET** 樣板樣式、併入一些頁面內容 (以 **JET** 元件的形式)，並提供可建置 **JET** 型樣板與網站的入門元件。

### 基本程序

以下是使用入門樣板的基本程序：

1. 建立新網站並選取入門範本。新入門範本會隨新網站一起建立。



### 備註：

請謹慎選擇網站名稱。您賦予的網站名稱會複製為新主題的名稱，而使用者將可在您發布網站後看見該主題。主題建立後，您將無法變更主題名稱。

入門網站與主題包含一組開始網站和主題開發所需的資料夾與檔案。

2. 將新主題同步至您的桌面。您可以在桌面上工作，以使用版面配置和靜態資產來擴充和自訂主題。由於這是主題的複本且您的樣板使用特定名稱，因此您可以對主題進行變更，而不會影響到來源主題。

3. 在「網站產生器」中開啟網站。您可以使用「網站產生器」新增網站結構、將元件和互動新增至頁面，以及新增將成為新樣板預設網站之一部分的內容。您可以重複使用或移除入門樣板中提供的網站內容。
4. 當網站與主題準備就緒時，將變更與您的 Oracle Content Management 執行處理同步，然後從選取的網站建立新樣板。
5. 共用樣板，讓其他人也可以使用它。

### 使用入門樣板來建立樣板

使用入門樣板：

1. 按一下 Oracle Content Management 側邊導覽中的**網站**。  
就會顯示現有網站清單。
2. 按一下**建立**。
3. 在「建立網站」對話方塊中，選取要作為您網站基礎的入門樣板。  
新網站會使用入門樣板隨附的主題，您必須加以重新命名以符合新網站的名稱。
4. 在此對話方塊中，輸入網站的名稱。網站 URL 中會使用此名稱。名稱中可以包含字母、數字、底線 (\_) 以及連字號 (-)。如果您輸入空格，系統會自動以底線取代。

請勿使用下列作為樣板、主題、元件、網站、或網站頁面的名稱：`authsite`、`content`、`pages`、`scstemplate_*`、`_comps`、`_components`、`_compsdelivery`、`_idcservice`、`_sitescloud`、`_sitesclouddelivery`、`_themes`、`_themesdelivery`。您可以使用下列名稱作為網站頁面的名稱，但不可以作為樣板、主題、元件或網站的名稱：文件、網站。

#### 備註：

Oracle Content Management 網站 URL 的路徑區分大小寫。查詢或片段字串中的大小寫由開發人員在其自訂程式碼中控制。

```
https://host[:port]][/]path[?query][#fragment]
      \_____/      \_____/
      Location      Data
```

5. (選擇性) 輸入網站的描述。
6. 準備就緒時，請按一下**建立**。

畫面會出現一個進度列，顯示新網站的名稱和建立狀態。網站建立之後，其名稱就會顯示在網站清單中。它的最初狀態會是離線。

若要在清單中快速找到新建立的網站，請依**上次更新時間**排序清單。您剛剛建立的網站將會出現在清單頂端。

#### 備註：

系統會自動將您所建立網站的管理員角色指定給您。

7. 使用桌面應用程式將主題同步至您的桌面。

您現在應該會看到主題的資料夾階層與檔案。

8. 入門主題包含一組最基本的資料夾與檔案，如下：

```
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` 檔案是 **Cookie** 許可即現式視窗的預設關閉圖示。

`/layouts` 資料夾包含一個入門頁面版面配置檔案 (`index.html`)，其中具有下列內容：

- 一組允許使用檔案作為頁面版面配置的 **HTML** 標記。
- 一個內建指示文字的單一槽位，像是如何將主題同步至您的桌面、如何新增頁面版面配置、如何將元件新增至頁面版面配置，以及如何使用新頁面版面配置來建置網站階層。
- 一個簡單的 **JavaScript** 導覽檔案範例，說明如何使用呈現器 **API JavaScript** 函數和物件。在網站階層間穿梭以及產生網站導覽功能所需的 **HTML** 標記，都必須使用呈現 **API**

另請參閱[基本主題結構](#)。

9. 完成主題修改時，將主題資料夾與檔案同步至您的 **Oracle Content Management** 執行處理。若要查看主題的變更對網站外觀與行為的影響，請在「網站產生器」中開啟網站。您可能需要反覆執行此處理作業。
10. 若要檢視或修改網站，請選取網站並從滑鼠右鍵功能表中選擇**開啟**，或按一下動作列中的 。
11. 將編輯器模式切換至**編輯**，以便對網站進行變更。您可以使用入門主題中提供的頁面版面配置，以修改現有的頁面和新增頁面。

 **備註：**

如果您熟悉頁面版面配置的結構和使用方式，可以刪除入門樣板中所提供但您不想要的區段，並將想要區段的版面配置切換至其中一種新的版面配置。

12. 完成對網站的變更時，**儲存**網站，然後按一下**發布**以將更新合併至基本網站。
13. 選取網站並根據該網站建立樣板。這會為新樣板導入資產與主題。
14. 與成員共用樣板，讓其他成員也可以使用此樣板。

## 從 Bootstrap 或網站設計樣板建立網站樣板

開放的 Oracle Content Management 架構代表可以使用在其他程式碼架構 (例如 Foundation 或 Bootstrap) 完成的工作。只要少許變更，便可以將 Bootstrap 樣板轉換為主題，讓其成為 Oracle Content Management 樣板的一部分。

### 基本程序

以下是之後各節中詳述的步驟概要：

1. [先決條件](#)
2. [建立網站](#)
3. [同步主題資料夾](#)
4. [建立基本主題](#)
5. [更新網站頁面](#)
6. [更新導覽](#)
7. [更新網站版面配置](#)
8. [發布網站](#)
9. [建立新樣板](#)

### 先決條件

- 安裝並執行 Oracle Content Management 桌面應用程式，以便將資料夾和檔案同步至您的本機電腦。
- 將 Bootstrap 樣板主題資料夾、檔案及內容下載至您的本機電腦備用。

### 建立網站

從 Oracle Content Management 入門樣板建立網站：

1. 按一下側邊導覽中的**網站**。  
就會顯示現有網站清單。
2. 按一下**建立**。
3. 在「建立網站」對話方塊中，選取 **JET 入門樣板**或**入門樣板**，作為網站的基礎。
4. 在此對話方塊中，輸入網站的名稱。網站 URL 中會使用此名稱。名稱中可以包含字母、數字、底線 ( \_ ) 以及連字號 (-)。如果您輸入空格，系統會自動以連字號取代。

請勿使用下列作為樣板、主題、元件、網站、或網站頁面的名稱：authsite、content、pages、scstemplate\_\*、\_comps、\_components、\_compsdelivery、\_idcservice、\_sitescloud、\_sitesclouddelivery、\_themes、\_themesdelivery。您可以使用下列名稱作為網站頁面的名稱，但不可以作為樣板、主題、元件或網站的名稱：文件、網站。

 **備註：**

Oracle Content Management 網站 URL 的路徑區分大小寫。查詢或片段字串中的大小寫由開發人員在其自訂程式碼中控制。

```
https://[host[:port]][/]path[?query][#fragment]
      \_____/      \_____/
      Location      Data
```

5. (選擇性) 輸入網站的描述。
6. 準備就緒時，請按一下**建立**。

畫面會出現一個進度列，顯示新網站的名稱和建立狀態。網站建立之後，其名稱就會顯示在網站清單中。它的最初狀態會是離線。

若要在清單中快速找到新建立的網站，請依**上次更新時間**排序清單。您剛剛建立的網站將會出現在清單頂端。

### 同步主題資料夾

從入門樣板建立網站時，會建立一個入門樣板主題的複本，並使用網站名稱加上主題名稱命名。例如，My\_New\_Site 的主題為 My\_New\_SiteTheme。

使用桌面應用程式將網站的主題資料夾和檔案同步至您的本機電腦。請參閱**運用 Oracle Content Management 進行文件協作**中的開始同步。

您就會在本機電腦上見到樣板主題資料夾階層和檔案。以下為範例：

```
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
```

### 建立基本主題

1. 將這些 **Bootstrap** 檔案複製到本機電腦上已同步的主題資料夾中，以覆寫現有的檔案。
  - html 檔案會複製到 `theme_name/layouts` 資料夾中
  - css 檔案會複製到 `theme_name/assets/css` 資料夾中
  - js 檔案會複製到 `theme_name/assets/js` 資料夾中
  - image 檔案會複製到 `theme_name/assets/images` 資料夾中，而且可依照背景、註腳、人物，將影像歸入所屬的子資料夾中
2. 修改 layout 資料夾中的 html 檔案，以更新相對路徑並增加必要元素。就典型的 **Bootstrap** 主題而言，`/assets` 資料夾會有許多相對路徑，因此您必須加以修改以指向主題資料夾。

請修正所使用 `css`、`js` 和 `images` 資料夾的路徑：

```
_scs_theme_root_/assets/css/  
_scs_theme_root_/assets/js/  
_scs_theme_root_/assets/images/
```

#### 備註：

此步驟完成之後，`_scs_theme_root` 部分將會自動調整為使用主題的環境。

3. 每個 **Oracle Content Management** 版面配置都有三個需要注意的地方：
  - a. html 檔案的 `<head>` 標記中需包含下列標記：
    - `<!--$SCS_RENDER_INFO-->`
    - `<!--$SCS_SITE_HEADER-->`
    - `<!--$SCS_PAGE_HEADER-->`
  - b. 版面配置檔案結尾的 `<body>` 標記內需包含呈現器命令檔。這兩個路徑在「網站產生器」和程式實際執行環境中都會自動調整。

```
<script data-main="/_sitescloud/renderer/renderer.js" src="/  
_sitescloud/renderer/require.js"></script>
```

- c. `renderer.js` 檔案的內容之後需包含下列標記：
  - `<!--$SCS_PAGE_FOOTER-->`
  - `<!--$SCS_SITE_FOOTER-->`

確認修改過的主題檔案與 **Oracle Content Management** 中的網站同步。

## 更新網站頁面

在「網站產生器」中開啟網站時，最初見到的是預設網站中的頁面。您可以刪除不想要的頁面，並將想要保留頁面的版面配置切換成其中一個新的版面配置。

1. 按一下側邊導覽中的**網站**。  
就會顯示現有網站清單。
2. 選取網站並從滑鼠右鍵功能表中選擇**開啟**，或按一下動作列中的 。
3. 輸入更新的名稱和選擇性的描述，然後按一下**建立**。  
針對更新項目名稱，您可以使用字母、數字、底線 ( \_ ) 和連字號 ( - )。如果您輸入空格，系統會自動以連字號取代。  
  
如果已經有網站的更新，請由清單中選取一個更新，然後按一下 。
4. 「網站產生器」便會以預覽模式開啟。若要在提要欄位中變更或使用導覽選項，請確定「編輯」開關  設為**編輯**。
5. 若要編輯特定頁面，請使用提要欄位中的網站樹狀結構或使用網站的專屬導覽來選擇頁面。
6. 若要移除不想要的頁面，請選取該頁面，然後按一下 。
7. 若要新增頁面，請按一下**新增頁面**。若要調整頁面在網站樹狀結構中的位置，請拖放頁面。
8. 若要變更頁面的版面配置，請從網站樹狀結構中選擇該頁面，然後按一下  以顯示頁面設定值。  
前往**頁面版面配置**欄位，然後從功能表中選取其他版面配置。頁面版面配置的數目和類型取決於網站所關聯的主題。
9. 按一下**儲存**以儲存目前更新的變更。您可以繼續對目前的更新進行作業，或建立新的更新 (若有需要)。

## 更新導覽

在網站預覽查看的導覽並不會與目前的階層相同，那是因為 **Bootstrap** 主題之版面配置中的導覽採用硬式編碼。

請更新導覽，將 **Bootstrap** 主題中的硬式編碼程式碼取代為從網站階層動態產生的程式碼。

編輯本機的主題檔案同步複本。

1. 移除版面配置的硬式編碼導覽程式碼。以下是必須從典型之 **Bootstrap** 主題移除的硬式編碼導覽範例。這是典型的標頭區段，包含標誌、頁面太窄時之「漢堡」功能表的「切換導覽」部分 (響應式部分)，以及導覽至其他頁面的硬式編碼頁面導覽。

```
<header id="header" class="header navbar-fixed-top">
  <div class="container">
    <h1 class="logo">
      <a href="index.html"><span class="text">Velocity</span></a>
    </h1><!--//logo-->
    <nav class="main-nav navbar-right" role="navigation">
```

```

        <div class="navbar-header">
            <button class="navbar-toggle" type="button" data-
toggle="collapse" data-target="#navbar-collapse">
                <span class="sr-only">Toggle navigation</span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
            </button><!--//nav-toggle-->
        </div><!--//navbar-header-->
        <div id="navbar-collapse" class="navbar-collapse
collapse">
            <ul class="nav navbar-nav">
                <li class="active nav-item"><a
href="index.html">Home</a></li>
                <li class="nav-item"><a
href="features.html">Features</a></li>
                <li class="nav-item"><a
href="pricing.html">Pricing</a></li>
                <li class="nav-item dropdown">
                    <a class="dropdown-toggle" data-
toggle="dropdown" data-hover="dropdown" data-delay="0" data-close-
others="false" href="#">Pages <i class="fa fa-angle-down"></i></a>
                    <ul class="dropdown-menu">
                        <li><a href="download.html">Download
Apps</a></li>
                        <li><a href="blog.html">Blog</a></li>
                        <li><a href="blog-single.html">Blog
Single</a></li>
                        <li><a href="blog-category.html">Blog
Category</a></li>
                        <li><a href="blog-archive.html">Blog
Archive</a></li>
                        <li><a href="about.html">About Us</a></li>
                        <li><a
href="contact.html">Contact</a></li>
                    </ul>
                </li><!--//dropdown-->
                <li class="nav-item"><a href="login.html">Log
in</a></li>
                <li class="nav-item nav-item-cta last"><a
class="btn btn-cta btn-cta-secondary" href="signup.html">Sign Up
Free</a></li>
            </ul><!--//nav-->
        </div><!--//navabr-collapse-->
    </nav><!--//main-nav-->
</div><!--//container-->
</header><!--//header-->

```

2. 撰寫 JavaScript 程式碼以瀏覽網站結構資訊並產生導覽程式碼，然後將此 JavaScript 程式碼包含到版面配置，例如：

```

<script type="text/javascript" src="_scs_theme_root_/assets/js/
navbar.js"></script>

```

3. 根據特定 CSS 應有的標記，修改 topnav.js 檔案的輸出。

## 更新網站版面配置

此時網站雖可運作，但版面配置中並無可編輯的區域 (槽位)。

1. 在版面配置中找出或新增 DIV 元素，並將其指定為槽位。

**槽位**是版面配置中的 DIV 元素，在類別屬性中的值為 "scs-slot"。每個槽位的 id 屬性都必須是唯一的。例如：

```
<div id="slot-content1" class="scs-slot scs-responsive"></div>
```

若要讓槽位能夠自動根據瀏覽器檢視區的大小調整，請包含類別屬性 "scs-responsive"。

2. 視需要重複此步驟，在此版面配置或其他版面配置中建立其他槽位。

## 發布網站

完成並儲存所有網站變更之後，您必須**發布**網站。

發布網站會取用目前更新的所有變更，並將這些變更合併至基本網站中，讓更新的基本網站成為新的基本網站。

## 建立新樣板

您現在已有一個具備關聯主題的可運作網站，因此您可以讓其成為樣板並與他人共用，讓他們能夠從樣板建立網站。

1. 按一下 Oracle Content Management 側邊導覽中的**開發人員**。
2. 按一下**檢視所有樣板**。
3. 按一下**建立**，然後選擇**從現有網站**。
4. 選取使用入門樣板和 Bootstrap 樣板所建立的新網站。
5. 輸入新樣板的名稱，然後按一下**建立**。
6. 若要封裝樣板以搭配其他 Oracle Content Management 執行處理使用，請選取樣板並選擇**匯出**功能表選項，建立可供下載的 .zip 檔案。

# 使用 Developer Cloud Service 開發樣板

您可以使用 Developer Cloud Service 開發 Oracle Content Management 樣板。

請依照以下步驟在 Developer Cloud Service 中開發樣板、在本機進行測試，然後將樣板匯至 Oracle Content Management：

1. 在本機機器上設定 [OCE 工具程式](#)。
2. 登入 Oracle Content Management 的 [Developer Cloud Service 主控台](#)。
3. 在 Developer Cloud Service 中建立專案。
4. 將 OCE 工具程式新增至新 Git 儲存區域中的專案程式碼。
5. 在 Developer Cloud Service 中建立樣板。

您可以建立新樣板來進行開發、在 [Developer Cloud Service](#) 中複製現有樣板或從 [Oracle Content Management](#) 匯入樣板。

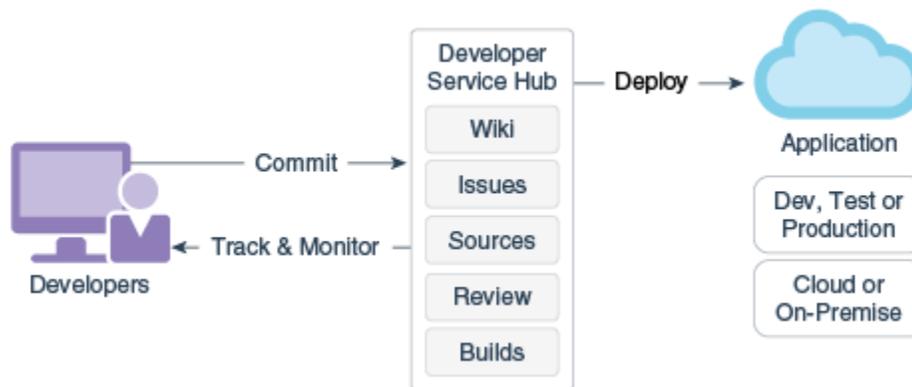
6. 在本機測試控制程式中測試樣板。
7. 合併變更。
8. 將樣板從 [Developer Cloud Service](#) 匯出至 [Oracle Content Management](#)。

## 登入 Oracle Content Management 的 Developer Cloud Service 主控台

在 [Developer Cloud Service](#) 主控台上開始開發適用於 [Oracle Content Management](#) 的自訂元件。

身為 [Oracle Cloud](#) 服務的管理員，您可以透過「我的服務管理」建立及管理您的雲端服務。如果您是 [Oracle Content Management](#) 的服務執行處理管理員和「標準開發人員服務」的服務管理員，便可以設定它們並開始使用：

1. 使用為您帳戶提供的資訊來登入 [Oracle Cloud](#)。
2. 登入「我的服務管理」來建立和管理您的 [Oracle Content Management](#) 執行處理與「標準開發人員服務」。



3. 依要求確認您的 [Oracle Developer Cloud Service](#) 電子郵件。
  4. 使用您服務的訂閱詳細資訊來設定您的 [Oracle Content Management](#) 執行處理，然後前往 [Oracle Content Management](#) 執行處理的 URL。
  5. 前往您的「標準開發人員服務」URL。
  6. 登入 [Oracle Developer Cloud Service](#) 帳戶。
- 存取 [Developer Cloud Service](#) URL 並登入主控台。

## 在 Developer Cloud Service 中建立專案

您可以使用 "Content Experience Cloud" 專案樣板在 [Developer Cloud Service](#) 中建立專案，或者使用空的 [Git](#) 儲存區域建立專案，然後從您的 [Oracle Content Management](#) 執行處理匯入 [OCE](#) 工具程式。

- 使用 [Oracle Content Management](#) 樣板建立 [Developer Cloud Service](#) 專案

- 使用從 [Oracle Content Management](#) 下載的 OCE 工具程式在 Developer Cloud Service 中建立專案
- 將 OCE 工具程式新增至新 Git 儲存區域中的專案程式碼

## 在 Developer Cloud Service 中建立樣板

您可以使用 `cec` 命令行公用程式，從提供的來源樣板建立 Oracle Content Management 樣板。

使用 `cec create-template` 命令，從其中一個可用的來源樣板建立樣板。在命令行輸入 `cec create-template -h` 會顯示可用的來源樣板。

以下是建立樣板的範例：

```
cec create-template CafeSupremoLite_yourname -f CafeSupremoLite
```

**Windows：**此命令會在您建立樣板時，建立一個可在外部 HTML WYSIWYG 編輯器中呈現主題的符號連結 (symlink) (例如 `_scs_theme_root_`)。若要在 Windows 中建立符號連結 (symlink)，您通常需要以管理權限執行命令行公用程式。如果您不是使用 WYSIWYG 編輯器來編輯主題，則無需管理權限即可執行，並請忽略符號連結 (symlink) 建立錯誤。

上述範例會建立 `CafeSupremoLite_yournameTheme` 樣板，並在 `cec-components/src/main/` 提供原始程式碼。下表顯示您在 Developer Cloud Service 中建立網站樣板後的原始程式碼位置。

樣板原始程式碼	主題	樣板的元件
<code>cec-components/src/main/templates</code>	<code>cec-components/src/main/themes</code>	<code>cec-components/src/main/components</code>

本機測試控制程式也會顯示元件，並可讓您依樣板和類型篩選元件。

您可以使用任何文字或程式碼編輯器來編輯主題和元件檔案。請參閱[使用本機測試控制程式進行測試](#)。編輯主題或元件之後，請重新整理瀏覽器以查看您的變更。

### ! 重要事項：

這些樣板、主題及元件的原始程式碼都位於 `src/main/`。您不應修改 `src/main` 之外的任何檔案，因為它們是 Oracle Content Management 本機伺服器運作所需的檔案。

## 在 Developer Cloud Service 中複製樣板

您可以在 Developer Cloud Service 中複製現有 Oracle Content Management 樣板。

若要從 `src/main/templates` 複製您的其中一個現有樣板，請使用 `cec copy-template` 命令。如果樣板包含來自其他儲存區域的資產，請選擇性地提供儲存區域對應，否則將不會複製這些資產。

以下範例會將 Temp1 樣板複製到新的 Temp2 樣板：

```
cec copy-template Temp1 -n Temp2
```

## 將樣板匯入 Developer Cloud Service

您可以從 Oracle Content Management 將樣板匯入 Developer Cloud Service 中，以進行進一步的開發。

若您有從 Oracle Content Management 伺服器建立的樣板壓縮檔，您可以將該檔案匯入 Developer Cloud Service 中做進一步的開發，例如編輯主題或元件。使用以下命令：

```
cec import-template <location of the template zip file>
```

指定 Oracle Content Management 中包含此壓縮檔的資料夾。

請參閱[關於樣板和匯出樣板](#)。

## 合併變更

建立元件、樣板或內容版面配置之後，或是在本機機器上編輯原始程式碼之後，您需要將新的和變更過的元件和樣板合併到專案的 Git 儲存區域中。

若要将變更合併至 Git 儲存區域中，請在終端機視窗中依序輸入下列命令。

```
cd cec-components git pull
git add .
git status
git commit -a -m "Your comments" git pull
git push
```

## 從 Developer Cloud Service 匯出樣板

您可以從 Developer Cloud Service 匯出樣板壓縮檔，然後使用該檔案在 Oracle Content Management 建立網站。

樣板開發完成之後，您可以執行下列命令來匯出樣板。命令回應會通知您樣板壓縮檔在 Oracle Content Management 中的所在位置。

```
cec export-template CafeSupremoLite_yourname
```

請參閱[關於樣板和匯入樣板](#)。

# 21

## 開發主題

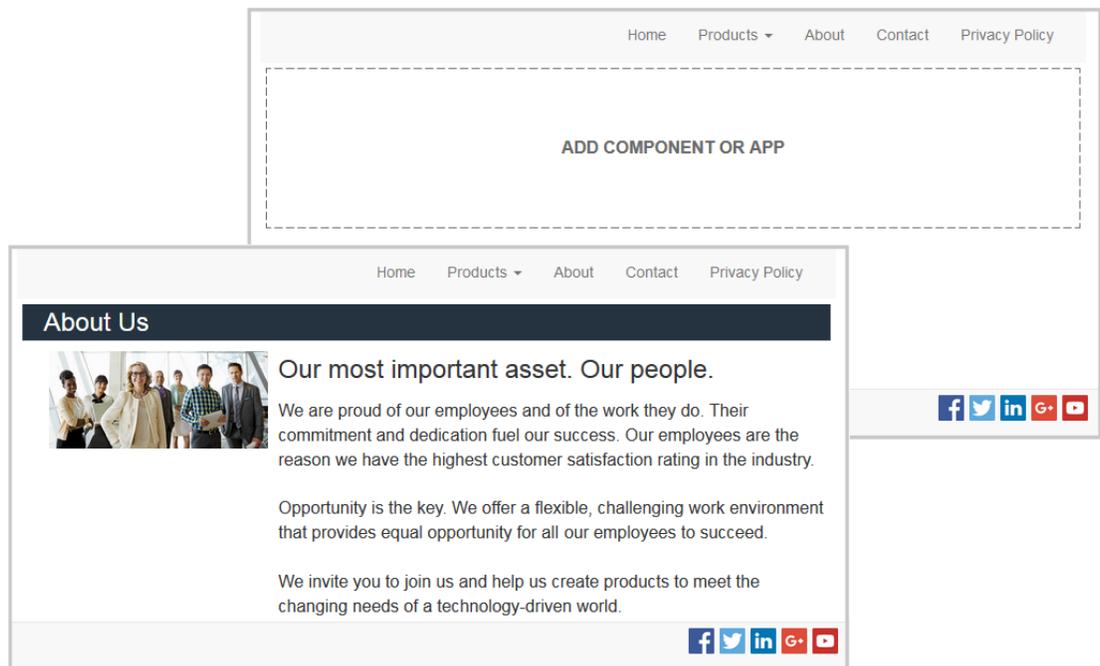
主題定義網站的一般外觀與感覺 (即整體樣式)，包括色彩配置、字型大小、字型類型及頁面背景。主題讓網站各頁面的視覺效果一致。您可以指定設計和範例內容來建立獨特的主題和主題變化，然後使用這些主題建立網站，以推廣您的品牌和願景。

- [關於主題](#)
- [基本主題結構](#)
- [網站導覽](#)
- [建立主題](#)
- [建立元件與主題的關聯](#)
- [Sites Rendering API](#)

## 關於主題

主題定義網站的一般外觀與感覺，包括內容、外觀以及行為。主題讓網站各頁面的視覺效果一致。

設計新主題代表指定版面配置、樣式、範例內容、導覽，以及作為新網站起始點的所有基本資訊。主題設計者需制定期望的網站外觀與行為。設計主題時應謹記主題未來的使用方式；例如，大多數或所有的使用者都將以行動裝置存取網站。您的使用者若想要建立許多類似網站，設計自訂主題會非常實用。您可以使用主題之間可共用的一般樣式頁面版面配置來設計主題。



主題包含用於設計網站內容、外觀與行為的頁面版面配置。您可以變更設計和設定值及新增內容，以建立可推銷風格、品牌和願景的網站。

主題包括：

- 屬於頁面版面配置的背景影像資產或其他內容 (影像、JavaScript 檔案等等)
- 網站的樣式設定值 (CSS)
- 數種頁面版面配置 (HTML 檔案)
- 建構網站導覽的程式碼 (JavaScript 檔案)
- 可搭配元件使用的基本樣式清單 (於 design.css 和 design.json 檔案中指定)

主題也可包含內建資料，以用來填入從其中一個頁面樣板建立的新頁面。例如，使用者可以針對「產品」區段建立一個新頁面，然後選擇 new\_product.html 頁面版面配置。主題若包含 new\_product-pageseed.json 檔案，新頁面在第一次建立之後，將會填入該頁面內建檔的內容。就如同範例內容，此內建資料也可以修改，它的目的僅僅是為建置頁面所提供的起點。

您可以從要搭配主題使用的元件中，建立使用那些元件子集的主題。使用者如果為他們的網站選擇使用該主題，就只會見到為該主題所指定的元件。請參閱[建立元件與主題的關聯](#)。

每個網站都會使用主題。若從樣板建立網站，會繼承樣板的主題。您隨時可以變更網站的主題。Oracle Content Management 提供一些樣板，這些樣板包含可供您用來建立網站的主題。

如果網站使用新的未發布主題，則會在第一次讓網站上線時自動發布網站的主題。如果您變更主題，並且想要更新線上網站來顯示變更，則必須明確地發布主題。只有主題擁有者或具有管理者權限的使用者才能明確地發布主題。

#### 備註：

如果您發布主題變更，則所有使用該主題的線上網站都將反映此變更。例如，如果您變更主題中所指定的預設字型，並發布主題，則所有使用該主題的網站都將使用新的預設字型。

另請參閱[管理主題](#)。

## 基本主題結構

主題的基本結構包括 Oracle Content Management 中儲存之資料夾所指定的設計、導覽以及樣式。主題是網站樣板的一部分。

當使用者選取樣板來建立新網站時，關聯的主題資料便會自動載入。如果您使用「JET 入門樣板」或「入門樣板」，系統會自動複製主題，而非參照主題。開發人員若使用入門樣板，請建立自己的主題複本。

主題以特定的資料夾和檔案結構加以組織，如以下範例的基本資料夾與檔案所示：

```
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
```

某些資料夾包含特定類型的資訊，包括下列資料夾：

- `assets`：JavaScript、階層式樣式表 (CSS)、影像，以及版面配置參照的其他支援檔案。
- `designs`：`design.css` 和 `design.json` 檔案，用於指定元件的樣式選項。
- `layouts`：頁面樣板的 HTML 檔案，用於顯示網站的頁面。

 **備註：**

所有 HTML 檔案的開頭都必須要有一個 DOCTYPE 元素，看起來如下：`<!DOCTYPE html>`

- `publish`：將主題用於發布的網站後，系統就會列出此目錄，其中包含檔案的複本。如果您曾經使用桌面應用程式同步主題，此目錄就會顯示；但主題若是隨著樣板匯出，就不會包含此目錄。
- `responsepages`：用於處理錯誤 (404) 的特殊頁面。

 **備註：**

假如網站中的某個頁面標記為「錯誤頁面」，系統將會忽略來自主題的 404 錯誤訊息，並改用指定的錯誤頁面。

- `viewport.json`：指定主題的檢視區設定值。
- `_folder.json`：指定主題的名稱和 GUID。例如：

```
{
  "themeName": "MarketingCampaignTheme",
  "itemGUID": "TB79D65F699B022AC4E11F4D4EE870070A1ADD86BBBB"
}
```

GUID 由 Oracle Content Management 在主題第一次匯入或複製後建立。主題名稱由主題開發人員在建立主題時指定。

- `components.json`：(已不再使用)。記錄主題內使用的自訂元件。僅保留供回溯相容性之用。

建立新主題時，將會用到兩個重要檔案。這兩個檔案主要設定元件樣式：

- design.css
- design.json

建議將導覽資訊放到一個 JavaScript 檔案中；例如，nav.js 檔案。主題的 /assets/js/ 資料夾對此類檔案而言是很適合的位置。

## 網站導覽

網站的階層儲存在網站的 structure.json 檔案中。階層會載入記憶體中，並且以 SCS.structureMap 物件形式在頁面相關資訊環境中提供使用。

網站產生器會讀取 structure.json 檔案，以在網站產生器中繪製網站樹狀結構。structure.json 檔案將包含網站頁面的程式碼。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
}
```

網站頁面中必須要有導覽 JavaScript 程式碼，才能讀取結構並繪製網站的導覽連結。Oracle Content Management 提供的樣板包含說明其運作方式的範例導覽 JavaScript 檔案。

Oracle Content Management 提供之部分主題中使用的 `topnav.js` 檔案便是一個例子，說明如何搭配呈現 API 呼叫 (例如 `SCSRenderAPI.getPageLinkData`) 使用 `SCS.structureMap` 物件，瀏覽網站結構並且在頁面中繪製出呈現導覽功能表所需的 HTML 標記。以下是範例 `topnav.js` 檔案的程式碼：

```
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();
            }
        }
    );
}
```

您可以使用呈現 API 呼叫，產生可在網站編輯和預覽模式以及已發布之線上網站中正常運作的導覽連結。請參閱[呈現 API 參考](#)。

將導覽資訊放入 JavaScript 檔案 (例如 `topnav.js`) 中會是一個不錯的做法。JavaScript 檔案通常儲存在主題的 `/assets/js/` 資料夾中，就像 Oracle Content Management 所提供的範例主題一樣。

## 建立主題

您可以複製現有主題，然後變更複本來建立新的主題。也可以匯入 Bootstrap 內容並在新主題中使用。

### 備註：

不論是建立新主題還是更新現有主題，請一律複製該主題，然後對其複本進行作業。使用範例網站或真實網站的複本來測試主題，以確保主題可正確運作。請注意，對現有主題所做的變更會在修訂過的主題發布後，立即在使用該主題的所有網站實行。

### 複製主題

1. 在首頁中，按一下**開發人員**。

便會顯示**開發人員**頁面。

2. 按一下**檢視所有主題**。

就會顯示現有的主題清單。您可以按一下檢視圖示並從清單中選取選項，來控制主題的顯示方式。

3. 選取主題並從滑鼠右鍵功能表中選擇**複製**，或按一下動作列中的 。

便會複製主題的所有資料夾和檔案，包括任何範例頁面和內容。

#### **備註：**

當您複製主題時，不會複製主題的共用資訊。

4. 輸入所複製主題的名稱。您不能使用其他主題已經使用的名稱。

網站名稱可包含字母、數字、底線 ( \_ ) 以及連字號 ( - )。如果您輸入空格，系統會自動以連字號取代。

請勿使用下列作為樣板、主題、元件、網站、或網站頁面的名稱：`authsite`、`content`、`pages`、`scstemplate_*`、`_comps`、`_components`、`_compsdelivery`、`_idcservice`、`_sitescloud`、`_sitesclouddelivery`、`_themes`、`_themesdelivery`。您可以使用下列名稱作為網站頁面的名稱，但不可以作為樣板、主題、元件或網站的名稱：文件、網站。

5. (選擇性) 輸入主題的描述。

6. 按一下**複製**。

畫面會出現一個進度列，顯示新的主題名稱和複製狀態。複製了主題之後，其名稱就會出現在主題清單中。按一下主題清單中的主題名稱，即可瀏覽構成主題的資料夾和檔案。

7. 使用 Oracle Content Management 桌面應用程式，將主題資料夾和檔案同步到您的本機系統。如此，您就可以瀏覽本機資料夾並直接使用檔案。您對主題所做的變更會自動同步。您可以使用慣用的 HTML、程式碼或文字編輯工具來進行變更。

### 設定網站內是否可使用自訂樣式

主題管理員可以指定主題中的樣式或自訂樣式。

主題的 `components.json` 檔案中的設定，指定是否可在「網站產生器」設定自訂樣式。這是為了控管讓網站提供者在建置網站時能夠遵守網站的樣式，不會使用其他字型 and 色彩。

若不允許在網站中使用自訂樣式：

1. 隱藏元件**設定值**面板中的**自訂**選項。  
在主題的 `components.json` 檔案中，增加下列物件與其他元件定義：

```
[
  {
    "showStyleClassOnly": "true"
  }
]
```

2. 自訂 CKEditor 的工具列群組和按鈕以移除自訂樣式：

在主題的 `components.json` 檔案中，您可以指定 CKEditor 的 `toolbarGroups` 和 `removeButtons` 特性，以自訂 Title 和 Paragraph 元件的群組和群組內的按鈕。

例如，增加下列物件與其他元件定義，以移除 Styles 群組中的樣式、字型 and 色彩：

```
[
  {
    "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. 避免將樣式貼入 CKEditor 中。

貼上的內容會採用設定的樣式和按鈕。在以上範例中，會將 Font 從貼上的資料中移除。

若不允許使用自訂樣式，會產生以下結果：

- 在所有的內建元件中，**樣式**頁籤會隱藏「自訂」選項。

- 在 CKEditor 中，會隱藏用以設定樣式的工具列按鈕，字型系列、字型色彩等設定都會被覆寫。
- 此外在 CKEditor 中，也無法設定貼上之文字的樣式。可以使用粗體或斜體。

### 隱藏主題的元件和區段版面配置

您可以編輯主題的 `component.json` 檔案來隱藏元件和區段版面配置，讓網站開發人員無法使用它們。建議您這麼做，才能升級為一致的外觀與感覺。請參閱[隱藏主題的元件和區段版面配置](#)。

### 在網站驗證主題

完成編輯主題後，您需要檢查主題是否可如預期般在網站運作。

1. 確定使用 Oracle Content Management 桌面應用程式將複製的資料夾和檔案同步，並且確定所有的修改都已經儲存。
2. 在「網站產生器」中開啟網站 (可以是測試網站或現有的離線網站)，然後將網站切換成使用新主題。
3. 使用此新主題測試網站，方法為新增頁面及使用設定值、檢視影像、檢查導覽，以及新主題中所有已變更的項目。
4. 確定所有項目都正確呈現，包括所有檔案和連結。
5. 發布主題，以便讓網站實行主題變更，然後在程式實際執行環境中檢查該主題。

請參閱[發布主題](#)。

### 使用 Bootstrap 主題

由於 Oracle Content Management 主題與 Bootstrap 主題之間的相似性，因此您可以轉換現有的 Bootstrap 主題頁面和內容，供運用於 Oracle Content Management 主題中。

Bootstrap 是用於建立網站和 Web 應用程式的免費開源工具集合。它包含適用於介面元件和 JavaScript 擴充功能的 HTML 和 CSS 式設計樣板。

您可以依照相同的指示來建立主題，但需透過程式碼 (利用所選擇的編輯器) 匯入及編輯 Bootstrap 主題的任何頁面或內容。

您不太可能會完全使用 Bootstrap 主題與其所有頁面和內容。一般來說，您只會選取主題的幾個部分在 Oracle Content Management 主題中運用，也許只有幾個頁面樣板中的某些部分而已。例如，某個 Bootstrap 主題包含了樣板內的數個區塊，包括標頭、導覽區塊、主體區塊以及頁尾區塊。由於 Bootstrap 主題是以可直接從檔案系統檢視 (不需要 Web 伺服器) 的檔案集合形式提供，因此它們的每個頁面內包含許多重複項目 (因為未使用頁面組件引擎，所以這是必然的)。當您在 Oracle Content Management 主題中使用這項資訊時，您需要取得這些部分並將它們加到可重複使用的 Oracle Content Management 頁面樣板，然後使用 Oracle Content Management 將這些樣板中的多個頁面組合起來。

將資訊加到可重複使用樣板的一個範例便是導覽區段。典型 Bootstrap 主題中的導覽是在所有頁面中重複出現，不過當您將該導覽移到 Oracle Content Management 主題中時，您必須使用 JavaScript 程式碼來動態瀏覽網站階層及產生導覽結構。接著，您只要在所有頁面中包含該命令檔，頁面即可導覽，當頁面新增至網站或自網站中移除時，導覽也會配合調整。

請參閱[從 Bootstrap 或網站設計樣板建立網站樣板](#)。

## 隱藏主題的元件和區段版面配置

依照預設，開發網站的人員都可以使用所有元件和區段版面配置。您有時候可能會想要隱藏元件或區段版面配置，以便使用特定主題時，無法使用這些元件或區段版面配置。例如，如果想要不使用任何按鈕元件或滑桿區段版面配置來升級成一致的外觀與感覺，您可以藉由編輯主題的 `components.json` 檔案，以在主題中將其隱藏。

若要隱藏主題中的元件和區段版面配置：

1. 在 Oracle Content Management 中，按一下左側導覽功能表中的**開發人員**。
2. 在「開發人員」頁面中，按一下**檢視所有主題**。
3. 在「主題」頁面中，選取要修改的主題，然後從滑鼠右鍵功能表中選擇**開啟**，或按一下動作列中的 。
4. 選取 `components.json` 檔案，然後按一下**下載**。
5. 在文字編輯器中開啟 `components.json` 檔案。
6. 新增物件來指定元件類型和 ID，並將特性 `hidden` 設為 `true`。  
例如，下列範例項目會隱藏按鈕元件：

```
"type": "scs-button",  
"id": "scs-button",  
"hidden": true
```

同樣地，下列範例項目會隱藏滑桿區段版面配置：

```
"type": "scs-sectionlayout",  
"id": "scs-sl-slider",  
"hidden": true
```

`components.json` 檔案中隱藏按鈕元件和滑桿區段版面配置的完整項目如下：

```
[  
  {  
    "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. 進行所需的變更之後，請將 `components.json` 檔案上傳至主題以作為新的版本。

## 隱藏主題的元件對齊方式、寬度或間距選項

依照預設，建置網站的人員可在元件的設定值對話方塊中，指定大多數元件的對齊方式、寬度以及間距選項。不過，開發人員可能會將自訂元件的這些選項隱藏起來，讓提供者無法以與網站主題不一致的形式呈現內容。例如，若想要將按鈕置中並將間距設為 30 像素，以提昇成一致的外觀與感覺，您可以編輯主題的 `components.json` 檔案，隱藏設定值中變更這兩個樣式的選項。

若您想要覆寫自訂元件的最初預設值，也可以編輯 `components.json` 檔案。對最初值的所有變更只會套用至編輯過的 `components.json` 檔案上傳為新版本之後新增至頁面的元件。

隱藏主題中元件的對齊方式、寬度或間距設定值選項：

1. 在 Oracle Content Management 中，按一下左側導覽功能表中的**開發人員**。
2. 在「開發人員」頁面中，按一下**檢視所有主題**。
3. 在「主題」頁面中，選取要修改的主題，然後從滑鼠右鍵功能表中選擇**開啟**，或按一下動作列中的 。
4. 選取 `components.json` 檔案，然後按一下**下載**。
5. 在文字編輯器中開啟 `components.json` 檔案。
6. 新增物件以指定元件類型和 ID，並將想要隱藏的特性設為 `true`。  
例如，下列為一個在設定值對話方塊中隱藏按鈕元件之對齊方式、寬度和間距特性的項目：

```

"type": "scs-button",
"id": "scs-button",
"hideAlignmentAndWidth": true,
"hideSpacing": true,

```

若想要覆寫該按鈕元件的預設對齊方式和間距特性，以下為如何指定最初值的範例：

```
"initialData": {
  "alignment": "center",
  "marginTop": 30,
  "marginBottom": 30
```

`components.json` 檔案中指定主題內按鈕元件之設定值對話方塊中預設值及隱藏對齊方式、寬度與間距選項的完整項目如下：

```
[
  {
    "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. 進行所需的變更之後，請將 `components.json` 檔案上傳至主題以作為新的版本。

## 建立元件與主題的關聯

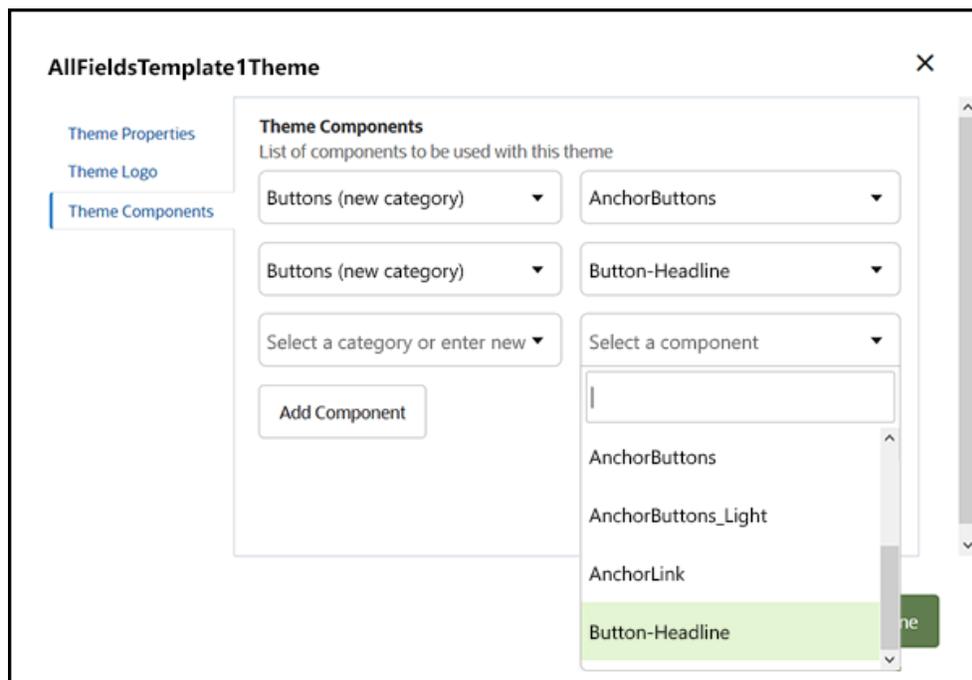
您也可以建立元件、區段版面配置和元件群組與主題的關聯，以便將特定的元件子集搭配該主題使用。

開發人員可以為樣板建立主題和元件，讓行銷人員用以建立促銷公司產品的網站。建立元件與主題的關聯後，使用者在「網站產生器」中選取**主題元件**時即可使用該元件。在**主題元件**頁籤

中，使用者只會見到與該主題關聯的元件。建立元件與主題的關聯可確保即使並未在網站中使用此元件，此元件仍會隨著網站樣板匯出。

建立元件與主題的關聯：

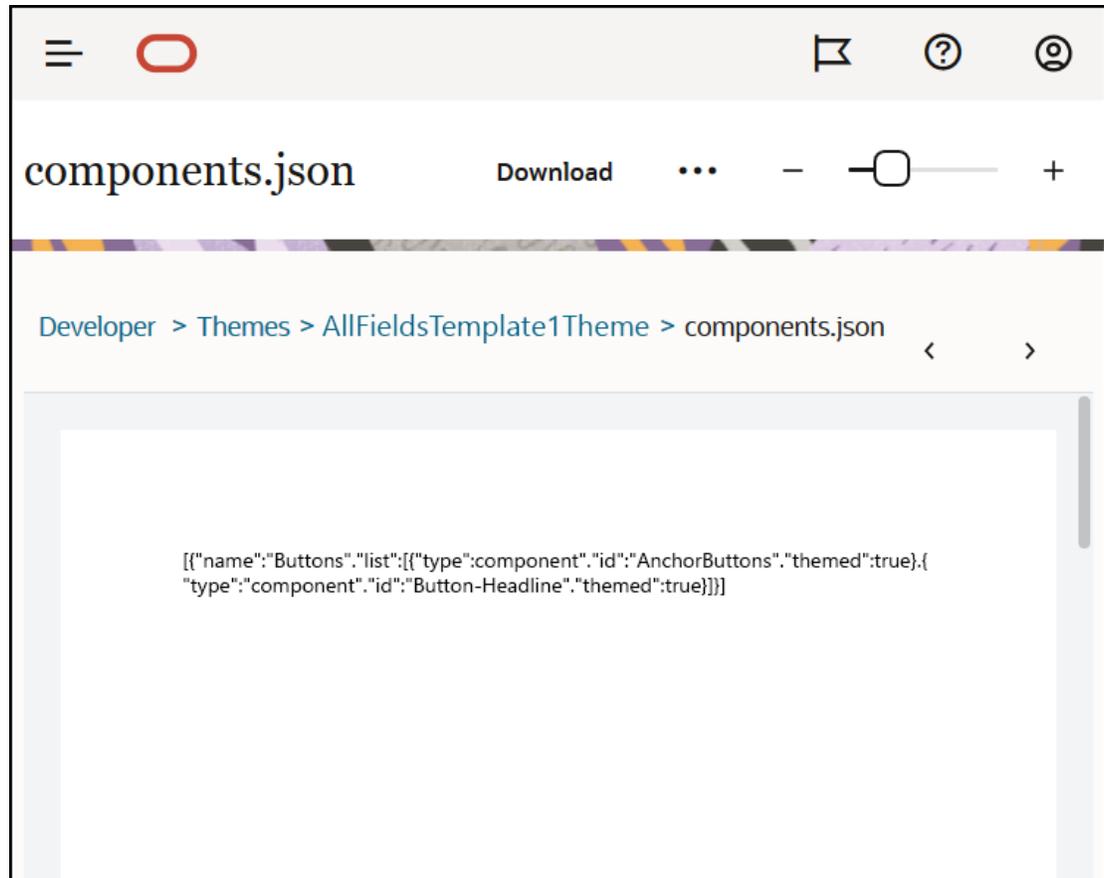
1. 在 Oracle Content Management 中，按一下左側導覽功能表中的**開發人員**。
2. 在**開發人員**頁面中，按一下**檢視所有主題**。
3. 從「主題」頁面中選取一個主題，然後按一下滑鼠右鍵功能表或動作列中的**特性**。
4. 按一下**主題元件**以開啟**主題元件**頁籤。
5. 選取一或多個要與主題關聯的元件，並將元件分類：
  - a. 從**選取元件**下拉式清單中選擇一個自訂元件。
  - b. 針對第一個選擇的元件輸入類別名稱，以建立元件類別。
  - c. 針對選擇的每個其他元件，選取現有的類別或建立新類別。
  - d. 按一下**新增元件**。



6. 按一下**儲存**以建立所選元件與主題的關聯。
7. 在**主題**頁面中，選取一個主題並按一下「特性」，重新開啟**主題元件**頁籤。
8. 確認所選的元件已經儲存。

若要移除與主題關聯的元件，請按一下**主題元件**頁籤中元件名稱旁的 **X** 符號，然後按一下**移除元件**對話方塊中的**是**。

您可以從樣板的**詳細資訊**頁面中，查看與樣板目前所使用主題關聯的元件清單。與主題關聯之元件的參照儲存在主題的 `components.json` 檔案中。



設定主題關聯之後，您即可將樣板部署到 Oracle Content Management，並將其分享給行銷團隊使用。行銷人員針對其網站選擇使用該主題時，只會見到您在「網站產生器」的**主題元件**頁籤中為該主題所指定的元件。**自訂**和**全部**元件清單中也會包含關聯的主題元件。

#### 在 OCE 工具程式中，建立元件與主題的關聯。

若要在 OCE 工具程式中建立元件與主題的關聯，可以使用下列 `cec` 命令。元件將會顯示在「網站產生器」的**主題元件**頁籤中，以及**自訂**和**全部**元件清單中。

```
cec add-component-to-theme <component>
```

下列 `cec` 命令可移除元件與主題的關聯。

```
cec remove-component-from-theme <component>
```

如需 OCE 工具程式的相關資訊，請參閱使用 OCE 工具程式進行開發。

## Sites Rendering API

Oracle Content Management **Sites Rendering API** (SCSRenderAPI) 是存在於所有 Oracle Content Management 網頁上的全域視窗物件。它主要負責呈現頁面的槽位和元件，並提供存在於主題版面配置上 JavaScript 程式碼的介面。開發人員在使用主題或元件，可以利用這些事

件和函數。**程式實際執行** SCSRenderAPI 可呈現檢視和預覽顯示模式，而**設計階段** SCSRenderAPI 可呈現預覽、編輯和附註顯示模式。

在程式實際執行，[Sites Rendering API](#) 具有三種一般用途：

- 使用元件和內容，填入頁面上的槽位。
- 符合頁面版面配置上其 **JavaScript** 程式碼所進行的資訊要求。
- 在頁面呈現處理作業的生命週期期間引發事件。

在設計階段，[Sites Rendering API](#) 具有四種一般用途：

- 使用元件和內容，填入頁面上的槽位。
- 符合頁面版面配置上其 **JavaScript** 程式碼所進行的資訊要求。
- 在頁面呈現和編輯處理作業的生命週期期間引發事件。
- 與「網站產生器」互動，以允許頁面編輯和附註。

將下列命令檔標記放在主題版面配置上，即可將 [Sites Rendering API](#) 載入 Oracle Content Management 網頁中：

```
<script data-main="/_sitescloud/renderrer/renderrer.js" src="/_sitescloud/renderrer/require.js"></script>
```

這通常放在版面配置的底端。(請注意，此標記中的 URL 將自動調整為適當的環境：設計階段或程式實際執行)。SCSRenderAPI 物件會以非同步方式載入；自訂 **JavaScript** 程式碼可以透過處理 `scsrenderstart` 事件，來監聽 [Sites Rendering API](#) 的可用性。

## 開發版面配置

版面配置定義內容在頁面上的排列方式，並用來產生 Oracle Content Management 網站中所使用頁面的 HTML。

- [關於版面配置](#)
- [搜尋引擎最佳化 \(SEO\)](#)
- [瞭解 components.json 檔案與格式](#)
- [自訂網站產生器中的工具列群組](#)
- [限制槽位中的元件](#)
- [將版面配置內容設為可編輯](#)
- [建立區段版面配置](#)
- [建立支援遲緩載入的區段版面配置](#)
- [使用 API 開發自訂區段版面配置](#)
- [開發內容版面配置](#)

## 關於版面配置

版面配置定義網站頁面中內容的排列方式。不同的版面配置可以包含不同數目的指定槽位。槽位是橫跨頁面寬度的區域，可包含一或多個類型的內容。

每個主題都有數個頁面版面配置。請參閱[關於主題](#)。

當您將頁面新增至網站時，會選取該頁面要使用的版面配置。每個版面配置在頁面上都有可在其中拖放內容的區域 (稱為槽位)。這些槽位的內容完全由您所決定。它可以是任何項目，從標題、文字和分隔線到多媒體、圖庫和社群媒體等。

版面配置包含有效的 HTML 建構以及 Oracle Content Management 呈現器可辨識的特殊標記。版面配置必須以 DOCTYPE 敘述句為開頭，才能將瀏覽器設定為以符合標準規範的模式呈現頁面；例如：`<!DOCTYPE html>`。某些元件必須要有此敘述句才能達到最佳結果。

此範例程式碼顯示最精簡的版面配置：

```
1 <!DOCTYPE html>
3 <head>
4     <meta http-equiv="X-UA-Compatible" content="IE=edge">
5
6     <script src="/_themes/ [--$SCS_THEME_NAME--]/assets/js/
topnav.js"></script>
7     <link rel="stylesheet" type="text/css"
8         href="/_themes/ [--$SCS_THEME_NAME--]/assets/css/main.css">
9     <link rel="stylesheet" type="text/css"
10        href="/_themes/ [--$SCS_THEME_NAME--]/designs/
[!--$SCS_DESIGN_NAME--]/design.css">
```

```
11
12     <!--$SCS_RENDER_INFO-->
13     <!--$SCS_SITE_HEADER-->
14     <!--$SCS_PAGE_HEADER-->
15 </head>
16 <body>
17     <div id="topNavigation"></div>
18     <div id="mainContentSlot" class="scs-slot scs-responsive"></div>
19     <script data-main="/_sitescloud/renderer/renderer.js"
20           src="/_sitescloud/renderer/require.js"></script>
21     <!--$SCS_SITE_FOOTER-->
22 </body>
23 </html>
```

頁面在瀏覽器中呈現時，各種不同的記號都會展開。

- `[!--$SCS_THEME_NAME--]`  
這會展開至目前選擇的網站主題名稱。使用此記號即可複製主題，因為使用此記號的 URL 將會參照目前的主題。
- `[!--$SCS_DESIGN_NAME--]`  
這會展開至目前選擇的網站設計名稱。這讓版面配置可供主題內的多個設計使用。
- `[!--$SCS_RENDER_INFO--]`  
這會展開至命令檔標記，其中保留頁面的頁面階層和元件呈現資訊。此記號應放在版面配置的 `<head>` 區段中。
- `[!--$SCS_SITE_HEADER--]`  
這會展開至「搜尋引擎最佳化 (SEO)」特性的「標頭」欄位中所指定的網站標頭值。您可以在此處輸入要放置於所有頁面上的全網站標記。請參閱[設定搜尋引擎特性](#)。
- `[!--$SCS_PAGE_HEADER--]`  
這會展開至在「網站產生器」之「頁面設定值」特性的「頁面標頭」欄位中找到的頁面標頭值。您可以在此處輸入要針對此頁面設定的頁面特定標記。
- `[!--$SCS_SITE_FOOTER--]`  
這會展開至「搜尋引擎最佳化 (SEO)」特性之「頁尾」欄位中的網站頁尾值。您可以在此處輸入要放置於所有頁面上的全網站標記。請參閱[設定搜尋引擎特性](#)。

#### 備註：

這些記號也可以使用 `<!--$` 前置碼和 `-->` 尾碼作為分界字元，以取代 `[!--$` 和 `--]`。

網站若為線上狀態，連結中的下列記號會取代為目前使用這些記號之相關資訊環境的真實值。如此，連結在網站為編輯狀態或在上線的已發布網站中時才能夠運作。

- `/_sitescloud/` 會取代為 `/_sitesclouddelivery/`

· `/_themes/` 會取代為 `/_themesdelivery/`

**槽位**是版面配置中的 **DIV** 元素，在類別屬性中的值為 "scs-slot"。槽位是使用者可新增元件以填入網站內容的區域。您可以指定 "scs-slot" 類別屬性，將多個 **DIV** 元素指定為槽位。每個槽位的 `id` 屬性都必須是唯一的。

#### 備註：

槽位不能採用巢狀結構，但您可以使用 **CSS** 讓它們成為浮動層以提供建議。若要這麼做，請使用元件群組或區段版面配置。請參閱[建立區段版面配置](#)。

瀏覽器檢視區的解析度變更時，額外包含 "scs-responsive" 類別屬性值的槽位將會重新整理其內容。對於大型的桌面顯示或小型的行動裝置，這讓槽位可以響應式的方式呈現。

最後一個 `<script>` 標記 (範例中的第 19 行) 會載入 **Oracle Content Management** 呈現程式碼。此程式碼的作用為在頁面上繪製元件，也可讓自訂程式碼存取呈現 **API**。如果沒有這個 `<script>` 標記，「網站產生器」就無法變更以版面配置為基礎的頁面。

## 搜尋引擎最佳化 (SEO)

您可以提供關鍵字，協助搜尋引擎識別您的網站內容。

### SEO 設定值

網站層級和頁面層級都可以定義搜尋引擎最佳化 (SEO) 設定值。SEO 文字將會併入到所有內建樣板的頁尾。

請參閱[設定搜尋引擎特性](#)。

### 網站訪客的 Cookie

為達到 SEO 目的，網站訪客會使用 **Cookie**，基於計費考量，每個瀏覽器及每個網站使用一個 **Cookie**。每個 **Cookie** 需要每小時或 24 小時更新一次。

每個網站都必須有一個通知訪客有關使用 **Cookie** 的即現式視窗。



Footer

Text entered above will be displayed in the Cookie consent message.

Oracle Content and Experience uses cookies for the purpose of identifying unique user visits. To comply with EU Cookie Regulations your site should display the Cookie consent message and include a separate page explaining use of cookies on the site to visitors. For more information about use of cookies in Oracle Content and Experience, see your Service Agreement.

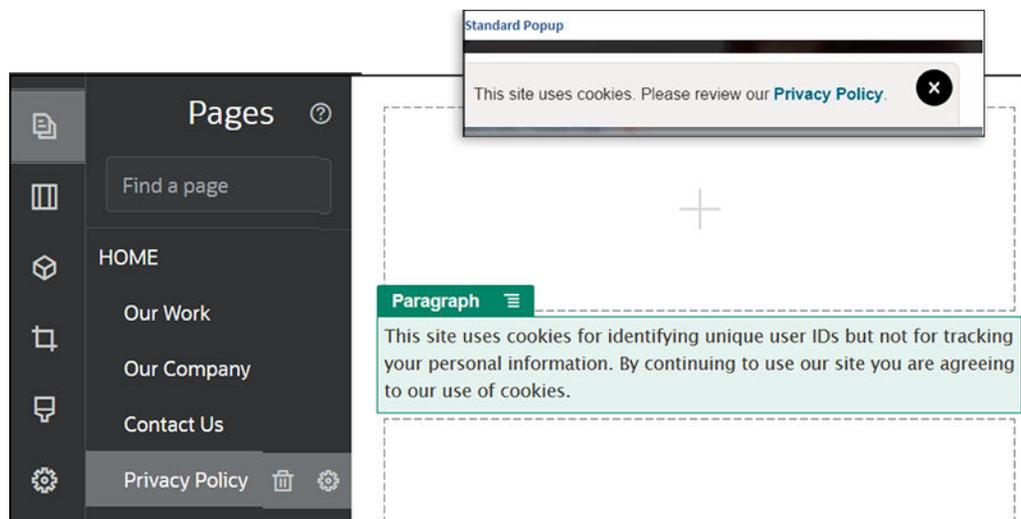
此即現式視窗應包含下列資訊：

- SEO 標頭和頁尾文字
- 具有特定 ID 的頁尾 Div，系統將會提取其中文字並對其設定連結

```
<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>
```

### 隱私權原則頁面

本文字將連結至含有公告文字的隱私權原則頁面。



### 搜尋引擎最佳化的內部 HTML

當您將元件資料儲存在「網站產生器」時，您可以一併儲存內部 HTML (這是元件在程式實際執行時所呈現產生)。此內部 HTML 儲存在頁面資料中，因此當頁面呈現時，可將資料插入頁面中以取代元件。這可以在頁面轉譯的初期進行，讓搜尋引擎有更多機會能順利蒐集到頁面內容。

## 瞭解 components.json 檔案與格式

主題必須要有一個指定主題中所使用元件的 `components.json` 檔案，它的位置在 `/ThemeName/components.json`。

此檔案必須包含有效的 JSON，而且最低極限為必須包含一個空的 JSON 陣列 `[]`。

`components.json` 檔案語法會列出所有本機元件並完全支援元件分類。(遠端元件在「元件目錄」中註冊。)

不論您在主題層級的 `components.json` 檔案中增加了哪些元件 (包括未增加任何元件)，**Oracle Content Management** 都會提供一組預設的元件供使用者使用。此預設集合在原始程式碼中定義。下列清單顯示這些元件和內嵌框架中呈現的 (內建) 元件。此外，在服務層級註冊並提供給執行處理中使用者使用的所有遠端元件，也都在「網站產生器」中使用。

Oracle Content Management 包含下列本機元件。

名稱	類型	ID
標題	scs-title	scs-title
段落	scs-paragraph	scs-paragraph
影像	scs-image	scs-image
圖庫	scs-gallery	scs-gallery
圖庫格線	scs-gallerygrid	scs-gallerygrid
文件	scs-document	scs-document
按鈕	scs-button	scs-button
地圖	scs-map	scs-map
分隔線	scs-divider	scs-divider
空白間隔	scs-spacer	scs-spacer
YouTube	scs-youtube	scs-youtube
社群列	scs-socialbar	scs-socialbar
視訊	scs-video	scs-video
文章 (自訂元件)	scs-component	scs-comp-article
大標題 (自訂元件)	scs-component	scs-comp-headline
影像與文字 (自訂元件)	scs-component	scs-comp-image-text

Oracle Content Management 包含以下這些在內嵌框架呈現的元件。其中並不包括註冊的遠端元件。

名稱	類型	ID
對話	scs-app	對話
文件管理程式	scs-app	文件管理程式
資料夾清單	scs-app	資料夾清單
檔案清單	scs-app	檔案清單
Facebook 按讚	scs-app	Facebook 按讚
Facebook 推薦	scs-app	Facebook 推薦
Twitter 跟隨	scs-app	Twitter 跟隨
Twitter 分享	scs-app	Twitter 分享

### 一般格式

`components.json` 檔案的一般格式如下：

- 元件特性在各元件內指定。最上層 "components" 或 "apps" 特性已不再使用。
- 每個元件都有 "type" 特性。元件只能有某些值 (所有可能的值都列在預設元件表格中)。

- 每個元件都有 "id" 特性且必須是唯一的。此特性用來區分 "type" 相同的元件。應用程式先前有 "appName" 特性。雖然 "appName" 在無 "id" 特性時仍可使用，但 "appName" 特性其實已不再使用。
- 每個元件都有 "name" 特性，此為使用者介面中的顯示名稱。若未指定後備值，對元件而言，此值為相對應之預設元件的名稱；對遠端元件而言，此值為 ID。

以下為 components.json 檔案範例：

```
[
  {
    "name": "COMP_CONFIG_TEXT_CATEGORY_NAME",
    "list": [
      {
        "type": "scs-title",
        "id": "my-headline",
        "name": "My Headline",
        ...
      },
      {
        ...
      },
      ...
    ],
  },
  {
    "name": "My own category name",
    "list": [ ... ]
  }
]
```

一般結構為類別物件的 JSON 陣列。每個類別物件都有一個 "name" 特性和 "list" 特性。"name" 特性可以是對應至某個本地化字串的鍵值。如果這些預設類別不夠用，您可以提供自己的類別名稱，但這些類別不會被本地化。下表列出可用的預設類別和相對應的鍵值。

鍵值	類別名稱 (英文)
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

每個類別物件中的 "list" 特性都包含一個元件物件陣列。每個元件或物件都必須要有 "type" 和 "id" 特性。其他特性則為選擇性。

- "type" 特性必須是預設元件中的其中一種類型。"type" 若不存在，該元件就不會顯示。
- "id" 特性在所有元件中必須是唯一的。"id" 若已經存在，該元件就不會顯示。
- "name" 特性是元件在使用者介面中的顯示名稱。這取代了先前應用程式 (現在為遠端元件) 的 "appName" 特性。

- 所有其他特性則如先前版本等同視之。

### 新增元件至 components.json

您不能修改預設元件。但是，可以根據現有的預設元件建立新元件。例如，可以根據 "scs-title" 元件 (已設定一些預設文字) 建立一個新元件。新增元件的最低要求是需指定 "type" 和 "id" 特性。

- "type" 必須是預設元件中的其中一種類型。"type" 若不存在，該元件就不會顯示。
- "id" 在所有元件中必須是唯一的。"id" 若已經存在，該元件就不會顯示。

以下是新增標題元件的程式碼範例。此元件將會與預設標題元件一起顯示。

```
[
  {
    "name": "COMP_CONFIG_TEXT_CATEGORY_NAME",
    "list": [
      {
        "type": "scs-title",
        "id": "my-headline"
      }
    ]
  }
]
```

以下是新增標題元件 (含顯示名稱和預設文字) 的程式碼範例。

```
[
  {
    "name": "COMP_CONFIG_TEXT_CATEGORY_NAME",
    "list": [
      {
        "type": "scs-title",
        "id": "my-headline",
        "name": "My Headline",
        "initialData": {
          "userText": "This is a second title component"
        }
      }
    ]
  }
]
```

請注意，標題元件會以預設標題元件的所有特性為基礎，然後在預設元件套用主題層級的修改，以建立新元件。

### 回溯相容性

先前格式的 components.json 檔案仍可供讀取。

- 檔案包含最上層 "components" 或 "apps" 特性。
- 檔案若包含 "apps" 特性，系統仍會載入此特性底下的使用者定義遠端元件。

- 檔案若包含最上層 "apps" 特性，系統會假設此特性底下之所有遠端元件的類型都是 "scs-app" 。
- 若有 "appName" 特性，請將 "id" 設為 "appName" 值。如果指定，顯示名稱會與 "name" 相同，否則為 "id" 值。

## 自訂網站產生器中的工具列群組

對於自訂元件，您可以自訂「網站產生器」工具列中顯示的內建工具列群組複本。

您可以對具有限制集且可使用自訂標記的內建元件，建立自己的元件版本。任何自訂內容都不會影響內建元件。

您可以對具有限制集且可使用自訂標記的內建元件，建立自己的元件版本，這樣做並不會影響任何內建元件。

工具列群組定義當您按一下「標題」或「段落」元件來進行編輯時，您在工具列中看到的內容。您可以移除或重新排列「標題」或「段落」元件支援的內容，但不允許您新增 Plug-in 。

若要自訂「網站產生器」中的工具列群組，您必須使用與 CKEditor 用於 toolbarGroups 組態的相同語法。請參閱 <https://ckeditor.com/latest/samples/toolbarconfigurator/index.html#basic> 。

「標題」與「段落」元件支援下列群組：

- "basicstyles" - 限制為粗體/斜體/底線 <https://ckeditor.com/cke4/addon/basicstyles>
- "styles" - 字型樣式 [https://docs.ckeditor.com/ckeditor4/latest/guide/dev\\_styles.html](https://docs.ckeditor.com/ckeditor4/latest/guide/dev_styles.html)
- "colors" - 文字和背景色彩 [https://docs.ckeditor.com/ckeditor4/latest/guide/dev\\_colorbutton.html](https://docs.ckeditor.com/ckeditor4/latest/guide/dev_colorbutton.html)
- "undo" - 還原/重做 CKEditor 中目前的執行處理
- "links" - 連結自訂 Plugin 對話方塊
- "paragraph" - 項目符號/編號清單與縮排支援
  - "list"
  - "indent"
- "align" - 靠左/靠右/置中
- "cleanup" - 移除所選文字的全部樣式

「段落」元件也支援影像和表格插入 Plug-in：

- "insert"
  - "image"
  - "table"

此外，還可以使用資料列區隔符號項目：

- "/"

 **備註：**

若您在工具列群組組態中設定任何其他值，該值會在「網站產生器」工具列建立之前先被移除。您不能提供 "extraPlugins"。僅支援 "name"/"groups" 組態。所有 "items" 項目都會被忽略。

例如，如果您想要防止使用者定義字型、色彩、樣式或大小，可以依下列方式更新工具列組態。對於 "id" 值，您必須指定與內建值不同的自訂值。

```
[{
  "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;"
    }
  }
]}]
```

## 驗證

以上述程式碼取代您的 `components.json` 檔案，然後編輯您的網站 (如果正在編輯，請重新整理瀏覽器)。此時，當您編輯「標題」元件時，它將不再顯示可供選取的字型樣式或色彩。「段落」元件則會繼續顯示這些項目，而可用的字型大小清單將會限制為 16、24 及 48。

## 預設工具列群組

「標題」與「段落」的預設工具列群組如下：

- 標題

```
[{
  "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"]
  }
]

```

## 限制槽位中的元件

您可以為任何版面配置槽位，指定對槽位中所允許元件的特定限制。

如果您限制槽位中的元件，任何使用者只要拖曳不允許的元件，就會看到警告訊息，而無法對該槽位新增或移除元件。

若要設定此限制，您需編輯主題中的版面配置 (例如版面配置檔案 `themes\theme_name\layouts\oneslot.htm`)，然後將自訂資料屬性新增至槽位 `div`。

這是自訂屬性的格式。與舊版本主要的差異在於，使用者只能指定元件的 `id`。若要在舊版中限制元件，使用者必須使用冗長的 "`<type> <id>`" 語法 (目前仍支援)。

```

data-allowed-items='["<id>:", "<type>", "<type>:<id>", ...]'
data-disallowed-items='["<id>:"<type>", "<type>:<id>", ...]'

```

### 備註：

`data-allowed-items` 與 `data-disallowed-items` 的值可以使用雙引號 (") 或單引號 (')。在以下範例中，`data-allowed-items` 使用單引號括住 JSON 陣列，而 `data-disallowed-items` 則使用雙引號括住 JSON 陣列：

```

<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 Oracle Content Management 隨附的元件及其個別 ID。為了防止發生任何命名衝突，請勿在任何自訂的 (本機或遠端) 元件 ID 前面加上 scs-，或使用此表格中所列出的任何類型或 ID。

名稱	類型	ID
文件管理程式	scs-app	文件管理程式
Facebook 按讚	scs-app	Facebook 按讚
Facebook 推薦	scs-app	Facebook 推薦
檔案清單	scs-app	檔案清單
資料夾清單	scs-app	資料夾清單
Twitter 跟隨	scs-app	Twitter 跟隨
Twitter 分享	scs-app	Twitter 分享
按鈕	scs-button	scs-button
文章 (自訂元件)	scs-component	scs-comp-article
大標題 (自訂元件)	scs-component	scs-comp-headline
影像與文字 (自訂元件)	scs-component	scs-comp-image-text
元件群組	scs-componentgroup	scs-componentgroup
內容搜尋	scs-contentsearch	scs-contentsearch
內容清單	scs-contentlist	scs-contentlist
內容預留位置	scs-component	scs-contentplaceholder
內容項目	scs-component	scs-contentitem
分隔線	scs-divider	scs-divider
文件	scs-document	scs-document
圖庫	scs-gallery	scs-gallery
圖庫格線	scs-gallerygrid	scs-gallerygrid
影像	scs-image	scs-image
地圖	scs-map	scs-map
段落	scs-paragraph	scs-paragraph
社群列	scs-socialbar	scs-socialbar
空白間隔	scs-spacer	scs-spacer
標題	scs-title	scs-title
YouTube	scs-youtube	scs-youtube

使用者可以建立本機或遠端元件。此範例中提供的名稱 (My\_Local\_Component) 是可用來指定要在槽位內限制此元件的 ID。

### Create Local Component ✕

Provide a name for your component

Use only letters, numbers, hyphens, and underscores in component names.

Provide a description for your component (optional)

Component Type

## 將版面配置內容設為可編輯

您可以對版面配置中的某些文字或影像內容進行設定，讓使用者在處理以此版面配置為基礎的頁面時能夠編輯這些文字和影像。

您可以在任何主題中使用此功能，包括 **Bootstrap** 主題。修改會自動採用原始頁面的樣式。

您可以在版面配置中的下列 **HTML** 標記加上簡單標記：<p>、<h1> 到 <h6>、<div> 以及 <img>。

文字和影像格式選項包括：

- 文字：使用者可以指定粗體、斜體、底線以及連結。
- 影像：使用者可以指定影像檔的連結。變更標題 (使用者將游標停駐在影像上方時所看到的內容) 的特性及新增替代文字以提供輔助。

修改文字或影像元素以讓使用者能夠進行編輯時，有以下兩個要項：

1. 新增 `scs-editable` 至 `class`。
2. 新增唯一的 `id` 屬性。

版面配置修改過後，此功能便可供以該版面配置為基礎的所有頁面 (甚至是新頁面) 使用。

複製並貼上頁面時，會將修改項目複製到新頁面。

### 修改標題標記

以下是如何修改版面配置中的標題標記，以便讓使用者能夠進行編輯的範例。

1. 將版面配置檔案同步到本機電腦，或是編輯 <h1> 標頭的 HTML 來源檔案。
2. 新增 `scs-editable` 至 `class`，並且在標題的程式碼行中新增 `id="test-heading"` 屬性，讓編輯器能夠與其連附。例如：

```
<h1 class="brand-heading scs-editable" id="test-heading">Sample  
Heading Value</h1>
```

3. 儲存檔案。
4. 與 Oracle Content Management 同步並重新載入瀏覽器。

當使用者將滑鼠游標停駐在標題上方時，其周圍便會出現黑色邊框，代表可以編輯該內容。使用者按一下標題後，邊框便會變成綠色，代表使用者現在可以編輯標題內容。

5. 主題若尚未發布，請加以發布。如果已經發布，會在您重新整理瀏覽器後顯示此變更。

主題發布之後，網站使用者便可以按一下標頭，在「網站產生器」中進行編輯。

系統會為 `scs-editable` 元素建立一個虛擬元件，讓您可以「網站產生器」中進行變更，並且與頁面資料一同儲存。在程式實際執行時，控制器會在頁面呈現之前，將 `scs-editable` 標記取代為您在「網站產生器」中設定的值。

### 修改影像標記

修改版面配置中的影像標記使其可供使用者編輯的程序，與修改文字標記的程序類似。

1. 新增 `scs-editable` 至 `class`。
2. 新增唯一的影像 `id`。

使用者可以按一下影像，然後變更特性以使用其他影像。

以下是可供使用者編輯的影像範例程式碼：

```

```

## 建立區段版面配置

您可以建立區段版面配置來排列網站頁面中槽位內的內容。

企業使用者可以根據開發人員提供的區段版面配置，排列網站上的內容項目。您可以從預設版面配置建立新的區段版面配置。

您可以將區段版面配置匯出進行離線修改，然後再將它匯入成新區段版面配置，或取代現有的區段版面配置。您可以個別匯出區段版面配置，或匯出為樣板套件的一部分 (包括自訂元件和版面配置)。

以下是可用的內建區段版面配置：

- 水平
- 兩欄
- 三欄

- 垂直
- 頁籤式
- 滑桿

您無須建立任何項目，即可立即在「網站產生器」中使用這些版面配置。

這些區段版面配置的檔案都有註解，提供更多關於區段版面配置檔案結構的詳細資訊。若要查看註解，可以根據內建區段版面配置來建立一個新的區段版面配置，然後將新版面配置匯出進行編輯，如以下程序所述。

建立區段版面配置：

1. 在首頁中，按一下**開發人員**。
2. 按一下**檢視所有元件**。
3. 從右邊的**建立**下拉式功能表中選擇**建立區段版面配置**。
4. 在**建立區段版面配置**對話方塊中，提供區段版面配置元件的名稱和描述。
5. 若要匯出區段版面配置進行編輯，請選取該區段版面配置，然後從滑鼠右鍵功能表中選擇**匯出**，或按一下動作列中的 。

- a. 瀏覽至現有資料夾，或按一下**建立**以建立新資料夾並提供名稱，以及視需要提供描述。
- b. 選取資料夾旁邊的核取方塊，然後按一下**確定**。
- c. 按一下資料夾的圖示或名稱，以開啟該資料夾。

系統就會在選取的資料夾中，以區段版面配置名稱加 .zip 副檔名，建立一個版面配置套件檔案。請將此檔案下載至您的開發環境以編輯其中的檔案。

您可以從[使用 API 開發自訂區段版面配置](#)獲得關於「區段版面配置 API」的資訊。

6. 將修改過的檔案匯入成新的區段版面配置，或取代現有的區段版面配置。
  - a. 在首頁中，按一下**內容**頁籤，然後按一下**文件**。
  - b. 將修改過的區段版面配置以 .zip 的檔案格式 (包含與匯出相同的資料夾和檔案名稱) 上傳到資料夾。
  - c. 在**開發人員**頁面中，按一下**檢視所有元件**。
  - d. 從**建立**功能表中選擇**匯入元件**。
  - e. 選取已上傳壓縮檔 (包含修改過的區段版面配置) 旁邊的核取方塊，然後按一下**確定**。

修改過的區段版面配置便會匯入您選取的資料夾中。

您也可以匯出區段版面配置，並將它複製或移到另一個 Oracle Content Management 執行處理，然後在該處匯入。

## 建立支援遲緩載入的區段版面配置

內容清單可以呼叫含有所查詢之其他元件的區段版面配置。

呈現內容清單時，可以選擇選取區段版面配置來呈現所有傳回的內容項目。這可讓您針對內容項目建立各種不同的版面配置，例如表格、滑桿或八欄版面配置。這些自訂區段版面配置也可以加入到更進階的分頁功能中。

內容清單支援下列分頁功能：

- 分頁
- 捲動時載入
- 按一下時載入

對於標準的分頁功能，區段版面配置不需要執行任何作業。當使用者按下一頁時，它將會重新呈現下一組項目。但若是**捲動時載入**和**按一下時載入**，就不是讓區段版面配置重新呈現，而是在區段版面配置中加入其他元件。這大多用於無限捲動模型，也就是先載入前  $n$  個項目，當使用者向下捲動頁面時，再擷取並呈現下一組項目。若要支援**捲動時載入**和**按一下時載入**，自訂區段版面配置需要執行：

1. `render.js`：實行 `addComponent()` API。將對每個要增加至區段版面配置的新元件呼叫此檔案。

```
// 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`：包含下列內容，讓內容清單知道區段版面配置支援 `addComponent()` api。

```
"contentListData": {
  "addComponent": true
},
```

`appInfo.json` 更新之後，當使用者在設定值面板中選取此區段版面配置，他們將會在分頁畫面中見到**按一下時載入**和**捲動時載入**選項。

## 使用 API 開發自訂區段版面配置

您可以在 Oracle Content Management 中使用「區段版面配置 API」開發自訂區段版面配置，此 API 包括呈現 API 和編輯 API。

如需查看入門檔案，請參閱[建立區段版面配置](#)。區段版面配置的入門檔案包含關於區段版面配置檔案結構的詳細註解。

### 呈現 API

呈現 API 是從 `render.js` 模組載入，在「網站產生器」中和程式實際執行時使用。

呈現 API	描述	輸入參數	傳回結果
<b>(建構子)</b>	起始「區段版面配置」呈現模組。	包含下列特性的 JavaScript 物件： <ul style="list-style-type: none"> <li>· <b>sectionlayoutData (物件)</b>：頁面模型中的區段版面配置資料。</li> <li>· <b>componentId (字串)</b>：區段版面配置的 componentId 值，通常是 GUID。</li> <li>· <b>renderMode (字串，選擇性)</b>：呈現作業的呈現模式。</li> <li>· <b>customSettingsData (物件)</b>：sectionLayoutData 中的 customSettingsData 複本。</li> </ul>	起始的區段版面配置呈現 API。
<b>render</b>	將適合區段版面配置的 DOM 元素發送至頁面，包括子項元件的容器 DIV。	<b>container (元素)</b> ：應在其中呈現區段版面配置之標記的 DOM 元素。	此方法傳回之後，便會呈現子項元件。 您可以尋找子項 div[id] 元素來識別子項元件
<b>addComponent</b>	搭配內容清單元件使用，以動態將子項元件新增至區段版面配置。 此為選擇性函數。	<b>container (元素)</b> ：應在其中呈現新元件的 DOM 元素。 <b>componentId (字串)</b> ：要新增至區段版面配置之新元件的 ID。	此方法傳回之後，便會呈現 ID 與 componentId 輸入相同的元素。

## 編輯 API

區段版面配置之 appinfo.json 檔案中的 hasEditHandlers 特性若設為 true，便會載入 edit.js 模組。

編輯 API 主要用於網站產生器。

此模組中除了建構子外，其他的所有函數均為選擇性。

編輯 API	描述	輸入參數	傳回結果
<b>(建構子)</b>	起始區段版面配置編輯模組。	包含下列特性的 JavaScript 物件： <ul style="list-style-type: none"> <li>· <b>componentId (字串)</b>：區段版面配置的 componentId 值，通常是 GUID。</li> </ul>	會起始區段版面配置編輯 API。

編輯 API	描述	輸入參數	傳回結果
<b>getCapabilities</b>	傳回描述區段版面配置之編輯功能的物件。	<p>描述區段版面配置之編輯功能的 JavaScript 物件。</p> <p>如果輸入，會將預設功能提供給函數。此函數可視需要修改「功能」物件。</p> <p>「功能」物件可包含下列功能：</p> <ul style="list-style-type: none"> <li>• <b>title (字串)</b>：要對使用者顯示的區段版面配置標題。</li> <li>• <b>settingsTitle (字串)</b>：「設定值面板」對話方塊中所要顯示的標題。</li> <li>• <b>hasSettings (布林值)</b>：指示區段版面配置是否支援「設定值面板」。</li> <li>• <b>allowMove (布林值)</b>：指示區段版面配置是否允許移動子項項目。</li> <li>• <b>allowDelete (布林值)</b>：指示區段版面配置是否允許刪除子項項目。</li> <li>• <b>isHidden (布林值)</b>：指示區段版面配置目前是否因為使用者選項而隱藏。</li> <li>• <b>dropTarget (布林值)</b>：指示區段版面配置是否為拖放作業的目標。</li> <li>• <b>customMenuOptions (陣列)</b>：要新增至「區段版面配置」相關內容功能表的自訂功能表選項。每個功能選項物件都具有以下特性： <ul style="list-style-type: none"> <li>- <b>label (字串)</b>：功能表項目的顯示文字。</li> <li>- <b>action (函數)</b>：點選功能表項目後所呼叫的函數。</li> <li>- <b>disabled (布林值)</b>：指示功能表項目是否應以停用狀態顯示。</li> <li>- <b>icon (字串)</b>：要顯示在功能表項目標籤旁的 URL。(此特性保留供日後使用。)</li> <li>- <b>checkmark (布林值)</b>：指示是否應在功能表項目標籤旁邊顯示勾選記號。</li> <li>- <b>subMenuItems (陣列)</b>：子功能表中所要顯示的功能表選項。</li> </ul> </li> </ul>	(物件)：區段版面配置的功能。
<b>getCaptionContent</b>	傳回使用者介面元素中將顯示的區段版面配置顯示名稱。	無。	(字串)：區段版面配置的顯示名稱。

編輯 API	描述	輸入參數	傳回結果
<b>filterCapabilities</b>	<p>允許區段版面配置在對使用者顯示功能表之前修改「功能」物件。</p> <p>您可以使用此 API 調整或移除功能表選項。(另請參閱 <b>getCapabilities</b>。)</p>	<p>描述區段版面配置之編輯功能的 JavaScript 物件。如果輸入，會將預設功能提供給函數。</p>	<p>(物件) 區段版面配置的功能。</p>
<b>onDragOver</b>	<p>在進行拖放作業時呼叫，指示是否可將拖曳的項目放置在區段版面配置中。</p>	<p><b>eventObject (事件物件)</b>：保留拖曳事件資訊的事件物件。</p> <p><b>dataTransfer (DataTransfer 物件)</b>：保留拖曳到區段版面配置上項目之相關資訊的 DataTransfer 物件。</p>	<p>(布林值) 此值指示區段版面配置是否接受拖曳的項目。</p> <p>區段版面配置若接受拖曳的項目，則會傳回 <i>true</i>，否則傳回 <i>false</i>。</p>
<b>onDrop</b>	<p>在進行拖放作業時的放置部分呼叫，指示是否應將拖曳的項目放置在區段版面配置內。</p>	<p><b>eventObject (事件物件)</b>：保留放置事件資訊的事件物件。</p> <p><b>dataTransfer (DataTransfer 物件)</b>：保留放置在區段版面配置上項目之相關資訊的 DataTransfer 物件。</p>	<p>(布林值) 此值指示放置作業是否由區段版面配置處理。</p> <p>傳回 <i>true</i> 表示會略過預設邏輯。</p>
<b>onAddComponent</b>	<p>讓區段版面配置知道拖放作業已在區段版面配置中新增項目。</p>	<p><b>eventObject (事件物件)</b>：保留拖曳事件資訊的事件物件。</p> <p><b>dataTransfer (DataTransfer 物件)</b>：保留放置在區段版面配置上項目之相關資訊的 DataTransfer 物件。</p> <p><b>componentId (字串)</b>：新增之項目的 componentId 值。</p>	<p>區段版面配置通知。</p>
<b>onMoveComponent</b>	<p>讓區段版面配置知道拖放作業已將項目移入區段版面配置中。</p>	<p><b>eventObject (事件物件)</b>：保留拖曳事件資訊的事件物件。</p> <p><b>dataTransfer (DataTransfer 物件)</b>：保留放置在區段版面配置上項目之相關資訊的 DataTransfer 物件。</p> <p><b>componentId (字串)</b>：移動之項目的 componentId 值。</p>	<p>區段版面配置通知。</p>
<b>getSettingsData</b>	<p>允許區段版面配置在「設定值面板」顯示之前變更設定值資料。</p>	<p><b>settingsData (物件)</b>：針對區段版面配置運算的預設設定值資料</p>	<p>(物件) 區段版面配置的設定值資料。</p>
<b>updateSettings</b>	<p>允許區段版面配置在「設定值面板」關閉之後變更其設定值。</p> <p>此 API 會在將設定值儲存到頁面模型中之前呼叫。</p>	<p><b>parameters (物件)</b>：從「設定值面板」傳回的原始參數物件。</p> <p><b>sectionLayoutData (物件)</b>：將儲存的區段版面配置資料。將會從 parameters (物件) 產生預設資料，並將其傳送至此參數中的函數。</p>	<p>(物件) 要儲存到頁面模型中的區段版面配置資料。</p>

編輯 API	描述	輸入參數	傳回結果
<b>dispose</b>	<p>允許編輯模組釋出記憶體、取消附加事件，以及取消配置與編輯處理程式關聯的資源。</p> <p>區段版面配置需要完全重新繪製時 (例如還原/重做作業)，便會呼叫此 API。</p>	無。	啟用重新繪製區段版面配置。

## 開發內容版面配置

內容版面配置可協助使用者透過網站頁面中所使用的內容清單或內容預留位置元件，檢視內容項目中的資料。您可以為一種內容類型建立多個內容版面配置，以建立不同檢視或呈現內容項目的不同部分。

例如，「部落格文章」內容類型可能會依部落格文章內容的使用方式和位置的不同，而需要不同的內容版面配置。網站的首頁可能顯示部落格文章項目清單，但按一下首頁中的部落格文章後，會在**詳細資訊**頁面顯示該部落格文章的詳細資訊。

首頁有一個設定成列出「部落格文章」內容類型項目的內容清單，其中使用「部落格文章摘要」內容版面配置作為項目檢視。

「詳細資訊」頁面的內容預留位置中使用「部落格文章標頭」內容版面配置來顯示標頭影像和標題。兩欄區段版面配置包含兩個內容預留位置，分別佔 70% 和 30% 的寬度，使用的是「部落格文章內容」和「部落格文章作者」內容版面配置。如您所見，這裡使用了四種不同的內容版面配置來呈現相同的內容類型。

您可以採用下列兩種方式之一來建立內容版面配置：

- 在 Oracle Content Management 中，選擇**開發人員 > 檢視所有元件 > 建立 > 建立內容版面配置**。
- 在「OCE 工具程式」專案中，使用 `cec create-contentlayout` 命令。

Oracle Content Management 會建立內容類型的預設內容版面配置。若要修改預設內容版面配置，您可以編輯下列檔案：

- `assets/layout.html`  
編輯此檔案以變更 HTML 檢視。
- `assets/design.css`  
編輯此檔案以設定內容版面配置的樣式。
- `assets/render.js`  
編輯此檔案以變更 `layout.html` 中使用的資料，或將動態行為新增至內容版面配置。

內容版面配置元件會呈現來自 Oracle Content Management 伺服器的內容項目。大多數資產都儲存在 Oracle Content Management 伺服器中。有時您可能會想要使用內容版面配置本身在本機提供的靜態資產，例如使用背景影像來設定樣式。例如，在接下來的內

容版面配置中，可在 `render.js` 中產生 `images/background.jpg` 的絕對 URL，並在 `layout.html` 中使用該 URL。

產生絕對 URL 的最簡單方式是使用內建的 `requirejs` 檔案。將相依性定義為 `'require'` 並使用 `require.toURL()` 產生 URL，如下列程式碼所示：

```
define([
  'require',
  'jquery',
  'mustache',
  'text!./layout.html',
  'css!./design.css'
], function (require, $, Mustache, templateHtml, css) {
  'use strict';

  var imageURL = req.toUrl('./images/background.jpg');
```

下列主題說明如何開發內容版面配置：

- [使用 Oracle Content Management 建立內容版面配置](#)
- [將版面配置檢視傳送給內容版面配置](#)
- [使用 API 產生網站詳細資訊頁面 URL](#)
- [運用 Developer Cloud Service 在本機開發內容版面配置](#)
- [擴充內容清單查詢巨集](#)
- [開發完善的內容版面配置](#)
- [建立範例部落格樣板](#)
- [將內容版面配置對應新增至樣板](#)
- [使用本機測試控制程式測試內容版面配置](#)
- [將含有內容版面配置的樣板匯入 Oracle Content Management](#)

### 相關主題

如需有關如何管理內容版面配置的資訊，請參閱[管理自訂元件與版面配置](#)。

如需有關在網站中使用數位資產及其他內容項目的資訊，請參閱[運用 Oracle Content Management 管理資產](#)中的使用資產和管理數位資產。

## 使用 Oracle Content Management 建立內容版面配置

您可以建立內容版面配置，用於編排內容項目中的欄位在網站頁面上顯示時的版面配置。當內容項目新增至頁面時，它將會使用選擇的內容版面配置。

企業使用者可以依據開發人員提供的內容類型和版面配置，建立及使用內容項目。您可以從預設版面配置建立新的內容版面配置。內容類型若有多個關聯的內容版面配置，網站設計者不需要變更內容，就可以顯示不同相關資訊環境中的內容項目。

如果您在某個內容清單元件中使用內容版面配置，此內容版面配置會在每個內容項目重複出現一次。內容版面配置接著便由區段版面配置來編排。

您可以將內容版面配置匯出進行離線修改，然後再將它匯入並取代現有的內容版面配置。

建立內容版面配置：

1. 在 Oracle Oracle Content Management 首頁中，按一下側邊導覽中的**開發人員**。  
**開發人員**頁面便會顯示。
2. 按一下**檢視所有元件**。
3. 從右邊的**建立**下拉式功能表中選擇**建立內容版面配置**。
4. 在**建立內容版面配置**對話方塊中，選取將使用該版面配置的內容類型，選擇要顯示的欄位，以及如果您希望網站建立者在頁面中增加內容項目和內容清單元件時能夠加入自訂設定值與樣式，請啟用**在網站中使用時新增對自訂設定值的支援**。

### 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. 提供內容版面配置元件的名稱和描述，然後按一下**建立**。  
此內容版面配置就會新增至您的元件。

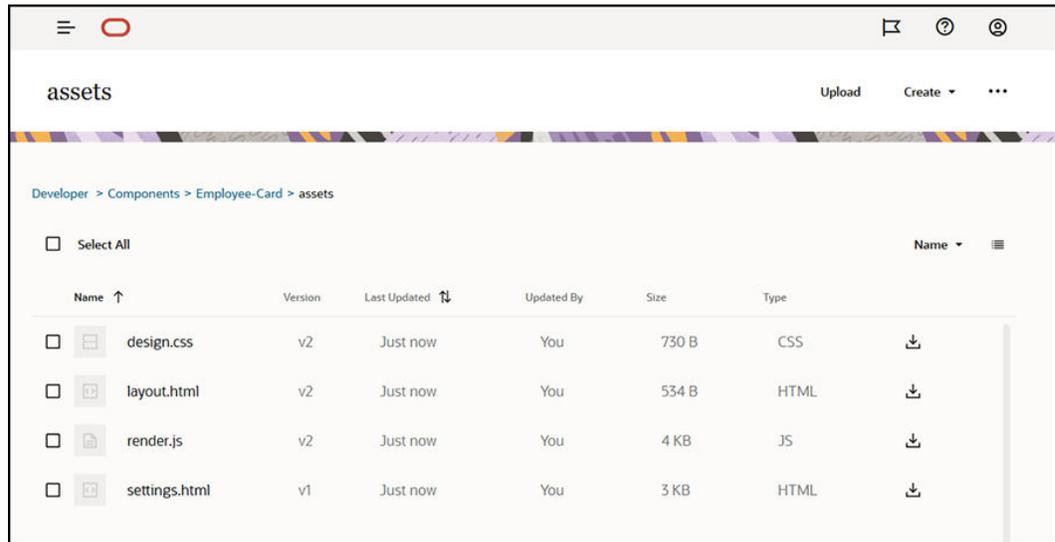
 **備註：**

內容版面配置標題中只能使用文數字字元、連字號以及底線。

6. 若要匯出內容版面配置進行編輯，請選取該內容版面配置，然後從滑鼠右鍵功能表中選擇**匯出**，或按一下動作列中的 。

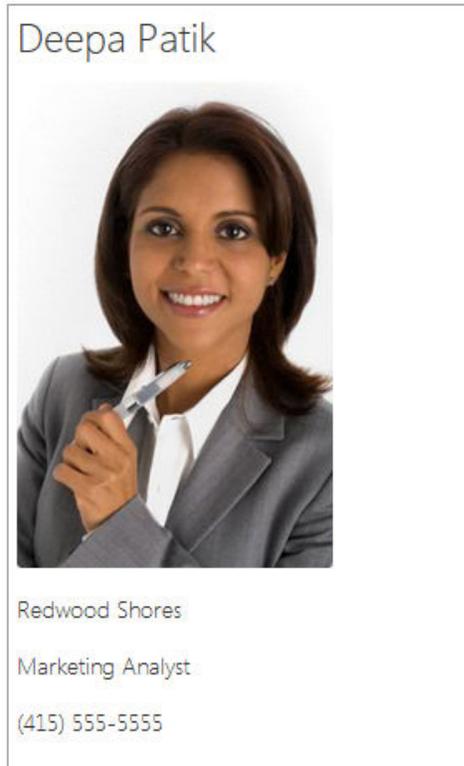
- a. 瀏覽至現有資料夾，或依序選取**建立和資料夾**以建立新資料夾並提供名稱，以及視需要提供描述。
- b. 選取資料夾旁邊的核取方塊，然後按一下**確定**。
- c. 按一下資料夾的圖示或名稱，以開啟該資料夾。

系統就會在選取的資料夾中，以內容版面配置名稱加 .zip 副檔名，建立一個版面配置套件檔案。請將此檔案下載至您的開發環境以編輯其中的檔案。



這些檔案主要控制使用員工證內容版面配置之內容項目中欄位的版面配置。如果已啟用**在網站中使用時新增對自訂設定值的支援**，則系統還會另外建立一個名為 **settings.html** 的檔案，用來提供單一內容項目的預設呈現，以顯示該內容項目。

## Default Layout



## Card Layout



7. 編輯 `design.css`、`layout.html` 以及 `render.js` 檔案，以獲取想要的內容版面配置。

例如，下列檔案指定員工證的內容版面配置：

- a. 編輯 `design.css` 檔案：

```
.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. 編輯 layout.html 檔案：

```
{{#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. 編輯 render.js 檔案：

```
/* 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;
});

```

如需有關編輯 `render.js` 和其他檔案的資訊，請參閱[開發元件](#)。

8. 匯入修改過的檔案以取代現有內容版面配置。

a. 在首頁中，按一下**文件**。

b. 將修改過的內容版面配置以 **.zip** 的檔案格式 (包含與匯出相同的資料夾和檔案名稱) 上傳到資料夾。

若要將它匯入為新的內容版面配置，您需要變更該內容版面配置在 `_folder.json` 中的 GUID。

c. 在首頁中，按一下**開發人員**。

**開發人員**頁面便會顯示。

d. 按一下**檢視所有元件**。

e. 從**建立功能表**中選擇**匯入元件**。

f. 選取已上傳壓縮檔 (包含修改過的元件) 旁邊的核取方塊，然後按一下**確定**。

修改過的內容版面配置便會匯入**元件**中。

您也可以匯出內容版面配置，並將它複製或移到另一個 Oracle Content Management 執行處理，然後在該處匯入。

## 將版面配置檢視傳送給內容版面配置

開發內容版面配置時，若網站中使用內容版面配置，您即可取得相關元件的特性。

就此特定使用案例而言，您可以從內容清單中取得替內容類型選擇的內容版面配置檢視。接著，您就可以根據類別來更改元件的呈現方式。若不存取此特性，您將需要建立兩個實際上是一樣的內容版面配置。

對內容項目元件和內容清單元件呈現的內容版面配置都含有 `contentLayoutCategory` 特性。您可以透過 **Sites SDK** 存取此特性，如下所示。

```

scsData.SitesSDK.getProperty('contentLayoutCategory', function
(layoutCategory)
    { console.log(layoutCategory);});

```

請參閱 Oracle Content Management SDK。

內容版面配置從 Oracle Content Management 網站中的內容項目或內容清單呈現時，才會有此特性。內容版面配置若是從第三方應用程式呈現，便沒有此特性。

## 使用 API 產生網站詳細資訊頁面 URL

在您擁有的查詢中，若是在內容版面配置中呈現內容項目清單，則可以為內容項目建立詳細資訊頁面連結。您可以使用 Sites SDK `SCSRenderAPI.getPageLinkData` API 來產生網站詳細資訊頁面 URL。

`detailPageId` 特性是內容項目或內容清單中所選詳細資訊頁面的 ID。若未設定此值，傳回的值會是 `SiteStructureMap` 中已設定 `isDetailPage` 特性的第一個頁面。

若要存取此特性，您可以使用 Sites SDK。只有 Oracle Content Management 網站使用內容版面配置時，才能這麼做。而且，只能透過 `scsData` 存取，在建立內容版面配置後，它會以其中一個引數的形式傳送。例如：

```
scsData.SitesSDK.getProperty('detailPageId', function (detailPageId)
{ console.log(detailPageId);});
```

取得 `detailPageId` 之後，您就可以用它來建構詳細資訊頁面的連結。

`SCSRenderAPI` 的 `getPageLinkData()` 函數會獲取 `pageId` 和其他選項，然後建構透過選項傳送的頁面 URL。此函數的簽章如下：

```
SCSRenderAPI.getPageLinkData(pageId,
options);
```

它包含下列參數：

- **pageId**：與 Sites SDK `detailPageId` 特性傳回的 `detailPageId` 相同。
- **選項**：
  - `contentType`
  - `contentId`
  - `contentName`

傳回值為包含以下特性的物件：

- `hideInNavigation`
- `href`
- `href`

以下的範例將所有項目集結在一起：

```
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);
});
```

這將列印出："/sites/{site}/{detailPageName}/{contentType}/{contentId}/{contentSlug}"

如果 `pageId` 不是詳細資訊頁面，內容值就不會加到 URL。

請參閱 [Oracle Content Management SDK](#)。

## 運用 Developer Cloud Service 在本機開發內容版面配置

您可以使用「OCE 工具程式」中的 [Developer Cloud Service](#)，在本機建立、編輯、設定及測試 [Oracle Content Management](#) 的內容版面配置。

請依下列步驟準備使用 [Developer Cloud Service](#) 開發內容版面配置：

1. 在本機機器上設定 [OCE 工具程式](#)。
2. 登入 [Oracle Content Management](#) 的 [Developer Cloud Service](#) 主控台
3. 在 [Developer Cloud Service](#) 中建立專案。
4. 將 [OCE 工具程式](#) 新增至新 [Git](#) 儲存區域中的專案程式碼。
5. 建立內容版面配置

下列主題描述如何使用「OCE 工具程式」的 [Developer Cloud Service](#) 在本機開發內容版面配置：

- [建立 Developer Cloud Service 的內容版面配置](#)
- [定義 RequireJS 模組](#)
- [設定建構子函數參數](#)
- [呈現內容版面配置](#)
- [編輯 Mustache 樣板中的內容版面配置](#)
- [新增動態 DOM 操控](#)
- [在 design.css 檔案中定義樣式](#)
- [取得參照項目](#)
- [取得媒體 URL](#)
- [引發觸發程式](#)
- [以搜尋查詢方式瀏覽至搜尋頁面](#)
- [擴充巨集及呈現 RTF 內容](#)
- [連結至詳細資訊頁面](#)

使用 [Developer Cloud Service](#) 樣板開發內容版面配置後，請[合併變更](#)至專案的 [Git](#) 儲存區域。

## 建立 Developer Cloud Service 的內容版面配置

您可以使用 `cec create-contentlayout` 命令，為 Developer Cloud Service 專案中的內容類型建立內容版面配置。

若要在 Developer Cloud Service 專案中建立內容版面配置：

1. 在終端機視窗中，前往 `cec-components` 目錄。
2. 輸入 `cec create-contentlayout` 以查看此命令的選項和範例：

```
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. 輸入下列命令以查看伺服器上可用的內容類型：

```
cec list-server-content-types
```

4. 如「用法」中所示，您可以為 **Oracle Content Management** 伺服器中的內容類型或為 `cec-components/src/main/templates` 底下之樣板中的內容類型建立內容版面配置。例如，下列命令會為伺服器中的類型建立內容版面配置：

```
cec create-contentlayout Blog-Post-Overview-Layout -c Blog-Post -r
```

您可以編輯下列檔案來修改內容版面配置：

- `assets/layout.html`

此檔案指定 HTML 檢視。請參閱[編輯 Mustache 樣板中的內容版面配置](#)。

- `assets/design.css`

此檔案指定內容版面配置的樣式。請參閱在 [design.css](#) 檔案中定義樣式。

- `assets/render.js`

此檔案指定 `layout.html` 中使用的資料，並且可讓您新增內容版面配置的動態行為。請參閱[定義 RequireJS 模組](#)。

如果您建立允許使用自訂設定值的版面配置 (在建立版面配置時使用 `-a` 選項)，系統還會另外建立一個 `settings.html` 檔案，用於提供單一內容項目的預設呈現，以顯示該內容項目。

## 定義 RequireJS 模組

您可以在 `render.js` 檔案中定義 **RequireJS** 模組。網站便會載入相依項，例如 **jQuery**、**Mustache**、**RequireJS Text Plugin** 以及 **RequireJS CSS Plugin**。

```
define([
    'jquery',
    'mustache',
    'text!./layout.html',
    'css!./design.css'
], function ($, Mustache, templateHtml, css) {
```

您可以使用 **Mustache** 樣板系統來呈現版面配置。

內容版面配置的 `assets/render.js` 檔案具有下列特性：

- 必須是 **RequireJS** 模組。
- 應傳回 **JavaScript** 建構子函數。網站會傳送參數物件來呼叫建構子函數。此參數物件包含呈現版面配置所需的內容項目資料和 **API**。
- 此建構子函數應含有可處理內容版面配置呈現的 `render(parentObj)` 方法。它應該將內容版面配置 **DOM** 物件附加到傳送給 `render()` 方法的 `parentObj` 物件。
- **RequireJS** 模組可以使用相依項，包括 **jQuery**、**Mustache**、**RequireJS Text Plugin** 以及 **RequireJS CSS Plugin**。網站將會載入這些相依項。您也可以使用其他程式庫。

## 設定建構子函數參數

網站在建立新的建構子函數執行處理時，會傳送包含 `contentItemData`、`scsData` 以及 `contentClient` 的參數，以協助內容版面配置的開發。

以下為建構子函數的程式碼範例：

```
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;
```

建構子函數參數包含下列物件：

- **params.contentItemData**：包含內容項目，包括其名稱、描述、ID 以及資料。例如，可以使用 `params.contentItemData.data['blogpost_title']` 存取內容項目中的 'blogpost\_title' 欄位。
- **params.scsData**：此物件會在從網站內呼叫建構子時傳入資訊。第三方應用程式中呈現的內容版面配置不會有此物件。此物件包含一個 Sites SDK 物件、用於引發觸發程式的 `contentTriggerFunction` 方法以及[詳細資訊](#)頁面連結。
- **params.contentClient**：這是從 Content SDK 建立的 `contentClient` 物件，用於呼叫內容版面配置。因此，已設定了適當的內容伺服器參數。如果需要對內容伺服器進行其他呼叫，可以使用 `contentClient` 物件，無需自行建立。此物件包含內容的從屬端 API。這些 API 可用於查詢、搜尋及取得內容項目與其內容類型。另外也提供其他協助程式 API，例如 `expandMacros()` 可展開 RTF 內容中使用的巨集。

## 呈現內容版面配置

`ContentLayout` 的 `render(params)` 方法可呈現來自樣板的內容版面配置。對於內容版面配置，預設會使用 **Mustache** 樣板，但您可以使用任何想要的樣板技術。

`ContentLayout` 的 `render(params)` 方法可以使用下列程式碼來呈現含有資料的樣板：

```
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);
}
```

您可以將所需資料新增至從 `params.contentItemData` 建立的 `content` 物件。Oracle 建議您將來自 `params.scsData` 的特性合併到此物件中，以便讓樣板也能使用這些特性。所呈現的樣板應該附加至傳送給 `render()` 方法的父項物件。

## 編輯 Mustache 樣板中的內容版面配置

您可以在 `assets/layout.html` 檔案中編輯預設的 Mustache 樣板，此檔案包含預設的內容版面配置。

呈現部落格標題的簡單樣板如下：

```
<h1>
  {{blogTitle}}
</h1>
```

## 新增動態 DOM 操控

呼叫 `Mustache.render()` 並將樣板附加到父項物件後，即可新增動態 DOM 操控至 `render.js`。

例如，您可以附加監聽器動態變更樣式：

```
// Dynamic DOM Manipulation can be done here
$('h1').click(function (event) {
  alert('Title is : ' + $(this).text());
});
```

## 在 design.css 檔案中定義樣式

您可以在 `design.css` 檔案中，定義內容版面配置樣板中使用的所有樣式。

系統會使用 RequireJS CSS Plugin 將 `design.css` 檔案載入模組定義中。

## 取得參照項目

您可以取得其參照資料欄位參考其他內容類型之內容類型的參照項目。

例如，「部落格張貼項目」內容類型的 `Author` 欄位為「作者」內容類型的參照。在「部落格張貼項目」的內容版面配置中，`contentClient.getItems()` 會將「作者」參照項目的詳細資訊與目前的「部落格張貼項目」建立關聯。

```
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  
});
```

## 取得媒體 URL

您可以使用 `contentClient.getRenditionURL()` 取得預設的數位資產轉譯，例如影像。

```
blogHeaderImage = contentClient.getRenditionURL({  
    'id': data['blog-post_header_image'].id  
}),
```

如果需要其他轉譯 (例如 `thumbnail`)，您可以使用 `contentClient.getItems()` 取得數位資產，然後參考 `item.data.renditions.default` 和 `item.data.renditions.thumbnail`。

## 引發觸發程式

您可以使用 `scsData.contentTriggerFunction(payload)` 從內容版面配置引發觸發程式。

以下是作者內容版面配置範例，它會在按一下作者名稱時引發觸發程式：

```
{{#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}}
```

有效負載是目前所選作者的搜尋查詢，可供頁面上的其他內容清單項目監聽。

## 以搜尋查詢方式瀏覽至搜尋頁面

常見的使用案例是，按一下內容版面配置內的連結後，以動態搜尋查詢方式瀏覽至搜尋頁面。

例如，假設想要在按一下內容版面配置中的「此作者的更多文章」連結並傳送搜尋有效負載後，瀏覽至「作者」搜尋頁面。以下程式碼將可達成此目的。請注意，在網站頁面內執行時，可以在內容版面配置中使用 `SCS` 和 `SCSRenderAPI` 全域物件。

```
$('.more-from-author').click($.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));

```

如果希望在相同頁面中多次重複使用同樣的內容版面配置，最好在 **CSS** 選取器中使用唯一 ID，而不要使用類別選取器，像是 `$('.more-from-author').click(...)`。

例如：

```

template.html
  <div id="{{navigateId}}">...</div>

render.js
  content.navigateId = this.scsData.id + 'detailTrigger';
  $('##' + navigateId).click(...)

```

## 擴充巨集及呈現 RTF 內容

內容項目的 RTF 內容可以內嵌數位影像。

為了在內容版面配置中正確呈現此 RTF 內容，RTF 欄位需使用 `contentClient.expandMacros()` API。此 API 可解析 RTF 內容中的所有數位資產參照。

```

data["starter-blog-post_content"] =
  contentClient.expandMacros(data["starter-blog-post_content"]);

```

如果利用 **Mustache** 呈現，應該使用 `{{{ }}}` 呈現 RTF 內容值，因為 RTF 內容包含 HTML。使用 `{{{ }}}` 括住變數時，**Mustache** 不會遁離 HTML。

## 連結至詳細資訊頁面

透過 `scsData.detailPageLink` 即可取得「詳細資訊」頁面連結。

例如，如果要在按一下部落格標題時瀏覽至「詳細資訊」頁面，以顯示部落格詳細資訊，您可以使用「詳細資訊」頁面連結，如下：

```
<a href="{{scsData.detailPageLink}}">
  <h2 class="post-title">
    {{blogTitle}}
  </h2>
</a>
```

## 擴充內容清單查詢巨集

您可以在內容清單查詢中，定義執行頁面時所計算特性的值，以顯示最近更新的內容。

網站內大多數的元件特性都是靜態的。由使用者選取或輸入元件任一特性的固定字串或值，而且無論頁面在何時何地執行都不會改變。但是，您可以定義執行頁面時所計算特性的值。這對於顯示內容查詢中最近更新的內容非常實用。使用者可以輸入日期，像是「過去 3 天內」。

您可以在數個特性中插入 **Mustache JS** 擴充。這些字串中參照的值衍生自頁面執行時所執行的模型。內建的模型可處理適用於 **Content REST API** 呼叫的日期格式。您可以依據使用者需求添加其他值來擴充此模型。

以下是您可以針對特性輸入的字串範例：

```
Content List component:
  Additional Query String property:
    updatedDate gt "{{#content.date}}today - 3 days{{/
content.date}}"
```

此 **Mustache** 日期項目會在程式實際執行時評估，因此傳回值會根據執行的時間改變（亦即擴展為 `updatedDate gt "2220181002060000000"`）。如此，使用者便可建置任何複雜的日期字串，而不只是輸入預先定義的值。

### 支援的元件特性

下列特性支援 **Mustache JS** 樣板語法：

- 內容清單
  - 其他查詢字串
    - 例如：`updatedDate gt "{{#content.date}}today - 3 days{{/content.date}}"`
- 標題/段落/文字
  - 透過 **CKEditor** 輸入的 RTF 內容
    - 例如：`"Content REST API format for date: {{#content.date}}now{{/content.date}}"`

 **備註：**

若無 **Mustache** 樣板的自訂模型，標題/段落/文字的擴充就不是那麼實用。但是，對於驗證在「其他查詢字串」中輸入的內容卻是非常實用，因為當您在編輯與檢視之間切換時會進行評估，而且立即可以看見。

**支援的元件語法**

內建即支援 `content.date` 物件。此物件使用以下兩個主參數：`today` 和 `now`。

`today` 值採用目前的瀏覽器時間，將其轉換為今晚午夜，然後再將該值轉換為 **UTC** 時間。

- `{{#content.date}}today{{/content.date}}` 會擴展為今晚午夜的瀏覽器值，轉換為 **UTC** 值並將格式設為 **Content REST API** 日期格式。例如：

```
2220181008065959999
```

- 可以使用以下方式提供更多選擇：

```
today +/- [day | week | month | year]
```

- 當您增減 `today` 值時，它的行為方式也會有所不同。如果是減，將會使用早上的時間。如果是加，將會使用午夜的時間。例如：

- `{{#content.date}}today - 1 day{{/content.date}}` 會擴展至昨天早上。
- `{{#content.date}}today + 2 days{{/content.date}}` 會擴展至後天的午夜。

`now` 值採用目前的瀏覽器時間，不做任何調整地將其轉換為 **UTC** 時間。

- `{{#content.date}}now{{/content.date}}` 會擴展為轉換成 **UTC** 值並將格式設定為 **Content REST API** 日期格式的目前瀏覽器時間。

- `now` 也可以加入 `hour`。方式如下：

```
now +/- [hour | day | week | month | year]
```

- 例如：

- `{{#content.date}}now + 2 hours{{/content.date}}` - 從現在開始 2 小時，轉換為 **Content REST API** 日期格式的 **UTC** 時間
- `{{#content.date}}now - 1 day{{/content.date}}` - 昨天的這個瀏覽器時間，轉換為 **Content REST API** 日期格式的 **UTC** 時間

**使用支援的元件語法**

若要在其他查詢字串中使用巨集擴充，假設要傳回過去 3 週的所有項目，可以輸入：

```
updatedAt gt "{{#code.date}}today - 3 weeks{{/code.date}}"
```

只會傳回日期，供用於 **Content REST API** 呼叫，會在您建構查詢字串時加上引號，就如同在輸入靜態值時一樣。

## MustacheJS

如需相關語法，請參閱 [Mustache JS 樣板頁面](http://mustache.github.io/mustache.5.html)，網址為 [mustache.github.io/mustache.5.html](http://mustache.github.io/mustache.5.html)。

擴展字串時所執行的 **Mustache** 執行處理有一項變更。**Mustache** 同時提供使用 `{{ }}` 的 `text` 擴充，以及使用 `{{{ }}}}` 的 `html` 擴充。這兩者之間的差異在於文字擴充會對字串進行 **HTML** 編碼；亦即若值擴展為 `a < b`，結果會是 `a &lt; b`。這不是您在建構 **URL** 字串時所要的結果。您可以告訴使用者使用 **HTML** 擴充，但這只是額外工作並且會產生更多問題，例如解釋他們為何需要使用 `{{{ }}}}`。

為了避免此情形，已對 **Mustache** 做了設定，讓其在使用 `{{ }}` 時不會遁離值。這表示 `{{ }}` 和 `{{{ }}}}` 的行為方式是相同的。如有需要，也可以將任何結果的編碼保留給使用者作為練習。

## OOTB Mustache 模型

**Mustache** 需要將模型套用至樣板以進行擴充。在前面的範例中，內建已經定義 `{{#content.date}}`，開發人員可以新增新的特性，例如 `{{person}}`。使用者若在 **Mustache** 樣板中輸入模型中沒有的值，結果將會是空的字串。因此，以 `Hello {{person}}` 為例，除非開發人員在模型中新增 `person`，否則只會擴展為 `Hello`。

使用的模型物件是名為 `SCSMacros` 的全域物件。開發人員可以任意將任何其他項目新增到此物件中。評估樣板時，會將此物件傳送至 **Mustache**。

內建的模型物件目前僅支援 `content.date` 物件：

```
{
  content: {
    date: <lambda implementation>
  }
}
```

## 自訂 Mustache 模型

開發人員可以根據其需求來增強支援的物件。因此，他們可以導入 `lastTwoDays` 物件，並且將擴充簡化為 `{{lastTwoDays}}`。

若要將模型擴充為支援先前範例中 `Hello {{person}}` 之類的内容，您必須將 `person` 物件新增至 `SCSMacros`。必須在頁面執行之前先執行此作業。您可以在頁面版面配置的開頭增加命令檔標記，即可在頁面版面配置內完成此作業。例如：

```
<script type="text/javascript">
window.SCSMacros = window.SCSMacros || {}; // define/get the SCSMacros
object
window.SCSMacros.person = "World";
</script>
```

進行此變更之後，`Hello {{person}}` 樣板會擴充為 `Hello World`。

若要將值傳送至此物件 (例如 `Hello {{#person}}personId{{/person}}`)，您必須實行 `mustache lambda`，並且包裝並擴充實行內的值。

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>
```

進行此變更之後，`Hello {{#person}}111{/person}}` 樣板會擴充至 `Hello Small`，`Hello {{#person}}222{/person}}` 樣板則擴充至 `Hello Big`。

#### 備註：

**Mustache** 擴充會同步執行。若您需要擷取非同步值，在您嘗試執行 **mustache** 擴充之前必須先在模型內解析這些值，雖然目前並不支援這麼做，但可以透過訂製實行來實現。

## 開發完善的內容版面配置

內容版面配置必須是健全的，才能處理 **Content REST** 呼叫取得的三種回應資料類型：

- 內容項目：若是具有 `expand=all` 參數，將會展開參照並有大量文字欄位。
- 內容項目：若沒有 `expand=all`，將不會展開參照但會有大量文字欄位。
- 內容查詢將不會展開參照，也不會有大量文字欄位。

為提升效能，從 **Oracle Content Management 19.2.3** 版開始，擷取資產資料的 **Content REST** 呼叫已不再包含 `expand=all` 參數。`expand` 參數會告訴 **Oracle Content Management** 在回應中不僅擷取目前的項目，也向下展開並擷取所有參照的項目。自訂內容版面配置的資料若是透過 `expand` 參數擷取，必須經過更新，才能處理所擷取資料不包含參照之欄位值的情況。

## 呈現內容項目

使用內容版面配置呈現內容項目。內容版面配置會接收內容項目資料，讓資料以 **HTML** 呈現，然後插入頁面中。

內容版面配置預設雖會利用 **Mustache** 樣板來呈現內容項目，但還是可以使用任何 **JavaScript** 技術導入。為了讓 **Mustache** 樣板呈現，資料必須為特定格式。內容版面配置 `render.js` 檔案必須確保傳送至樣板的模型為該格式。

內容版面配置呈現有數種使用案例：

- 在 **Oracle Content Management** 資產管理使用者介面中使用時，資料可處於「已編輯」狀態，讓使用者在儲存前能夠預覽變更。
- 在 **Oracle Content Management** 網站的內容清單或內容項目中使用時，會在資料中額外加入執行網站的其他資訊。
- 透過 **Content SDK** 的 `contentClient.renderLayout()` 呼叫使用時，**Content SDK** 的使用者可將任何資料直接傳送到內容版面配置。

為了效能起見，通常會在建立可傳回所有必要資料的單一查詢或是建立多個查詢之間做取捨，以便在快速的初次查詢時盡可能地快速呈現輪廓，然後透過後續查詢陸續填入各個區域。選擇的模型依資料和使用案例而定。

此外，依資料擷取方式的不同，傳送至內容版面配置的資料在格式上可能也會有所不同。例如，如果您使用 **Content REST** 搭配 `expand` 參數，則也會傳回其他內容項目的欄位參照 (個別或以群組方式)。如果未包含參照的內容項目，您需要進行額外的 **REST** 呼叫。

為了處理所有案例，內容版面配置開發人員對於接收的資料格式方面必須保有彈性。此外，開發人員可以在必要時擷取其他資料，並強制將資料轉換為呈現樣板所預期的格式。

## 內容版面配置的資料結構標準化

內容版面配置開發人員需要將內容版面配置接收的資料結構標準化。

假如全部的資料都存在，內容版面配置可以只呈現元件。如果資料未全數存在，內容版面配置可能需要進行額外的查詢。在所有情況下，內容版面配置都不應假設使用某一特定資料格式，而是應強制讓資料轉換成可呈現的格式。

您需要確保所有想要的資料都存在。如果資料不存在，您將需要進行額外的查詢。資料中可能會遺漏的欄位如下：

- 參照欄位的 "fields" 項目
- 大量文字欄位

內容版面配置都是針對特定內容類型所設計，因此內容版面配置的開發人員知道所需的欄位清單。系統必須擷取這些欄位的資料，內容版面配置才能夠呈現。您有以下的兩個選項：擷取遺漏的資料，然後呈現完整資料；或者，立即呈現，再擷取遺漏的資料來填入空白區域。

### 選項 1：擷取遺漏的資料，然後呈現完整資料

建立一個擷取所需資料的承諾，然後當所有承諾傳回時繼續呈現。

例如，以下列的內容類型與相對應的欄位為例：

- `starter-blog-author`
  - 欄位
    - \* `starter-blog-author_name` - 文字欄位
    - \* `starter-blog-author_bio` - 文字欄位
- `starter-blog-post`
  - 欄位
    - \* `starter-blog-post_title` - 文字欄位

- \* starter-blog-post\_content - 大量文字欄位
- \* starter-blog-post\_author - starter-blog-author 項目的參照

內容版面配置使用以下樣板呈現這些預期的欄位值：

```

{{#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}}

```

可使用下列查詢中的資料呼叫內容版面配置：

- 含 "expand" 的項目查詢 - 提供所有資料
  - /content/published/api/v1.1/items/{id}?expand=fields.starter-blog-post\_author&channelToken=8dd714be0096ffaf0f7eb08f4ce5630f
  - 若要在樣板中順利填入所有值，這是所需的資料格式。如果使用的是另一種查詢，則需要進行額外作業來擷取資料並將其轉換成此格式。
  - "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": "..."
  }
}
          
```
- 不含 "expand" 的項目查詢 - 沒有參照的項目欄位 **"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 查詢 - 沒有大量文字欄位 **"starter-blog-post\_content"**、沒有參照的項目欄位 **"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"
  }
}

```

為了讓這些查詢的呈現一致，內容版面配置的 `render.js` 必須確定所有的參照欄位都展開且大量文字欄位存在。

若沒有的話，就必須往回查詢這些欄位、修正資料，然後呈現完整的資料。

範例 `render()` 函數：

```

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
      referedFields = ['starter-blog-post_author'], // names of
reference fields
      largeTextFields = ['starter-blog-post_content'], // large text
fields in this asset
      fieldsData = content.fields;
  // See if we need to fetch any referenced fields
  referedFields.forEach(function (fieldName) {
    if(fieldsData[fieldName] && fieldsData[fieldName].fields) {

```

```
        // got data already, nothing else to do
    } else {
        // fetch this item
        fetchIDs.push(fieldsData[fieldName].id);
    }
    });

    // See if we need to fetch any large text fields
    for(var i = 0; i < largeTextFields.length; i++) {
        if(!fieldsData[largeTextFields[i]]) {
            // need to fetch this content item directly to get all the large
text fields
            fetchIDs.push(content.id);
            break;
        }
    }
    // now we have the IDs of all the content items we need to fetch, get
them all before continuing
    getRefItems(contentClient, fetchIDs).then(function (referenceData) {
        var items = referenceData && referenceData.items || [];

        // add the data back in
        items.forEach(function (referencedItem){
            // check if it's the current item
            if(referencedItem.id === content.id) {
                // copy across the large text fields
                largeTextFields.forEach(function (fieldName) {
                    fieldsData[fieldName] = referencedItem.fields[fieldName];
                });
            } else{
                // check for any referenced fields
                for (var i = 0; i < referedFields.length; i++) {
                    if(referencedItem.id === fieldsData[referedFields[i]].id)
                {
                    // copy across the fields values
                    fieldsData[referedFields[i]].fields =
referencedItem.fields;
                    break;
                }
            }
        }
    });

    // now data is fixed up, we can continue as before
    try{
        // Mustache
        template = Mustache.render(templateHtml, content);

        if(template) {
            $(parentObj).append(template);
        }
    } catch (e) {
        console.error(e.stack);
    }
}
```

```
    });  
}
```

### 選項 2：立即呈現，再擷取遺漏的資料來填入空白區域

將可能不存在的項目分離出來，然後在第二次傳送時呈現這些項目，以讓效能獲得提升。這將需要用到兩個 Mustache 樣板，第一個執行最初的呈現並將「空洞」留下，待資料完備後再於第二次呈現時填入。

這需要將 Mustache 樣板設定為支援多次傳送，其方式有兩種：為「空洞」提供個別的樣板，或讓模型傳回樣板巨集，而非實際值。無論是哪一種方式，在擷取到資料並接著將資料填入空洞以透過適當的 UI 動畫顯示之前，您都需要先將這些空洞隱藏起來，以免頁面不停地「跳動」。

## 建立範例部落格樣板

BlogTemplate 範例主要展示內容版面配置功能。

您可以在 Developer Cloud Service 中建立樣板、檢查樣板中的內容版面配置以及測試搜尋功能：

1. 建立 BlogTemplate 類型的樣板：

```
cec create-template MyBlogTemplate -f BlogTemplate
```

系統就會複製 BlogTemplate 並建立一個新的 MyBlogTemplate 樣板。

2. 開啟 `http://localhost:8085/`，然後依序按一下**樣板**和 **MyBlogTemplate**。
3. 您會見到部落格張貼項目清單。請按一下其中一個張貼項目。您就會前往「詳細資訊」頁面，這是利用三種不同內容版面配置呈現的頁面。
4. 按一下**更多此作者的文章**。您就會前往搜尋頁面。
5. 按一下作者姓名。該內容版面配置便會引發觸發程式並在右側顯示文章。
6. 前往首頁重新測試搜尋。

## 將內容版面配置對應新增至樣板

建立內容版面配置之後，您可以藉由新增內容版面配置對應，將此內容版面配置新增至 Developer Cloud Service 專案中的本機樣板。

使用 `cec add-contentlayout-mapping` 命令為樣板新增內容版面配置對應。必須提供內容版面配置所依據的內容類型 (-c)，以及對應所適用的樣板 (-t)。例如：

```
cec add-contentlayout-mapping Blog-Post-Detail-Layout -c Blog-Post -t  
BlogTemplate
```

預設的內容版面配置對應是桌面的預設樣式。您可以指定 `-s <layoutstyle>` 選項來指定不同的版面配置樣式，例如名為 **Overview** 或 **Details** 的版面配置樣式：

```
cec add-contentlayout-mapping Blog-Post-Detail-Layout -c Blog-Post -t  
BlogTemplate -s Details
```

您也可以使用 `-m` 選項為行動裝置設定對應：

```
cec add-contentlayout-mapping Blog-Post-Detail-Layout -c Blog-Post -t  
BlogTemplate -m
```

## 使用本機測試控制程式測試內容版面配置

將內容類型和內容版面配置對應新增至樣板後，您可以在本機測試控制程式中測試內容版面配置。

請參閱[使用本機測試控制程式進行測試](#)。

## 使用本機測試控制程式進行測試

先在本機測試控制程式中執行自訂元件、樣板和內容版面配置，再將它們匯入 Oracle Content Management。

開始本機測試控制程式：

1. 在終端機視窗中輸入 `cd cec`。
2. 輸入 `cec develop &` 或 `cec develop --server <server-name> &`
3. 在瀏覽器中開啟 `http://localhost:8085`，查看在本機測試控制程式中執行的元件、樣板及內容版面配置。
4. 您可以在下列目錄中找到您的元件、樣板、主題等：
  - `cec/src/main/components`
  - `cec/src/main/templates`
  - `cec/src/main/themes`

## 將含有內容版面配置的樣板匯入 Oracle Content Management

開發並測試您的內容版面配置之後，可以從 Developer Cloud Service 專案匯出包含內容版面配置的樣板，然後再將該樣板匯入至 Oracle Content Management。

請參閱[匯出樣板](#)。

# 23

## 開發元件

您可以開發用於 **Oracle Content Management** 網站的元件。

- [關於元件](#)
- [關於開發元件](#)
- [建立元件](#)
- [使用 Developer Cloud Service 開發自訂元件](#)
- [開發多語言網站適用的可翻譯元件](#)
- [建置含設定值面板的 H1 元件](#)
- [使用 HTML 樣板建立簡單元件](#)
- [本機元件與遠端元件的比較](#)
- [呈現元件設定值](#)
- [本機元件實行](#)
- [元件的樣式類別](#)
- [如何設定內建元件的樣式](#)
- [設定元件特性](#)
- [內嵌框架中呈現的元件](#)
- [關於內嵌框架中呈現之元件的執行處理 ID 和結構](#)
- [遠端元件安全性](#)
- [註冊遠端元件](#)
- [刪除元件](#)
- [Sites SDK](#)

## 關於元件

元件是一種可以新增至網站頁面的特定類型內容。**Oracle Content Management** 中的元件包括像段落、標題、影像、分隔線等等的項目。

**Oracle Content Management** 支援下列元件類型：

- 本機元件：檔案儲存在 **Oracle Content Management** 中
- 遠端元件：檔案儲存在遠端伺服器上

本機元件可以設為直接在頁面內呈現，或是在頁面的內嵌框架中呈現。遠端元件則一律是在內嵌框架中呈現。

**Oracle Content Management** 對每個樣板 (同時包含主題和網站) 都提供一組預設元件。您可以建立新元件，也可以在您的元件內使用這些預設元件。新增元件至網站後，即可根據需求指定

頁面內容、字型與字體大小、影像框架與位置及其他樣式，以編輯元件的特性設定值。您可以編輯的設定值取決於元件類型。

Oracle Content Management 包含以下元件。

名稱	類型	ID
標題	scs-title	scs-title
段落	scs-paragraph	scs-paragraph
文字	scs-text	scs-text
影像	scs-image	scs-image
圖庫	scs-gallery	scs-gallery
圖庫格線	scs-gallerygrid	scs-gallerygrid
文件	scs-document	scs-document
按鈕	scs-button	scs-button
地圖	scs-map	scs-map
分隔線	scs-divider	scs-divider
空白間隔	scs-spacer	scs-spacer
YouTube	scs-youtube	scs-youtube
社群列	scs-socialbar	scs-socialbar
文章 (自訂元件)	scs-component	scs-comp-article
大標題 (自訂元件)	scs-component	scs-comp-headline
資料夾清單	scs-app	資料夾清單
檔案清單	scs-app	檔案清單
文件管理程式	scs-app	文件管理程式
Facebook 按讚	scs-app	Facebook 按讚
Twitter 跟隨	scs-app	Twitter 跟隨
Twitter 分享	scs-app	Twitter 分享
Facebook 推薦	scs-app	Facebook 推薦
影像與文字 (自訂元件)	scs-component	scs-comp-image-text

請參閱 *運用 Oracle Content Management 建置網站* 中的排列頁面內容。

## 關於開發元件

開發自己的自訂元件可讓您使用任何選擇的頁面技術，開發可內嵌於網站頁面內的複合元素。這實際上可讓您擴充 Oracle Content Management 隨附的元件清單。

所有已註冊元件的項目都儲存在「元件目錄」中，這個資料夾位於 Oracle Content Management，其中包含所有已註冊的元件項目。

請勿使用下列作為樣板、主題、元件、網站、或網站頁面的名稱：authsite、content、pages、scs-template\_\*、\_comps、\_components、\_compsdelivery、\_idcservice、\_sitescloud、\_sitesclouddelivery、\_themes、\_themesdelivery。您可以使用下列名稱作為網站頁面的名稱，但不可以作為樣板、主題、元件或網站的名稱：文件、網站。

## 元件類型

「元件目錄」支援下列元件類型：

- 本機元件
- 在內嵌框架中呈現的本機元件
- 遠端元件

此元件類型會儲存為元件資料夾的擴充屬性 "xScsAppType"。以下是有效的值。

類型	描述
本機元件	所有相依項都應該在本機。
在內嵌框架中呈現的本機元件	系統會從與 <b>Oracle Content Management</b> 相同的網域提供元件。
遠端元件	系統會從遠端位置提供元件。

## 元件檔案結構

每一種元件類型在「元件目錄」中建立時，依據其實行方式會有一組不同的檔案。

### 本機元件：

```
/Components/component-name
  appinfo.json
  _folder_icon.jpg
  assets
    settings.html
    render.js
```

### 使用內嵌架構的本機元件：

```
/Components/component-name
  appinfo.json
  _folder_icon.jpg
  assets
    settings.html
    render.js
  js
    sites.min.js
    knockout.min.js
    jquery.min.js
```

### 遠端元件：

```
/Components/component-name
  appinfo.json
  _folder_icon.jpg
  keys.json
```

## 資料夾描述資料

元件註冊資料儲存在資料夾描述資料中。下列特性可用來唯一識別元件及其類型。

特性	描述
應用程式名稱	指定元件名稱的資料夾名稱。
應用程式描述	資料夾的描述。
應用程式 GUID	每個元件都與一個 GUID 關聯，並且會儲存為擴充屬性 <code>xScsItemGUID</code> 。元件建立時，將由伺服器產生 GUID。
應用程式類型	此特性會指定元件類型。它會儲存為擴充屬性 <code>xScsApType</code> 。

### 備註：

儲存為擴充屬性 `xScsAppIconUrl` 的 `iconUrl` 特性已不再使用。

## appinfo.json 檔案

每一種元件類型的 `appinfo.json` 註冊檔皆只包含資料夾描述資料中未提供的資料。資料夾描述資料中已定義的元件特性不會複製到 `appinfo.json` 檔案中。

### 本機元件：

```
{
  "settingsData": {
    "settingsHeight": 80,
    "settingsRenderOption": "dialog",
    "settingsWidth": 300,
    "componentLayouts": [],
    "triggers": [],
    "actions": []
  },
  "initialData": {
    "customSettingsData": []
  }
}
```

### 在內嵌框架中呈現的本機元件：

```
{
  "endpoints": {
    "settings": {
      "height": "300",
      "width": "400"
    }
  }
  "initialData": {
    "customSettingsData": {}
  }
}
```

```
    }  
  }  
}
```

#### 遠端元件：

```
{  
  "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": {}  
  }  
}
```

本機元件沒有已註冊的 "url" 值。它們使用特定檔案，您可以編輯其內容，但無法變更位置或名稱。

- 本機元件使用 `assets/render.js` 和 `assets/settings.html` 檔案。
- 在內嵌框架中呈現的本機元件使用 `assets/render.html` 和 `assets/settings.html` 檔案。
- 遠端元件使用所指定的任何 "url" 值。

## 建立元件

您可以建立在 Oracle Content Management 中使用的自訂元件。

雖然 Oracle Content Management 提供許多建置網站的預先定義元件，但開發人員還是可以建立能夠存取相同特色與功能的自訂元件。這些元件類型包括本機元件、在內嵌框架呈現的本機元件，以及在內嵌框架呈現的遠端元件。

### 本機元件

建立本機元件時，系統會提供您一個功能完整的範例元件，作為建立自己專屬元件的基礎。選取**預設**即可建立 Knockout 版本的元件，或者選取**樣板**以建立 Mustache 版本的元件。請參閱 *運用 Oracle Content Management 建置網站* 中的建立本機元件或版面配置。

### 使用內嵌框架的本機元件

建立使用內嵌框架的本機元件時，系統會提供您一個功能完整的範例元件，作為建立自己專屬元件的基礎。建立此類型元件的指示與建立本機元件的相同，但需另外選取**封閉測試環境**選項，以建立儲存在本機的內嵌框架版本元件。請參閱 [內嵌框架中呈現的元件](#)。

### 遠端元件

建立使用內嵌框架的遠端元件時，請選取**封閉測試環境**選項。

將建立的檔案複製到遠端伺服器，然後註冊此遠端元件。接著，測試此元件後，再將它納入發布的網站中。

請參閱[內嵌框架中呈現的元件](#)和[註冊遠端元件](#)。

## 使用 Developer Cloud Service 開發自訂元件

Oracle Developer Cloud Service 可協助您開發適用於 Oracle Content Management 的樣板、主題及自訂元件。

Developer Cloud Service 與 Oracle Content Management 整合後所提供的樣板中，含有可開發樣板與元件的工具。其中也提供可快速上手的單元測試範例。此整合包含 Git 儲存區域和工具，可協助開發樣板和元件，還包含可快速、迭代開發樣板、主題及自訂元件的本機測試控制程式。

Developer Cloud Service 可協助您執行下列工作：

- 設定本機開發環境，以運用 Oracle Content Management 執行處理在本機進行樣板、主題和元件的開發與測試
- 從範例或入門項目建立樣板與元件、在測試控制程式中的這些樣板與元件中執行、探索它們，以及在 Developer Cloud Service 環境中開發樣板、主題及元件
- 將建立好的樣板或元件從 Oracle Content Management 匯入至 Developer Cloud Service 環境，以進行來源管理和進一步開發
- 從 Developer Cloud Service 環境匯出樣板或元件，以匯入至 Oracle Content Management 便於在網站中使用這些樣板或元件
- 複製現有元件
- 撰寫單元測試
- 最佳化元件
- 將元件部署到 Oracle Content Management

下列主題說明如何使用 Developer Cloud Service 的整合來為 Oracle Content Management 進行開發：

- [開發 Oracle Content Management 自訂元件](#)
- [最佳化元件 \(精簡\) 以提升效能](#)
- [執行連續整合工作](#)

## 開發 Oracle Content Management 自訂元件

您可以使用 Developer Cloud Service 和本機機器，開發 Oracle Content Management 自訂元件。

下列主題描述使用 Developer Cloud Service 開發及測試 Oracle Content Management 元件的步驟：

1. [在本機機器上設定 OCE 工具程式](#)。
2. [登入 Oracle Content Management 的 Developer Cloud Service 主控台](#)。
3. [在 Developer Cloud Service 中建立專案](#)。

4. 將 OCE 工具程式新增至新 Git 儲存區域中的專案程式碼。
5. 開發您的自訂元件
6. 使用本機測試控制程式進行測試
7. 撰寫並執行單元測試

## 開發您的自訂元件

您可以使用 `cec` 命令行公用程式來建立新元件、在本機開發元件，然後將元件匯至 Oracle Content Management。

### 建立元件

若要在本機建立及開發自訂元件，請使用下列命令：

```
cec create-component <component-name> -f <source>
```

為 `source` 選擇下列其中一個值：

- 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

範例：

```
cec create-component MyLocalComponent1 -f local
```

系統會在您 Git 儲存區域的 `cec-components/src/main/components` 底下建立元件。

`src/main/components` 目錄是 `Sample-To-Do` 元件的內建目錄。您建立的所有元件都會儲存到此目錄中。

## 複製元件

您可以使用 `cec copy-component` 命令在 Developer Cloud Service 中複製元件：

```
cec copy-component <source> [<destination>]
```

此命令會將名為 `<source>` 的現有元件複製到 `<destination>`。

## 將元件匯出或部署至 Oracle Content Management

在本機伺服器開發及測試過元件後，您可以使用下列命令匯出元件。這會建立元件壓縮檔。您可以手動將此元件壓縮檔匯入 Oracle Content Management。

```
cec export-component <component name>
```

您也可以使用下列命令，直接將元件從 Developer Cloud Service 部署到 Oracle Content Management。

```
cec deploy <component name>
```

`cec deployAll` 命令將會部署 `src/main/components` 中的所有元件。

## 將元件匯入 Developer Cloud Service

如果元件壓縮檔是從 Oracle Content Management 伺服器建立，您可以將它匯入 Developer Cloud Service 並進一步開發。使用以下命令：

```
cec import-component <location of the component zip file>
```

### ❗ 重要事項：

您元件的原始程式碼位於 `src/main/components`。您不應修改 `src/main/components` 之外的任何檔案，因為它們是 Oracle Content Management 本機伺服器運作所需的檔案。

## 撰寫並執行單元測試

從範例單元測試著手，為您的自訂 Oracle Content Management 元件撰寫並執行單元測試。

### 從範例單元測試著手

Oracle Content Management 的 Developer Cloud Service 範例包含下列可協助您撰寫單元測試的檔案：

- **src/test/unit**：包含 `Sample-To-Do` 元件的單元測試。這是撰寫元件 JavaScript 程式碼 (包含 `RequireJS` 模組) 的單元測試範例。其中使用了 JavaScript 單元測試的 `Mocha` 和 `Chai` 架構。

- **index.html**：執行單元測試。它會載入 Mocha、Chai 及主要測試檔案 `test-main.js`。
- **test-main.js**：載入單元測試模組 `Sample-To-Do-Test` 並執行 Mocha 單元測試。
- **Sample-To-Do-Test.js**：
  - 定義實際的測試。它會使用 RequireJS 來載入元件原始程式碼 `components/Sample-To-Do/assets/render.js`。
  - 測試 `add()`、`delete()`、`title()` 及 `placeholder()` 方法。

### 為元件撰寫單元測試

若要為自訂元件撰寫每個單元測試，請依照下列步驟：

1. 撰寫類似於 `Sample-To-Do-Test.js` 的單元測試。
2. 將您撰寫的測試載入至 `test-main.js`。

### 在本機執行單元測試

若要在本機執行單元測試，請依照下列步驟：

1. 將 Git 儲存區域複製到本機。
2. 在本機機器的終端機視窗中，輸入 `npm install`。
3. 輸入 `npm start`。
4. 輸入 `npm test` 或在瀏覽器中開啟 `http://localhost:8085/unit/`。

## 最佳化元件 (精簡) 以提升效能

使用 Developer Cloud Service 來精簡 JavaScript 程式碼，以提升元件效能和縮減下載大小。

### 最佳化元件

針對 `Sample-Text-With-Image` 元件，系統提供一個將元件原始程式碼最佳化的範例 (選擇性)。您可以使用 RequireJS 最佳化處理程式來精簡及合併原始程式碼。精簡元件會移除 JavaScript 程式碼中非必要的空間，使效能提升並縮減下載大小。最佳化涉及下列動作：

- 精簡 JavaScript。
- 壓縮 CSS。
- 將 JavaScript、HTML 及 CSS 合併至單一檔案 `render.js`。

此最佳化作業將縮減下載大小並提升效能。

### 為元件啟用最佳化

在 `Sample-Text-With-Image` 元件中，最佳化會將 `render.js` 中標示為相依項的所有檔案 (`template.html`、`data-defaults.js` 及 `design.css`) 精簡並合併至 `render.js` 而成為單一檔案。其他已經支援最佳化的元件如下：

- `Sample-Folder-List`
- `Sample-File-List`
- `Sample-Documents-Manager`
- `Sample-Process-Start-Form`

- Sample-Process-Task-List
- Sample-Process-Task-Details

若要為本機元件啟用最佳化，請執行下列步驟 (必要)：

1. 將 `gulpfile.js` 從 `Sample-Text-With-Image` 元件複製到您的元件。
2. 針對 `render.js` 相依項，修正 `gulpfile.js` 中的模組名稱。請參考 `gulpfile.js` 中的註解。

#### 備註：

如果您的現有元件位於 `assets` 底下的 `css` 資料夾，請將 `css` 資料夾重新命名為 `styles` 以避免在最佳化時發生錯誤。

## 執行連續整合工作

您可以執行連續整合工作，讓網站保持最新狀態。

請參閱 [Developer Cloud Service](#) 中的 "Getting Started with CEC Custom Components Development" Wiki。您可以設定連續整合的組建工作。

## 開發多語言網站適用的可翻譯元件

自訂元件的開發人員可以指定多語言 (MLS) 網站的頁面翻譯應包含自訂元件內的哪些字串。

若要這麼做，在將資料儲存到 `customSettingsData` 物件時，您需要在最上層導入 `nls` 特性。

例如：

```
SitesSDK.setProperty('customSettingsData', {
  'nls': {
    linkText: 'More...'
  }
});
```

翻譯工作建立後，**Oracle Content Management** 將會檢查 `customSettingsData` 物件的最上層特性，並且匯出頁面上各個自訂元件執行處理的整個 `nls` 物件。翻譯人員將翻譯這些值，之後再將翻譯的值匯回網站。

網站翻譯匯入之後，正確版本的 `nls` 物件將會傳回已翻譯地區設定的 `customSettingsData` 物件。

例如，若將網站翻譯成法文，接著以法文地區設定呈現頁面，傳送給自訂元件之 `customSettingsData` 物件中的 `nls` 物件值將更新成：

```
{
  'nls': {
    'linkText': 'Plus...'
  }
}
```

```
    },  
  },  
}
```

customSettingsData 中 nls 物件的格式應限定為名稱/值組。這有助於翻譯工作並可確保當頁面以翻譯的地區設定呈現時，翻譯的值能夠正確套用到網站中的基本值。

## 建置含設定值面板的 H1 元件

您可以建立一個最小的 Oracle Content Management 元件，其中含有一個簡單的 HTML 樣板和 CSS。此 H1 元件有一個簡單設定值面板以及一個代表 design.json 中主題的項目，讓其他 Oracle Content Management 使用者在編輯器使用此元件時，有三個內建樣式可供選擇。

建立新元件時，您可以使用一組立即可運作的內建檔案。這些內建檔案涵蓋了產品內元件的大部分功能。您可以變更內建程式碼來建立自己的元件，這只需要一小部分的內建程式碼就可達成最終結果。

只要以下五個步驟，您就能夠使用設定值面板建置一個 H1 元件：

1. 建立新本機元件
2. 建置基本 H1 元件
3. 新增元件的 CSS
4. 新增設定值面板以變更標題文字
5. 更新主題以讓其他人挑選 H1 元件樣式

## 建立新本機元件

您可以在 Oracle Content Management 建立可立即放置在頁面上的本機元件。這是建立任何新元件的起點。

建立新本機元件：

1. 依序按一下側邊導覽中的**開發人員**和**檢視所有元件**。
2. 選取**建立 > 建立本機元件**。
3. 輸入名稱，例如 H1\_Component，以及選擇性的描述。
4. 按一下**建立**以建立新元件。

現在，您已順利建立一個元件，而且應該會在元件頁面的元件清單以及您建立之任何網站的**新增 > 自訂**元件選盤中見到此元件。請使用下列步驟驗證您的元件建立 (檢查點 1)：

1. 使用任何內建的樣板建立一個新網站；例如使用 StarterTemplate 樣板建立 ComponentTest 網站。
2. 選取**編輯**選項，然後建立網站更新以在編輯器中開啟。
3. 編輯您所建立之網站內的頁面。
4. 按一下左側工具列上的**新增**按鈕 (+)，然後選取**自訂**以開啟自訂元件清單。
5. 從自訂元件選盤中選取 **H1\_Component**，並將其拖放至頁面上。

現在，您應該會看到建立的本機元件以預設的方式呈現。

6. 選取元件的相關內容功能表。

7. 從下拉式功能表中選擇**設定值**。  
您可以變更設定值，看看內建元件的呈現方式會如何變更。  
您可以修改內建檔案來建立新的自訂元件。

## 建置基本 H1 元件

您可以移除內建檔案中的大部分內容來建立 H1 元件。它會顯示您建立 `viewModel` 時內建的標題文字。之後您便可以為元件提供設定值與樣式。

複查本機元件的結構：

1. 使用 **Oracle Content Management** 桌面同步應用程式找出元件，並將其與檔案系統同步。
  - 在新版的桌面同步應用程式中，選擇**開始同步**或**選取要同步的資料夾**選項。
  - 如果沒有桌面同步應用程式，您可以在 **Oracle Content Management** 的**元件**頁籤上選取該元件，然後向下展開來查看檔案。
2. 如果列出元件底下的檔案，您可以看到這些檔案：
  - `assets` 資料夾中的元件檔案：
    - `render.js`
    - `settings.html`
  - `appinfo.json`：含有元件描述的 **JSON** 檔案。  
請參閱[關於開發元件](#)。
  - `folder_icon.jpg`：在「元件目錄」中顯示的圖示。

建置 H1 元件：

1. 開啟 `appinfo.json` 檔案，然後以下列幾行取代其內容：

```
{
  "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. 在您慣用的文字編輯器中，開啟 `assets` 資料夾中的 `render.js` 檔案。

## 3. 將 render.js 的內容變更成下列幾行：

```
/* 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 資料夾中，建立新檔案 render.html 以作為元件的簡單 HTML 樣板。

5. 在 `render.html` 檔案中使用下列內容：

```
<h1 data-bind="text: headingText()">
</h1>
```

元件 `assets` 資料夾現在包含三個檔案。

- `render.html`
- `render.js`
- `settings.html`

將新的 H1 元件新增至您的頁面 (檢查點 2)。

## 新增元件的 CSS

您可以新增 CSS，以提供預設的元件樣式。

新增 CSS：

1. 新增 `design.css` 檔案至元件的 `assets` 資料夾，其中包含以下內容：

```
.h1-component-default-style .scs-component-content {
font-family: "Helvetica Neue", "Helvetica", "Arial", sans-serif;
font-size: 24px;
color:red;
font-weight: normal; }
```

2. 新增至 `appinfo.json`，以宣告將用來設定元件樣式的樣式類別前置碼。如果新增 `h1-component` 的 `styleClassName`，當元件放到頁面上時，預設樣式會是 `h1-component-default-style`。 `appinfo.json` 的新內容如下：

```
{
  "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": [ ]
  }
}
```

- 將第 2 行 (`define(['knockout', 'jquery', 'text!./render.html'], function(ko, $, template) {})` 取代為以下內容，以更新 `render.js` 來載入您的 CSS 檔案。

```
define(['knockout', 'jquery', 'text!./render.html', 'css!./design.css'], function(ko, $, template, css) {
```

 **備註：**

由於您已變更 `appinfo.json` 檔案，因此必須重新整理瀏覽器並再次將元件新增至頁面以取得變更。

確認您的元件將會從 `appinfo.json` 檔案取得預設資料。

## 新增設定值面板以變更標題文字

更新 `settings.html` 檔案以提供設定值面板，用於設定 H1 元件的文字。

新增用以變更標題文字の設定值面板：

- 更新 `settings.html` 檔案，使其包含下列內容：

```
<!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. 從「網站產生器」中選取您的元件，然後按一下**設定值**。

設定值面板不大，而且設為內嵌於元件設定值面板的最上方。請尋找「標題文字」區段。

確認可以變更標題文字 (檢查點 4)：

在您變更設定值面板中的標題文字後，元件便會更新為顯示新文字。

## 更新主題以讓其他人挑選 H1 元件樣式

您可以在主題登錄元件的樣式，以便讓其他使用者能夠在設定值面板或**樣式**頁籤中切換您提供的元件樣式。

更新主題以讓其他使用者挑選元件樣式：

1. 在元件的 `design.css` 檔案中增加其他樣式。在每個樣式的前面加上 `appinfo.json` 中定義之元件的登錄 `styleClassName`。就此元件而言，前置碼為 `h1-component`。

而且，另外新增了 `h1-component-gothic-style` 和 `h1-component-courier-style` 這兩個樣式。

新的 `design.css` 內容現在會像以下這樣：

```
.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. 在主題的 `design.json` 檔案中登錄您的樣式。此檔案在網站所使用的主題中。依序向下展開至 `designs` 資料夾、`defaults` 資料夾中的主題檔案，然後在 `design.json` 中為您的元件新增一個區段。

以下的**粗體**字即為新增內容的範例：

```
      "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": [  

```

新增之 `design.json` 程式碼片段中的名稱 ("Plain"、"Courier"、"Gothic") 將會顯示在樣式頁籤中的元件設定值面板中，如下所示。選取之後，它們將會分別套用您元件的相對應樣式 ("`h1-component-default-style`"、"`h1-component-courier-style`"、"`h1-component-gothic-style`")。

## 使用 HTML 樣板建立簡單元件

您可以建立一個 Oracle Content Management 元件，此元件使用 HTML 和 CSS 搭配 Mustache JS 樣板以根據您在設定值面板上輸入的資料呈現內容。

您提供的設定值面板可允許編輯將在 HTML 樣板中呈現的資料。

此程序的範例使用 Mustache JS 樣板。

透過使用 HTML 樣板，您只要兩個步驟就可以建立簡單元件：

1. [建立 HTML 元件](#)
2. [將使用者提供的資料新增至 HTML 元件](#)

## 建立 HTML 元件

您可以在「Oracle Content Management 元件目錄」中建立新的本機元件，然後透過在 `assets` 資料夾中新增及變更檔案來建置 HTML 元件。

建立及建置 HTML 元件：

1. 建立一個元件並將它命名為 HTML Component。  
請參閱[建立新本機元件](#)。
2. 新增 `mustache.min.js` 檔案至 HTML 元件的 `assets` 資料夾。
3. 在 `assets` 資料夾中建立一個名為 `render.html` 的新檔案，其中包含元件主體的 HTML。就本範例來說，建立包含以下內容的 `render.html` 檔案：

```
<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` 資料夾中建立一個名為 `design.css` 的新檔案，此為元件的 `CSS`。就本範例來說，新增下列命令行以作為 `design.css` 檔案的內容：

```

.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. 開啟 `assets` 資料夾中的 `render.js` 檔案，將其內容變更如下。不論在前面步驟使用的 `HTML` 和 `CSS` 為何，下列 `render.js` 檔案都會將 `HTML` 和 `CSS` 呈現為頁面：

```

/* 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
```

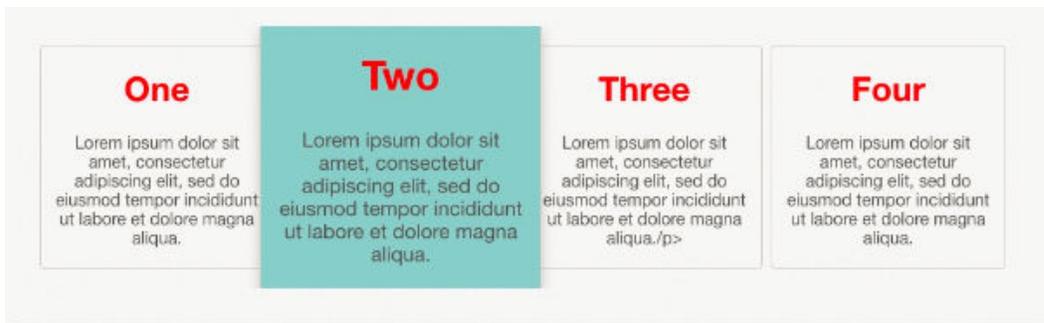
```
//
this.dispose = $.proxy(function() {
    // nothing required
}, this);
};
// -----
// Create the factory object for your component
// -----
var sampleComponentFactory = {
    createComponent: function(args, callback) {
        // return a new instance of the component
        return callback(new SampleComponentImpl(args));
    }
};
return sampleComponentFactory;
});
```

確認已建置 HTML 元件 (檢查點 1)：

1. 確認元件的 `assets` 資料夾中包含以下五個檔案。
  - `design.css`
  - `mustache.min.js`
  - `render.html`
  - `render.js`
  - `settings.html`
2. 將新的 HTML 元件加到測試網站的頁面中。在編輯器中，您應該會在「編輯」和「預覽」模式看到頁面中的元件以下列方式呈現。  
「編輯」模式



「預覽」模式



## 將使用者提供的資料新增至 HTML 元件

您可以新增一個**自訂設定值**面板，其中提供了您在 HTML 元件中定義的所有資料元素欄位。使用者可接著在**自訂設定值**面板中，為頁面上的元件執行處理填入值。

將使用者提供的資料新增至 HTML 元件：

1. 更新 `render.html` 檔案中的 HTML 樣板，以包含您想要的任何資料元素。以下範例使用 Mustache JS 樣板建立語言，因此必須使用 `{{` 與 `}}` 語法來新增資料元素，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>
  <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. 藉由變更 HTML 元件中 `appinfo.json` 檔案的內容，為下列資料欄位提供預設值：

```
{
  "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. 新增**自訂設定值**面板，此面板將尋找您 HTML 樣板 `render.html` 檔案中的資料值，以建立可供使用者在編輯器中變更的設定值。將 `settings.html` 檔案的內容變更成下列幾行：

```

<!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);

            };

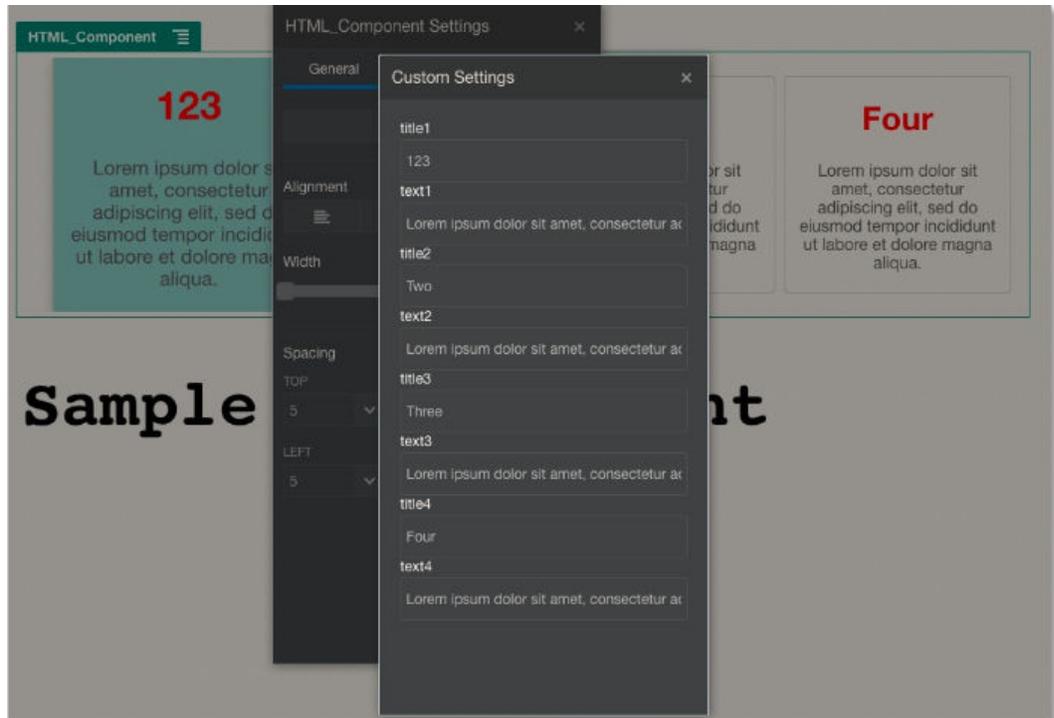
            // apply the bindings
            ko.applyBindings(new SettingsViewModel());
        </script>
    </body>
```

 **備註：**

由於您已變更 appinfo.json 檔案，因此必須重新整理瀏覽器並再次將元件新增至頁面以取得變更。

確認您的元件現在會從 appinfo.json 檔案取得預設資料，且您可以在**自訂設定值**面板中變更值 (檢查點 2)：

1. 在您變更 appinfo.json 檔案之後，請記得重新整理瀏覽器並再次將元件新增至頁面以取得變更。
2. 開啟 HTML 元件中的**自訂設定值**面板並輸入資料。



3. 確認您的 Oracle Content Management 元件使用 HTML 和 CSS 搭配 Mustache JS 樣板，並根據您在自訂設定值面板上輸入的資料呈現內容。

## 本機元件與遠端元件的比較

本機元件和遠端元件的實行方式不同。

下表列出這兩種元件在實行方式上的差異之處。這可以協助您決定要使用在內嵌框架中呈現的本機元件或遠端元件。

本機元件實行	遠端元件實行
以內容為中心的複雜使用者介面整合到 Oracle Content Management	將應用程式邏輯整合到 Oracle Content Management
執行 JavaScript 來呈現頁面中的內容	使用 <iframe> HTML 標記呈現頁面中的內容
使用任何 JavaScript 技術堆疊	可使用任何技術，不限於 JavaScript
運用 Oracle Content Management JavaScript 堆疊，而且可重複使用 Oracle Content Management 元件	未與 Oracle Content Management JavaScript 堆疊整合
從儲存在 Oracle Content Management 伺服器的 /assets 文件中加上資產相依項	所有資產存取都需有 HTTPS URL 和相對應的憑證
與樣板一起由 Oracle Content Management 代管，主題則各自發布	URL 端點需有代管的中間層伺服器
程式碼在頁面內執行，因此若元件程式碼中斷的話，頁面也會隨之中斷	頁面呈現與內嵌框架的內容無關，因此元件如果失敗，頁面也不會中斷
元件的呈現速度如果緩慢，也會讓頁面的呈現速度變慢	頁面載入與內嵌框架內容無關，因此頁面將以最快的速度載入，接著載入任何內嵌框架內容

 **備註：**

雖然本機元件與遠端元件的呈現方式不同，但「網站產生器」中的「設定值」面板實行是一樣的。這兩種類型元件的「設定值」面板都是利用內嵌框架呈現，而「設定值」面板使用相同的 JavaScript SDK，因此可跨界和跨網域通訊。

## 呈現元件設定值

若要呈現元件設定值，您可以使用元件設定值 URL 和元件設定值呈現選項。

### 元件設定值 URL

元件設定值 URL 使用內嵌框架呈現，並且使用參數呼叫，讓特定設定值的實際元件能夠放到頁面上。設定值 URL 的格式如下：

```
{Component Settings URL}?instance=<app-
instance>&width=<width>&currCompId=<id of the app associated with the
settings panel>&locale=<locale>
```

名稱	類型	描述
Component Settings URL	URL	元件的元件設定值 URL
width	數字	「設定值」內嵌框架的寬度 (像素)
currCompId	字串	「設定值」面板中目前編輯元件的元件 ID
locale	字串	主機網站目前的地區設定 (網站產生器)。格式為 <language>_<dialect>。範例：En_us。

### 本機元件設定值呈現選項

根據內嵌框架大小與複雜度，本機元件「設定值」面板中呈現內嵌框架的選項共有三個。每個選項都是在 settingsRenderOption 特性中指定。

 **備註：**

這些選項僅供本機元件使用。遠端元件「設定值」面板一律在對話方塊中呈現。

- inline
  - 只有在使用者需要輸入幾個小特性時才使用此選項。
  - 內嵌框架會插入**一般**頁籤，以取代原本用來瀏覽至「設定值」面板的按鈕。
  - 提供整合度最高的解決方案，使用者需要的點選動作最少，但它的空間有限。
- panel

- 清單中的特性若比較多，但仍可在標準 300 像素的「設定值」面板內適當顯示時，請使用此選項。
  - 內嵌框架將會滑入檢視中並顯示「上一步」按鈕，讓使用者回到「一般」頁籤。
  - 提供整合式解決方案，可在頁面中與內嵌框架互動。
- dialog
    - 需要複雜使用者介面以處理更為普遍設定值版面配置時的預設模式。
    - 在頁面上的強制回應對話方塊中顯示內嵌框架。

## 本機元件實行

元件執行處理由「元件處理站」物件所建立。

元件註冊檔中的「元件處理站」物件項目為 `initData.componentFactory`。這是 **RequireJS** 用來載入元件處理站之 **JavaScript** 檔案的參照。它必須傳回實行 `componentFactory` 介面的 **JavaScript** 物件。

「元件處理站」物件必須實行 `customComponentFactory.createComponent(args)` API，此 API 會建立元件的每個執行處理。

`args` 包含以下設定值：

- `SitesSDK`：Oracle Content Management Sites SDK。
- `id`：新增至頁面之元件的唯一 ID (GUID)。
- `viewMode`：頁面目前的呈現模式。頁面若在編輯中，會為「編輯」模式。頁面若在預覽中，會為「瀏覽」模式。在程式實際執行時 (亦即網站已發布)，值為未定義。您可以根據每一種模式應顯示的功能，提供不同的實行。例如，頁面以「編輯」模式執行時，連結不應為作用中。

### 備註：

您不一定要對您的元件使用 **jQuery** 或 **Knockout**，但若想要運用 **Oracle Content Management** 功能 (如巢狀元件)，就必須使用 **Oracle Content Management** 所提供的 **Knockout** 版本。此 **Knockout** 版本提供其他地方無法取得的已擴充元件註冊和處理程式。

就元件本身而言，**SDK** 會在元件建立後傳入，因此元件能夠與頁面生命週期通訊。頁面生命週期函數必須由元件實行，並且由 **Oracle Content Management** 呼叫以在頁面上呈現元件。

提供必要和選擇性的 API，以用於實行元件。

### 必要的 API

`customComponent.render(container)`：會要求元件將本身插入提供的 **DOM** 容器元素中。

- `container`：自訂元件 **HTML** 的 **DOM** 容器元素。

## 選擇性的 API

`customComponent.dispose()`：會在將元件自頁面移除時呼叫。提供讓元件移除任何已不再需要之資源的機會。

# 元件的樣式類別

您可以建立定義的樣式清單，讓使用者能夠套用至您的元件。

您元件的預先定義樣式清單應遵循與 **Oracle Content Management** 透過主題的設計檔案所提供元件之定義樣式類別相同的模型。請在元件的 `appinfo.json` 檔案中指定您自訂樣式類別的名稱。

如需定義額外樣式，請在 `design.css` 和 `design.json` 檔案中定義。`json` 檔案提供將在使用者介面顯示之名稱與 `css` 類別名稱下之實際名稱的對應，`css` 檔案則提供每個樣式類別的詳細資訊。

`design.json` 檔案的元件結構如下：

```
{
  "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": {
      "styles": []
    },
    "scs-gallery": {
      "styles": []
    },
    "scs-youtube": {
      "styles": []
    },
    "scs-socialbar": {
```

```
        "styles": []
      },
      "scs-document": {
        "styles": []
      }
    }
  }
}
```

每個 "styles":[] 項目都可包含該特定元件的樣式清單。"name" 可以是內建本地化字串的參照，或是要使用的指定值。例如，「標題」元件提供以下預設樣式：

```
{
  "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"
  }
]
}
```

name 值會對應至使用者介面中顯示的實際內容，如下所示：

```
"COMP_STYLE_FLAT": "Flat",
"COMP_STYLE_HIGHLIGHT": "Highlight",
"COMP_STYLE_DIVIDER": "Divider",
```

css 檔案提供類別值的定義：

```
.scs-title-default-style {
  color: #333333;
  display: block;
  font-family: "Helvetica Neue", "Helvetica", "Arial", sans-serif;
  font-size: 24px;
  font-weight: normal;
}
```

例如，您可以在主題的 design.json 檔案中，根據在 components.json 檔案中定義的 initialData.componentId 值，為您的元件新增項目：

```
"componentId": "news-article"
```

design.json 檔案中相對應的項目會是：

```
"componentStyles": {
  "news-article": {
    "styles": [{
      "name": "News Article 1",
      "class": "news-article-default-style"
    },
    {
      "name": "News Article 2",
      "class": "news-article-style-1"
    }
  ]
},
```

design.css 檔案中相對應的項目會是：

```
.news-article-default-style .scs-image {...}
.news-article-style-1 .scs-image {...}
```

## 如何設定內建元件的樣式

您可以覆寫及擴充內建樣式來打造自己的外觀與感覺，為 Oracle Content Management 的內建元件建立不同樣式。

內建元件的視覺樣式取自以下兩個位置：

- comp.css：這是內建的 CSS 檔案，指定每個元件的基本外觀
- design.css：這是屬於網站目前所使用主題之一部分的 CSS 檔案

在 design.css 檔案中，您可以覆寫及擴充內建的 comp.css 樣式，以打造自己的外觀與感覺。在主題中，design.css 檔案位於 designs/default 目錄中。

下列主題描述 comp.css 檔案中所有內建元件通用的類別，並且提供定義主題的概要：

- [元件樣式設定基本資訊](#)
- [元件特定樣式設定](#)
- [設定元件特性](#)

## 元件樣式設定基本資訊

所有內建的網站元件都共用相似的 CSS 類別結構。

每個元件的最外層 <div> 元素會套用下列三個 CSS 類別：

```
scs-component scs-type design-style
```

type 是元件類型 (例如 image、gallery 或 divider)。design-style 是為元件選擇的樣式類別，如主題檔案中所定義。

## 主題樣式基本資訊

主題的 `design.json` 檔案會列出可套用至每一種元件類型 (例如 `button` 或 `image`) 的所有樣式 (`frame`、`shadow`、`highlighted` 等)。每一種樣式都有顯示名稱和類別名稱。顯示名稱會顯示在**設定值**面板的**樣式**頁籤上。類別名稱會參照主題之 `design.css` 檔案中的 **CSS** 選取器。例如，`button` 元件的項目如下：

```
"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"
  }
]
},
```

內建元件的名稱會經過轉譯，因此您會看到一個可從資源組合取得樣式名稱的索引鍵。如果您將 `button` 元件新增至頁面，然後在**設定值 > 樣式**面板中選擇**簡單**樣式，`design.json` 檔案就會將顯示名稱**簡單** (索引鍵為 `COMP_STYLE_SIMPLE`) 與類別名稱 `scs-button-style-3` 建立關聯。`button` 將以下列類別呈現：

```
scs-component scs-button scs-button-style-3
```

如果沒有為指定元件選擇任何樣式，則會使用預設樣式 `scs-type-default-style`。在上述範例中，按鈕將以下列類別呈現：

```
scs-component scs-button scs-button-default-style
```

## scs-component-content 樣式

針對每個內建元件，在先前提到的 `scs-component <div>` 內，都有一個含有 `scs-component-content` **CSS** 類別的 `content <div>`。換句話說，就是：

```
scs-component scs-type design-style
scs-component-content
```

在 `design.css` 檔案中，`scs-component-content` 類別通常用來設定包圍元件的「方塊」樣式 (例如，使用框線或陰影)。

值得注意的是，在內建的 `comp.css` 檔案中，通用 `scs-component-content` 類別除了其他 **CSS** 特性之外，還定義了 `position: relative` 和 `display: inline-block`。

雖然 `scs-component-content` 對於設定包圍每個元件的「方塊」樣式相當實用，但需要特定元件類別才能完整設定元件的樣式。請參閱[元件特定樣式設定](#)。

## 元件特定樣式設定

您可以在影像、按鈕、文件、段落、標題、地圖以及其他元件套用特定樣式。

### 影像元件

「影像」元件具有 `scs-component-content` 類別的下列 CSS 類別結構：

```
scs-image-container  
scs-image-link  
scs-image-image  
scs-image-caption
```

`scs-image-image` 類別會套用至 `<img>` 標記本身。`scs-image-caption` 類別則用於設定說明文字的樣式 (若有說明文字時)。

影像有附加的連結時，才會顯示 `scs-image-link` 類別。此類別和 `scs-image-container` 類別通常都不需要自訂樣式。

影像說明文字預設會以橫跨影像底部的半透明浮動層方式呈現。



```
.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;  
}
```

若要將說明文字放在影像頂端及變更色彩，需要在 `design.json` 檔案中為影像元件新增額外樣式，然後在 `design.css` 檔案中定義它的 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;  
}
```

### 按鈕元件

「按鈕」元件的類別結構如下：

```
scs-button-button  
  scs-button-text
```

`scs-button-button` 類別是可點選的 `<div>`，其樣式與按鈕相似。`scs-button-text` 類別可用於設定按鈕內的文字樣式。

例如，可在 `design.json` 檔案中為按鈕元件新增額外樣式以測試其外觀與感覺的變更，然後在 `design.css` 檔案中定義它的 CSS。



```
.design-style .scs-button-button {
  background-image: linear-gradient(
    to top, #E3E7E9 0%, #E7EBED 50%, #F1F3F3 100%);
  border: 1px solid #c4ced7;
  color: #000000;
}

.design-style .scs-button-button:hover {
  background: #f7f8f9;
  border: 1px solid #c4ced7;
  color: #0572ce;
}

.design-style .scs-button-button:active {
  background: #0572ce;
  border: 1px solid #0572ce;
  color: #ffffff;
}
```



```
.scs-button-style-4 .scs-button-button {
  background-image: radial-gradient(
    red, yellow, green
  );
  border: 1px solid #c4ced7;
  color: #000000;
}

.scs-button-style-4 .scs-button-button:hover {
  background: #f7f8f9;
  border: 1px solid #c4ced7;
  color: #0572ce;
}
```

```
.scs-button-style-4 .scs-button-button:active {  
  background: #0572ce;  
  border: 1px solid #0572ce;  
  color: #ffffff;  
}
```

## 文件

「文件」元件的類別結構如下：

```
scs-document-container  
  scs-document-cap  
    scs-document-title  
    scs-document-desc
```

`scs-document-container` 類別會讓文件檢視器自動換行，通常不需設定樣式。

## 圖庫

「圖庫」元件具有包含基礎 JSSOR 滑桿元件的單一類別：

```
scs-gallery-container
```

JSSOR 滑桿使用數個也可用於設定樣式的類別：

```
jssorb14 (navigator)  
jssora02l (left arrow)  
jssora02r (right arrow)  
jssort07 (thumbnails)
```

## 圖庫格線

「圖庫格線」元件的類別取決於**設定值**面板中所選取的版面配置和剪裁：

```
scs-gallerygrid-container scs-gallerygrid-layout  
  scs-gallerygrid-cell  
    scs-image (multiple)
```

根據為「圖庫格線」所選之剪裁和版面配置的不同，`layout` 的值將會是 `stretch`、`crop`、`fit` 或 `flowing`。

「欄」版面配置才會顯示 `scs-gallerygrid-cell` 類別。

## 社群列

「社群列」元件的類別結構如下：

```
scs-socialbar-container  
  scs-socialbar-icon
```

`scs-socialbar-icon` 類別會套用至社群列的每個 `<img>` 標記。

## 段落

「段落」元件只有包含實際段落文字的單一類別：

```
scs-paragraph-text
```

例如，若要讓在「段落」元件提供的文字具有金屬雕刻文字效果，請在 `design.json` 檔案中新增額外的樣式類別，然後在 `design.css` 檔案中定義它的 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;  
}
```

或者若想要花俏一點，可以像下一個範例這樣。



```
.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;
}
```

## 標題

「標題」元件也是只有包含實際文字的單一類別：

```
scs-title-text
```

## 地圖

「地圖」元件具有包含地圖呈現的單一類別：

```
scs-map-content
```

此類別通常不需設定樣式。

## 自訂本機元件

「自訂本機元件」只有包含實際元件的單一類別：

```
scs-custom-component-wrapper
```

您可以完全控制用於呈現自訂本機元件之自訂檢視所需的 CSS 樣式。本機元件會以內嵌方式呈現，也就是說您可以直接套用主題或 `design.css` 檔案中定義的 CSS 樣式。

## 自訂遠端元件

「自訂遠端元件」只有包含其 `iframe` 的單一類別：

```
scs-app-iframe-wrapper
```

除了套用自訂遠端元件中定義的 CSS 樣式之外，您也可以利用 Sites SDK 從主機網站擷取 `design.css` 檔案。

```
// 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，您可以讓元件沿用主機樣式的樣式設定。

## 分隔線

「分隔線」元件雖然沒有元件特定類別，但是 `<hr>` 標記本身是可以設定樣式的。

例如，可以建立虛線分隔線：

```
.design-style .scs-divider hr {  
border-top: 1px dotted #333333;  
}
```

## 視訊、YouTube、空白間隔

視訊、YouTube 或空白間隔元件沒有元件特定類別。

# 設定元件特性

您可以設定網站中要使用的元件特性。

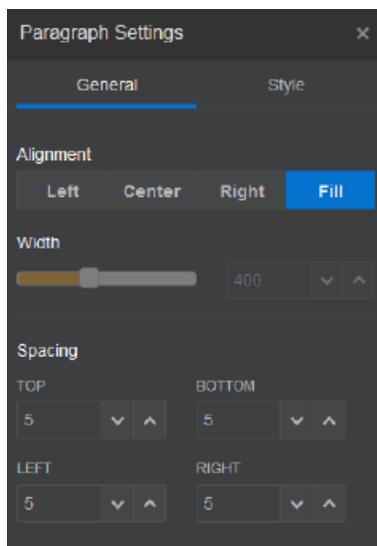
元件是網頁的個別部分，其中包括文字、標題、影像、按鈕、分隔線、地圖、圖庫、視訊等等。當您建立主題的設計時，還必須指定每個類型元件的預設值。每個元件都有定義元件外觀和行為的設定值，例如大小、對齊方式、間距、色彩和框線。設定值會依元件而有所不同。您也可以選擇使用者是否能夠變更網站新主題中的元件特性。

例如，以下步驟說明如何設定「段落」元件的設定值：

1. 以「編輯」模式開啟開發網站，選取其中含有段落元件的頁面，或新增一個段落元件。
2. 按一下段落元件，然後按一下此元件角落的 ，接著選擇**設定值**。  
**設定值**面板便會顯示，供您選擇此元件的值。

### 備註：

每種類型的元件都有特定的設定值選項。如果使用本機或遠端自訂元件，您將會見到**自訂設定值**連結。



- 例如，您可以變更對齊方式、寬度、間距、字型、樣式、色彩、大小等設定值。元件外觀便會依新設定值改變。
- 完成之後，請按一下 。您的設定值就會套用到頁面。

## 內嵌框架中呈現的元件

只要註冊及新增外部伺服器的元件 (稱為**遠端元件**)，或在建立本機元件時選取內嵌框架選項，就可以在 **Oracle Content Management** 頁面中指定在內嵌框架中呈現的元件。這類型的元件可擴充網站的功能，例如新增社群元件或購物車結帳元件。

對於遠端元件，您必須指定端點 **URL**。對於在內嵌框架中呈現的本機元件，**URL** 會衍生自「元件目錄」中元件的名稱。

- 端點 **URL**：會從此 **URL** 擷取小工具的內容，然後內嵌在內嵌框架中。
- 設定值 **URL**：此 **URL** 會在內嵌框架中呈現，用以設定放置於頁面上的元件。

元件可以呈現靜態或動態的資料視覺化項目，以及顯示表單或其他可擴充網站功能的互動式使用者介面。請參閱[呈現元件設定值](#)。

### Oracle Content Management 提供的元件

Oracle Content Management 提供一組在內嵌框架中呈現的元件。

名稱	類型	ID
資料夾清單	scs-app	資料夾清單
檔案清單	scs-app	檔案清單
文件管理程式	scs-app	文件管理程式
Facebook 按讚	scs-app	Facebook 按讚
Twitter 跟隨	scs-app	Twitter 跟隨
Twitter 分享	scs-app	Twitter 分享

名稱	類型	ID
Facebook 推薦	scs-app	Facebook 推薦

### 元件註冊

若要在網站中使用在內嵌框架中呈現的遠端元件，必須使用 **HTTPS** 註冊端點 **URL**。這項資訊儲存在「元件目錄」中。端點必須允許讓 **URL** 在內嵌框架中顯示；請勿在標頭中設定 `X-Frame-Options="sameorigin"`。至於本機元件，由於檔案是儲存在 **Oracle Content Management** 伺服器上，所以此端點條件自動符合。

在內嵌框架中呈現的元件註冊後，系統會產生一個代表此元件的新 **GUID**。如果重複在某個 **Oracle Content Management** 執行處理中註冊此類元件，將會產生多個 **GUID**，因為它們代表該元件的每一次註冊。元件註冊之後，會將其 **GUID** 的描述儲存在 **Oracle Content Management** 中；此元件還是會從其遠端端點執行。請參閱[註冊遠端元件](#)。

在內嵌框架中呈現的元件註冊之後，還會產生一個執行處理 **ID**。此執行處理 **ID** 代表在某個 **Oracle Content Management** 租用戶註冊的元件。在同一個 **Oracle Content Management** 租用戶內或在不同的 **Oracle Content Management** 租用戶重複註冊的相同元件，將會有不同的執行處理 **ID**。

### 遠端元件設定值保存

在「設定值」面板對話方塊中顯示於內嵌框架中呈現之元件的設定值 **URL** 時，會同時提供執行處理 **ID** 和元件 **ID**。這讓該元件可以選擇將其本身的所有設定值保存在自己的伺服器，並且以執行處理 **ID** 和元件 **ID** 編製索引。或者，也可以使用 **Oracle Content Management Sites SDK**，將元件 **ID** 的 **JSON** 資料 (最多可達 1.5 KB) 儲存在網站頁面的頁面模型中。

對於在內嵌框架中呈現的元件而言，使用 **Sites SDK** 保存設定值有兩項優點：

- 元件可輕鬆參與頁面版本、頁面更新以及網站發布模型。
- 元件可包含在瀏覽器中執行的 **HTML** 端點，而非在後端系統執行的端點。

請參閱 **Oracle Content Management SDK**。

## 關於內嵌框架中呈現之元件的執行處理 ID 和結構

元件執行處理 **ID** 是在網站內嵌框架中所呈現元件的唯一 **ID**。

當使用者將內嵌框架中呈現的元件從「元件目錄」拖放到網站頁面上時，系統會對 **Oracle Sites Cloud Service** 進行佈建呼叫以產生新的唯一元件執行處理 **ID**。系統會保證此 **ID** 是唯一的，且在該服務上佈建的所有這類元件執行處理都會取得相同的執行處理 **ID**。

元件執行處理 **ID** 包含額外資訊，可用來保護在內嵌框架中呈現之元件的設定值和使用，確保元件設定值的更新是來自受信任的位置。

執行處理 **ID** 參數可讓開發人員識別網站及認證呼叫方。認證呼叫者時，會透過驗證使用元件秘密金鑰產生的數位簽章來認證。此秘密金鑰是在元件註冊過程中所產生。

元件執行處理會由以 ‘.’ 分界字元區隔的兩個部分組成：資料與結構。

### 元件執行處理：資料

在內嵌框架中呈現的元件，其執行處理的資料部分是一個 Base64 JSON 編碼字串。以下是該 JSON 字串的結構：

```
{
  "instanceid": "BBDC7614F693B75110D811E6C0B77C935FAEC5112E5E",
  "permissions": "",
  "entitlements": "",
  "signdate": "1435426735293",
  "sitedomain": "service1-tenant4.localhost"
}
```

欄位名稱	描述
instanceid	在 Oracle Content Management 租用戶的內嵌框架中呈現之元件的唯一 ID。
signdate	簽章產生日期。
sitedomain	Oracle Content Management 執行處理的網域名稱。
permissions	網站成員的一組權限。在編輯模式下，其值為 "SITE_OWNER"；否則不會有值。
entitlements	網站擁有者所購買的進階功能清單。

### 元件執行處理：簽章

元件執行處理的資料部分序列化之後，才會以 APP\_SECRET\_KEY 簽署。此秘密金鑰必須在註冊元件時產生並對開發人員顯示。計算簽章時，會透過以秘密金鑰產生元件執行處理資料部分 (為序列化的 JSON 結構) 的雜湊來計算，如下所示：

```
$signature = HMAC (serialized JSON structure, APP_SECRET_KEY)
```

系統將使用 SHA256 雜湊演算法產生簽章。接著會由序列化的 JSON 結構和產生的簽章串接成權杖，如下所示：

```
$instance = {base64encoded serialized JSON structure}.
{base64encoded $signature}
```

範例：

```
//base64 encoded serialized object //signature
eyJpbmN0YW5jZWlkIjoiQTRGOTE3REY5OTZEN0Q3ODBCMjUzODZFOTFEMDA3ODJGMjVBRjY2Rjc3O
TIIiLCJzaWduZGF0ZSI6IjE0NDU2MzcwNTk5MTciLCJzaXRlZG9tYWluIjoic2VydmljZTEtdGVuYW
50MS51cy5vcnFjbGUuY29tIiwicGVybWlzc2lvdnMiOiJTSVRFX09XTkVSIiwizW50aXRzZW1lbnR
zIjoiIn0=.5p3of7t110uwysF3zpm+YgICSHH8C/BHczdbVZx2VH8=
```

## 遠端元件安全性

Oracle Content Management 除了讓第三方開發人員可以將他們的自訂元件整合到 Oracle Content Management 平台，也能將元件儲存在遠端伺服器上。

每個遠端元件在 Oracle Content Management 都必須要有註冊的設定值和呈現端點。除了端點之外，開發人員也必須提供註冊元件的唯一秘密金鑰。

Oracle Content Management 會呼叫註冊元件的端點，以顯現網站頁面中的內容。由於這些端點在公用網際網路上都是公開的，因此開發人員應確認已註冊遠端元件的端點都是從 Oracle Content Management 呼叫。驗證呼叫者的真實性時，系統會將已簽署的權杖傳遞至某個 URL 的已註冊端點。系統會驗證已簽署權杖中內嵌的數位簽章與註冊過程中提供之遠端元件的祕密金鑰來認證呼叫方。

權杖的格式為：

```
{base64 encoded serialized JSON data}.{base64 encoded signature}
```

以下是傳送給已註冊應用程式端點的範例權杖：

```
eyJpbmN0YW5jZWlkIjoiQTRGOTE3REY5OTZEN0Q3ODBCMjUzODZFOTFEMDA3ODJGMjVBRjY2Rjc3OTIiLCJzaWduZGF0ZSI6IjE0NDU2MzcwNTk5MTciLCJzaXRlZG9tYWluIjoic2VydmljZTEtdGVuYW50MS51cy5vcnFjbGUuY29tIiwicGVybnVlc2lvdnMiOiJTSVRFX09XTkVSIiwuZW50aXRzZWl1bnRzIjoibn0=.5p3of7t110wuy5F3zpm+YgICSHH8C/BHczdbVZx2VH8=
```

權杖包含兩個不同的部分：資料和簽章 (以 ':' 分界字元區隔)。

一般的準則是，開發人員在授予某個元件之已註冊端點的存取權之前，應一律在「編輯」或「預覽」模式認證權杖。此外，在設定值端點認證呼叫方時，開發人員應一律負責尋找權杖之**權限**欄位中的 SITE\_OWNER 值。權杖的**權限**欄位只有在「編輯」模式才會顯示 SITE\_OWNER 值。在編輯階段作業期間產生的權杖一律不會保存回頁面模型，而且會變換為**權限**欄位值為 NULL 的程式實際執行權杖。

### 資料

執行處理的資料部分為 Base64 JSON 編碼字串。以下是該 JSON 字串的結構：

```
{
  "instanceid": "BBDC7614F693B75110D811E6C0B77C935FAEC5112E5E",
  "permissions": "",
  "entitlements": "",
  "signdate": "1435426735293",
  "sitedomain": "servicel-tenant4.localhost"
}
```

欄位名稱	描述
instanceid	Oracle Content Management 租用戶元件的唯一 ID。
signdate	簽章產生日期。
sitedomain	Oracle Content Management 執行處理的網域名稱。

欄位名稱	描述
permissions	網站成員的一組權限。在「編輯」模式中，它的值為 "SITE_OWNER"；否則不會有值。
entitlements	網站擁有者所購買的進階功能清單。

### 簽章

遠端元件執行處理的資料部分會在以 APP\_SECRET\_KEY 簽署之前先進行序列化。此秘密金鑰必須在註冊元件時產生並對開發人員顯示。此簽章是利用秘密金鑰來產生元件執行處理資料部分 (序列化的 JSON 結構) 的雜湊計算出來，如此處所示：

```
$signature = HMAC (serialized JSON structure, APP_SECRET_KEY)
```

系統將使用 SHA256 雜湊演算法產生簽章。接著會由序列化的 JSON 結構和產生的簽章串接成權杖，如下所示：

```
$instance = {base64encoded serialized JSON structure}.
{base64encoded $signature}
```

## 註冊遠端元件

您必須先在 Oracle Content Management 中註冊遠端元件，才能在網站中使用該元件。

您可以註冊第三方遠端元件和自行開發的遠端元件。

註冊要在 Oracle Content Management 執行處理中使用的遠端元件：

- 依序按一下 **開發人員** 和 **檢視所有元件**。
- 按一下 **建立**，然後選擇 **註冊遠端元件**。
- 在「註冊遠端元件」視窗中，輸入或選取資訊，包括：
  - 名稱**：使用者將會見到之元件的名稱。
  - 描述**：使用者將會見到之元件的描述。
  - 元件 URL**：iframe 中所使用的端點，用於呈現頁面中的元件內容。它必須是 HTTPS。
  - 設定值 URL**：iframe 中所使用的端點，用於呈現新增到頁面之遠端元件的設定值。它必須是 HTTPS。
  - 設定值寬度**：設定元件設定值面板的預設寬度 (像素)。
  - 設定值高度**：設定元件設定值面板的預設高度 (像素)。
  - 金鑰**：遠端元件的 192 位元 AES 金鑰，元件佈建之後，會使用此金鑰建立已簽署的雜湊權杖。此金鑰主要用來加密元件設定值，並確保可以安全地讀取及寫入這些設定值。
- 按一下 **註冊**。

遠端元件建立之後，其名稱會顯示在元件清單中。只要按一下元件清單中的元件名稱，您即可瀏覽註冊該元件所使用的檔案。

元件註冊資訊會儲存在網站 (這些網站都建立於相同的 Oracle Content Management 執行處理中) 所使用的目錄中，但是元件則繼續維持為遠端服務。

若您是元件擁有者，「網站產生器」的「自訂元件」面板中會新增元件圖示，其中包含您指定給該元件的名稱。您可以與其他使用者共用此元件，他們將會在「網站產生器」的「自訂元件」面板中見到此元件。

## 刪除元件

如果您有適當的權限，就可以刪除元件管理程式中的元件，讓它不再可供使用。當您刪除元件時，元件資料夾與其所有關聯的資料夾和檔案都會移至資源回收。

您若是元件的建立者 (即元件擁有者)，或某人與您共用元件且授予您管理者角色，您就能夠刪除元件管理程式中的元件。

### 備註：

您不能刪除任何網站或更新項目 (包括在資源回收中的網站或更新項目) 仍在使用中的元件。

刪除元件：

1. 在首頁中，按一下**開發人員**。
2. 按一下**檢視所有元件**。  
就會顯示目前已註冊的所有元件。
3. 選取元件名稱並從滑鼠右鍵功能表中選擇**刪除**，或按一下動作列中的 。  
系統會提示您確認動作。
4. 若要確認刪除動作，請按一下**是**。若要停止刪除動作，請按一下**否**。  
如果您確認刪除，系統就會將該元件與其所有關聯的資料夾和檔案都移至資源回收。

已刪除的元件資料夾仍會保留在資源回收中，直到：

- 您回復資料夾。
- 您永久地刪除資料夾。
- 達到資源回收配額。
- 系統根據服務管理員所設定的間隔自動清空資源回收。預設值為 90 天。

## Sites SDK

為 Oracle Content Management 開發的元件在網站中會呈現為元件，且可被拖放至網站特定頁面上的任何位置。

Sites SDK 會處理元件與頁面之間的所有通訊。

- `Sites.Settings.getProperty(propertyName, callbackFunction)`：提供回呼以針對自訂元件執行處理擷取所要求的特性。
- `Sites.Settings.setProperty(propertyName, propertyValue)`：依據自訂元件執行處理儲存所要求的特性。

針對所有自訂元件特性使用 `Sites.Settings.getProperty` 或 `Sites.Settings.setProperty`，然後使用 `SitesSDK.publish` 和 `SitesSDK.subscribe` 來監聽訊息事件。

請參閱 [Sites SDK 參考資料](#)。

## 自訂控制器檔案

在 Oracle Content Management 網站中，控制器檔案可用來顯示網站的每一頁。瀏覽器傳送網頁要求後，伺服器會傳送控制器檔案複本作為回應。針對要求的每一頁，伺服器都會傳遞相同的控制器檔案複本。

- [關於控制器檔案](#)
- [修改預設 controller.html 檔案](#)
- [關於 SCS 物件](#)
- [不應自訂的控制器檔案區段](#)
- [使用權杖允許自訂控制器檔案可攜性](#)
- [自訂控制器檔案範例](#)

## 關於控制器檔案

控制器檔案是小型 HTML 頁面，可動態起始剩餘頁面的呈現順序。這是可最先套用自訂項的位置，以影響網站每個頁面的行為。

控制器檔案的主要工作是為控制器 JavaScript 提供和代管執行環境。控制器 JavaScript 隨後會載入和顯示頁面。控制器檔案中的自訂項可覆寫和影響控制器 JavaScript 的作業。

## 預設控制器檔案

首次建立 Oracle Content Management 網站時，會在網站與預設控制器檔案 controller.html 之間建立關聯。

```
<!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>
```

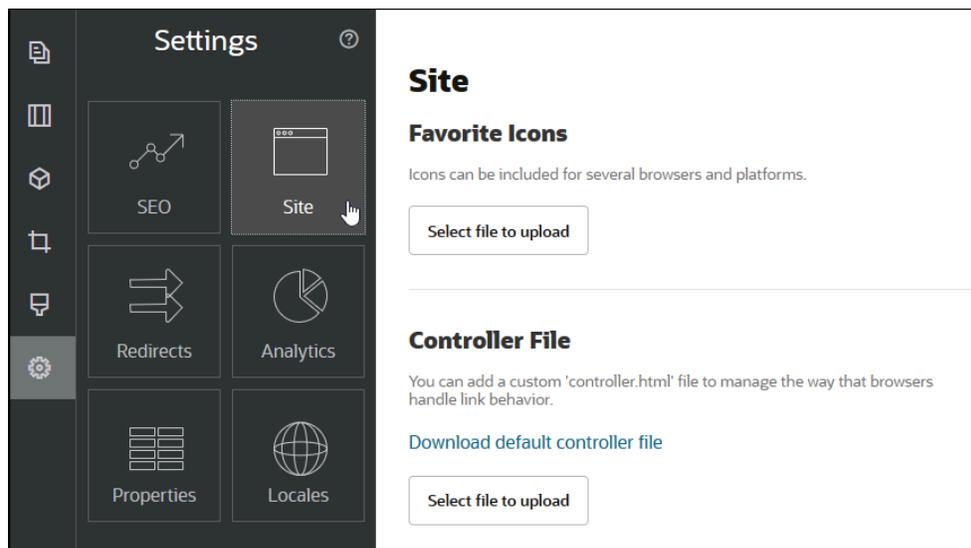
```
<!-- The body tag of the controller must have the id scsControllerBody.  
This identifier is used by the Controller JavaScript. If JavaScript is  
not enabled on the browser, the noscript tag content is displayed.-->  
<body id="scsControllerBody"><noscript>This site requires JavaScript to  
be enabled.</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>
```

## 修改預設 controller.html 檔案

下載網站的 controller.html 檔案並進行修改，即可影響每個網站頁面的行為。下載網站的 controller.html 檔案：

1. 在網站產生器中開啟您要修改的網站，然後將其設定為**編輯**。
2. 選取現有的更新或建立新的更新。
3. 在側邊導覽功能表依序按一下**設定值**和**網站**。



4. 在「控制器檔案」區段中，按一下**下載預設控制器檔案**。
5. 儲存到本機磁碟機，並透過編輯自訂 controller.html 檔案。
6. 完成編輯後回到網站的設定值頁面，然後在「控制器檔案」區段按一下**選取要上傳的檔案**。
7. 瀏覽到已修改的控制器檔案並將其上傳。

 備註：

控制器檔案的自訂項只會用於線上網站。這些自訂項不會在編輯網站或網站預覽中使用。若要在上傳網站自訂 `controller.html` 檔案後讓變更生效，必須確認網站更新並發布網站。

## 關於 SCS 物件

控制器 JavaScript 使用全域 SCS 物件中定義的變數和特性來呈現網頁。必須在控制器檔案中定義全域 JavaScript 物件才能包含 `controller.js` 檔案。SCS 物件中有兩個主要特性可用來進行自訂：

- `SCS.sitePrefix`
- `SCS.preInitRendering`
- `SCS.getDeviceInfo`

### SCS.sitePrefix

`SCS.sitePrefix` 變數可定義線上網站的路徑前置碼。通常此預設值為 `site/<siteName>/`。

控制器 JavaScript 使用這個值決定要顯示的網頁。例如，如果瀏覽器要求位於 `/site/SampleSite/products/index.html` 的頁面，`sitePrefix /site/SampleSite/` 就會允許 JavaScript 計算應顯示網站的 `products/index.html` 頁面。請注意，`sitePrefix` 必須是字串值，開頭和結尾都要是 `"/` 字元。

如有代理主機、其他基礎架構 (如 Akamai) 和 URL 對應規則，請修改此變數以允許自訂網站前置碼。例如，將 `sitePrefix` 設定為 `/intranet/ExampleSite/` 可允許瀏覽器使用該路徑名稱傳遞網站，而不是使用預設的 `/site/SampleSite/` 路徑前置碼。

 備註：

如果瀏覽器中的 URL 與網站前置碼不符，預設控制器 JavaScript 會在內部使用 `"/` 作為 `sitePrefix`。這允許在沒有自訂的情況下，使用虛名網域提供網站。

### SCS.preInitRendering

如果定義，控制器 JavaScript 程式碼會先呼叫 `SCS.preInitRendering` 函數，再執行任何運算邏輯。您可以在自訂控制器檔案中定義這個實用的函數，以覆寫控制器 JavaScript 的基本作業。

如果定義，`SCS.preInitRendering` 必須是函數。呼叫該函數不需任何引數，而且不會預期或處理任何傳回值。

### SCS.getDeviceInfo

`SCS.getDeviceInfo` 函數允許自訂控制器 JavaScript 內部的裝置偵測邏輯。這項偵測功能可用來判斷是否應傳遞行動或響應式版本的頁面。

SCS.getDeviceInfo 函數不使用任何引數，而且會傳回擁有兩個特性的 JavaScript 物件：

- **isMobile**—這是布林值特性，指出目前的裝置為行動從屬端，如智慧型手機。
- **isIOS**—這是布林值特性，指出目前的裝置正在執行 iOS 型作業系統。

如果自訂控制器未覆寫 SCS.getDeviceInfo，則會使用內建的預設實行。

## 不應自訂的控制器檔案區段

控制器檔案中的特定區段由 Oracle Content Management 伺服器進行處理，以便使用 CDN 及定義 URL 快取區段。下方的範例有內嵌的附註。

```
<!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>
```

## 使用權杖允許自訂控制器檔案可攜性

建立網站時，會在控制器檔案中使用網站的名稱。這會將控制器檔案綁定至具名網站，並使控制器檔案無法跨網站使用。

如果在網站使用的控制器檔案與控制器檔案中使用的名稱不同，網站將無法正常運作。這也適用於從擁有自訂控制器檔案的網站樣板建立的網站。下列動態評估的權杖允許控制器檔案與多個網站搭配使用。

權杖	描述
<code>[!-\$SCS_SITE_PREFIX-]</code>	<p>這可評估目前網站的網站前置碼。範例值可能包含：</p> <pre>/site/MySite/</pre> <p>或</p> <pre>/site/authsite/MySecureSite/</pre>
<code>[!-\$SCS_SITE_PATH-]</code>	<p>這可評估目前產品 CDN 位置。該值結尾不能使用 '/' 字元。範例值可能包含：</p> <pre>/site/MySite/_cache_0000</pre> <p>或</p> <pre>/site/authsite/MySecureSite</pre>
<code>[!-\$SCS_PRODUCT_PATH-]</code>	<p>這可評估目前產品 URL，包含快取索引鍵 (如果有的話)。該值結尾不能使用 '/' 字元。範例值可能包含：</p> <pre>https://www.example.com/cdn/cec/v21.1.2.23</pre> <p>或</p> <pre>https://www.example.com/cdn/cec/v21.1.3.18</pre>

## 自訂控制器檔案範例

提供下列自訂控制器檔案的範例：

- [變更網站前置碼](#)
- [自訂等待圖形](#)
- [自訂 Favicon](#)
- [自訂非 JavaScript 蒐集程式的 <noscript> 和 <meta> 標記](#)
- [預先擷取 JavaScript 檔案](#)

- 使用其他標記驗證網站擁有權
- 增強裝置偵測
- 使用權杖增強 [controller.htm](#) 可攜性

## 變更網站前置碼

下列範例控制器檔案定義 `preInitRendering` 函數，允許以多個前置碼傳遞網站。

### 備註：

若要使用此範例，必須設定 **CDN** 或其他代理主機以回應定義的前置碼。此外，如果函數中定義的其他前置碼都與瀏覽器 **URL** 不相符，則會使用預設行為。

```
<!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>
```

## 自訂等待圖形

下列範例控制器檔案可自訂當網路延遲導致呈現延遲時要顯示的等待圖形。



### 備註：

這個範例使用影像的資料 URL，而不是在呈現期間發出另一個要求。

```
<!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>
```

## 自訂 Favicon

下列範例控制器檔案可定義網站的自訂 **favicon**。如果瀏覽器不會從呈現頁面的文件物件模型動態載入 **favicon**，這個功能就很實用。

```
<!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>
```

## 自訂非 JavaScript 蒐集程式的 `<noscript>` 和 `<meta>` 標記

下列範例控制器檔案可針對不處理 JavaScript 的蒐集程式自訂 `<noscript>` 訊息和 `<meta>` 標記。

### 備註：

由於控制器檔案適用於網站的每一頁，因此每一頁都會顯示相同的描述標記。

```
<!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>
```

## 預先擷取 JavaScript 檔案

有些瀏覽器允許宣告目前或後續瀏覽所需的資源。控制器檔案中的標記可宣告應預先擷取或預先載入的資源。

### 備註：

由於每個 OCE 網頁都使用 `require.js` 和 `renderer.js`，因此是使用預先載入和預先擷取技術的最佳選擇。此外，如果 `<link>` 標記先列出 `href`，OCE 伺服器就會修正該值以便從 CDN 傳遞。

```
<!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>
```

## 使用其他標記驗證網站擁有權

第三方蒐集程式和搜尋引擎可能需要其他標記才能驗證網站擁有權。您可以視需要將標記插入控制器檔案。

```
<!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>

```

## 增強裝置偵測

在控制器中自訂裝置偵測代碼，即可辨識新的行動裝置或自訂的裝置使用者代理程式 ID。

### 備註：

- 控制器會呼叫 `getDeviceInfo` 函數以判斷是否正在使用行動裝置。這個範例會覆寫預設 `getDeviceInfo` 呼叫。
- `isMobile` 特性可決定呈現頁面時要使用的版面配置。
- `isIOS` 特性則決定與地圖元件搭配使用的提供者。
- 內建的預設 `isMobile` 測試為

```

/Mobi|iPhone|iPod|BlackBerry|IEMobile|Opera Mini/
i.test(userAgent) && !/iPad/i.test(userAgent)

```

- 內建的預設 `isIOS` 測試為

```

/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>
```

## 使用權杖增強 controller.htm 可攜性

根據預設，網站名稱會以硬式編碼的形式寫入 `controller.html` 檔案，導致難以重新命名網站或在多個網站重複使用相同的自訂控制器。這個問題可以透過權杖緩解。下列範例顯示如何使用可攜性權杖。

### 備註：

範例中的 `<link>` 標記僅供示範之用。基準 `controller.html` 不會包含這些標記。

```
<!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

## 使用其他工具開發網站

Oracle Content Management 提供數種方式，使用其他第三方工具與在 Oracle Content Management 內或 Oracle Content Management 外開發的應用程式和體驗互動。

### 使用 OCE 工具程式

Oracle Content Management OCE 工具程式和 SDK 可協助您開發自訂應用程式以使用在 Oracle Content Management 儲存區域管理的內容。這些應用程式可以在 Oracle Content Management 或使用第三方工具開發。

OCE 工具程式可協助您執行以下工作：

- 設定本機開發環境，以運用 Oracle Content Management 執行處理在本機進行元件、樣板、主題及內容版面配置的開發與測試
- 從範例建立元件、網站樣板及內容版面配置，在測試控制程式中執行與探索，以及在 Developer Cloud Service 環境中開發元件、樣板、主題及內容版面配置
- 將在 Oracle Content Management 中建立的元件和網站樣板匯入 Developer Cloud Service 專案與環境，進行來源管理並做進一步的開發
- 匯出 Developer Cloud Service 環境中的元件、樣板與內容版面配置，以便在 Oracle Content Management 中使用
- 複製現有元件、樣板或內容版面配置
- 撰寫單位測試
- 最佳化元件
- 將元件和樣板部署到 Oracle Content Management

可以在 <https://github.com/oracle/content-and-experience-toolkit#readme> 找到其他資訊。

### 體驗協調流程

如果您使用 Oracle Content Management 以外的工具來建立體驗，則可以將 Oracle Content Management 儲存區域連線至這些體驗，讓內容建立者可以在工作時預覽網站變更，以及可以在變更或發布內容時觸發自動建置。此體驗協調流程可自動執行內容提供者與網站開發人員之間的工作流程處理作業，以簡化體驗管理和發布。

## 使用 OCE 工具程式進行開發

OCE 工具程式可協助您開發 Oracle Content Management 的網站樣板、主題、自訂元件及內容版面配置。

使用 OCE 工具程式，您可以在自己的開發環境中工作，並使用 Oracle Content Management 中的資產儲存區域、檔案和資料夾。OCE 工具程式提供建立及開發自訂元件和網站樣板 (包括主題和內容版面配置) 的工具。包括可進行快速、迭代開發以及可快速上手範例單元測試的本機測試控制程式。

下列主題描述如何設定 OCE 工具程式，以及在本機機器或以 Developer Cloud Service 專案方式使用 OCE 工具程式進行開發：

- [在本機機器上設定 OCE 工具程式](#)
- [升級至 jQuery 3.5.x](#)
- [使用 Developer Cloud Service 開發 Oracle Content Management](#)
- [使用 OCE 工具程式將變更從測試環境傳輸到生產環境](#)
- [從樣板建立網站並保持相同的內容 GUID](#)
- [匯入及匯出分類](#)
- [匯入及匯出建議](#)
- [使用 OCE 工具程式開發自訂欄位編輯器](#)
- [將網站從一個伺服器移轉或更新至另一個伺服器](#)
- [使用 OCE 工具程式編製網站頁面索引](#)
- [使用 OCE 工具程式編製多語言網站的索引](#)
- [建立簡化的元件以便於元件開發](#)
- [編譯網站以提昇網站頁面的程式實際執行效能](#)
- [在 Oracle Content Management 伺服器中建立新網站或資產翻譯工作](#)
- [使用語言服務提供者翻譯網站](#)

## 在本機機器上設定 OCE 工具程式

在您的本機機器上，您可以從 [GitHub](#) 下載 OCE 工具程式。

若要設定 OCE 工具程式，請依照以下網址的指示

```
https://github.com/oracle/content-and-experience-toolkit/blob/master/README.md
```

完成設定，並準備使用 OCE 工具程式：

1. [透過 npm 安裝相依項](#)
2. [使用 cec 命令行公用程式](#)
3. [使用本機測試控制程式進行測試](#)

## 透過 npm 安裝相依項

使用 npm (節點套件管理程式) 為您的專案安裝網站相依項。

如果您使用代理主機存取網際網路，請使用 `npm config` 命令設定 npm 的代理主機。請參閱 <https://docs.npmjs.com/misc/config>。若要設定瀏覽器的代理主機，請參閱 <https://bower.io/docs/config/>。

### 備註：

請確定已在本機電腦上安裝 Node.js 8.0.0 或更新版本 (<https://nodejs.org/>)。

## 使用 cec 命令行公用程式

cec 跨平台的命令行公用程式提供建立和管理樣板與元件的命令。

使用 cec 命令行公用程式之前，請先在本機開發環境中為需連線至 Oracle Content Management 伺服器的命令建立來源。

整合式的協助提供搭配範例執行命令的資訊。若要檢視 cec 整合式的協助，您可以在命令行中輸入 cec 命令：

- 在終端機視窗中，前往 cec 目錄。
- 輸入 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
  or folder hierarchy on OCM server.      [alias: cfd]
  cec share-folder <name>                 Shares folder with
  users and groups on OCM server.        [alias: sfd]
  cec unshare-folder <name>              Deletes user or
  group access to a shared folder on OCM server. [alias: usfd]
  cec list-folder <path>                  Displays folder
  hierarchy on OCM server.                [alias: lfd]
  cec download-folder <path>              Downloads folder
  from OCM server.                        [alias: dlfd]
```

<pre>cec upload-folder &lt;path&gt; server. cec delete-folder &lt;path&gt; server. cec download-file &lt;file&gt; from OCM server. cec upload-file &lt;file&gt; OCM server. cec delete-file &lt;file&gt; server.</pre>	<pre>Uploads folder to OCM [alias: ulfd] Deletes folder on OCM [alias:  ] Downloads file &lt;file&gt; [alias: dlf] Uploads file &lt;file&gt; to [alias: ulf] Deletes file on OCM [alias:  ]</pre>
--	---

#### Components

<pre>cec create-component &lt;name&gt; &lt;name&gt;. cec copy-component &lt;source&gt; [&lt;destination&gt;] component named &lt;source&gt; to &lt;destination&gt;. cec import-component &lt;zip&gt; from &lt;zip&gt;. cec export-component &lt;name&gt; &lt;name&gt; as a zip file. cec download-component &lt;names&gt; components &lt;names&gt; from the OCM server. cec upload-component &lt;names&gt; &lt;names&gt; to the OCM server. cec control-component &lt;action&gt; &lt;action&gt; on components on OCM server. cec share-component &lt;name&gt; users and groups on OCM server. cec unshare-component &lt;name&gt; 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 &lt;name&gt; &lt;name&gt;. cec create-template-from-site &lt;name&gt; &lt;name&gt; from site &lt;site&gt; on the OCM server. cec download-template &lt;name&gt; &lt;name&gt; from the OCM server. cec compile-template &lt;source&gt; within the template. cec copy-template &lt;source&gt; [&lt;destination&gt;] template named &lt;source&gt; to &lt;destination&gt;. cec import-template &lt;zip&gt; &lt;zip&gt;. cec export-template &lt;name&gt; &lt;name&gt; as a zip file. cec upload-template &lt;name&gt; &lt;name&gt; to the OCM server. cec delete-template &lt;name&gt; &lt;name&gt; on the OCM server. cec share-template &lt;name&gt; users and groups on OCM server. cec unshare-template &lt;name&gt; access to a template on OCM server. cec update-template &lt;action&gt;</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</pre>
---	--

local template.	[alias: ut]
cec describe-template <name>	Describes the
template <name> package.	[alias:
dst]	
cec create-template-report <name>	Generates an asset
usage report for the template <name> package.	[alias: ctttr]
 Themes	
cec add-component-to-theme <component>	Adds a component
to a theme.	[alias: actt]
cec remove-component-from-theme <component>	Removes a
component from a theme.	[alias:
rcft]	
cec control-theme <action>	Performs action
<action> on theme on OCM server.	[alias: ctt]
cec share-theme <name>	Shares theme with
users and groups on OCM server.	[alias: sth]
cec unshare-theme <name>	Deletes user or
group access to a theme on OCM server.	[alias: usth]
 Sites	
cec create-site <name>	Creates Enterprise
Site <name>.	[alias: cs]
cec copy-site <name>	Copies Enterprise
Site <name>.	[alias: cps]
cec update-site <name>	Update Enterprise
Site <name>.	[alias: us]
cec transfer-site <name>	Transfers a site
from one OCM server to another.	[alias: ts]
cec transfer-site-content <name>	Creates scripts to
transfer site content from one OCM server to another.	[alias: tsc]
cec validate-site <name>	Validates site
<name>.	[alias: vs]
cec control-site <action>	Performs action
<action> on site on OCM server.	[alias: cts]
cec share-site <name>	Shares site with
users and groups on OCM server.	[alias: ss]
cec unshare-site <name>	Deletes user or
group access to a site on OCM server.	[alias: uss]
cec get-site-security <name>	Gets site security
on OCM server.	[alias: gss]
cec set-site-security <name>	Sets site security
on OCM server.	[alias: sss]
cec index-site <site>	Index the page
content of site <site> on OCM server.	[alias: is]
cec create-site-map <site>	Creates a site map
for site <site> on OCM server.	[alias: csm]
cec create-rss-feed <site>	Creates RSS feed
for site <site> on OCM server.	[alias: crf]
cec create-asset-report <site>	Generates an asset
usage report for site <site> on OCM server.	[alias: car]
cec upload-static-site-files <path>	Uploads files to
render statically from a site on OCM server.	[alias: ulss]
cec download-static-site-files <site>	Downloads the
static files from a site on OCM server.	[alias:

dlss]	
cec delete-static-site-files <site>	Deletes the static
files from a site on OCM server.	[alias: ]
cec refresh-prerender-cache <site>	Refreshes pre-render
cache for a site on OCM server.	[alias: rpc]
cec migrate-site <site>	Migrates a site from
OCI IC server to EC server.	[alias: ms]
Assets	
cec download-content	Downloads content from
OCM server.	[alias: dlc]
cec upload-content <name>	Uploads local content
to a repository on OCM server.	[alias: ulc]
cec control-content <action>	Performs action
<action> on channel items on OCM server.	[alias: ctct]
cec transfer-content <repository>	Creates scripts to
transfer content from one OCM server to another.	[alias: tc]
cec list-assets	Lists assets on OCM
server.	[alias: la]
cec create-digital-asset	Creates digital
asset	[alias: cda]
cec update-digital-asset <id>	Updates digital
asset	[alias: uda]
cec copy-assets <repository>	Copies assets to
another repository on OCM server.	[alias: ca]
cec create-asset-usage-report <assets>	Generates an asset
usage report for assets on OCM server.	[alias: caur]
Content	
cec create-repository <name>	Creates a repository on
OCM server.	[alias: cr]
cec control-repository <action>	Performs action
<action> on repositories on OCM server.	[alias: ctr]
cec share-repository <name>	Shares repository with
users and groups on OCM server.	[alias: sr]
cec unshare-repository <name>	Deletes user or group
access to a repository on OCM server.	[alias: usr]
cec create-channel <name>	Creates a channel on
OCM server.	[alias: cch]
cec share-channel <name>	Shares channel with
users and groups on OCM server.	[alias: sch]
cec unshare-channel <name>	Deletes user or group
access to a channel on OCM server.	[alias: usch]
cec create-localization-policy <name>	Creates a localization
policy on OCM server.	[alias: clp]
cec list-server-content-types	Lists all content types
from server.	[alias: lsct]
cec share-type <name>	Shares type with users
and groups on OCM server.	[alias: st]
cec unshare-type <name>	Deletes user or group
access to a type on OCM server.	[alias: ust]
cec download-type <name>	Downloads types from
OCM server.	[alias: dltp]
cec upload-type <name>	Uploads types to OCM
server.	[alias: ultp]

cec update-type <action> <action> on a type	Performs action [alias: utp]
cec download-recommendation <name> recommendation from the OCM server. dlr]	Downloads a [alias:
cec upload-recommendation <name> recommendation to the OCM server. [alias: ulr]	Uploads a
cec create-contentlayout <name> layout based on a content type.	Creates a content [alias: ccl]
cec add-contentlayout-mapping <contentlayout> type and content layout mapping.	Creates content [alias: aclm]
cec remove-contentlayout-mapping <contentlayout> layout mapping.	Removes a content [alias: rclm]
cec add-field-editor <name> editor to a field in a content type. afe]	Adds a field [alias:
cec remove-field-editor <name> editor from a field in a content type.	Removes a field [alias: rfe]
cec migrate-content <name> from OCI IC server to EC server.	Migrates content [alias: mc]
cec compile-content content items generating HTML renditions. cmpr]	Compiles the [alias:
cec upload-compiled-content <path> compiled content to OCM server. ulcc]	Uploads the [alias:
Taxonomies	
cec download-taxonomy <name> taxonomy from OCM server. dltx]	Downloads a [alias:
cec upload-taxonomy <taxonomy> to OCM server.	Uploads a taxonomy [alias: ultx]
cec control-taxonomy <action> taxonomy on OCM server.	Performs action on [alias: cttx]
Translation	
cec list-translation-jobs jobs.	Lists translation [alias: ltj]
cec create-translation-job <name> translation job <name> for a site on OCM server. [alias: ctj]	Creates a
cec download-translation-job <name> translation job <name> from OCM server. [alias: dtj]	Downloads
cec submit-translation-job <name> translation job <name> to translation connection <connection>. [alias: stj]	Submits
cec refresh-translation-job <name> translation job <name> from translation connection. [alias: rtj]	Refreshes
cec ingest-translation-job <name> job <name> from translation connection and ingest.	Gets translated [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:

--help, -h	Show Help
--server, -s	The registered OCM server
--channelToken, -c	The channel access token to use for content URLs
--type, -t	The type of content to retrieve from the serve [published   draft]
--pages, -p	The list of pages to compile
--recurse, -r	Compile all child pages of those specified in the page list
--debug, -d	Start the compiler with "--inspect-brk" option to debug compilation
--noDetailPages, -e	Do not generate compiled detail pages
--noDefaultDetailPageLink, -o	Do not generate compiled detail page for items/content lists that use the default detail page
--targetDevice, -a	The target device type when using adaptive layouts [desktop   mobile]
--siteName, -n	The target site name to use when compiling the template
--secureSite, -u	The target site is a secure site

```

--includeLocale, -l          Include default locale when creating pages
--verbose, -v                Run in verbose mode to display all warning
messages during compilation.
--ignoreErrors, -i          Ignore compilation errors when calculating
the exit code for the process.

```

Examples:

```

cec compile-template Templ          Compiles the
site in template Templ using content stored in the template.
cec compile-template Templ -c channelToken      Compiles the
site in template Templ using the given channelToken for any content URLs.
cec compile-template Templ -c channelToken -s UAT -t draft  Compiles the
site in template Templ retrieving draft content from the specified server.
cec compile-template Templ -p 104,112,183 -r      Compiles the
specified pages in the site in template Templ including all child pages.
cec compile-template Templ -d                Waits for the
debugger to be attached. Once attached, compiles the site in template Templ.

```

```

-----
cec copy-template
-----
Usage: cec copy-template <source> [<destination>]

```

Copies an existing template named <source> to <destination>. <source> is a folder name from src/templates

Options:  
--help, -h Show Help

Examples:  
cec copy-template Templ Temp2 Copies Templ to Temp2.

```

-----
cec import-template
-----
Usage: cec import-template <zip>

```

Imports a template from <zip>. Specify the absolute path of the zip file. The zip file name will be used as the template name.

Options:  
--help, -h Show Help

Examples:  
cec import-template /home/Templ.zip Imports the template Templ.

```

-----
cec export-template
-----
Usage: cec export-template <name>

```

Exports the template <name> as a zip file and provides the location of the zip file.

Options:

```
--help, -h      Show Help
--optimize, -o  Optimize the template
```

Examples:

```
cec export-template Templ  Exports the template Templ.
```

```
-----
cec upload-template
-----
```

Usage: cec upload-template <name>

Uploads the template <name> to the Content Management server. Specify the server with -s <server> or use the one specified in cec.properties file. Optionally specify -f <folder> to set the folder to upload the template zip file. Optionally specify -p to publish theme and components after import.

Options:

```
--help, -h          Show Help
--folder, -f        <folder> Folder to upload the template
zip file
--server, -s        <server> The registered OCM server
--optimize, -o      Optimize the template
--excludecontenttemplate, -x  Exclude content template
--excludecomponents, -e      Exclude components
--publish, -p       Publish theme and components
```

Examples:

```
cec upload-template StarterTemplate          Uploads the
template StarterTemplate.
cec upload-template StarterTemplate -s UAT    Uploads the
template StarterTemplate to the registered server UAT.
cec upload-template StarterTemplate -f Import/Templates  Uploads file
StarterTemplate.zip to folder Import/Templates and imports the template
StarterTemplate.
cec upload-template StarterTemplate -p       Publish the
theme and all components in StarterTemplate.zip after import
cec upload-template StarterTemplate -o      Optimizes
and uploads the template StarterTemplate.
cec upload-template StarterTemplate -x      Exclude the
"Content Template" from the template upload. "Content Template" upload
can be managed independently.
cec upload-template StarterTemplate -e      Exclude all
components from the template upload. Components can be uploaded
independently.
```

```
-----
cec delete-template
-----
```

Usage: cec delete-template <name>

Deletes the template <name> on the Content Management server. Specify the server with -s <server> or use the one specified in cec.properties file. Optionally specify -p to permanently delete the template.

```
Options:
  --help, -h      Show Help
  --server, -s    <server> The registered OCM server
  --permanent, -p flag to indicate to permanently delete the template
```

```
Examples:
  cec delete-template BlogTemplate
  cec delete-template BlogTemplate -p
  cec delete-template BlogTemplate -s UAT
```

```
-----
cec share-template
-----
Usage: cec share-template <name>
```

Shares template with users and groups on OCM server and assign a role. Specify the server with -s <server> or use the one specified in cec.properties file. The valid roles are

```
manager
contributor
downloader
viewer
```

```
Options:
  --help, -h      Show Help
  --users, -u     The comma separated list of user names
  --groups, -g    The comma separated list of group names
  --role, -r      The role [manager | contributor | downloader | viewer] to
assign to the users or groups [required]
  --server, -s    <server> The registered OCM server
```

```
Examples:
  cec share-template Templat1 -u user1,user2 -r manager
Share template Templat1 with user user1 and user2 and assign Manager role
to them
  cec share-template Templat1 -u user1,user2 -g group1,group2 -r manager
Share template Templat1 with user user1 and user2 and group group1 and
group2 and assign Manager role to them
  cec share-template Templat1 -u user1,user2 -r manager -s UAT
Share template Templat1 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 Site1 -t Site1Copy          Copies a
standard site
cec copy-site Site1 -t Site1Copy -r Repository1    Copies an
enterprise site
cec copy-site Site1 -t Site1Copy -r Repository1 -x sitelc  Copies an
enterprise site and sets the site prefix to sitelc
```

```
-----
cec update-site
-----
```

Usage: cec update-site <name>

Update Enterprise Site on OCM server using the content from the template. Specify the server with -s <server> or use the one specified in cec.properties file.

Options:

```
--help, -h          Show Help
--template, -t      <template> Template [required]
--excludecontenttemplate, -x  Exclude content template
--server, -s        <server> The registered OCM server
```

Examples:

```
cec update-site Site1 -t Templatel      Updates site Site1 using the
content from template Templatel
cec update-site Site1 -t Templatel -x  Updates site Site1 using the
content from template Templatel excluding the "Content Template"
```

```
-----
cec transfer-site
-----
```

Usage: cec transfer-site <name>

Transfers a site from one OCM server to another. By default all assets are transferred, optionally specify -p to transfer only published assets. Specify the source server with -s <server> and the destination server with -d <destination>. If the site contains assets from other repositories, optionally provide the repository mapping otherwise those assets will not be transferred.

Options:

```
--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:

<code>--help, -h</code>	Show Help
<code>--server, -s</code>	The registered OCM server the site is from [required]
<code>--destination, -d</code>	The registered OCM server to transfer the content [required]
<code>--repository, -r</code>	The site repository [required]
<code>--publishedassets, -p</code>	The flag to indicate published assets only
<code>--addtositecollection, -l</code>	Add assets to the site collection
<code>--repositorymappings, -m</code>	The repositories for assets from other repositories
<code>--number, -n</code>	The number of items in each batch, defaults to 500
<code>--execute, -e</code>	Execute the scripts

Examples:

```
cec transfer-site-content Site1 -s DEV -d UAT -r
Repository1
Generate script Site1_downloadcontent and Site1_uploadcontent
cec transfer-site-content Site1 -s DEV -d UAT -r Repository1 -
e                               Generate
script Site1_downloadcontent and Site1_uploadcontent and execute them
cec transfer-site-content Site1 -s DEV -d UAT -r Repository1 -n
200                               Set batch size
to 200 items
cec transfer-site-content Site1 -s DEV -d UAT -r Repository1 -
p                               Only the
published assets will be transferred
cec transfer-site-content Site1 -s DEV -d UAT -r Repository1 -
l                               The assets
from the site repository will be added to site default collection on
destination server
cec transfer-site-content Site1 -s DEV -d UAT -r Repository1 -m
"Shared Images:Shared Images,Shared Video:Shared Video"
```

```
-----
cec validate-site
-----
Usage: cec validate-site <name>
```

Validates site <name> on OCM server before publish or view publishing failure. Specify the server with `-s <server>` or use the one specified in `cec.properties` file.

Options:

```
--help, -h      Show Help
--server, -s    <server> The registered OCM server
```

Examples:

```
cec validate-site Site1           Validate site Site1 on the server
specified in cec.properties file
cec validate-site Site1 -s UAT    Validate site Site1 on the registered
server UAT
```

```
-----
cec control-site
-----
```

Usage: cec control-site <action>

Perform <action> on site on OCM server. Specify the site with -s <site>. Specify the server with -r <server> or use the one specified in cec.properties file. The valid actions are

```
publish
unpublish
bring-online
take-offline
```

Options:

```
--help, -h          Show Help
--site, -s          <site> Site [required]
--usedcontentonly, -u Publish used content only
--compilesite, -c   Compile site after publish
--staticonly, -t    Only publish site static files
--fullpublish, -f   Do a full publish
--server, -r        <server> The registered OCM server
```

Examples:

```
cec control-site publish -s Site1           Publish site Site1 on the
server specified in cec.properties file
cec control-site publish -s Site1 -u       Publish the site and all
assets added to the site's pages
cec control-site publish -s Site1 -c      Compile and publish site
Site1
cec control-site publish -s Site1 -t      Only publish the static
files of site Site1
cec control-site publish -s Site1 -f      Do a full publish of Site1
cec control-site publish -s Site1 -r UAT  Publish site Site1 on the
registered server UAT
cec control-site unpublish -s Site1 -r UAT Unpublish site Site1 on the
registered server UAT
cec control-site bring-online -s Site1 -r UAT Bring site Site1 online on
the registered server UAT
cec control-site take-offline -s Site1 -r UAT Take site Site1 offline on
the registered server UAT
```

```
-----
cec share-site
-----
```

Usage: cec share-site <name>

Shares site with users and groups on OCM server and assign a role. Specify the server with -s <server> or use the one specified in cec.properties file. The valid roles are

```
manager
contributor
downloader
viewer
```

Options:

```
--help, -h    Show Help
--users, -u    The comma separated list of user names
--groups, -g   The comma separated list of group names
--role, -r     The role [manager | contributor | downloader | viewer]
to assign to the users or groups [required]
--server, -s  <server> The registered OCM server
```

Examples:

```
cec share-site Site1 -u user1,user2 -r manager
Share site Site1 with user user1 and user2 and assign Manager role to
them
cec share-site Site1 -u user1,user2 -g group1,group2 -r manager
Share site Site1 with user user1 and user2 and group group1 and group2
and assign Manager role to them
cec share-site Site1 -u user1,user2 -r manager -s UAT
Share site Site1 with user user1 and user2 and assign Manager role to
them on the registered server UAT
```

```
-----
cec unshare-site
-----
```

Usage: cec unshare-site <name>

Deletes user or group access to a site on OCM server. Specify the server with -s <server> or use the one specified in cec.properties file.

Options:

```
--help, -h    Show Help
--users, -u    The comma separated list of user names
--groups, -g   The comma separated list of group names
--server, -s  <server> The registered OCM server
```

Examples:

```
cec unshare-site Site1 -u user1,user2
cec unshare-site Site1 -u user1,user2 -g group1,group2
cec unshare-site Site1 -u user1,user2 -s UAT
```

```
-----
cec get-site-security
-----
```

Usage: cec get-site-security <name>

Gets site security on OCM server. Specify the server with `-s <server>` or use the one specified in `cec.properties` file.

Options:

`--help, -h` Show Help  
`--server, -s <server>` The registered OCM server

Examples:

```
cec get-site-security Sitel
cec get-site-security Sitel -s UAT
```

```
-----
cec set-site-security
-----
```

Usage: `cec set-site-security <name>`

Makes the site publicly available to anyone, restrict the site to registered users, or restrict the site to specific users. Specify the server with `-r <server>` or use the one specified in `cec.properties` file. Optionally specify `-a <access>` to set who can access the site. The valid group names are

```
Cloud users
Visitors
Service users
Specific users
```

Options:

`--help, -h` Show Help  
`--signin, -s` If require sign in to access site: yes | no [required]  
`--access, -a` The comma separated list of group names  
`--addusers, -u` The comma separated list of users to access the site  
`--deleteusers, -d` The comma separated list of users to remove access from the site  
`--server, -r <server>` The registered OCM server

Examples:

```
cec set-site-security Sitel -s no
make the site publicly available to anyone
cec set-site-security Sitel -s no -r UAT
make the site publicly available to anyone on server UAT
cec set-site-security Sitel -s yes
Require everyone to sign in to access this site and any authenticated user
can access
cec set-site-security Sitel -s yes -a "Visitors,Service users"
Require everyone to sign in to access this site and all service visitors and
users can access
cec set-site-security Sitel -s yes -a "Specific users" -u user1,user2
Require everyone to sign in to access this site and only user1 and user2 can
access
cec set-site-security Sitel -s yes -d 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 Template1 -t -r Rep01 -c channel1
Upload content from template Template1 to repository Rep01 and add to
channel channel1
cec upload-content ~/Downloads/content.zip -f -r Rep01 -c channel1
Upload content from file ~/Downloads/content.zip to repository Rep01
and add to channel channel1
```

```
-----
cec control-content
-----
```

Usage: cec control-content <action>

Performs action <action> on channel items on OCM server. Specify the channel with -c <channel>. Specify the server with -s <server> or use the one specified in cec.properties file. The valid actions are

```
publish
unpublish
add
remove
```

Options:

```
--help, -h          Show Help
--channel, -c       Channel
--repository, -r    Repository, required when <action> is add
--collection, -l    Collection
--query, -q         Query to fetch the assets
--assets, -a        The comma separated list of asset GUIDS
--server, -s        The registered OCM server
--date, -d          Date to publish items
--name, -n          Name of the scheduled publishing job to create
```

Examples:

```
cec control-content publish -c
Channel1
           Publish all items in channel Channel1 on the
server specified in cec.properties file
cec control-content publish -c Channel1 -a
GUID1,GUID2
           Publish asset GUID1 and GUID2 in channel Channel1
cec control-content publish -c Channel1 -s
UAT
           Publish all items in channel Channel1 on the registered server
UAT
cec control-content unpublish -c Channel1 -s
UAT
           Unpublish all items in channel Channel1 on the registered server
UAT
cec control-content add -c Channel1 -r Rep01 -s
UAT
           Add all items in repository Rep01 to channel Channel1 on the
registered server UAT
cec control-content add -c Channel1 -r Rep01 -q 'type eq "BlogType"' -
s UAT
           Add all items in
```

```

repository Repol, matching the query to channel Channel1 on the registered
server UAT
  cec control-content add -c Channel1 -r Repol -q 'channels co
"CHANNELF43508F995FE582EC219EFEF03076128932B9A3F1DF6"' -s UAT Add all items
in repository Repol and Channel2 to channel Channel1 on the registered
server UAT
  cec control-content add -c Channel1 -r Repol -a GUID1,GUID2 -s
UAT Add asset GUID1
and GUID2 in repository Repol to channel Channel1
  cec control-content remove -c Channel1 -s
UAT
  Remove all items in channel Channel1 on the registered server UAT
  cec control-content add -l Collection1 -r Repol -s
UAT Add
all items in repository Repol to collection Collection1 on the registered
server UAT
  cec control-content remove -l Collection -s
UAT
  Remove all items in collection Collection1 on the registered server UAT
  cec control-content publish -c C1 -r R1 -s UAT -d "2021/9/21 0:30:00 PST" -
n Name Create a publishing job
called Name to publish all items in channel C1 on the specified date.
Requires server version: 21.2.1

```

```

-----
cec transfer-content
-----
Usage: cec transfer-content <repository>

```

Creates scripts to transfer content from one OCM server to another. This command is used to transfer large number of content items and the items are transferred in batches. By default the scripts will not be executed by this command. By default all assets are transferred, optionally specify `-p` to transfer only published assets. Specify the source server with `-s <server>` and the destination server with `-d <destination>`. Optionally specify `-n` for the number of items in each batch, defaults to 200.

Options:

<code>--help, -h</code>	Show Help
<code>--server, -s</code>	The registered OCM server the content is from
[required]	
<code>--destination, -d</code>	The registered OCM server to transfer the content
[required]	
<code>--channel, -c</code>	The channel
<code>--publishedassets, -p</code>	The flag to indicate published assets only
<code>--number, -n</code>	The number of items in each batch, defaults to 200
<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:

<code>--help, -h</code>	Show Help
<code>--collection, -l</code>	Collection name
<code>--channel, -c</code>	Channel name
<code>--query, -q</code>	Query to fetch the assets
<code>--assets, -a</code>	The comma separated list of asset GUIDS
<code>--target, -t</code>	The target repository [required]
<code>--server, -s</code>	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:

<code>--help, -h</code>	Show Help
-------------------------	-----------

```
--output, -o Output the report to a JSON file
--server, -s The registered OCM server
```

Examples:

```
cec create-asset-usage-report GUID1
cec create-asset-usage-report GUID1 -s UAT
cec create-asset-usage-report GUID1 -o                               The report
will be saved to GUID1AssetUsage.json
cec create-asset-usage-report GUID1,GUID2 -o                       The report
will be saved to GUID1_GUID2AssetUsage.json
cec create-asset-usage-report GUID1,GUID2 -o ItemReport.json      The report
will be saved to ItemReport.json
```

```
-----
cec create-repository
-----
```

Usage: cec create-repository <name>

Creates a repository on OCM server. Specify the server with `-s <server>` or use the one specified in `cec.properties` file. Optionally specify `-d <description>` to set the description. Optionally specify `-t <contenttypes>` to set the content types. Optionally specify `-c <channels>` to set the publishing channels. Optionally specify `-l <defaultlanguage>` to set the default language. Optionally specify `-p <type>` to set the repository type. The valid repository types are

```
asset
business
```

Options:

```
--help, -h           Show Help
--description, -d    The description for the repository
--type, -p           The repository type [asset | business]. Defaults to
asset
--contenttypes, -t  The comma separated list of content types for the
repository
--channels, -c      The comma separated list of publishing channels to
use in this repository
--defaultlanguage, -l The default language
--server, -s        The registered OCM server
```

Examples:

```
cec create-repository Repol
cec create-repository BusinessRepo -p business
cec create-repository Repol -d "Blog Repository" -t BlogType,AuthorType -c
channel1,channel2 -l en-US -s UAT
```

```
-----
cec control-repository
-----
```

Usage: cec control-repository <action>

Performs action <action> on repositories on OCM server. Specify the server with `-s <server>` or use the one specified in `cec.properties` file. The valid

actions are

```
add-type
remove-type
add-channel
remove-channel
add-taxonomy
remove-taxonomy
```

Options:

```
--help, -h          Show Help
--repository, -r    The comma separated list of content repositories
[required]
--contenttypes, -t  The comma separated list of content types
--channels, -c      The comma separated list of publishing channels
--taxonomies, -x    The comma separated list of promoted taxonomies
--server, -s        The registered OCM server
```

Examples:

```
cec control-repository add-type -r Repo1 -t Blog,Author
cec control-repository add-type -r Repo1,Repo2 -t Blog,Author
cec control-repository add-type -r Repo1 -t Blog,Author -s UAT
cec control-repository remove-type -r Repo1 -t Blog,Author
cec control-repository add-channel -r Repo1 -c channel1,channel2
cec control-repository remove-channel -r Repo1 -c channel1,channel2
cec control-repository add-taxonomy -r Repo1 -x Taxonomy1,Taxonomy2
cec control-repository remove-taxonomy -r Repo1 -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:

--help, -h	Show Help
--description, -d	The description for the channel
--type, -t	The channel type [public   secure]
--publishpolicy, -p	The publish policy [anythingPublished   onlyApproved]
--localizationpolicy, -l	The localization policy for the channel
--server, -s	The registered OCM server

Examples:

```
cec create-channel channel1                Create public
channel channel1 and everything can be published
cec create-channel channel1 -s UAT         On registered
server UAT, reate public channel channel1 and everything can be
published
cec create-channel channel1 -l en-fr       Create public
channel 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:

--help, -h	Show Help
--users, -u	The comma separated list of user names
--groups, -g	The comma separated list of group names
--role, -r	The role [manager   contributor   viewer] to assign to the users or groups [required]
--server, -s <server>	The registered OCM server

Examples:

```
cec share-channel Channel1 -u user1,user2 -r
manager                Share channel Channel1 with user user1 and
user2 and assign Manager role to them
cec share-channel Channel1 -u user1,user2 -g group1,group2 -r
manager                Share channel Channel1 with user user1 and user2 and group
group1 and group2 and assign Manager role to them
```

```
cec share-channel Channel1 -u user1,user2 -r manager -s UAT
Share channel Channel1 with user user1 and user2 and assign Manager role to
them on the registered server UAT
```

```
-----
cec unshare-channel
-----
```

Usage: cec unshare-channel <name>

Deletes user or group access to a channel on OCM server. Specify the server with `-s <server>` or use the one specified in `cec.properties` file.

Options:

```
--help, -h      Show Help
--users, -u     The comma separated list of user names
--groups, -g    The comma separated list of group names
--server, -s    <server> The registered OCM server
```

Examples:

```
cec unshare-channel Channel1 -u user1,user2
cec unshare-channel Channel1 -u user1,user2 -g group1,group2
cec unshare-channel Channel1 -u user1,user2 -s UAT
```

```
-----
cec create-localization-policy
-----
```

Usage: cec create-localization-policy <name>

Creates a localization policy on OCM server. Specify the server with `-s <server>` or use the one specified in `cec.properties` file. Specify `-r <requiredlanguages>` to set the required languages. Specify `-l <defaultlanguage>` to set the default language. Optionally specify `-o <optionallanguages>` to set the optional languages. Optionally specify `-d <description>` to set the description.

Options:

```
--help, -h      Show Help
--requiredlanguages, -r The comma separated list of required languages
for the localization policy [required]
--defaultlanguage, -l The default language [required]
--optionallanguages, -o The comma separated list of optional languages
for the localization policy
--description, -d   The description for the repository
--server, -s       The registered OCM server
```

Examples:

```
cec create-localization-policy en-us -r en-US -l en-US
cec create-localization-policy en-fr -r en-US,fr-FR -l en-US
cec create-localization-policy multi -r en-US,fr-FR -l en-US -o zh-CN -d
"Policy for Blog" -s UAT
```

```
-----
cec list-server-content-types
-----
```

Usage: cec list-server-content-types

Lists all content types from server.

Options:

--help, -h Show Help  
--server, -s <server> The registered OCM server

Examples:

cec list-server-content-types  
cec list-server-content-types -s UAT

-----  
cec share-type  
-----

Usage: cec share-type <name>

Shares type with users and groups on OCM server and assign a role. Specify the server with -s <server> or use the one specified in cec.properties file. The valid roles are

manager  
contributor

Options:

--help, -h Show Help  
--users, -u The comma separated list of user names  
--groups, -g The comma separated list of group names  
--role, -r The role [manager | contributor] to assign to the users or groups [required]  
--server, -s <server> The registered OCM server

Examples:

cec share-type BlogType -u user1,user2 -r manager  
Share type BlogType with user user1 and user2 and assign Manager role to them  
cec share-type BlogType -u user1,user2 -g group1,group2 -r manager  
Share type BlogType with user user1 and user2 and group group1 and group2 and assign Manager role to them  
cec share-type BlogType -u user1,user2 -r manager -s UAT  
Share type BlogType with user user1 and user2 and assign Manager role to them on the registered server UAT

-----  
cec unshare-type  
-----

Usage: cec unshare-type <name>

Deletes user or group access to a type on OCM server. Specify the server with -s <server> or use the one specified in cec.properties file.

Options:

--help, -h Show Help  
--users, -u The comma separated list of user names  
--groups, -g The comma separated list of group names

```
--server, -s <server> The registered OCM server
```

Examples:

```
cec unshare-type BlogType -u user1,user2
cec unshare-type BlogType -u user1,user2 -g group1,group2
cec unshare-type BlogType -u user1,user2 -s UAT
```

```
-----
cec download-type
-----
```

```
Usage: cec download-type <name>
```

Downloads types from OCM server. The content field editors and forms for the types will also be downloaded. Specify the server with `-s <server>` or use the one specified in `cec.properties` file.

Options:

```
--help, -h    Show Help
--server, -s <server> The registered OCM server
```

Examples:

```
cec download-type BlogType           Download content type BlogType and
save to local folder src/types/BlogType
cec download-type BlogType,BlogAuthor Download content type BlogType and
BlogAuthor and save to local folder
cec download-type BlogType -s UAT
```

```
-----
cec upload-type
-----
```

```
Usage: cec upload-type <name>
```

Uploads types to OCM server. The content field editors and forms for the types will also be uploaded. Specify the server with `-s <server>` or use the one specified in `cec.properties` file.

Options:

```
--help, -h    Show Help
--server, -s <server> The registered OCM server
```

Examples:

```
cec upload-type BlogType
cec upload-type BlogType -s UAT
cec upload-type BlogAuthor,BlogType Place the referenced types first
```

```
-----
cec update-type
-----
```

```
Usage: cec update-type <action>
```

Performs action `<action>` on a type in a local template or on OCM server. Specify the server with `-s <server>` or use the one specified in `cec.properties` file. The valid actions are

```
add-content-form
```

remove-content-form

Options:

--help, -h Show Help  
--objectname, -o the content form [required]  
--contenttype, -c the content type [required]  
--template, -t The template the content type is from  
--contenttemplate, -t Flag to indicate the template is a content  
template  
--server, -s The registered OCM server

Examples:

cec update-type add-content-form -o form1 -c BlogPost -t  
BlogTemplate Associate content form form1 with content  
type BlogPost from local template at src/templates/BlogTemplate  
cec update-type add-content-form -o form1 -c BlogPost -t  
BlogTemplateContent -n Associate content form form1 with content  
type BlogPost from local template at src/content/BlogTemplateContent  
cec update-type add-content-form -o form1 -c BlogPost -s  
UAT Associate content form form1 with content  
type BlogPost on the registered server UAT  
cec update-type add-content-form -o form1 -c BlogPost -  
s Associate content form form1 with content  
type BlogPost on the server specified in cec.properties file  
cec update-type remove-content-form -o form1 -c BlogPost -t  
BlogTemplate Change not to use form1 when create or edit  
items of type BlogPost from local template at src/templates/BlogTemplate  
cec update-type remove-content-form -o form1 -c BlogPost -t  
BlogTemplateContent -n Change not to use form1 when create or edit  
items of type BlogPost from local template at src/content/  
BlogTemplateContent  
cec update-type remove-content-form -o form1 -c BlogPost -s  
UAT Change not to use form1 when create or edit  
items of type BlogPost on the registered server UAT

-----  
cec download-recommendation  
-----

Usage: cec download-recommendation <name>

Downloads a recommendation from the Content Management server. Specify the server with -s <server> or use the one specified in cec.properties file. Optionally specify repository with -r <repository>. Optionally specify -p to download the published version.

Options:

--help, -h Show Help  
--repository, -r The repository  
--published, -p The flag to indicate published version  
--channel, -c Channel name, required when <published> is set  
--server, -s <server> The registered OCM server

Examples:

cec download-recommendation Recommendation1 Downloads

```
Recommendation1
  cec download-recommendation Recommendation1 -p -c Channel1 Downloads
Recommendation1 published to channel Channel1
  cec download-recommendation Recommendation1 -s UAT
  cec download-recommendation Recommendation1 -r Repol
```

```
-----
cec upload-recommendation
-----
```

Usage: cec upload-recommendation <name>

Uploads a recommendation to repository <repository> on OCM server. Specify the server with -s <server> or use the one specified in cec.properties file.

Options:

```
--help, -h          Show Help
--repository, -r    The repository [required]
--server, -s        <server> The registered OCM server
```

Examples:

```
cec upload-recommendation Recommendation1 -r Repol
cec upload-recommendation Recommendation1 -r Repol -s UAT
```

```
-----
cec create-contentlayout
-----
```

Usage: cec create-contentlayout <name>

Creates a content layout based on a content type from a local template or from OCM server. By default, an "overview" content layout is created. Optionally specify -s <style> to create in a different style.

Valid values for <style> are:

```
detail
overview
```

Options:

```
--help, -h          Show Help
--contenttype, -c    <contenttype> Content layout is based on
[required]
--template, -t      <template> Content type is from
--server, -r        The registered OCM server
--style, -s         <style> Content layout style: detail | overview
--addcustomsettings, -a Add support for custom settings when used in Sites
```

Examples:

```
cec create-contentlayout Blog-Post-Overview-Layout -c Blog-Post -t
BlogTemplate
cec create-contentlayout Blog-Post-Detail-Layout -c Blog-Post -t
BlogTemplate -s detail
cec create-contentlayout Blog-Post-Overview-Layout -c Blog-Post -t
BlogTemplate -a          Add custom settings when used in Sites
cec create-contentlayout Blog-Post-Overview-Layout -c Blog-Post -
r                          Use content type Blog-Post from the server specified
```

```
in cec.properties file
cec create-contentlayout Blog-Post-Overview-Layout -c Blog-Post -r
UAT -s detail          Use content type Blog-Post from the registered
server UAT
```

```
-----
cec add-contentlayout-mapping
-----
```

Usage: cec add-contentlayout-mapping <contentlayout>

Creates content type and content layout mapping. By default, the mapping is set for "Default". Optionally specify -s <layoutstyle> to name the mapping. By default, the mapping is set for desktop. Optionally specify -m to set the mapping for mobile.

Options:

```
--help, -h          Show Help
--contenttype, -c <contenttype> Content layout is based on
[required]
--template, -t      <template> The mapping is for
--layoutstyle, -s <style> Content layout style
--mobile, -m        mobile mapping
--server, -r        <server> The registered OCM server
```

Examples:

```
cec add-contentlayout-mapping Blog-Post-Detail-Layout -c Blog-Post -t
BlogTemplate
cec add-contentlayout-mapping Blog-Post-Detail-Layout -c Blog-Post -t
BlogTemplate -m
cec add-contentlayout-mapping Blog-Post-Detail-Layout -c Blog-Post -t
BlogTemplate -s Details
cec add-contentlayout-mapping Blog-Post-Overview-Layout -c Blog-Post -
t BlogTemplate -s "Content List Default"
cec add-contentlayout-mapping Blog-Post-Overview-Layout -c Blog-Post -
t BlogTemplate -s Overview
cec add-contentlayout-mapping Blog-Post-Overview-Layout -c Blog-Post -
r UAT          Set "Content Item Default" to
Blog-Post-Overview-Layout for content type Blog-Post on server UAT
```

```
-----
cec remove-contentlayout-mapping
-----
```

Usage: cec remove-contentlayout-mapping <contentlayout>

Removes a content layout mapping. By default, all mappings for the content layout are removed. Optionally specify -s <layoutstyle> to name the mapping and -m to indicate the mobile mapping.

Options:

```
--help, -h          Show Help
--contenttype, -c <contenttype> 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
```

## 使用本機測試控制程式進行測試

先在本機測試控制程式中執行自訂元件、樣板和內容版面配置，再將它們匯入 Oracle Content Management。

開始本機測試控制程式：

1. 在終端機視窗中輸入 `cd cec`。
2. 輸入 `cec develop &` 或 `cec develop --server <server-name> &`
3. 在瀏覽器中開啟 `http://localhost:8085`，查看在本機測試控制程式中執行的元件、樣板及內容版面配置。
4. 您可以在下列目錄中找到您的元件、樣板、主題等：
  - `cec/src/main/components`
  - `cec/src/main/templates`
  - `cec/src/main/themes`

## 升級至 jQuery 3.5.x

開發人員可以使用 OCE 工具程式識別未正確關閉的 `html` 標記。

升級至 jQuery 3.5.x 以取得 HTML 剖析器中的安全修正。

<https://blog.jquery.com/2020/04/10/jquery-3-5-0-released>

在 jQuery 3.5.x 中剖析特定 HTML 字串與在 3.4.x 中剖析相同字串將會得到不同的結果。這些字串包括自行關閉、違反 HTML 標準的標記；例如："<div />"。這類字串，尤其是較大標記序列的一部分時，在 3.5.x 和 3.4.x 中可能會以不同方式剖析。

您可以執行 `cec create-asset-report <site>`，尋找頁面 JSON 檔案、元件 HTML 和 JS 檔案中未正確關閉的 html 標記。

## 使用 Developer Cloud Service 開發 Oracle Content Management

Developer Cloud Service 整合的 OCE 工具程式可協助您開發 Oracle Content Management 網站樣板、主題、自訂元件以及內容版面配置。

使用 OCE 工具程式時，您可以使用 Oracle Content Management 中的資產儲存區域、檔案及資料夾。OCE 工具程式提供建立及開發自訂元件和網站樣板 (包括主題和內容版面配置) 的工具。其中包括 Git 儲存區域及本機測試控制程式，可進行快速、迭代開發以及可快速上手的範例單元測試。

OCE 工具程式可協助您執行以下工作：

- 設定本機開發環境，以運用 Oracle Content Management 執行處理在本機進行元件、樣板、主題及內容版面配置的開發與測試
- 從範例建立元件、網站樣板與內容版面配置、在測試控制程式中執行與探索，以及在 Developer Cloud Service 環境中開發元件、樣板、主題以及內容版面配置
- 將在 Oracle Content Management 中建立的元件和網站樣板匯入 Developer Cloud Service 環境，進行來源管理並做進一步的開發
- 匯出 Developer Cloud Service 環境中的元件、樣板與內容版面配置，以便在 Oracle Content Management 中使用
- 複製現有元件、樣板或內容版面配置
- 撰寫單元測試
- 最佳化元件
- 將元件和樣板部署到 Oracle Content Management

下列主題描述如何設定 Developer Cloud Service 以便開發自訂元件、網站樣板、主題與內容版面配置：

1. [關於使用 Developer Cloud Service](#)
2. [登入 Oracle Content Management 的 Developer Cloud Service 主控台](#)
3. [在 Developer Cloud Service 中建立專案](#)
4. [將 OCE 工具程式新增至新 Git 儲存區域中的專案程式碼](#)
5. [在本機測試控制程式中測試自訂元件、樣板及內容版面配置](#)

## 6. 合併變更

下列主題提供關於使用 Oracle Content Management OCE 工具程式的資訊：

- 使用 `cec` 命令行公用程式
- 使用 Developer Cloud Service 開發自訂元件
- 使用 Developer Cloud Service 開發樣板
- 開發內容版面配置

## 關於使用 Developer Cloud Service

Oracle Developer Cloud Service 是一種雲端軟體開發「平台即服務」(PaaS)，也是一種代管應用程式開發基礎架構的環境。它提供運用開放原始碼標準的整合，能夠在 Oracle Cloud 內開發、協作及部署應用程式。

Developer Cloud Service 是由 Oracle Cloud 代管的軟體與服務集合，透過與 Git、Maven、問題及 Wiki 的整合，協助您有效地管理應用程式開發生命週期。使用 Oracle Developer Cloud Service，您可以將應用程式原始程式碼確認至 Oracle Cloud 上的 Git 儲存區域、線上追蹤指定的問題與瑕疵、使用 Wiki 頁面分享資訊、對等複查原始程式碼以及監督專案組建。順利測試之後，您即可將專案部署到 Oracle Content Management。

## 登入 Oracle Content Management 的 Developer Cloud Service 主控台

在 Developer Cloud Service 主控台上開始開發適用於 Oracle Content Management 的自訂元件。

身為 Oracle Cloud 服務的管理員，您可以透過「我的服務管理」建立及管理您的雲端服務。如果您是 Oracle Content Management 的服務執行處理管理員和「標準開發人員服務」的服務管理員，便可以設定它們並開始使用：

1. 使用為您帳戶提供的資訊來登入 Oracle Cloud。
2. 登入「我的服務管理」來建立和管理您的 Oracle Content Management 執行處理與「標準開發人員服務」。



3. 依要求確認您的 Oracle Developer Cloud Service 電子郵件。
4. 使用您服務的訂閱詳細資訊來設定您的 Oracle Content Management 執行處理，然後前往 Oracle Content Management 執行處理的 URL。

5. 前往您的「標準開發人員服務」URL。
6. 登入 Oracle Developer Cloud Service 帳戶。  
存取 Developer Cloud Service URL 並登入主控台。

## 在 Developer Cloud Service 中建立專案

您可以使用 "Content Experience Cloud" 專案樣板在 Developer Cloud Service 中建立專案，或者使用空的 Git 儲存區域建立專案，然後從您的 Oracle Content Management 執行處理匯入 OCE 工具程式。

- 使用 [Oracle Content Management 樣板建立 Developer Cloud Service 專案](#)
- 使用從 [Oracle Content Management 下載的 OCE 工具程式在 Developer Cloud Service 中建立專案](#)
- 將 [OCE 工具程式新增至新 Git 儲存區域中的專案程式碼](#)

## 使用 Oracle Content Management 樣板建立 Developer Cloud Service 專案

在 Developer Cloud Service 中建立開發自訂元件、樣板、主題及內容版面配置的專案。

建立專案：

1. 登入 Developer Cloud Service 主控台後，按一下**新專案**。
2. 在樣板清單中，選擇**內容管理**，然後按一下**下一步**。
3. 在「專案特性」底下的特性中，選擇 **Wiki 標記** 欄位中的 **CONFLUENCE**。

## 使用從 Oracle Content Management 下載的 OCE 工具程式在 Developer Cloud Service 中建立專案

在 Developer Cloud Service 中建立開發自訂元件、樣板、主題及內容版面配置的專案。

建立專案：

1. 登入 Developer Cloud Service 主控台後，按一下**新專案**。
2. 命名您的專案，輸入或選取您所要的其他專案詳細資訊，然後按一下**下一步**。
3. 在樣板清單中，選擇**初始儲存區域**，然後按一下**下一步**。
4. 在「專案特性」下的特性中，選擇「初始儲存區域」的 Empty Repository。按一下**完成**。

## 將 OCE 工具程式新增至新 Git 儲存區域中的專案程式碼

您可以將 OCE 工具程式新增至新的空專案 Git 儲存區域。

1. 在新專案的 **REPOSITORIES** 底下，複製專案 Git 儲存區域的 HTTP URL。
2. 開啟終端機視窗並輸入以下命令：`git clone <your-project> .git`
  - a. 在系統要求時，輸入您的 Developer Cloud Service 密碼。

- b. 如果見到 "git is not a command" 錯誤，請從 <https://git-scm.com/downloads> 安裝 Git，然後重新輸入 git clone 命令。
3. git clone git@github.com:oracle/content-and-experience-toolkit.git  
或者，可以從以下位置下載：<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

## 在本機測試控制程式中測試自訂元件、樣板及內容版面配置

先在本機測試控制程式中執行自訂元件、樣板和內容版面配置，再將它們匯入 Oracle Content Management。

開始本機測試控制程式：

1. 在終端機視窗中輸入 cd cec-components。  
輸入 npm start &
2. 在瀏覽器中開啟 <http://localhost:8085>，查看在本機測試控制程式中執行的元件、樣板及內容版面配置。

在本機伺服器測試元件時，您可以選擇使用本機樣板或 Oracle Content Management 伺服器的內容。

## 合併變更

建立元件、樣板或內容版面配置之後，或是在本機機器上編輯原始程式碼之後，您需要將新的和變更過的元件和樣板合併到專案的 Git 儲存區域中。

若要將變更合併至 Git 儲存區域中，請在終端機視窗中依序輸入下列命令。

```
cd cec-components git pull
git add .
git status
git commit -a -m "Your comments" git pull
git push
```

## 使用 OCE 工具程式將變更從測試環境傳輸到生產環境

開發網站樣板之後，您可以使用 OCE 工具程式的命令行介面 (CLI) 將樣板從開發環境傳輸至 Oracle Content Management 伺服器的測試環境再到生產環境中。

若要傳輸變更，您可以使用工具程式命令建立網站，以及管理其在開發、測試和生產伺服器上的生命週期。您可以對開發環境中的網站進行變更，然後將這些變更傳輸至測試和生產環境。您也可以將這組命令行公用程式併入命令檔環境，以管理部署。您可以使用 CLI 公用程式來推新項目 (例如資產和元件) 以及現有內容的更新。

下列步驟顯示如何使用 OCE 工具程式 CLI 將您的變更從開發環境傳輸至測試再到生產環境：

1. 使用相同儲存區域和本地化規定設定開發、測試和生產伺服器。

若要將開發伺服器的變更傳輸至測試伺服器再到生產伺服器，您需要在三個伺服器上都設定相同名稱的儲存區域和本地化規定。資產儲存區域的預設本地化規定為 en-US，但如果三個伺服器的本地化規定都相同，您可以使用其他本地化規定。

請參閱 [Set Up Asset Repositories](#)。

2. 使用 Oracle Content Management 註冊您的開發、測試和生產伺服器。

在您將變更傳輸至網站之前，您必須先註冊每個伺服器。您可以使用 OCE 工具程式提供的 `cec register-server` 命令註冊伺服器：

```
cec register-server <name>
```

請指定下列命令選項：

- `-e <endpoint>` 表示伺服器 URL
- `-u <user>` 和 `-p <password>` 用於連線至伺服器
- `-t <type>`，設定伺服器類型 (選擇性)。預設值為 `pod_ec`。

連線至 Oracle Public Cloud 上的 Oracle Content Management 租用戶時，只能使用 `pod_ec`。

例如，以下命令會註冊屬於 Oracle Public Cloud 租用戶的伺服器：

```
cec register-server DEV -e https://DEV.example.com -u user1 -p  
<password>
```

下一個命令會註冊 Oracle Content Management 的獨立開發執行處理：

```
cec register-server DEV -e https://DEV.git.oraclecorp.example.com -  
u user1 -p <password>
```

註冊 Oracle Content Management 伺服器之後，便可以使用 `cec list` 工具程式命令列出其內容。

以下命令會列出開發伺服器的內容：

```
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:
  Published   Online           Theme                               Type
  Name                               StarterSiteTheme                 Enterprise

Templates:
  Name                               Theme                               Type
  StarterTemplate                     StarterTheme                       Standard

```

**3.** 將網站樣板上傳至開發伺服器，並從樣板建立網站。

您可以使用 `cec create-template` 命令建立網站樣板，然後將樣板上傳至開發伺服器。接著便可使用 `cec create-site` 命令從該樣板建立網站。以下命令會建立一個樣板並上傳該樣板：

```

- cec create-template blog -f BlogTemplate
- cec upload-template blog -s DEV

```

下一個命令會從上傳的樣板建立名為 `blog` 的網站：

```

cec-components> cec create-site blog -t blog -r r -l "en-US" -d "en-US" --
server DEV
- Logged in to remote server: <https:<host:<port>
- establish user session
- get template
- get repository
- get localization policy
- creating enterprise site . . .
  name           blog
  template       blog
  site prefix    blog
  repository     r
  localization policy en-US
  default language en-US
- submit create site site
- create site in process: percentage 95
- site created

```

**4.** 發布網站並讓該網站在您的開發伺服器上線。

建立網站之後，您可以使用 `cec control-site` 命令發布網站並讓網站上線：

```
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. 若要將網站從 **DEV** 移至 **UAT**，您需要將該網站封裝起來。在伺服器之間移動網站的封裝模型為樣板。從您在開發伺服器上建立的網站建立新樣板，然後下載該樣板。

下列範例中的 `cec create-template-from-site` 命令會從 `blog` 建立名為 `blog2` 的網站樣板。

```
cec create-template-from-site blog2 -s blog
```

使用 `cec download-template` 命令從開發網站下載您建立的樣板：

```
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. 上傳樣板以建立內容類型和內容版面配置對應。

```
cec upload-template blog2 --server UAT
```

7. 上傳樣板，但是從樣板排除內容項目 (內容樣板)。

```
cec upload-template blog2 --server UAT -x
```

執行此步驟以建立一個內容具有與原始網站相同 **GUID** 的網站。當您從包含內容的樣板建立網站時，新網站中的所有內容都會具有新的 **GUID**。因為我們要允許更新內容而不是建立新內容，所以您需要從樣板中排除內容。

8. 從樣板建立網站。

```
cec create-site blog -t blog2 -r r -l "en-US" -d "end-US" --server UAT
```

9. 將內容樣板上傳至網站的通道和集合。您必須執行此步驟，因為您已在步驟 7 中將其從樣板中排除。

```
cec upload-content blog2 -t -r r -c blog -l "blog site" --server UAT
```

10. 發布網站並讓該網站在您的測試伺服器上線。

使用 `cec control-site` 命令發布網站並讓該網站上線：

```
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. 如果之後有對 DEV 伺服器上的 blog 網站進行變更，您可以將變更傳輸至已在 UAT 伺服器上建立的網站。
12. 從您的網站建立另一個樣板，讓該樣板包含您的變更。

```
cec create-template-from-site blog3 -s blog --server DEV
```

13. 下載樣板。

```
cec download-template blog3 -s DEV
```

14. 上傳此樣板並從中建立一個網站，以將變更傳輸至您的測試環境。

```
cec upload-template blog3 -s UAT
```

此命令會建立或更新已變更的任何元件和主題，但是內容除外。

15. 現在，使用 `update-site` 命令取得內容並更新頁面。

```
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. 檢查網站以確認已傳輸變更。
17. 執行將網站從 DEV 移至 UAT 的相同步驟，從 UAT 伺服器移至 PROD 伺服器。
18. 在生產伺服器上建立網站，使其上線，然後確認變更。

您可以使用 `cec list` 命令檢視生產網站的內容，確定其中包含您在開發環境中所作的變更。此外，您可以檢查網站以確認變更已傳輸至生產環境。

## 加密密碼

向 OCE 工具程式註冊伺服器時，您必須加密密碼，伺服器才能在本機使用。

1. 使用包含純文字密碼的 `cec register-server` 命令註冊 Oracle Content Management 伺服器。
2. 使用 `cec create-encryption-key` 命令加密密碼。

```

cec create-encryption-key <file> [alias: cek]
Create an encryption key to encrypt/decrypt password for servers.

```

3. 使用加密金鑰再次註冊伺服器，讓伺服器在本機進行開發與測試。
- 加密的密碼會儲存在伺服器連線檔案中。當您連線至已註冊伺服器，密碼就會進行解密。

## 註冊伺服器

您可以在 OCE 工具程式註冊伺服器。

使用 `cec register-server` 命令搭配加密金鑰來註冊 Oracle Content Management 伺服器，以進行本機開發與測試。

註冊伺服器時，加密密碼可讓伺服器搭配 OCE 工具程式使用。請參閱[加密密碼](#)。

## 建立網站的使用狀況和權限報表

您可以建立報表來驗證與修正目標伺服器成員的權限，以將網站從測試環境移轉到生產環境。

使用 `cec create-asset-report`，並且依下列步驟進行檢查：

1. 檢查所有網站使用者自建物件的成員和通道指定：
  - 主題
  - 樣板
  - 元件
  - 內容類型

2. 標記您可以解決的問題。

例如：

```
cec create-asset-report blog1 -s <registered-server> -o
cec create-asset-report trbcent -s <registered-server> -o
```

此報表會使用 JSON 檔案格式產生，您可用以檢查使用狀況和權限問題。以下命令適用於權限修正：

- `cec share-type`：與 Oracle Content Management 伺服器中的使用者共用類型。
- `cec unshare-type`：移除 Oracle Content Management 伺服器中指定使用者對類型的存取權。
- `cec share-repository`：共用 Oracle Content Management 伺服器中的儲存區域 (以及該儲存區域所使用的類型)。
- `cec unshare-repository`：移除使用者對 Oracle Content Management 伺服器中儲存區域的存取權。
- 例如：

```
cec share-repository Rep01 -u <user-name1>,<user-name2> -r manager -t -s <registered-server>
```

## 下載與上傳文件和資料夾

您可以在 Oracle Content Management 伺服器下載與上傳文件和資料夾。

下列命令適用於文件和資料夾下載與上傳：

```
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>
```

若為 `cec-share-folder <name>`，您可以與 Oracle Content Management 伺服器的使用者共用資料夾，並且指定角色給使用者。使用 `-s <server>` 指定伺服器，或使用 `cec.properties` 檔案中指定的伺服器。有效的角色如下：

- manager
- contributor
- downloader
- viewer

若為下載，您可以指定資料夾階層。

## 從樣板建立網站並保持相同的內容 GUID

開發人員可以使用 OCE 工具程式命令從樣板建立 Oracle Content Management 網站，保留相同的內容 GUID。

使用以下 OCE 工具程式命令：

```
update create-site-from-template --reuse-content
```

此命令會在 Oracle Content Management 伺服器中建立一個網站，並且保留建立網站時的內容 ID。保留內容 ID 對於多次執行從測試環境移轉到生產環境的動作是必要的工作，這樣可避免在目標伺服器上產生重複的內容項目。

## 從標準網站建立企業樣板

開發人員可以使用 OCE 工具程式 CLI 命令，從標準網站建立企業樣板。

`create-template` 命令預設會在網站是標準網站時建立標準樣板，網站是企業網站時建立企業樣板。您也可以從標準網站建立企業樣板。

使用新的 `-enterprise` 選項執行 `cec create-template`：

```
cec create-template EnterpriseTempl -s StandardSite1 -e
```

## 匯入及匯出分類

請使用 OCE 工具程式命令將分類從本機機器匯入 Oracle Content Management 伺服器，或從伺服器匯出至本機機器。

`cec download-taxonomy <name>` 命令會從 Oracle Content Management 匯出分類。它會從 Oracle Content Management 伺服器下載分類。

您可以在此命令中使用下列選項：

- `--status, -t [promoted | published] [required]`：指定分類狀態。

- `--id, -i`：若其他分類有相同的名稱，請指定分類 ID。
- `--server, -s`：指定已註冊的 Oracle Content Management 伺服器，或使用在 `cec.properties` 檔案中指定的伺服器。
- `--help, -h`：顯示命令的說明。

以下是一些 `download-taxonomy` 命令的範例：

```
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>` 命令會將分類匯入 Oracle Content Management。它會將分類上傳到 Oracle Content Management 伺服器。

您可以在此命令中使用下列選項：

- `--createnew, -c`：建立一個新分類。
- `--name, -n`：指定新分類的名稱。
- `--abbreviation, -a`：指定新分類的縮寫。
- `--description, -d`：指定新分類的描述。
- `--file, -f`：指示該分類是否來自檔案。
- `--server, -s`：指定已註冊的 Oracle Content Management 伺服器，或使用在 `cec.properties` 檔案中指定的伺服器。
- `--T2P`：檢查目標 Oracle Content Management 執行處理上是否有相同 ID 的資產或分類存在。若為是，則會將資產新增為新版本，並將分類作為草稿 (覆寫現有草稿)。否則，系統會建立相同 ID 的新資產或分類。
- `--New`：一律在目標 Oracle Content Management 執行處理上建立新的資產或分類。
- `--help, -h`：顯示命令的說明。

以下是一些 `upload-taxonomy` 命令的範例：

```
cec upload-taxonomy Taxonomy1
```

Create a new taxonomy or a draft of an existing taxonomy on upload

```
cec upload-taxonomy Taxonomy1 -s UAT
```

Create a new taxonomy or a draft of an existing taxonomy on upload on the registered server UAT

```
cec upload-taxonomy Taxonomy1 -c
```

Create a new taxonomy on upload

```
cec upload-taxonomy Taxonomy1 -c -n Taxonomy1_2 -a t12 -d
```

Create a new taxonomy on upload with the given name, abbreviation

```
"Taxonomy1 copy" and description cec upload-taxonomy
```

```
Create a new taxonomy or a draft of an existing taxonomy in <file-name>.json  
-f and upload the JSON file
```

您可以使用 OCE 工具程式從測試環境移轉到生產環境 CLI 公用程式，從來源 Oracle Content Management 伺服器將資產、內容模型及其相依項一起自動匯入或匯出至目標 Oracle Content Management 伺服器。

管理者或內容管理員可以從測試環境移轉到生產環境中使用 OCE 工具程式命令匯入或匯出分類。有了管理者權限，您可以新增分類的草稿。有了內容管理員權限，您可以建立新的分類。

分類生命週期作業 (例如升級、指定給儲存區域及發布分類)，可供從測試環境移轉到生產環境使用。您可以將資產、分類資訊及分類一起匯入至匯入檔，您也可以從匯出檔將資產、分類資訊和分類一起匯出。

## 匯入及匯出建議

使用 OCE 工具程式命令將建議從 Oracle Content Management 伺服器匯入本機機器，或從本機機器匯出至伺服器。

`cec download-recommendation <name>` 命令會從 Oracle Content Management 匯出建議。此命令會從 Oracle Content Management 伺服器下載建議。

您可以在此命令中使用下列選項：

- `--status, -t [promoted | published] [required]`：指定建議狀態。
- `--id, -i`：如果其他建議有相同名稱，則指定建議 ID。
- `--server, -s`：指定已註冊的 Oracle Content Management 伺服器，或使用在 `cec.properties` 檔案中指定的伺服器。
- `--help, -h`：顯示命令的說明。

以下是一些 `download-recommendation` 命令的範例：

```
cec download-recommendation Recommendation1 -t promoted
```

```
cec download-recommendation Recommendation1 -i  
6A6DE836572C468B90F2A1C17B7CE5E4 -t promoted
```

```
cec download-recommendation recommendation -t published -s UAT
```

`cec download-recommendation <name>` 命令會將建議匯入 Oracle Content Management。此命令會將建議上傳到 Oracle Content Management 伺服器。

您可以在此命令中使用下列選項：

- `--createnew, -c`：建立新建議。
- `--name, -n`：指定新建議的名稱。
- `--abbreviation, -a`：為新建議指定縮寫。
- `--description, -d`：指定新建議的描述。
- `--file, -f`：表示建議是否來自檔案。

- `--server, -s`：指定已註冊的 Oracle Content Management 伺服器，或使用在 `cec.properties` 檔案中指定的伺服器。
- `--help, -h`：顯示命令的說明。

以下是一些 **upload-recommendation** 命令的範例：

```
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

## 新增或移除集合內容

身為開發人員，您可以使用 OCE 工具程式命令，在集合中新增或移除內容。

下列 `cec-control-content` 命令將 **Repo1** 儲存區域中的所有項目新增至 **UAT** 已註冊伺服器的 **Collection1** 集合：

```
cec control-content add -l Collection1 -r Repo1 -s UAT
```

下列 `cec-control-content` 命令會從 **UAT** 已註冊伺服器中移除 **Collection1** 集合的所有項目：

```
cec control-content remove -l Collection -s UAT
```

## 使用 OCE 工具程式開發自訂欄位編輯器

OCE 工具程式提供開發「欄位外觀」類型之元件的支援。開發人員可以建立及管理自訂欄位編輯器。

對於「欄位外觀」類型的元件，您可以執行下列工作：

- 開啟、複製或刪除元件
- 發布或取消發布元件

- 匯出或匯入元件
- 新增或移除元件上的成員
- 檢視特性
- 選擇元件標誌

您可透過「欄位外觀」類型篩選元件清單。

下列 OCE 工具程式命令可用於開發自訂欄位編輯器：

```
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
```

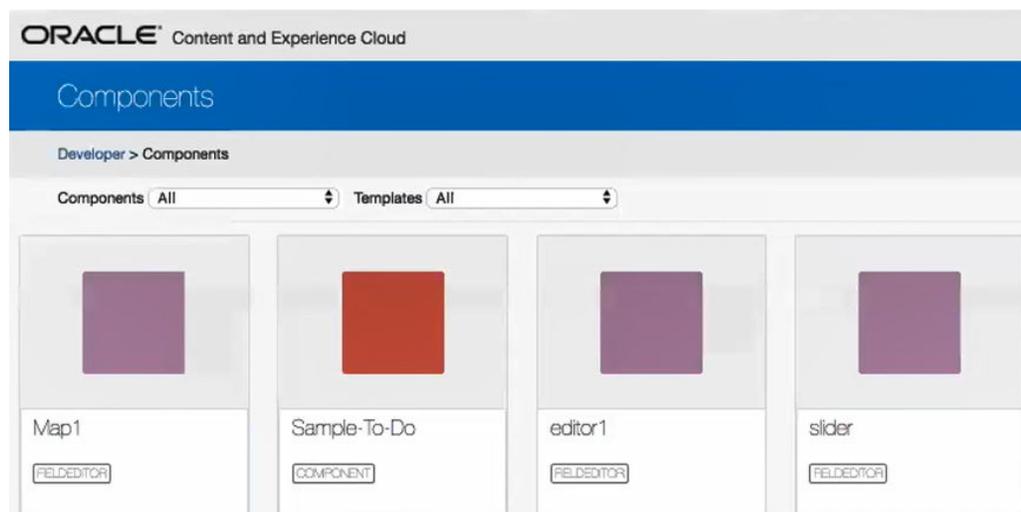
OCE 工具程式包含這些「欄位外觀」元件的範例：

- TextFieldEditor
- SliderFieldEditor
- MapFieldEditor

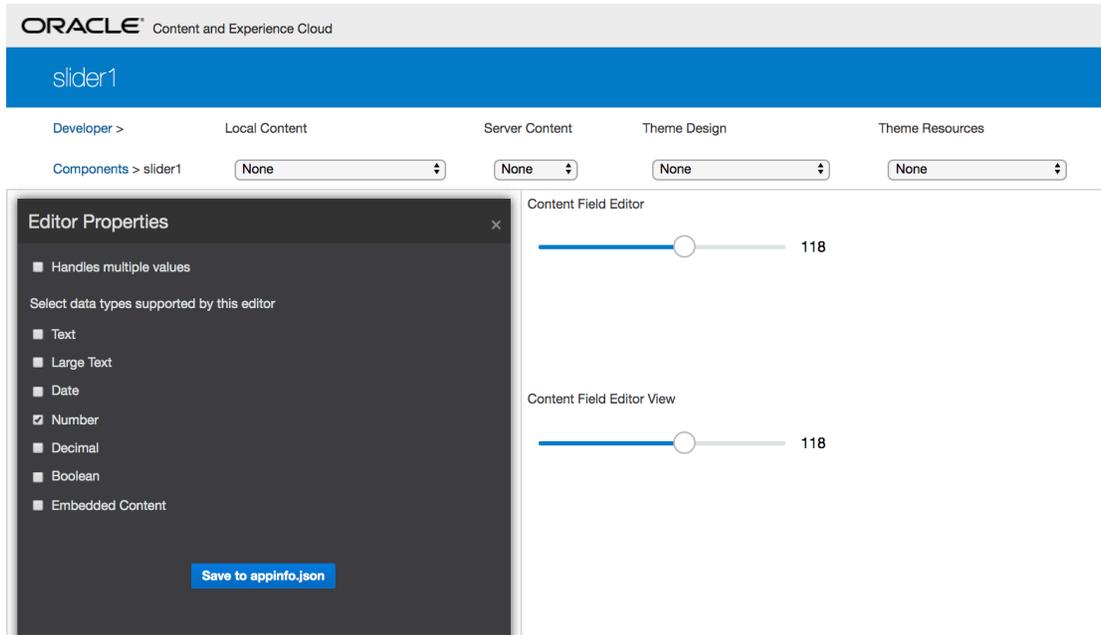
下列影像顯示您可用來開發範例「欄位外觀」元件的 OCE 工具程式命令。

```
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
```

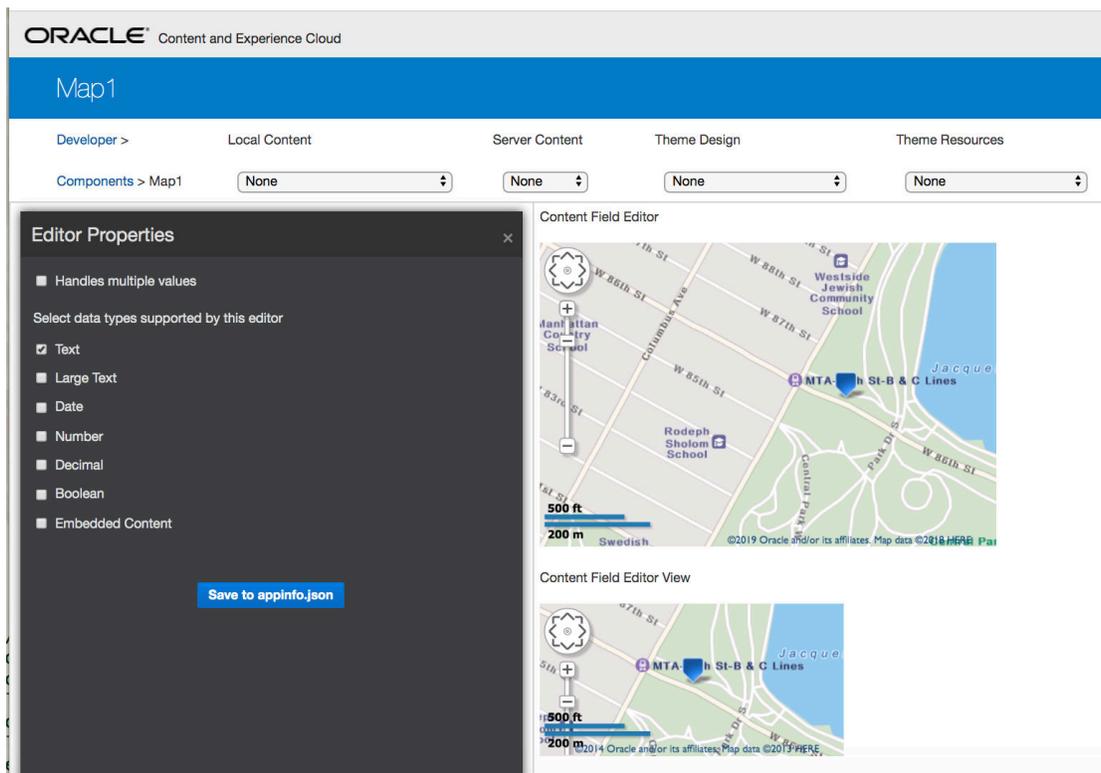
您可以在本機伺服器上建立內建的「欄位外觀」元件、進行測試，並將它們上傳到 Oracle Content Management 執行處理中。下列影像顯示 localhost:8085 中的這些元件。



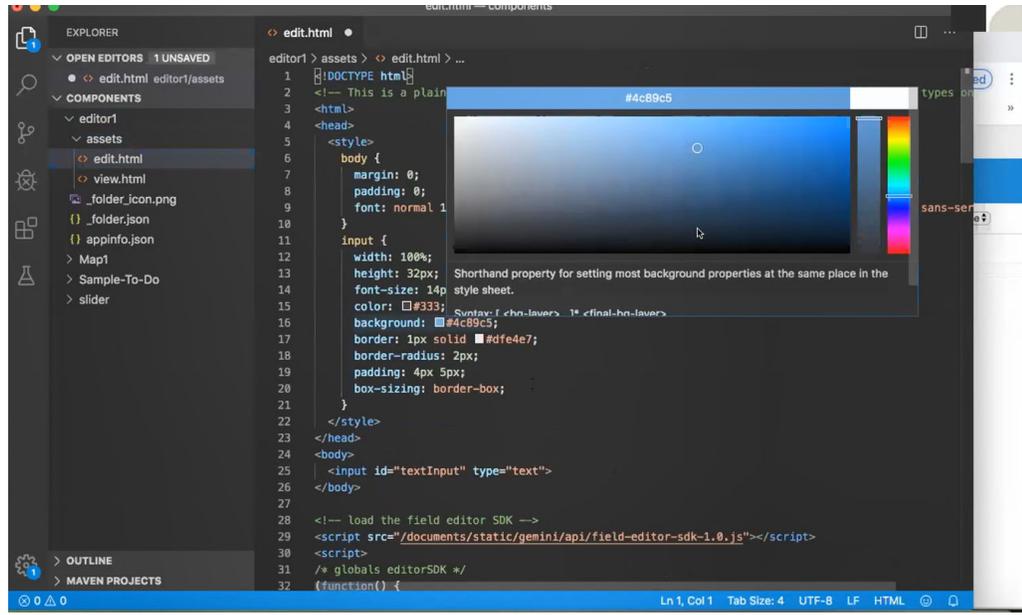
您可以在本機伺服器上測試每個元件，例如滑桿。您可於該處選取元件的特性並將其儲存。



對於地圖元件，您可以點擊地圖來提供位置以作為編輯器值。



您可以編輯元件的 HTML 檔案來變更其設定值，例如背景色彩。



您可以使用 OCE 工具程式命令以建立欄位編輯器與本機內容類型欄位的關聯：

```
cec add-field-editor editor1 -t SimpleContent -n -c SimpleType -f title
```

```
cec add-field-editor slider -t SimpleContent -n -c SimpleType -f value
```

完成自訂欄位編輯器的設定與測試後，您可以使用 OCE 工具程式命令將其上傳至 Oracle Content Management 執行處理：

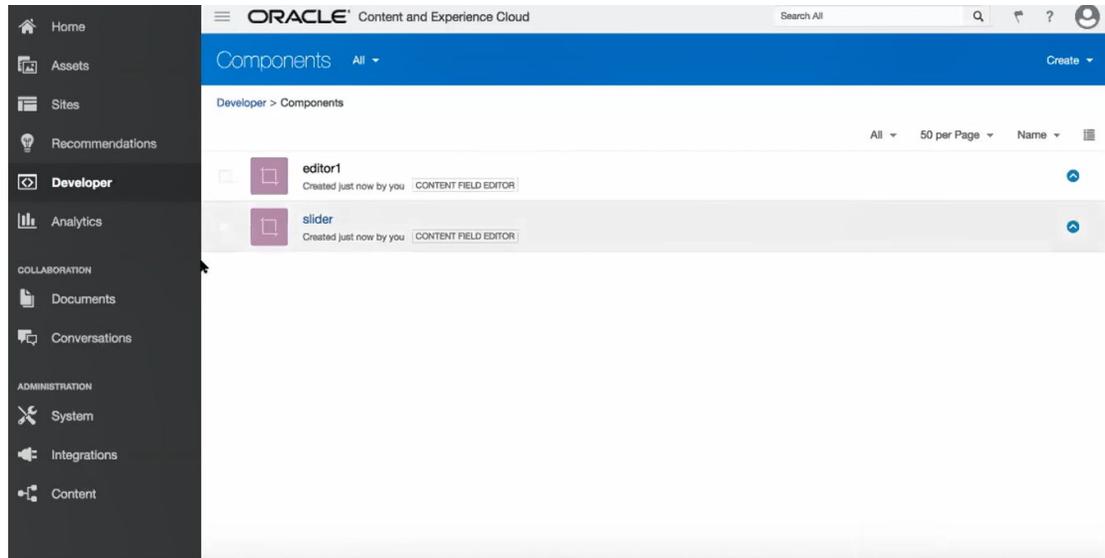
```
# upload editors  
cec ulcp editor1,slider -p -s Latest
```

```
# upload content
```

```
cec cr Repo5 -s
```

```
cec upload-content SimpleContent -r Rpo5 -s
```

上傳自訂欄位編輯器後，您的元件便會匯入 Oracle Content Management 執行處理中。



## 將網站從一個伺服器移轉或更新至另一個伺服器

開發人員可以使用 OCE 工具程式命令，從伺服器 A 建立或更新網站與其內容至伺服器 B。

根據預設，所有資產都會移轉，您也可以選擇性地指定 `-p` 僅移轉發布的資產。使用 `-s <server>` 指定來源伺服器，以及使用 `-d <destination>` 指定目標伺服器。如果網站包含來自其他儲存區域的資產，請選擇性地提供儲存區域對應，否則將不會移轉這些資產。

您可以使用下列命令更新網站或將網站從測試環境移轉至生產環境：

```
cec transfer-site site --from server --to server --repository r
--localization-policy l
```

如需其他選項，請參閱[使用 cec 命令行公用程式](#)。

## 傳輸無內容項目的網站

身為開發人員，您可以使用 OCE 工具程式將無內容項目的網站從一個 Oracle Content Management 伺服器傳輸至另一個伺服器。

若要傳輸包含大量內容資產的網站，您必須區隔網站與其內容。您可以將 `--excludecontent` 選項 (快速鍵 `-x`) 新增至 `transfer-site` 命令。

若已設定此選項，則僅能傳輸網站。

例如：

```
cec transfer-site Site1 -s DEV -d UAT -r Repository1 -l LocalizationPolicy1 -x
```

## 下載或上傳群組中網站的內容項目

若為開發人員，當您從 Oracle Content Management 伺服器傳輸網站至另一個伺服器時，您可以下載或上傳群組中網站的內容項目。

若要傳輸包含大量內容資產的網站，您必須區隔網站與其內容。傳輸不包含 (cec transfer-site --excludecontent) 內容的網站後，請使用 cec transfer-site-content 以傳輸網站的內容。

例如：

```
cec transfer-site-content <name>
```

此命令建立命令檔以將企業網站內容從 Oracle Content Management 伺服器傳輸至另一個伺服器。該命令用於傳輸大量的內容項目，且這些項目為批次傳輸。依照預設，此命令將不會執行命令檔，並會傳輸所有資產。您可指定 -p 來僅傳輸已發布資產。

使用 -s <server> 指定來源伺服器，以及使用 -d <destination> 指定目標伺服器。

### 選項

```
--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]
```

若未設定選項 --execute，cec transfer-site-content 完成後，請執行產生的命令檔 <site name>\_downloadcontent 以從來源伺服器下載內容，並執行 <site name>\_uploadcontent 以將下載的內容上傳到目標伺服器。

### 範例

```
cec transfer-site-content Site1 -s DEV -d UAT -r Repository1 Generate  
script Site1_downloadcontent and Site1_uploadcontent  
cec transfer-site-content Site1 -s DEV -d UAT -r Repository1 -e  
Generate script Site1_downloadcontent and Site1_uploadcontent and  
execute them  
cec transfer-site-content Site1 -s DEV -d UAT -r Repository1 -n 200  
cec transfer-site-content Site1 -s DEV -d UAT -r Repository1 -p
```

## 使用 OCE 工具程式編製網站頁面索引

您可以使用 OCE 工具程式建立網站頁面上文字的內容項目，並且啟用網站的頁面搜尋。

下列小節描述如何使用 OCE 工具程式來編製網站頁面索引：

1. 建立網站頁面文字的內容類型
2. 使用 OCE 工具程式建立頁面索引內容項目
3. 在 Oracle Content Management 中新增網站的內容搜尋

## 建立網站頁面文字的內容類型

若為內容類型，您可指定名稱、必要的欄位值、該類型的預設內容版面配置。

- 類型名稱  
指定任何有效的內容類型名稱。
- 欄位  
需有下列欄位。

欄位名稱	欄位類型	資料欄位值數目	描述
site	文字	單一值	網站名稱
pageid	文字	單一值	頁面 ID
pagename	文字	單一值	頁面名稱
pageurl	文字	單一值	頁面 URL
pagedescription	文字	單一值	頁面描述
keywords	文字	多個 (無上限)	所有頁面上的所有文字，和頁面上內容項目的所有文字欄位值，都是由 OCE 工具程式 index-site 命令取得

```
{#fields}
<div class="indextype"></div>
<div>
  <a href="{{pageFullURL}}" title="{{pagename}}">{{pagename}}</a>
</div>
{/fields}
```

```
content.fields.pageFullURL = SCSRenderAPI.getSitePrefix() +
content.fields.pageurl;
```

- 建立該類型的內容版面配置。  
內容版面配置應顯示網站名稱和 URL 才能瀏覽至頁面。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，產生頁面完整 URL：

```
content.fields.pageFullURL =  
SCSRenderAPI.getSitePrefix() +  
content.fields.pageurl;
```

- 將內容版面配置設為該類型的預設內容版面配置。

```
content.fields.pageFullURL =  
SCSRenderAPI.getSitePrefix() + content.fields.pageurl;
```

## 使用 OCE 工具程式建立頁面索引內容項目

您可以使用 OCE 工具程式命令建立頁面索引內容項目。

先決條件：

- 您的本機機器上已安裝並設定 OCE 工具程式。
- 已發布 Oracle Content Management 上的網站。
- 網站頁面上的內容項目已發布至網站通道。

在命令行介面輸入下列 OCE 工具程式命令：

```
cec index-site site name -c content type name -p
```

命令中的 *site name* 是網站名稱，*content type name* 是為頁面文字建立的內容類型；選項 `-p` 表示建立後發布頁面索引內容項目。

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
```

若要查看使用狀況，您可以輸入 `cec index-site -h`

## 在 Oracle Content Management 中新增網站的內容搜尋

您可以使用搜尋頁面和搜尋欄位在 Oracle Content Management 網站中新增內容搜尋。

若要將內容搜尋新增至網站：

1. [新增搜尋頁面至網站](#)
2. [新增搜尋欄位至主題](#)

### 新增搜尋頁面至網站

您可以新增搜尋頁面至網站，以及新增「內容清單」元件至搜尋頁面。

新增搜尋頁面：

1. 新增頁面至網站，並將其設為搜尋頁面。
2. 新增「內容清單」元件至搜尋頁面。
3. 將**內容類型**設為先前建立的頁面索引內容類型。

### 新增搜尋欄位至主題

若要在網站的每個頁面上顯示搜尋欄位，您可以新增搜尋欄位至主題的版面配置 HTML 頁面。

例如：

```
<div align="center">  
<input id="searchonpage" type="text" size="30" placeholder="Search on  
page. . ."/>  
</div>
```

1. 新增輸入欄位：

```
<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 主體的結尾新增 JavaScript。

## 使用 OCE 工具程式編製多語言網站的索引

您可以使用 OCE 工具程式，針對翻譯、搜尋頁面和內容項目編製多語言網站 (MLS) 的索引。

您可以在發布網站之前，建立多語言網站索引並加以測試。請使用 OCE 工具 `cec index-site` 命令編製多語言網站的索引。前往 `cec-components` 目錄，不使用任何選項來發出此命令，即可檢視此命令的說明資訊：

```
cec-components> cec index-site
Usage: cec index-site <site>
```

```
Create content item for each page with all text on the page. If the
page index content item already exists for a page, updated it with
latest
text on the page. Specify -c <contenttype> to set the page index
content type. Optionally specify -p to publish the page index items
after
creation or update.
```

Options:

```
--contenttype, -c <contenttype> page index content type
--publish, -p      publish page index items
--help, -h        Show
help
```

[boolean]

Examples:

```
cec index-site Site1 -c PageIndex
cec index-site Site1 -c PageIndex -p
```

每個頁面和每種語言都會有頁面索引項目。為每種語言建立的頁面索引內容項目，會以預設語言頁面索引項目的翻譯方式建立。當您在執行中網站進行查詢時，搜尋和內容清單便會從網站 URL 取得語言。這會自動篩選搜尋。

您必須先編製索引並加以翻譯，才能夠發布多語言網站，因此需要翻譯工作。請參閱在 [Oracle Content Management 伺服器中建立新網站或資產翻譯工作](#)。

必須有預設語言英文。對於每種支援的語言 (必要和選擇性)，請執行索引建立作業，並且建立索引項目的翻譯。如果執行兩次索引，則只會進行更新。

使用 OCE 工具程式命令編製索引、翻譯以及發布多語言網站：

1. 建立網站的內容類型，並讓它可在儲存區域中使用。請參閱[建立網站頁面文字的內容類型](#)。
2. 選取驗證原則。
  - a. 按一下左側導覽功能表中的**資產**。
  - b. 在**資產**功能表中選擇**本地化規定**。
  - c. 選取一個本地化規定。
  - d. 必要時修改本地化規定，以包含您想要用於編製索引與翻譯網站的語言。例如，如果規定只有英文，您可以新增法文和西班牙文。  
所有的翻譯都是從英文進行。
3. 下載翻譯工作。您只能翻譯網站中使用的資產。
4. **翻譯**網站。網站翻譯可以手動或透過整合式連線器完成。
5. 翻譯完成後上傳翻譯工作。
6. 使用 `cec index-site` 命令對網站進行編製索引。指定 `-c <contenttype>` 以設定頁面索引內容類型。

您也可以指定 `-p` 選項以發布網站。然後您可以在發布對即時網站的變更之前，驗證編製索引與翻譯。

例如，下列 `cec index-site` 命令會為使用英文、法文和西班牙文的網站建立網站索引。網站所支援的語言是來自指定的本地化規定，包括預設語言。

```
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 Navigtion
- create page index item for Detail Page
- create page index item for Pages
- create page index item for Page Content
- create page index item for Developing Templates
- create page index item for Themes
- add page index items to site channel
- remove page index items for page Search from site channel
- will create/update translate for fx-FR,es-ES
- query site stucture with locale fr-FR
```

- query page data (fr-FR)
- query content on the pages (fr-FR)
- will create 11 page index items (fr-FR)
- will update 0 page index items (fr-FR)
- will remove 1 page index items (fr-FR)
- create page index item for Themes (fr-FR)
- create page index item for Navigation (fr-FR)
- create page index item for Pages (fr-FR)
- create page index item for Detail Page (fr-FR)
- create page index item for Search (fr-FR)
- create page index item for Page Content (fr-FR)
- create page index item for Components (fr-FR)
- create page index item for Developing Templates (fr-FR)
- create page index item for Blog (fr-FR)
- create page index item for Home (fr-FR)
- create page index item for Privacy Policy (fr-FR)
- add page index items to site channel
- set page index items in fr-FR as translated
- remove page index items for page Search from site channel
- query site structure with locale es-ES
- query page data (es-ES)
- query content on the pages (es-ES)
- will create 11 page index items (es-ES)
- will create 0 page index items (es-ES)
- create page index item for Pages (en-ES)
- create page index item for Home (en-ES)
- create page index item for Themes (en-ES)
- create page index item for Components (en-ES)
- create page index item for Privacy Policy (en-ES)
- create page index item for Detail Page (en-ES)
- create page index item for Page Content (en-ES)
- create page index item for Navigation (en-ES)
- create page index item for Developing Templates (en-ES)
- create page index item for Search (en-ES)
- create page index item for Blog (en-ES)
- add page index items to site channel
- set page index items in es-ED as translated
- publish job submitted
- publish in proogress
- publish in progress
- publish page index items finished

7. 發布網站以包含翻譯。

## 建立簡化的元件以便於元件開發

您可以使用 OCE 工具程式建立簡化的元件，以輕鬆進行開發。

SimpleHTML 是 OCE 工具程式中的一個簡化元件，可作為輕鬆開發自訂元件的起點：

```
cec create-component -f SimpleHTML
```

此外，也提供您可快速上手的 JET 元件範例：

```
cec create-component MyComp -f JET-CCA-Demo-Card
```

## 設定網站編譯服務

您可以為編譯伺服器建立 Docker 映像檔，以在 Oracle Content Management 中設定網站編譯服務。這項服務可讓您選擇在發布前先定義要使用的編譯伺服器。

編譯伺服器在 **管理 > 系統 > 網站和資產** 頁面的 **編譯端點 URL** 欄位中有 URL 形式的端點。您可以在該處輸入要向伺服器註冊的完整 URL，然後按一下 **測試** 以驗證此端點。請參閱設定編譯端點 URL。

「網站編譯服務」會編譯網站頁面，因此在存取已發布網站時，會傳回 HTML 頁面。Oracle Content Management 伺服器發布網站時，如果已設定服務，會呼叫「網站編譯服務」來編譯頁面。

指定網站的端點後，您可以在 **網站特性** 對話方塊的 **靜態網站傳遞** 頁籤上啟用自動編譯。在您發布網站時，網站會透過編譯伺服器進行編譯。

網站編譯服務是工具程式命令的擴充功能。您可以自己執行 `cec compilation-server`，但 Docker 映像檔可讓您使用標準模式建立編譯伺服器，然後調整服務的組態。

若要建立 Docker 映像檔並發布 Docker 檔案：

1. 從 GitHub 下載 Docker 映像檔資訊。  
在您從 GitHub 下載資訊時，編譯伺服器底下會有三個額外檔案。其中一個 Docker 映像檔是編譯伺服器，包含一個 Docker 檔案與一個讀我檔案。
2. 建立 Docker 映像檔後會執行 `run.sh` 命令。  
此命令會下載 OCE 工具程式、進行安裝，然後建立來源目錄。
3. 接著，您就可以自訂環境並執行 `run.sh` 命令。您可以指定使用哪個連接埠、逾時項目，以及任何要變更的項目，然後更新並使用 `.sh` 命令。
4. 下載 Docker 映像檔資訊並變更編譯伺服器目錄後，您可以建立 Docker 檔案。  
如果已下載 OCE 工具程式，您可以移除 `no-cache` 選項，這樣就不會重新下載所有項目。
5. 列出 Docker 映像檔。
6. 可以使用 Docker 映像檔後，請註冊該映像檔。
7. 確定映像檔可以運作。  
前往異常狀況檔案 REST API，這可讓您查看支援哪些版本。此測試可讓您驗證檔案確實執行。
8. 您現在必須註冊伺服器。這需要伺服器名稱的第一個部分，其他部分會自動處理。
9. 回到您的系統目錄與網站資產，您可以在其中註冊編譯端點並按一下 **儲存**。
10. 前往網站特性指定要發布的項目與發布時間。

## 編譯網站以提昇網站頁面的程式實際執行效能

在 Oracle Content Management 中編譯網站可提升網站頁面的程式實際執行效能和行為。編譯作業會為網站中的每個頁面建立一個靜態 HTML 檔案 (其行為與原始頁面完全一樣)，以達成此目的。

## 網站編譯總覽

構成網站「頁面」的描述資料檔案會在編譯時合併，以避免通常在程式實際執行時所需的伺服器要求。如此，可以有效地將每頁檢視呈現成本變為單次編譯階段成本。

效能因程式實際執行要求數目減少而獲得改善。這包括避免所有內容查詢，因為您可以將這些查詢的結果直接編譯到頁面中。

行為則因為頁面會立即呈現而獲得改善。因此，可以避免像「無樣式內容閃爍」或元件以不確定的順序在頁面上顯示 (例如頁尾先顯示，然後在頁面上的其他項目隨之呈現後下移) 之類的問題。

若要編譯 Oracle Content Management 網站內的頁面，您需要將網站匯出為樣板，然後使用 OCE 工具程式編譯該樣板，再將產生的靜態頁面上傳到原始網站。下列為可引導您進行樣板編譯，以及如何使用此模型編譯網站的步驟。

## 與預先呈現互動

預先呈現的重點在於傳回適合搜尋引擎的結果。預先呈現所產生的頁面不會如原始頁面般執行，且不會執行任何 JavaScript。它只會傳回瀏覽器預先呈現的 HTML，以支援 SEO 文字搜尋。

透過網站編譯建立的靜態頁面預期會在瀏覽器中執行，並與原始動態網站頁面的行為完全一樣。這包含動態執行未編譯的元件以及執行程式實際執行查詢，其中使用者想要在結果中取得完全最新的資料，而不是在編譯時修正資料。

租用戶管理員啟用預先呈現解決方案時，靜態頁面也可透過預先呈現伺服器傳遞。這讓索引子和蒐集程式可處理靜態頁面中可能仍動態呈現的部分，例如內容清單。

內容查詢會傳回包含 RTF 欄位值的內容項目，因此當內容版面配置用於內容清單時，您無需個別擷取值。

## 呈現控制器網站頁面

`cec compile-template` 命令無需透過 Oracle Content Management 網站頁面控制器，即可讓網站頁面直接呈現在瀏覽器中。

Oracle Content Management 網站並非使用編譯器建立靜態 HTML 頁面，而是使用控制器模型呈現頁面。在頁面實際顯示於瀏覽器之前，這涉及數個取得網站資訊、網站內頁面，以及用於呈現頁面之樣板的要求。



### 備註：

當中還有其他要求，像是取得 `controller.js` 檔案。

## 呈現編譯的網站頁面

`cec compile-template` 命令可讓您將這些步驟全都編譯為靜態 HTML 檔案，藉由減少或免除這些伺服器要求，改進程式實際執行網站的效能。此外，您可以進一步減少無需在程式實際執行時進行變更之資源的要求，並將其編譯為靜態頁面。

預設編譯處理作業會執行現有程式實際執行控制器的工作，為網站中的每個 `page.json` 檔案建立一個實體 HTML 頁面。然後可以將這些頁面部署至網站，將會與網站一起發布，並且在點擊對應的頁面 URL 時使用。

## 樣板編譯

若要編譯網站，您必須能夠存取網站內的主題與元件。網站樣板是網站的封裝模型。樣板包含移轉和執行網站所需的所有資源。

您可以使用網站樣板將網站匯出至「OCE 工具程式」環境，在該環境中可編譯樣板內的網站。以下小節會逐步完成 OCE 工具程式中內建之 **BlogTemplate** 內的網站編譯。

## 設定

透過 OCE 工具程式即可使用 `cec compile-template` 命令，並在工具程式安裝時一併進行安裝。請依照標準指示安裝 OCE 工具程式。

安裝 OCE 工具程式後，您就可以執行 `cec` 命令以建立樣板。

1. 安裝 OCE 工具程式開發目錄：

```
mkdir cec-src
cd ./cec-src
cec install

# Start up the CEC Toolkit Development server for testing
cec develop &
```

2. 建立樣板。  
已使用自訂編譯器更新內建 `BlogTemplate`。從此來源建立樣板：

```
# create a new template (must be in the cec-src folder if not already
there)
cd ./cec-src
cec create-template BlogTemplate -f BlogTemplate
```

內建 **BlogTemplate** 會使用自訂編譯器擴充先前的樣板：

- 內容版面配置編譯器

```
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
```

- 頁面版面配置編譯器

```
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
```

## 編譯您的樣板

編譯您的樣板時，系統會針對樣板內的網站，為每個 `structure.json` 檔案中的頁面建立靜態 HTML 頁面。

編譯步驟結合頁面描述資料 (例如 `100.json`) 與頁面版面配置 (例如 `about.html`)，讓您在程式實際執行時無需控制器，便能立即顯示 HTML 頁面。自訂編譯器可讓您將先前的動態項目 (例如導覽和元件) 編譯至頁面中，進一步減少程式實際執行時 JavaScript 的執行。

`cec compile-template` 命令將會對網站中的每個頁面採取下列動作：

- 在網站和頁面描述資料中讀取
- 在頁面版面配置中讀取，並套用任何頁面版面配置編譯器
- 對於頁面中的每個槽位：
  - 使用 `page.json` 檔案所定義的格線展開槽位
  - 對於槽位內的每個元件：
    - \* 套用任何元件編譯器
    - \* 將產生的元件標記插入槽位內對應的位置中
- 在經過編譯的標記中展開任何巨集，並在程式實際執行時，插入 Oracle Content Management 呈現器所使用的 SCS JavaScript 物件
- 將經過編譯的頁面標記儲存在 `src/templates/<template>/static` 資料夾底下

若要編譯您的網站，請執行：

```
> 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.

## 編譯錯誤

您會在編譯期間得到三個類型的訊息：資訊、警告和錯誤。

1. 資訊訊息與您應該注意的問題有關，但是下列訊息可能為編譯的一部分：
  - 將會在程式實際執行時呈現的預留位置內容項目。
  - 遺漏頁面版面配置編譯器。頁面版面配置可能沒有任何動態元素，因此不會將遺漏版面配置編譯器視為問題。
  - 標示為「存取時呈現」的項目，也就是即使已編譯頁面，網站開發人員也想要在程式實際執行時動態呈現此元件。
2. 警告訊息與可能會影響執行中網站效能的問題有關，應盡可能進行修正。此類別的項目如下：
  - 遺漏內容版面配置或自訂元件編譯器。沒有這些編譯器，元件將會像之前一樣在頁面中以動態方式呈現。
  - 遺漏內容版面配置對應。您不太可能想要使用系統預設內容版面配置來呈現內容項目。
3. 錯誤訊息則與代表編譯失敗的問題有關。這些頁面將會在可能的情況下繼續編譯，但是整體編譯將會結束並發生錯誤。
  - 這很可能是自訂編譯器中的 **JavaScript** 錯誤所造成，必須解決任何錯誤。

 **備註：**

為了減少重複訊息，每次編譯時相同訊息只會出現一次，無論此錯誤是否發生在多個頁面上。

前面的範例有以下編譯結束狀態：

```
Compilation completed with 0 errors and 3 warnings.  
to display warnings, run with --verbose (-v) option.
```

如果使用 "-v" 選項重新執行編譯步驟，將會看到以下輸出：

```
> 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
```

## 詳細資訊頁面編譯

詳細資訊頁面編譯會整理在編譯期間遇到的所有內容。它會為找到的每個內容項目重新編譯詳細資訊頁面，並使用網址代稱值定義新詳細資料頁面的 URL。

前面的編譯輸出包含兩個部分：

1. 頁面編譯
2. 詳細資訊頁面編譯

在前面的範例中，您會看到關於 **Starter-Blog-Author** 頁面沒有內容版面配置對應項目的警告。但是，我們並不需要 **Starter-Blog-Author** 的詳細資訊頁面。詳細資訊頁面僅適用於

**Starter-Blog-Post** 內容項目。若要移除這些錯誤，如果內容項目在其設定值中沒有明確的詳細資訊頁面參照，可以使用以下選項從詳細資訊頁面編譯排除內容項目：

```
--noDefaultDetailPageLink, -o Do not generate compiled detail page for
items/content lists
    that use the default detail page.
```

所以重新執行編譯 (不包括使用預設詳細資訊頁面建立詳細資訊頁面) 會產生以下輸出：

```
> 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
```

## 編譯特定頁面

您不需要在樣板中編譯所有頁面，而是可以選取要編譯哪些頁面。當您只是在處理想要除錯或更新的特定頁面，而不是需要不斷重新編譯整個網站時，這項功能會非常實用。

若要編譯特定頁面，請使用 `--pages (-p)` 選項，且在後面接著想要編譯的頁面清單。

### 備註：

注意：內容項目若是在清單參照詳細資訊頁面中的頁面上，即使未明確包含詳細資訊頁面，還是會一併編譯詳細資訊頁面。

```
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
```

## 網站編譯

您已經從前面的步驟瞭解如何建立並編譯本機樣板的概要。一般而言，您編譯的將是實際的 Oracle Content Management 網站。

若要編譯 Oracle Content Management 網站，您需要先將網站匯出成樣板，然後再編譯該樣板。此樣板套件的內容將包含網站，以及隨樣板匯出之主題和元件中編譯網站所需的所有自訂程式碼。

## 網站編譯先決條件

下列步驟假設您已在 Oracle Content Management 伺服器中建立 **BlogSite** 網站。您可以上傳前面的 **BlogTemplate**，然後從該樣板建立此網站。

例如：

```
> cec upload-template BlogTemplate --sever UAT

> cec create-site BlogSite --template BlogTemplate --repository
<yourRepository> --
```

```
localizationPolicy <yourLocalizationPolicy> --defaultLanguage en-US --
server UAT
```

選取或建立網站後，您可以執行下列小節中的步驟來編譯網站。

## 編譯網站

OCE 工具程式中的 `cec create-template-from-site` 命令可以建立網站的樣板，因此您可以編譯網站。

```
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]
```

若要編譯網站：

1. 從網站建立樣板：

```
> cec create-template-from-site BlogTemplate --site BlogSite --
includeunpublishedassets -
server UAT
```

2. 下載您的樣板：

```
> cec download-template BlogTemplate --server UAT
```

3. 編譯您的樣板：

```
> cec compile-template BlogTemplate --noDefaultDetailPageLink --
verbose --server UAT
--channelToken e1bb88cdc1e025c8dd278f6b676877a3
```

### 備註：

您將必須為網站的發布通道在伺服器中取得網站的通道權杖 (`--channelToken (-c)` 選項)。這將用於樣板中網站內的所有查詢。

4. 上傳經過編譯的網站頁面。  
將經過編譯的靜態檔案複製到網站的靜態資料夾中：

```
> cec upload-static-site-files src/templates/BlogTemplate/static --
site BlogSite --server UAT
```

## 回復成未編譯行為

若要回復成未編譯行為，您必須移除上傳到網站的靜態檔案：

```
> cec delete-static-site-files BlogSite --server UAT
```

如果目前已發布網站，請重新發布網站。這會從網站移除「靜態」資料夾，但不會移除網站。

## 自訂編譯器

可以使用自訂編譯器進一步增強編譯處理作業。您可以呼叫自訂編譯器來編譯頁面中的頁面版面配置、區段版面配置、自訂元件或內容版面配置，而不需要在程式實際執行時動態新增元件。

如果沒有元件的自訂編譯器，或元件編譯器沒有傳回標記，則元件將會在程式實際執行時呈現，就像未曾進行編譯一樣。

下列元件支援自訂編譯器：

- 頁面版面配置
- 區段版面配置
- 自訂元件
- 內容版面配置

下列範例顯示頁面版面配置和內容版面配置編譯器。區段版面配置和自訂元件會依照和內容版面配置編譯器相同的模型。

如果沒有自訂編譯器存在，元件會透過 `render.js` 檔案呈現。

如果有自訂編譯器，系統會呼叫編譯器，然後將產生的 HTML 插入頁面中。自訂編譯也可以指出元件在程式實際執行時需要 JavaScript 填入作業。如果是這種情況，會呼叫 `render.js` 檔案內元件的 `hydrate()` 函數。若元件不需要填入，則不會載入元件的 `render.js` 檔案。

### 備註：

編譯功能僅適用於程式實際執行時 (已發布網站)。若在編輯、瀏覽或預覽模式中檢視網站，頁面會正常運作，而所有元件會一律動態新增至頁面。

## 限制條件

`cec compile-template` 作業是一個在任何瀏覽器之外執行的 NodeJS 應用程式。因為頁面不是在瀏覽器中呈現，所以沒有 DOM 或視窗物件，jQuery、VueJS 或 KnockoutJS 這類從屬端 JavaScript 程式庫也無法運作。

儘管可以使用像 JSDOM 程式庫來建立 DOM 物件並執行這些從屬端程式庫，但這麼做並沒有益處。對於簡單的 HTML 剖析，您可以使用 NodeJS HTML 剖析器 (例如 cheerio)。

標記傳回的 HTML 必須是有效的 HTML。這會透過剖析器傳送，而且只會將經過剖析的 HTML 加到頁面。這是為了確認經過編譯的 HTML 不會有可能會中斷槽位的不相符標記。

## 除錯自訂編譯器

開發自訂編譯器時，您需要對您的程式碼進行除錯。

`cec compile-template` 命令隨附於 `--debug (-d)` 選項，將啟動已設定 `--inspect-brk` 旗標的編譯器，以便除錯程式能夠連附至處理作業。接著您可以依照標準節點進行除錯以檢查您的程式碼。

```
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，您可以使用 `--pages (-p)` 選項來限制將要編譯成特定頁面的頁面。

## 頁面版面配置編譯器

頁面版面配置編譯器為 **NodeJS (CommonJS) JavaScript** 模組，它會編譯對應的頁面版面配置。

特定頁面版面配置的頁面版面配置編譯器，是由副檔名為 `-compile.js` 的名稱關聯所定義。

```
· src  
  - themes  
    * <yourTheme>  
      * layouts  
        * <yourPageLayout>.html  
        * <yourPageLayout>-compile.js
```

若頁面版面配置沒有 `-compile.js` 存在，則不會套用自訂編譯。

頁面版面配置編譯器必須實行會傳回承諾的 `compile()` 介面，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();

```

## 元件編譯器

自訂「元件編譯器」全都遵循與頁面編譯器相同的模型，可針對區段版面配置、自訂元件和內容版面配置加以建立。

在編譯期間，`cec compile-template` 命令會尋找和元件的 `render.js` 檔案相同位置中的 `compile.js` 檔案：

```

· src
  - components
    * <yourComponent>
      * assets
        * render.js
        * compile.js

```

如果此檔案不存在，便不會編譯該元件，並且在程式實際執行時呈現該元件。

如果此檔案存在，則必須實行會傳回承諾的 `compile()` 介面。例如，下列 `Starter-Blog-Author-Summary` 是自訂內容版面配置編譯器：

```
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;
```

## 內容版面配置編譯器

發布指定內容類型的內容項目後，會使用內容版面配置編譯器輸出內容版面配置的 HTML。

在進行編譯時，`cec compile-content` 命令將會在元件的 `render.js` 檔案所在位置中尋找 `compile.js` 檔案：

```
· src
  - components
    * <yourComponent>
      * assets
        * render.js
        * compile.js
```

如果此檔案不存在，便不會編譯版面配置，但將會在程式實際執行時呈現該版面配置。

如果此檔案存在，則必須實行會傳回承諾的 `compile()` 介面。例如，下列是一個內容版面配置編譯器，後面接著產生的 `layout.html` 輸出：

```
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;

```

您可以將產生的 `layout.html` 插入 Web 應用程式，或任何需有內容版面配置靜態 HTML 輸出的位置。`layout.html` 會輸出至 `compile.js` 檔案所在的目錄。

```

{{{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

與 `SCSRenderAPI` 類似，`SCSCompileAPI` 會傳送給每個編譯函數。

這包含下列特性和函數。

- 特性：
  - **navigationRoot**：網站的根節點 ID。
  - **navigationCurr**：目前頁面節點的 ID。
  - **structureMap**：網站階層中所有可由 ID 存取的節點。
  - **siteInfo**：所有網站特性。
- 函數：
  - **getContentClient**：取得要在 Content API 呼叫中使用的 `contentClient` 執行處理。
  - **getCustomSiteProperty**：取得網站編譯期間的自訂特性。

您可以在網站層級的 `siteinfo.json` 檔案中儲存自訂資料和特性，以根據資料和特性自訂網站。例如，您可以變更頁面背景材質、頁尾連結或網站名稱。

## 元件填入

填入指的是當 HTML 在瀏覽器中呈現時，將 JavaScript 行為加回頁面中經過編譯之 HTML 的程序。

例如，假設您的頁面上有兩個要呈現為主要/明細的元件，當按一下主要元件中的某個項目時需要更新明細元件。此動作完全由在頁面中執行的 JavaScript 處理。若要這麼做，您需要在這兩個元件呈現到頁面後填入它們的 HTML，方法為替主要元件中的元素增加 on click 事件處理程式，替明細元件容器增加監聽器，以便於 on click 事件發生時，根據事件中傳送的有效負載重新呈現。

元件編譯器會在頁面中插入 HTML。您的元件在程式實際執行時若需要執行額外的 JavaScript 來加入像事件處理程式之類的項目，則您可以選擇使用內嵌 JavaScript 或 Hydrate 函數。應選擇哪一個解決方案取決於您的需求。

## 內嵌 JavaScript

您可以直接將 `<script>` 標記插入傳回的已編譯標記中。此命令檔將會在頁面執行時執行。

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,"$AutoPlayInterval":3000,"$SlideDuration":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 函數

相較於內嵌 JavaScript，您可以改為在 `render.js` 檔案中包含 `hydrate` 函數，並請注意傳回編譯的標記時，該元件在程式實際執行時需要填入。此填入可避免重複的 `<script>` 標記，讓您利用現有的 JavaScript 程式碼來管理事件。

即使已載入 `render.js`，填入期間也不會呼叫 `render()` 函數。只會呼叫 `hydrate()` 函數。

### 備註：

如果編譯的元件沒有說明需要填入，則不會載入元件的 `render.js` 檔案。

例如，自訂內容版面配置編譯器將會傳回 - `{ 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，必要時自訂編譯器可以新增在程式實際執行時尋找的 *hydrate* 特性。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}}">
```

最後，如果元件標註需要填入，則會在程式實際執行時載入該元件的 `render.js` 檔案並呼叫 `hydrate()` 函數，傳入包含已編譯標記的容器 `<div>` 中。

例如，render.js - 請參閱 hydrate() 函數：

```
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);
    }
}
}
```

## 發布

產生經過編譯的靜態頁面並上傳到網站的靜態資料夾後，您必須發布或重新發布網站，頁面才能變成作用中。同樣地，若要回復成未編譯網站傳遞行為，從網站移除靜態檔案後，您必須發布或重新發布網站。

上傳的靜態頁面可於發布期間進行傳遞。因為這些檔案是在發布處理作業期間複製，發布作業的效能可能會依檔案數目等比例地降低。

發布作業使用目前的靜態檔案集，並讓他們可用來傳遞。這些檔案可能會與動態網站中發生的任何變更同步，也可能會鏡射動態網站。就留給網站開發人員在適當的時間更新靜態檔案集合。

## 靜態網站傳遞優先順序

當網站已經與靜態檔案關聯，會傳遞這些檔案以將相符 URL 傳送至伺服器。如果內送的 URL 與靜態檔案不相符，要求會傳回網站的 `controller.html` 檔案。這是依照網站傳遞的現有動態模型。

Oracle Content Management 網站也可以透過關聯的 JSON 檔案，定義 301 和 302 重新導向。設定重新導向後，重新導向的優先順序會高於靜態檔案。如果 URL 與重新導向規則和靜態檔案相符，將會從伺服器傳遞重新導向。

網站傳遞的 URL 評估依照此流程：

1. URL 是否與設定的重新導向相符？  
如果相符，請發出重新導向回應。
2. URL 是否相對應於靜態檔案？  
網站若設定行動靜態使用者代理程式清單，且要求來自與清單相符的瀏覽器，就會傳遞行動靜態檔案。
3. 否則，將傳遞動態網站 `controller.html` 檔案。

### 備註：

如果行動靜態檔案與網站關聯，且客戶使用 CDN 進行傳遞，則必須設定 CDN (通常是 Akamai) 以從標準桌面要求個別快取行動瀏覽器要求。

如果 CDN 未設定個別行動/標準快取，行動瀏覽器可能會收到標準回應，桌面瀏覽器可能會收到適用於行動瀏覽器的回應。

## 快取標頭

Web 伺服器回應中的 HTTP 標頭有助於判斷瀏覽器如何快取頁面。靜態頁面也會與快取標頭一起傳遞，以協助加快瀏覽器快取。

若為安全的網站，下列標頭將隨回應一起傳送：

- Cache-Control: no-store
- Pragma: no-cache

若為標準的非安全網站，將會傳送下列標頭：

- Cache-Control: max-age=300
- Edge-Control: !no-store,max-age=2592000,downstream-ttl=1800  
Edge-Control 標頭有助於加速 CDN 快取行為

如果您在這二者的其中一個區域中自訂了標頭，回應將包含自訂標頭，而不是此處所列的標準標頭。

您可以在租用戶層級或網站層級控制這些回應。

## 詳細資訊頁面

Oracle Content Management 網站的詳細資訊頁面可在單一頁面中顯示一些內容項目的資訊。

例如，可使用相同的詳細資訊頁面處理一些 URL。這些 URL 會顯示相同的頁面結構，但分別顯示的則是其網址代稱值為 `item1.html`、`item2.html` 以及 `item3.html` 之內容項目的內容。以此情況為例，cec 樣板編譯器可能會建立以下四個檔案：

- `/detail/item1.html`
- `/detail/item2.html`
- `/detail/item3.html`
- `/detail.html`

最後一個檔案的目的為讓新發布的內容無須重新編譯並重新發布網站，即可顯示在網站中。在本範例中，網址代稱值為 `item4.html` 的內容項目會在網站上線後才發布。靜態的 `/detail.html` 頁面可讓新項目以動態方式顯示在網站中。`/detail/item4.html` URL 會傳遞 `detail.html` 頁面，但會顯示 `item4.html` 內容項目的內容。

cec 編譯器會產生顯示內容項目的 `detail.html` 頁面。基於此原因，經過編譯之 `detail.html` 頁面中的相對 URL 將會有額外的父項區段 (`../`)。因此，如果直接參照，`detail.html` 頁面本身將不會正確顯示。所以，不應該參照 `detail.html` 頁面本身或將其新增至頁面導覽。

## 新增內容項目至通道

您可以使用 OCE 工具程式 `control-content` 命令，將內容項目新增至 Oracle Content Management 伺服器中的通道。

`control-content <action>` 命令包含一個動作 - `add`，用於新增內容項目至 Oracle Content Management 通道：

```
cec control-content add -c Channel1 -r Repo1 -s UAT
```

此命令會將儲存區域 `Repo1` 中的所有項目新增至已註冊伺服器 `UAT` 的通道 `Channel1`。

您可以使用 `-s <server>` 指定伺服器，或是使用 `cec.properties` 檔案中指定的伺服器。

`content-usage` 命令的有效動作如下：

- `publish`
- `unpublish`
- `add`
- `remove`

`content-usage` 命令的選項如下：

- `--channel, -c 通道 [必要]`
- `--repository, -r 儲存區域 [<action> 為 add 時為必要]`
- `--server, -s 註冊的 Oracle Content Management 伺服器`

· --help, -h 顯示說明 [布林值]

control-content 命令的範例如下：

```
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 Repol -s UAT
```

Add all items in repository Repol 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

## 編譯行動裝置適用的網站

您可以使用 OCE 工具程式來編譯網站中網頁的行動版面配置。對於相同的內容，行動版面配置可能與桌面頁面版面配置不同。或者，行動版面配置也可以和桌面版面配置相同。

在網站編輯器中，您可以為行動裝置選擇與桌面版面配置相同的頁面版面配置，或是指定不同的頁面版面配置。透過 OCE 工具程式，您可以為行動裝置個別編譯靜態版面配置。

您可以在行動裝置上檢視不同的網站頁面。行動裝置上呈現的頁面，可能不會像桌面版面配置的頁面一樣有橫幅。

在 OCE 工具程式中，cec compile-template 的說明頁面顯示 targetDevice 選項，以在您編譯網站樣板時，將特定裝置設為目標：

```
C:\git\webclient\developer\test\sites-compiler\cec-install>cec compile-template --help
Usage: cec compile-template <source>

Compiles all the pages within the site of the template and places the compiled pages under the sites assets folder.
Optionally specify -s <server> to make content queries against this server (requires channelToken).
Optionally specify -c <channelToken> to use this channelToken when generating any content URLs.
Optionally specify -t <contentType> [draft | published] content to retrieve from the server type, defaults to published.
Optionally specify -p <pages> the set of pages to compile.
Optionally specify -d <debug> to start the compilation with --inspect-brk flag.
Optionally specify -r <recurse> recurse through all child pages of specified pages.
Optionally specify -l <includeLocale> include default locale when creating pages.
Optionally specify -a <targetDevice> [desktop | mobile] target device type when using adaptive layouts.
Optionally specify -v <verbose> to display all warning messages during compilation.

Options:
--server, -s          The registered CEC server
--channelToken, -c   The channel access token to use for content URLs
--type, -t           The type of content to retrieve from the 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>
```

編譯網站時您可以指定要編譯為 desktop 或 mobile 版本。桌面檔案會放在 static/\_files 底下。行動裝置檔案會放在 static/\_mobilefiles 底下。

```
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
```

在您為行動裝置編譯樣板後，OCE 工具程式命令 upload-static-site-files 將支援行動檔案。

## 網站生命週期和編譯過的頁面

從網站建立樣板時，樣板中不會包含編譯過的頁面。這是為了避免網站開發人員預期應是傳遞動態頁面，但傳遞的卻是靜態頁面的問題。

如果您隨後從樣板建立網站，您將需要編譯網站頁面並將其上傳至新網站。

## 在 Oracle Content Management 伺服器中建立新網站或資產翻譯工作

使用 OCE 工具程式可在 Oracle Content Management 中為網站或資產建立翻譯工作。

您必須先有翻譯工作，才能編製多語言網站的索引。若要建立翻譯工作：

1. 按一下網站頁面頂端功能表上的**翻譯**。
2. 在**建立翻譯工作**對話方塊中輸入工作名稱，然後選擇預設來源語言、目標語言以及翻譯工作內容。

您可以選擇讓翻譯套件包含所有網站內容和目標資產、僅限網站內容，或是僅限定位至網站發布通道的資產。

翻譯會排除設定為**不翻譯**文字設定值的任何內容項目。例如，產品名稱通常不翻譯。

3. 按一下**建立**即可建立翻譯工作。
4. 使用 OCE 工具程式命令列出可用的工作：

```
cec components> cec list-translation-jobs
Asset translation jobs:
Name                               Status
Source Language Target Languages Pending
Languages

Site translation jobs:
Name                               Status
Source Language Target Languages Pending
Languages

demo1                               READY      en-
US          fr-FR,es-ES             fr-FR,es-ES
searchdemo1                         TRANSLATED en-
US          fr-FR,es-ES
```

5. 下載您的翻譯工作：

```
cec components> cec download-translation-job demo1
- translation job downloaded to /Users/<user-name>/Dev/webclient/
  developers/sites-toolkit/cec-components/demo.zip
- update the translation job status to INPROGRESS.
cec components> cec translate dmo1.zip -l all -t demo1-xlate.zip
- target languages: fr-FR,ex-ES
- translation finished: /Users/<user-name>/Dev/webclient/
  developers/sites-toolkit/cec-components/demo1-xlate.zip
```

6. 開啟翻譯組合，並為您翻譯的語言建立資源資料夾：

```
Unzip demo1-xlate.zip
ARchive: emol-xlate.zip
```

```
replace assets/job.json? [n]o, [A]ll, [N]one, [r]ename: A
  inflating assets/job.json
  inflating site/job.json
  inflating assets/es-ES/CORE47653001483240C1AAF180C435F189AB-
search_siteSearch202.json
  inflating assets/es-ES/COREA570227E12194356BAA16A80A78A2670-entry1.json
  inflating assets/es-ES/CORED977BC199A3B494596F0D467CAADF7FA-entry2.json
  inflating assets/fr-FR/CORE47653001483240C1AAF18DC435F1B9A8-
search_siteSearch202.json
  inflating assets/fr-FR/COREA570227E12194356BAA16A80A78A2670-entry1.json
  inflating assets/fr-FR/CORED977BC199A3B494596F0D467CA4DF7FA-entry2.json
  inflating assets/root/CORE476530014B3240C1AAF18DC435F1B948-
search_siteSearch202.json
  inflating assets/root/COREA570227E12194356BAA16A80A7842870-entry1.json
  inflating assets/root/CORED977BC199A38494596F0D467CA4DF7FA-entry2.json
  inflating site/es-ES/10.json
  inflating site/es-ES/100.json
  inflating site/es-ES/110.json
  inflating site/es-ES/120.json
  inflating site/es-ES/130.json
  inflating site/es-ES/140.json
  inflating site/es-ES/150.json
  inflating site/es-ES/200.json
  inflating site/es-ES/201.json
  inflating site/es-ES/202.json
  inflating site/es-ES/203.json
  inflating site/es-ES/siteinfo.json
  inflating site/es-ES/structure.json
  inflating site/fr-FR/10.json
  inflating site/fr-FR/100.json
  inflating site/fr-FR/110.json
  inflating site/fr-FR/120.json
  inflating site/fr-FR/130.json
  inflating site/fr-FR/140.json
  inflating site/fr-FR/150.json
  inflating site/fr-FR/200.json
  inflating site/fr-FR/201.json
  inflating site/fr-FR/202.json
  inflating site/fr-FR/203.json
  inflating site/fr-FR/siteinfo.json
  inflating site/fr-FR/structure.json
  inflating site/root/10.json
  inflating site/root/100.json
  inflating site/root/110.json
  inflating site/root/120.json
  inflating site/root/130.json
  inflating site/root/140.json
  inflating site/root/150.json
  inflating site/root/200.json
  inflating site/root/201.json
  inflating site/root/202.json
  inflating site/root/203.json
  inflating site/root/siteinfo.json
  inflating site/root/structure.json
  inflating
```

```

inflating
inflating
inflating
inflating
inflating
inflating
inflating

```

#### 7. 匯入翻譯工作：

```

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

```

## 使用語言服務提供者翻譯網站

您現在可以使用 OCE 工具程式命令行介面和語言服務提供者 (LSP) 處理多種語言的網站翻譯。

預設語言 (例如英文 (美國) (en-US)) 以及網站的一或多個替代語言 (例如德文和法文)，均由網站的本地化規定指定。您可以將網站的文字字串翻譯成指定的替代語言。如果您在翻譯之前變更了網站的語言，文字字串仍將以預設語言顯示。

OCE 工具程式的命令行介面提供下列翻譯選項：

Translation	
cec list-translation-jobs	Lists translation jobs.
[alias: ltj]	[alias: ]
cec create-translation-job <name>	Creates a translation job <name> for a site on CEC server.
[alias: ctj]	[alias: ]
cec download-translation-job <name>	Downloads translation job <name> from CEC server.
[alias: dtj]	[alias: ]
cec submit-translation-job <name>	Submits translation job <name> to translation connection <connection>.
[alias: stj]	[alias: ]
cec ingest-translation-job <name>	Gets translated job <name> from translation connection and ingest.
[alias: itj]	[alias: ]
cec upload-translation-job <name>	Uploads translation job <name> to CEC server.
[alias: utj]	[alias: ]
cec create-translation-connector <name>	Creates translation connector <name>.
[alias: ctc]	[alias: ]
cec start-translation-connector <name>	Starts translation connector <name>.
[alias: stc]	[alias: ]
cec register-translation-connector <name>	Registers a translation connector.
[alias: rtc]	[alias: ]

您可以使用 `cec list-translation-jobs` 命令列出已經在伺服器上的翻譯工作。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
```

輸入任何不含參數或加上 `-h` 的 `cec` 命令，即可提供該命令的說明。請參閱[使用 cec 命令行公用程式](#)。

下列各節提供使用 LSP 翻譯網站的相關資訊：

1. [使用 OCE 工具程式建立翻譯工作](#)
2. [列出翻譯工作](#)
3. [建立翻譯連線器](#)
4. [產生多語言網站的網站地圖](#)
5. [將翻譯工作送至語言服務提供者](#)
6. [將翻譯工作上傳至伺服器](#)

## 使用 OCE 工具程式建立翻譯工作

您可以使用 OCE 工具程式命令在本機系統上建立網站翻譯工作。

若要建立新的網站翻譯工作，請使用 `cec create-translation-job` 命令。此命令會尋找網站的所有資產，並且建立一個包含該網站所有需要翻譯項目的壓縮檔。

```
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
```

若要瞭解翻譯選項，請參閱在 [Oracle Content Management 伺服器中建立新網站或資產翻譯工作](#)。

## 列出翻譯工作

您可以列出伺服器上的翻譯工作，確認已建立您的工作並且可供使用。

```
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
```

請注意，FridayDemo 工作處於 READY 狀態。

## 建立翻譯連線器

語言服務提供者 (LSP) 可協助您翻譯網站。透過 LSP 的翻譯連線器，您即可送出和擷取翻譯工作。

您必須先建立翻譯連線器，才能送出翻譯工作。若不透過 LSP 翻譯網站，您可以建立模擬翻譯連線器來執行。使用 `cec create-translation-connector` 命令建立翻譯連線器，使用 `cec start-translation-connector` 命令加以啟動：

```
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 工具程式在預期的 API 上執行，以測試翻譯連線器：

1. 向 OCE 工具程式註冊連線器。

```
>cec register-translation-connector
```

2. 開啟工具程式，然後前往「翻譯連線」頁面。

```
>http://localhost:8085/public/translationconnections.html
```

3. 逐步執行翻譯連線器驗證頁面中的步驟。這些步驟在連線器環境中使用 /data 資料夾中的 translationBundle.zip 檔案，以驗證連線器。

您可以使用翻譯連線器 SDK 開發 Oracle Content Management 的翻譯連線器。此 SDK 是翻譯連線器 API 的範例 NodeJS 實行。範例接受 Oracle Content Management 翻譯工作壓縮檔、翻譯檔案中所有的資源，然後傳回包含所有翻譯的新壓縮檔。

SDK 要求使用者必須能夠存取 LSP 才能進行實際字串翻譯。模擬 LSP 伺服器包含在 SDK 中，只需在字串前面加上目標地區設定，即可模擬 LSP 的回應。

翻譯連線器 SDK 是由三個主要模組組成。

- **連線器：**實行必要 Oracle Content Management 翻譯連線器 API 的翻譯連線器。
- **工作管理程式：**檔案系統的範例工作管理程式，在語言服務提供者進行翻譯工作時，可維護連線器工作的狀態。
- **提供者：**LSP 所需之特定 API 集的實行，以送出文件進行翻譯並擷取已翻譯的文件。

您可以複製模擬翻譯提供者 JS，並實行當中所有的方法。

## 產生多語言網站的網站地圖

使用 OCE 工具程式產生多語言網站的網站地圖，並將地圖發布至網站。

您可以使用 `cec create-site map <site>` 命令在 Oracle Content Management 伺服器上建立多語言網站的網站地圖。例如：

```
cec create-site-map Site1 -u http://www.example.com/site1
```

此命令會在 Oracle Content Management 伺服器上遍訪網站結構、產生符合網站頁面階層的網站地圖階層，以及在指定的網站 URL 建立網站地圖。

命令選項如下：

```
--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> 選項的有效值如下：

- always
- hourly
- daily
- weekly
- monthly
- yearly
- never

```

· auto
cec create-site-map 命令的範例如下：

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

```

若要發布網站地圖，必須建立網站更新、更新網站地圖，以及確認更新。

## 將翻譯工作送至語言服務提供者

您可以將 OCE 工具程式提供的壓縮檔傳送給「語言服務提供者」，以開始進行翻譯工作。

您可以透過翻譯連線器將翻譯工作送給 LSP。送出需要一些時間，因為連線器需要將檔案解壓縮，再將所有個別的檔案送給 LSP。接著，LSP 再為您的翻譯工作建立專案。等到檔案都匯入專案之後，您就可以開始選取要翻譯的檔案。接著，再由 LSP 開始監督翻譯的狀態。

若要檢查狀態，請使用 `cec list-translation-jobs` 命令且不搭配任何選項，在本機列出翻譯工作。工作的狀態若為 `READY TO INGEST`，您就可以從 LSP 下載一個壓縮檔以擷取翻譯工作。翻譯連線器將您的壓縮檔送至 LSP，LSP 對一系列的檔案進行了翻譯，然後連線器從 LSP 取回檔案並壓縮為可供您下載並擷取的壓縮檔。

```

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

```

擷取壓縮檔會將翻譯工作從連線器提取回您的 OCE 工具程式。

```

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

```

擷取壓縮檔之後，當您在本機列出翻譯工作時，翻譯工作的狀態會是 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
```

您可以將翻譯過的工作上傳至 Oracle Content Management 伺服器。一般來說，工作將會進行初始快速翻譯，然後送回給您進行複查。網站翻譯可能需要幾週的時間才能完成，其中包括擷取 LSP 傳回的翻譯工作、更正翻譯以及重新送出翻譯工作。

## 將翻譯工作上傳至伺服器

擷取翻譯工作之後，您可以將其上傳到 Oracle Content Management 伺服器，然後檢查您網站上的翻譯。

請使用 `cec upload-translation-job` 命令，將您的翻譯壓縮檔上傳至伺服器。

```
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
```

上傳翻譯工作之後，伺服器上工作的狀態會是 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
FridayDemo                           INPROGRESS      en-US           de-
DE,fr-FR
```

若要檢查翻譯，您可以檢查正在翻譯之網站上資產中的文字字串。

# 第 VI 部分

## 附錄

以下是可用的附錄：

- [教學課程：使用 Knockout 開發元件](#)
- [Sites SDK 參考資料](#)
- [疑難排解](#)

## 教學課程：使用 Knockout 開發元件

此教學課程將引導您使用一組運用標準 Knockout ViewModel 和樣板功能的 JavaScript 物件，建立儲存在 Oracle Content Management 元件目錄中的元件。

- [透過 Knockout 開發元件的簡介和先決條件](#)
- [步驟 1：建立元件](#)
- [步驟 2：複查本機元件呈現的結構](#)
- [步驟 3：複查本機元件設定值的結構](#)
- [步驟 4：在元件中顯示新特性](#)
- [步驟 5：註冊觸發程式](#)
- [步驟 6：引發觸發程式](#)
- [步驟 7：註冊動作](#)
- [步驟 8：執行動作](#)
- [步驟 9：為元件的每個執行處理建立不同的標題](#)
- [步驟 10：使用具有內嵌編輯的巢狀元件](#)
- [步驟 11：支援不同的版面配置](#)
- [步驟 12：定義自訂樣式](#)
- [步驟 13：在內嵌框架中呈現元件](#)
- [步驟 14：在內嵌框架中呈現元件時使用自訂樣式](#)
- [步驟 15：與頁面還原和重做行為的整合](#)
- [步驟 16：資產管理](#)
- [教學課程複習](#)

### 透過 Knockout 開發元件的簡介和先決條件

本教學課程呈現使用 JavaScript 物件建立範例元件的步驟和驗證程序，這是利用標準 Knockout JS ViewModel 和樣板功能。

您應該能夠採用這些步驟中所參照的程式碼 (於建立元件時內建檔案中提供)，並透過自己的程式碼僅更新 .html 樣板和 JavaScript viewModel。

 **備註：**

儘管 Oracle Content Management 並未控管用來建立元件的 JavaScript 技術，但是無論選擇的 JavaScript 架構為何，通常元件中每個實行的處理站 JavaScript 函數都會是相同的。

### 先決條件

本教學課程僅介紹元件的實行。若需要更多元件的一般資訊，請參閱開發元件。

若要在本教學課程中完成步驟，您必須符合下列要求：

- 您必須能夠存取 Oracle Content Management 執行處理，並具備建立網站和元件的權限。
- Oracle Content Management 執行處理伺服器已使用 Oracle Content Management 桌面應用程式或自訂元件同步到本機電腦。請參閱使用 Developer Cloud Service 開發自訂元件。

此外，您應該熟悉這些 JavaScript 概念和架構：

- JavaScript 瀏覽器除錯
- JavaScript 閉包
- JavaScript 非同步模組定義 (AMD) 開發
- RequireJS 和 KnockoutJS 架構

繼續進行 [步驟 1：建立元件](#)。

## 步驟 1：建立元件

此步驟說明如何在 Oracle Content Management 中建立自訂元件。

建立自訂元件時，必須註冊該元件才能供 Oracle Content Management 使用。為了將元件的相關資訊告知 Oracle Content Management，您必須使用「網站產生器」的「元件」頁面註冊該元件。

有兩種類型的元件需要註冊。

- **本機元件：**
  - 這類元件的檔案儲存在 Oracle Content Management 執行處理伺服器上。
  - 主要優點是檔案位於您的網站上，因此您無需擔心跨網域或跨協定問題。
  - 缺點是您無法在 Oracle Content Management 伺服器中執行任何中間層邏輯，所以必須使用 REST API 連線至支援 CORS 的遠端伺服器。
  - 此類型的元件可以直接內嵌至頁面中，也可以選擇使用內嵌框架在頁面上呈現元件。
- **遠端元件：**
  - 這類元件的檔案儲存在遠端伺服器上，只需在元件的「呈現和設定值」面板註冊 URL。
  - 遠端元件的優點是具有建立元件內容時必須執行的伺服器端邏輯。

- 缺點是您必須確定已解決跨網域和安全問題才能存取這些 URL。
- 遠端元件一律使用內嵌框架在頁面上呈現。

#### 建立並註冊本機元件：

1. 在 Oracle Content Management 首頁上，按一下**開發人員**。  
便會顯示**開發人員**頁面。
2. 按一下**檢視所有元件**。
3. 從功能表中選擇**建立本機元件**。
4. 輸入元件的名稱，例如 **A\_Local\_Component**。
5. 輸入選擇性的描述。
6. 按一下**建立**。

完成後，您將會在元件清單中看到名為 A\_Local\_Component 的元件。

#### 檢查步驟 1 的結果

現在，您已順利建立一個元件，應該會在您所建立任何網站的「元件」選盤中看到此元件。使用以下步驟驗證您的元件建立：

1. 建立一個 localComponentTest 網站。
2. 選取該網站並按一下**開啟**。
3. 按一下**編輯**。
4. 為此網站建立一個更新，並指定其名稱及描述 (選擇性)。
5. 選取網站上的一個頁面。
6. 按一下側邊選盤中的 ，然後選取**自訂**以顯示自訂元件清單。
7. 從自訂元件清單選取 A\_Local\_Component，並將其拖放至頁面上。

現在，您應該會看到建立的本機元件以預設的方式呈現。

8. 選取您剛拖放至頁面上之元件其橫幅中的 .
9. 選取**設定值**。
10. 變更對齊方式並設定元件的樣式。
11. 關閉「設定值」面板。

下列步驟說明如何建立自訂元件，以及如何針對您的用途進行修改。繼續進行[步驟 2：複查本機元件的結構](#)。

## 步驟 2：複查本機元件呈現的結構

我們將在此步驟中複查為本機元件建立之預設檔案的結構。

以簡單的 Hello World 例子來說，四個 JavaScript 物件和數行的程式碼看起來似乎太多，但這是為了要幫助您瞭解建置更為複雜元件的基礎，以及如何處理與 Oracle Cloud Sites Service 頁面生命週期的互動。

複查本機元件的結構：

1. 在 Oracle Content Management 首頁上，按一下**開發人員**。  
便會顯示**開發人員**頁面。
2. 按一下**檢視所有元件**。
3. 從功能表中選擇**建立本機元件**。
4. 輸入元件的名稱，例如 **A\_Local\_Component**。
5. 輸入選擇性的描述。
6. 按一下**建立**。

完成後，您將會在元件清單中看到名為 A\_Local\_Component 的元件。

1. 使用 Oracle Content Management 桌面同步從屬端找出元件，並將其與檔案系統同步。

如果沒有桌面從屬端，您可以在 Oracle Content Management 介面的「元件」頁面中檢視所有元件並從中選取該元件，然後向下展開以查看檔案。

2. 如果列出元件底下的檔案，您將會見到以下這些檔案：

```
assets
  render.js
  settings.html
appinfo.json
_folder_icon.jpg
```

3. 開啟 /assets 目錄下的 render.js 檔案。

render.js 檔案的重點為：

- 它的結構為 JavaScript AMD 模組，因此可以成為頁面中的「必要項目」。
- 他同時包含已載入為 Oracle Content Management 頁面之一部分的 KnockoutJS 和 JQuery 參照。

讓我們看一下 render.js 檔案的結構。

render.js 檔案的內容中有兩個實行必要 Oracle Content Management 元件 API 的 JavaScript 物件：sampleComponentFactory 和 SampleComponentImpl。這些物件都是建立任何 KnockoutJS 式元件的實行範例。這些元件的實行將會根據所使用的技術而改變。

- sampleComponentFactory
  - 此元件由 render.js AMD 模組傳回。
  - 這是一個非常簡單的「處理站」元件，實行單一 createComponent() 介面。
  - 更為複雜的實行可使用傳送的 args 值，根據 viewMode 參數傳回不同的元件實行。與「網站產生器」相比，這樣的方式讓元件在程式實際執行時的實行更為輕巧。
- SampleComponentImpl
  - 此物件內的主要函數為 render 函數，主要用於在頁面上呈現元件。  
若要將 Knockout 元件呈現到頁面中，render 函數會動態將樣板加到頁面，然後將 viewModel 連結套用至樣板。

- 實行的其餘部分則是處理 viewModel 參數和樣板的初始化，以及頁面與元件之間的訊息傳遞。

render.js 檔案中的最後兩個物件為 sampleComponentTemplate 和 SampleComponentViewModel，主要提供自訂的元件實行。這些物件的實行將會依據需求而有所不同。

- sampleComponentTemplate
  - 此物件提供 KnockoutJS 樣板建立。它會等到元件的所有資料都起始後，再嘗試顯示內容。
- SampleComponentViewModel
  - viewModel 會代替元件擷取 Oracle Content Management 儲存的資訊，然後根據該資料選取如何適當地排列元件。
  - 樣板使用一般 Knockout 可監測項目處理元件儲存之描述資料的存取：

```
self.imageWidth = ko.observable('200px');
self.alignImage = ko.observable();
self.layout = ko.observable();
self.showTopLayout = ko.observable();
self.showStoryLayout = ko.observable();
```

- 觸發程式與動作整合：

**觸發程式**：可從元件引發 Oracle Content Management 觸發程式的函數，可連結至頁面上其他元件的動作。

```
self.imageClicked = function (data, event) {
  self.raiseTrigger("imageClicked"); // matches appinfo.json
};
```

**動作**：當元件被指示使用指定的有效負載執行動作時，可處理回呼的函數。

```
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);
      }
    });
  }
};
```

**回呼**可視需要執行任何註冊的動作。

```
SitesSDK.subscribe(SitesSDK.MESSAGE_TYPES.EXECUTE_ACTION,
$.proxy(self.executeActionsListener, self));
```

- 元件生命週期訂閱：
  - \* 元件初始化：確定元件在所有資料都擷取後才呈現。這透過 Knockout 可監測項目處理。

```
self.componentLayoutInitialized = ko.observable(false);  
self.customSettingsDataInitialized = ko.observable(false);
```

取得所有必要特性的最初值。這由回呼處理，以擷取資料。

```
SitesSDK.getProperty('componentLayout',  
self.updateComponentLayout);  
SitesSDK.getProperty('customSettingsData',  
self.updateCustomSettingsData);
```

- \* 描述資料更新：每當元件儲存的元件描述資料變更時即進行回呼；例如，當使用者呼叫「設定值」面板並更新資料時。

```
SitesSDK.subscribe(SitesSDK.MESSAGE_TYPES.SETTINGS_UPDATED,  
$.proxy(self.updateSettings, self));
```

#### 備註：

由於 Oracle Content Management 伺服器一律會設定 .html 檔案的 mime 類型，因此您無法上傳 .html 檔案並使用必要的 "text!" Plug-in 將其載入。因此對於樣板，您可能需要運用 "text!" Plug-in 以使用不同的副檔名來載入樣板，或者以內建資料方式直接將樣板內嵌於 JavaScript 中。

### 檢查步驟 2 的結果

您現在應該大致瞭解自訂元件呈現器的結構是如何建立。接下來，需要驗證它是否可以運作：

1. 更新 render.js 檔案中的 sampleComponentTemplate 物件，以變更以下內容。請將以下程式碼：

```
'<!-- ko if: initialized -->'+
```

改為使用以下程式碼：

```
'<!-- ko if: initialized -->'+  
'<div data-bind="text:\'image width is: \'' + imageWidth()" data-bbox="250 746 840 775"/>
```

2. 將元件同步或上傳至 Oracle Content Management 執行處理伺服器。

3. 編輯網站內的頁面，將 A\_Local\_Component 自訂元件放入頁面中。

此時，您在元件中應該會見到 image width is: 260px。

4. 開啟「設定值」面板並按一下 **自訂設定值** 按鈕。

5. 將 **影像寬度** 欄位變更為 300px。

6. 此時，元件中將會有以下兩項改變：
  - a. 預設影像的大小將從 260px 擴增至 300px。
  - b. 您新增的文字將更新為 `image width is 300px`。

繼續進行[步驟 3：複查本機元件設定值的結構](#)。

## 步驟 3：複查本機元件設定值的結構

在此步驟中，我們會複查指定用於本機元件的設定值結構。

與 `/assets` 目錄中的 `render.js` 檔案類似，相同目錄中有預先建立的 `settings.html` 檔案。`settings.html` 檔案會呈現元件的任何自訂設定值資料。預設的實行中，自訂設定值資料內有單一特性 `imageWidth`。

複查本機元件的結構：

1. 使用 Oracle Content Management 桌面同步從屬端找出元件，並將其與檔案系統同步。  
如果沒有桌面同步從屬端，您可以在 Oracle Content Management Web 介面的[元件頁籤](#)上選取該元件，然後向下展開來查看檔案。
2. 如果列出元件底下的檔案，您將會看到這些檔案：

```
assets
  render.js
  settings.html
  appinfo.json
  _folder_icon.jpg
```

開啟 `/assets` 目錄底下的 `settings.html` 檔案，然後複查該內容。與 `render.js` 檔案不同，`settings.html` 檔案在「網站產生器」的「設定值」面板中使用內嵌框架，這也是為何該檔案需要存取支援檔案才能在內嵌框架中正確呈現的原因。您需要「網站產生器」來管理您的網站，以便將 JavaScript 程式碼中的任何錯誤與「網站產生器」隔離，這也是為何 `settings.html` 檔案要使用內嵌框架的原因。

這些是 `settings.html` 檔案的主要區域：

- 呈現「設定值」面板的 Knockout 樣板。

```
<!-- 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 -->
```

- 自訂連結處理程式，以便在「設定值」面板呈現後，調整內嵌框架的高度。

```
ko.bindingHandlers.scsCompComponentImpl
```

- 套用至 Knockout 樣板的 Knockout ViewModel。

```
SettingsViewModel
```

這些是 SettingsViewModel 的主要元素：

- 元件生命週期的訂閱。
- 元件初始化：
  - 請確定擷取所有資料後，才會呈現該元件。這透過 Knockout 可監測項目處理。

```
self.initialized = ko.observable(false);
```

- 請確定準備之後，才會嘗試更新該資料。

```
self.saveData = false;
```

- 取得所有必要特性的最初值。這由回呼處理，以擷取資料。

```
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;  
});
```

- 將任何特性變更儲存至自訂設定值資料。

```
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);
```

若要新增其他想要擷取的特性，需要幾個步驟：

1. 更新使用者介面以顯示新值。
2. 將值起始為元件目前儲存的值。
3. 將值的所有變更儲存回元件。

若要將其他特性新增至自訂元件，請對 `settings.html` 檔案進行這些變更：

1. 新增其他可監測項目以處理新特性。請將以下程式碼：

```
self.width = ko.observable();
```

改為使用以下程式碼：

```
self.width = ko.observable();
self.imageBannerText = ko.observable();
```

2. 第一次顯示設定值面板時，請取得新特性的目前所有值。請將以下程式碼：

```
self.width(data.width);
```

改為使用以下程式碼：

```
self.width(data.width);
self.imageBannerText(data.imageBannerText);
```

3. 將任何變更儲存至此新特性。請將以下程式碼：

```
'width': isNaN(self.width()) ? self.width() : self.width() + 'px'
```

改為使用以下程式碼：

```
'width': isNaN(self.width()) ? self.width() : self.width() + 'px',
'imageBannerText': self.imageBannerText()
```

4. 新增使用者介面以顯示新的欄位。請將以下程式碼：

```
<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">
```

改為使用以下程式碼：

```
<label id="widthLabel" for="width" class="settings-heading" data-
bind="text: 'Image Width'"></label>
<input id="width" data-bind="value: width" placeholder="example: 200px or
33%" class="settings-text-box">
```

```
<label id="imageBannerTextLabel" for="imageBannerText" class="settings-
heading" data-bind="text: 'Image Banner'"></label>
<input id="imageBannerText" data-bind="value: imageBannerText"
placeholder="Text to display above an image" class="settings-text-box">
```

5. 同步或上傳 `settings.html` 檔案。

若要現在執行，則會顯示欄位。但是，「設定值」面板的大小不會自動變更。因為您已增加了面板的大小，您必須將 `components.json` 註冊項目更新為新的大小。

1. 下載 `appinfo.json` 檔案，該檔案與您元件的 `assets/` 目錄屬於同一層級，然後更新設定值面板的大小。請將以下程式碼：

```
"settingsHeight": 90,
```

改為使用以下程式碼：

```
"settingsHeight": 160,
```

2. 同步或上傳 `appinfo.json` 檔案。

### 檢查步驟 3 的結果

您現在應能查看並輸入您新增至「設定值」面板的新特性。

1. 重新整理網站中的頁面，讓網站產生器能夠取得元件的變更。
2. 將頁面切換為「編輯」模式
3. 將您的元件拖放到頁面中。
4. 顯示您元件的「設定值」面板。
5. 按一下 **自訂設定值** 按鈕。

您將會看到 `settings.html` 檔案中，每個您擁有的特性所顯示的兩個欄位。

繼續進行 [步驟 4：在元件中顯示新特性](#)。

## 步驟 4：在元件中顯示新特性

本節結束時，您將可以在「設定值」面板中輸入新特性的值，並查看自訂元件變更以反映新值。特性的更新也將自動儲存在頁面。

您必須在 `render.js` 檔中更新元件中的兩個 JavaScript 物件：

- `SampleComponentViewModel`
- `sampleComponentTemplate`

編輯 `render.js` 並更新 `SampleComponentViewModel` 元件以包含新特性。請將此特性：

```
self.showStoryLayout = ko.observable();
```

改為使用以下特性：

```
self.showStoryLayout = ko.observable();  
self.imageBannerText = ko.observable();
```

更新 `SampleComponentViewModel` 以取得值中的任何變更。請將此特性：

```
self.imageWidth(customData && customData.width);
```

改為使用以下特性：

```
self.imageWidth(customData && customData.width);  
self.imageBannerText(customData && customData.imageBannerText);
```

變更 `sampleComponentTemplate` 以顯示新特性。請將此特性：

```
'<div data-bind="text: \'image width is: \' + imageWidth()"'></div>' +
```

改為使用以下特性：

```
'<div data-bind="text: imageBannerText"'></div>' +
```

將元件同步或上傳至 **Oracle Content Management** 伺服器。

您已更改元件以顯示新特性。因為元件不像「設定值」面板是內嵌在頁面上的內嵌框架，它是直接插入頁面中，所以隨著元件的大小增加，其可用的區域將自動增加。

#### 檢查步驟 4 的結果

若要查看顯示的新特性：

1. 重新整理網站中的頁面，讓網站產生器能夠取得元件的變更。
2. 將頁面切換為「編輯」模式
3. 將您的元件拖放到頁面中。
4. 顯示您元件的「設定值」面板。
5. 按一下 **自訂設定值** 按鈕。
6. 將 **Image Banner** 變更為 **Workspace**。

您將會看到頁面上的元件更新成 **Worksapce** 出現在影像上方。

繼續進行 [步驟 5：註冊觸發程式](#)。

## 步驟 5：註冊觸發程式

您將在此步驟中複習 **Oracle Content Management** 觸發程式的註冊方式，亦即可以使用您元件之「設定值」面板中「連結」頁籤內的「觸發程式動作」選項選取。

觸發程式是 **Oracle Content Management** 元件間通訊的一部分。任何元件都可引發任意數目的觸發程式。元件可以為觸發程式提供有效負載，然後有效負載會被傳送給引發該觸發程式時所執行的任何動作。使用者可以為每個觸發程式選取應該執行的動作。最後，建立成一起運作的元件，可以在無須使用者定義元件之間的互動情況下，自動引發觸發程式，對其他元件執行動作。

就您新增的元件而言，觸發程式會註冊成為元件之註冊資料的一部分。若要新增觸發程式，請更新元件支援之各個觸發程式的 "triggers" 特性陣列。此外，還必須指定觸發程式支援的有效負載，如此才能建立使用者介面，讓使用者可以將有效負載內的值對應至動作所支援的特性。

開啟 `appinfo.json` 檔案並複查 `"triggers":[]`, 項目。

```
"triggers": [{
  "triggerName": "imageClicked",
  "triggerDescription": "Image clicked",
  "triggerPayload": [{
    "name": "payloadData",
    "displayName": "Trigger Payload Data"
  }]
}],
```

在此項目中，您會見到下列各項：

- `triggerName` ("imageClicked")，這應為唯一值，通常會以您的自訂元件 ID 命名。
- `triggerDescription` ("Image clicked")，這由使用者介面對話方塊使用以顯示您的觸發程式。
- 您觸發程式的單一值 `triggerPayload` ("payloadData")。使用者將可以選取此有效負載中的項目，並將其對應至此動作中的欄位。

### 檢查步驟 5 的結果

您可以前往元件之「設定值」面板中的**連結**頁籤查看並選取觸發程式：

1. 重新整理網站中的頁面，讓網站產生器能夠取得元件的變更。
2. 將頁面切換為「編輯」模式
3. 將您的元件拖放到頁面中。
4. 顯示您元件的「設定值」面板。
5. 選取「設定值」面板頂端的**連結**頁籤。
6. 按一下**觸發程式動作**作為「連結類型」。
7. 按一下您註冊的**已點選影像**觸發程式。
8. 在對話方塊中，從「頁面動作」區段內拖曳**顯示警示**動作。(頁面動作都是 Oracle Content Management 所提供的內建動作。)
9. 在**訊息**欄位中，選取**觸發程式有效負載資料**值，此為註冊觸發程式時，您在有效負載中所看到的項目名稱。您可以將此名稱變更為任何您要的名稱。

現在您可以註冊觸發程式，並將觸發程式對應至透過值傳遞的內建動作。在下一個步驟中，我們將會複習引發觸發程式以執行動作的方式。

繼續進行**步驟 6：引發觸發程式**。

## 步驟 6：引發觸發程式

在這個步驟中，將會告訴您如何引發看到的已註冊觸發程式。

元件可在任意點引發觸發程式。觸發程式通常由使用者互動，透過點選按鈕或選取表格中的資料列引發。不過其實元件可根據任意條件引發觸發程式，例如因 REST 呼叫而造成資料變更時。

在此範例中，當您按一下影像後將會透過 `whoAreYou` 特性目前的值引發觸發程式。

複查 `render.js` 檔案並查看 `SampleComponentViewModel` 物件。

引發觸發程式：

1. 複查 `SampleComponentViewModel` 物件中呼叫 **Sites SDK** 以引發觸發程式的函數。

```
self.raiseTrigger = function (triggerName) {
  SitesSDK.publish(SitesSDK.MESSAGE_TYPES.TRIGGER_ACTIONS, {
    'triggerName': triggerName,
    'triggerPayload': {
      'payloadData': 'some data here'
    }
  });
};
```

2. 現在您需要在使用者介面中呼叫該函數來引發觸發程式。複查 `render.js` 檔案並更新 `sampleComponentTemplate` 物件以包含以下項目：

```
'<div data-bind="attr: {style: imageStyle, \'data-layout\':
alignImage()}, click: imageClicked">' +
```

在 `SampleComponentViewModel` 物件中，您會看到按一下影像時所呼叫的 **JavaScript** 函數。此函數會呼叫 **Sites SDK** 並指示它觸發為觸發程式 `imageClicked` 定義的所有動作，此為從步驟 2 中的 `click` 連結傳入的值。也會透過包含單一欄位 `payloadData` 的 `triggerPayload` 傳送，及透過靜態值 `'some data here'` 傳送。`imageClicked` 和 `whoAreYou` 與註冊觸發程式的 `appinfo.json` 檔案 (上一個步驟中) 中的這些值相符。

在範例程式碼中，觸發程式會透過 `click` 連結的 `data-bind` 引發，並傳入觸發程式名稱 `imageClicked`。根據使用者選擇的版面配置，目前有 3 種 `<scs-image>` 元件的呈現。為了確保會針對每個版面配置引發觸發程式，請編輯 `render.js` 檔案進行以下變更。

- 從不同版面配置引發觸發程式。找到此程式碼的兩個項目：

```
'<div data-bind="attr: {style: imageStyle, \'data-layout\':
alignImage()}">' +
```

將程式碼變更為：

```
'<div data-bind="attr: {style: imageStyle, \'data-layout\':
alignImage()}, click: imageClicked">' +
```

- 指定要傳送至觸發程式的有效負載。請將以下程式碼：

```
self.raiseTrigger = function (triggerName) {
  SitesSDK.publish(SitesSDK.MESSAGE_TYPES.TRIGGER_ACTIONS, {
    'triggerName': triggerName,
    'triggerPayload': {
      'payloadData': 'some data here'
    }
  });
};
```

改為使用以下程式碼：

```
self.raiseTrigger = function (triggerName) {
  SitesSDK.publish(SitesSDK.MESSAGE_TYPES.TRIGGER_ACTIONS, {
    'triggerName': triggerName,
    'triggerPayload': {
      'payloadData': self.imageBannerText() // pass banner text
    }
  });
};
```

- 將 `render.js` 檔案同步或上傳至 Oracle Content Management 執行處理伺服器。

現在您已複查所需的程式碼，可以連結此觸發程式，讓您的自訂元件可以在按一下按鈕時引發。

### 檢查步驟 6 的結果

您現在應該可以註冊要對觸發程式執行的動作，而且在引發觸發程式時也能執行該動作：

1. 重新整理網站中的頁面，讓網站產生器能夠取得元件的變更。
2. 將頁面切換為「編輯」模式
3. 將您的元件拖放到頁面中。
4. 顯示您元件的「設定值」面板。
5. 選取「設定值」面板頂端的**連結**頁籤。
6. 選取**觸發程式動作**作為「連結類型」。
7. 按一下您看到的已註冊 **imageClicked** 觸發程式。
8. 在對話方塊中，從**頁面動作**區段拖曳**顯示警示**動作。
9. 在**訊息**欄位中選取 **payloadData** 值，這是您在註冊觸發程式時輸入的有效負載。
10. 關閉「設定值」面板，並將網站產生器切換為「預覽」模式。
11. 按一下元件中的影像。  
便會出現警示，顯示 no message defined，因為您尚未指定 imageBannerText 值。
12. 將頁面切換為「編輯」模式，再次啟動該元件的「設定值」面板。
13. 按一下**自訂設定值**並輸入 workplace。
14. 關閉「設定值」面板，並將頁面切換為「預覽」模式。
15. 按一下元件中的影像。

現在應該會顯示更新的有效負載 workplace，這是從您對 click 連結所做的變更呼叫。

您可以在觸發程式引發後執行任意數目的動作。

 **備註：**

沒有預先定義的動作執行順序。雖然動作會依所列示的順序呼叫，但並不會等待動作完成後才呼叫下一個動作。如果某個動作進行非同步呼叫，在下一個動作執行前它也許還沒完成。

繼續進行**步驟 7：註冊動作**。

## 步驟 7：註冊動作

Oracle Content Management 觸發程式引發之後，便會在元件上呼叫動作。

元件可以註冊任意數目的動作，也可以定義動作所支援的有效負載。當使用者選取動作時，可以填入要傳送至動作的有效負載。

和註冊觸發程式一樣，您可以在 `appinfo.json` 檔案註冊資料中註冊元件所支援的動作。若要複查元件中範例動作的註冊，請開啟 `appinfo.json` 檔案並尋找程式碼 "actions"。

```
"actions": [{
  "actionName": "setImageWidth",
  "actionDescription": "Update the image width",
  "actionPayload": [{
    "name": "imageWidth",
    "description": "Image Width in pixels",
    "type": {
      "ojComponent": {
        "component": "ojInputText"
      }
    },
    "value": ""
  }
}]
}]
```

在元件「設定值」面板的**連結**頁籤中按一下觸發程式後，便會在呼叫的「動作」對話方塊中見到此註冊的動作。

### 檢查步驟 7 的結果

1. 重新整理網站中的頁面，讓網站產生器能夠取得元件的變更。
2. 將頁面切換為「編輯」模式
3. 將您的元件拖放到頁面中。
4. 將「按鈕」元件放置在頁面上。
5. 開啟「按鈕」元件的「設定值」面板。
6. 在「一般」頁籤上，將按鈕的標籤變更為**請按我**。
7. 選取「設定值」面板上的**連結**頁籤。
8. 選取**觸發程式動作**作為**連結類型**。
9. 按一下「按鈕」元件的**按一下按鈕**觸發程式。

10. 在對話方塊中，展開左側選盤中的 `A_Local_Component` 元件。
11. 將 `A_Local_Component` 元件的**更新影像寬度**動作拖放至頁面上。
12. 在**影像寬度 (像素)**欄位中輸入 **300px**。

現在您已瞭解如何註冊動作，以及該動作將如何在使用者介面中顯示。在下一個步驟中，您將會學習在呼叫動作時如何操控元件中的動作。

繼續進行[步驟 8：執行動作](#)。

## 步驟 8：執行動作

本主題結束時，您將瞭解如何在頁面上置放將於元件內執行動作的元件。這利用到您在先前步驟中建立的動作註冊。

要執行動作的元件必須監聽 `EXECUTE_ACTION` 訊息。此訊息也包含傳送至將擷取預期值之動作的有效負載。

若要監聽 `EXECUTE_ACTION` 訊息，請編輯 `render.js` 檔案，以下列項目更新 `SampleComponentViewModel` 物件：

```
SitesSDK.subscribe('EXECUTE_ACTION', $.proxy(self.executeActionsListener, self));
```

收到 `EXECUTE_ACTION` 訊息後，便會執行關聯的回呼函數：

```
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);
      }
    });
  }
}
```

這會建立一個執行動作的 **JavaScript** 函數，然後在每次引發 `EXECUTE_ACTION` 訊息時使用 **Sites SDK** 呼叫此函數。

每當 `EXECUTE_ACTION` 訊息發出後，便會呼叫此元件，而且元件只會處理它能夠處理的動作。因此，您必須檢查動作的名稱，確定它是元件可以處理的動作。

動作的有效負載是一個值陣列。您通常需要從陣列中找尋在意的有效負載值。

 **備註：**

由於動作監聽器是一種回呼，因此您應使用 JavaScript 閉包或適當地連結函數，以確保當執行函數後能夠存取 `viewModel`。

**檢查步驟 8 的結果**

1. 重新整理網站中的頁面，讓網站產生器能夠取得元件的變更。
2. 將頁面切換為「編輯」模式。
3. 將您的元件拖放到頁面中。
4. 將「按鈕」元件拖放到頁面中。
5. 開啟「按鈕」元件的「設定值」面板。
6. 在「一般」頁籤中，將按鈕的標籤變更為**請按我**。
7. 選取「設定值」面板頂端的**連結**頁籤。
8. 選取**觸發程式動作**作為「連結類型」。
9. 按一下「按鈕」元件的**按一下按鈕**觸發程式。
10. 在對話方塊中，展開左側的 `A_Local_Component` 元件。
11. 將**更新影像寬度**動作從 `A_Local_Component` 元件拖放到右側。
12. 在**影像寬度 (像素)**欄位中輸入 `300px`。
13. 將頁面切換為「預覽」模式。
14. 按一下**請按我**按鈕。

此時，您影像的大小將會擴增為 `300px`。

 **備註：**

觸發程式和動作的目的都是用來支援元件間的通訊。他們的用途不是建立或管理狀態。如果您重新整理頁面，頁面將會回復成其原始狀態，就像未曾引發觸發程式或執行動作一樣。

繼續進行**步驟 9：為元件的每個執行處理建立不同的標題**。

## 步驟 9：為元件的每個執行處理建立不同的標題

此步驟說明如何針對元件的不同執行處理建立不同的標題。

將元件放置在頁面上時，您會注意到元件的橫幅為：`A_Local_Component`。當使用者只有在頁面上放置其中一個元件時倒無所謂，但是您可能想要建立不同的標題，讓使用者能夠區分您元件的不同執行處理。

您可以使用 Sites SDK 更新元件的標題。在此步驟中，您將會根據 `"imageBannerText"` 特性進行更新。

若要更新標題，請編輯 `render.js` 檔案並將以下程式碼新增至您的 `SampleComponentViewModel` 物件：

```
self.updateDescription = ko.computed(function () {  
    SitesSDK.setProperty('description', self.imageBannerText());  
});
```

每當 `imageBannerText` 可監測項目變更時，此 Knockout 運算就會更新元件的描述。

### 檢查步驟 9 的結果

1. 重新整理網站中的頁面，讓網站產生器能夠取得元件的變更。
2. 將頁面切換為「編輯」模式
3. 將您的元件放置在頁面上。
4. 顯示您元件的「設定值」面板。
5. 按一下 **自訂設定值** 按鈕。
6. 將 **Image Banner** 變更為 **Workplace**。
7. 關閉「設定值」面板，並將游標停駐在元件上方以顯示橫幅。

現在應該會看到顯示 `A_Local_Component Workplace`。

繼續進行 [步驟 10：使用具有內嵌編輯的巢狀元件](#)。

## 步驟 10：使用具有內嵌編輯的巢狀元件

Oracle Content Management 元件使用 KnockoutJS 元件架構實行。這表示如果您使用 KnockoutJS 實行元件，就能將 Oracle Content Management 內建元件直接包含在樣板中。

### 備註：

因為 Oracle Content Management 內建元件僅能在 Oracle Content Management 頁面中執行，所以如果您的元件在內嵌框架中呈現，則無法使用巢狀元件。

若要利用巢狀元件：

1. 使用 KnockoutJS 實行您的元件。
2. 使用 RequireJS 包含您的元件，並使用 Oracle Content Management 建立的相同 Knockout "ko" 執行處理變數。

這是必要項目，因為 Oracle Content Management 會擴充具有元件的 Knockout，如果您使用自己的 KnockoutJS 執行處理，就無法使用這些元件。

您將在此步驟中複查 Oracle Content Management 「影像」、「段落」及「標題」元件在自訂元件中呈現的方式。使用者將可以在頁面內直接編輯，並存取巢狀元件的「設定值」面板。

若要查看如何將這些元件包含在您的樣板中，請編輯 `render.js` 檔案並查看 `sampleComponentTemplate` 物件。所呈現的預設區段如下所示：

```
'<!-- 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>` 巢狀物件時，您將會看到下列項目：

```
'<scs-image params="{ scsComponent: { \'renderMode\': mode, \'parentId\':
id, \'id\': \'imageId\', \'data\': imageData } }"></scs-image>' +
```

傳送至 `params` 樣板連結的 `scsComponent` 資料包含下列項目：

- `renderMode`：指的是「網站產生器」使用的模式。您可以透過它啟用和停用功能。例如，在 `<scs-title>` 元件使用此項目時，它會在執行 `edit` 模式時新增 **RTF** 編輯器。
- `parentId`：此為必要項目，以便 **Oracle Content Management** 元件知道其呈現為巢狀元件。所有巢狀元件的變更將儲存在自訂元件的資料中。
- `id`：巢狀元件的唯一 ID。這會以自訂元件的 ID 命名。
- `data`：巢狀元件的初始資料。如果未進一步修改元件，則會呈現此初始資料。

參照的 `id` 和 `mode` 值會傳入 `SampleComponentViewModel` 物件中的自訂元件，因此您不需要修改物件就能取得這些值：

```
// Store the args
self.mode = args.viewMode;
self.id = args.id;
```

所有其他支援的巢狀元件語法依照與 `<scs-paragraph>` 相同的樣式；例如：`<scs-image>`、`<scs-title>`、`<scs-button>`。

### 檢查步驟 10 的結果

1. 重新整理網站中的頁面，讓網站產生器能夠取得元件的變更。
2. 將頁面切換為「編輯」模式
3. 將您的元件拖放到頁面中。

4. 按一下元件的 *As a page author, you can edit. . .* 文字，並使用 RTF 編輯器更新描述。
5. 切換為「預覽」模式以查看您的更新。
6. 切換回「編輯」模式。
7. 顯示您元件的「設定值」面板。
8. 按一下現在出現的**元件**連結，因為它找到您的巢狀元件。
9. 按一下**段落**，這是它找到的巢狀元件。

您現在可以針對元件中的「段落」元件更新特性。

#### 備註：

Oracle Content Management 在元件建立後才會知道樣板中可能存在的任何巢狀元件。若要讓 Oracle Content Management 瞭解隱藏的巢狀元件，您可以使用 `SitesSDK.setProperty('visibleNestedComponents', []);` API。若要依照預設顯示隱藏的巢狀元件，您必須更新元件註冊中的 `"nestedComponents": []` 陣列。

繼續進行步驟 11：支援不同的版面配置。

## 步驟 11：支援不同的版面配置

在此步驟中，我們將會複查允許使用者更改元件顯示方式的版面配置。

自訂元件可以支援允許使用者選擇之任意數目的版面配置。這些版面配置都會更改自訂元件的顯示方式。版面配置為註冊資料的其他擴充功能。

若要複查範例程式碼中支援的三種版面配置，請在 `appinfo.json` 檔案中複查 `"componentLayouts"` 項目

```
"componentLayouts": [  
  {  
    "name": "default",  
    "displayName": "IMAGE_LEFT_LAYOUT"  
  },  
  {  
    "name": "right",  
    "displayName": "IMAGE_RIGHT_LAYOUT"  
  },  
  {  
    "name": "top",  
    "displayName": "IMAGE_TOP_LAYOUT"  
  }  
],
```

如果您顯示自訂元件的「設定值」面板，將會看到用來切換版面配置的選項。若要啟用元件以因應選取的變更，`render.js` 檔案中的程式碼可以取得目前選取的值並監聽此值的變更。

編輯 `render.js` 檔案並檢視 `SampleComponentViewModel` 物件。

- 其中有在樣板中參照的 `layout` 可監測項目：

```
self.layout = ko.observable();
```

- 其中有每當此值變更時，可以處理的更新函數：

```
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);
```

- 初始化程式碼會取得版面配置的原始值，並呼叫更新函數：

```
SitesSDK.getProperty('componentLayout', self.updateComponentLayout);
```

特性變更監聽器會檢查此特性的任何變更，並呼叫更新函數：

```
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));
```

最後，`sampleComponentTemplate` 樣板物件的程式碼可反映此值的變更：

```
'<!-- ko if: alignImage() === \'right\' -->' +
```

這些變更可一起讓您在「設定值」面板中選取版面配置，並更新元件。

### 檢查步驟 11 的結果

1. 重新整理網站中的頁面，讓網站產生器能夠取得元件的變更。
2. 將頁面切換為「編輯」模式
3. 將您的元件拖放到頁面中。
4. 顯示您元件的「設定值」面板。
5. 從「版面配置」特性選取 **影像靠右方**。

這時元件將更新以顯示 "`<scs-image>`" 元件。

繼續進行 [步驟 12：定義自訂樣式](#)。

## 步驟 12：定義自訂樣式

您所建立的元件會如用於網站之主題的 `design.json` 和 `design.css` 檔案中的任何其他元件等同視之。

若要為自訂元件新增自己的樣式，請確認您註冊元件時使用的 `id` 值。在 `appinfo.json` 檔案中；此為 `"id": "hello-world"`。

使用該值編輯主題的 `design.json` 檔案，並新增該 `id` 所要支援的新樣式。例如，編輯主題中的 `/designs/default/design.json` 檔案，並新增以下程式碼：

```
"hello-world": {
  "styles": [{
    "name": "Plain",
    "class": "hello-world-default-style"
  },
  {
    "name": "Gothic",
    "class": "hello-world-gothic-style"
  }
]
```

開啟元件的「設定值」面板時，應該會在「樣式」頁籤中看到列出兩個選項：**Plain (預設)** 和 **Gothic**。不過在 `design.css` 檔案中實際定義列出的樣式類別之前，切換這些選項並不會有任何效果。

編輯主題的 `design.css` 檔案，並新增您樣式的階層式樣式表 (CSS) 類別。例如，編輯主題中的 `/designs/default/design.css` 檔案，並新增以下程式碼：

```
.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; }
```

儲存您的檔案並將其同步至 Oracle Content Management 執行處理伺服器。

### 檢查步驟 12 的結果

1. 重新整理網站中的頁面，讓網站產生器能夠取得元件的變更。
2. 將頁面切換為「編輯」模式
3. 將您的元件拖放到頁面中。
4. 顯示您元件的「設定值」面板。
5. 前往「樣式」頁籤。

6. 切換您 `design.json` 檔案中定義的 **Gothic** 和 **Plain** 樣式。

您會注意到元件切換每次選取所套用的 CSS 類別時，元件中的字型大小會有所調整以反映變更。

繼續進行 [步驟 13：在內嵌框架中呈現元件](#)。

## 步驟 13：在內嵌框架中呈現元件

到目前為止，範例已在頁面中顯示內嵌呈現的本機元件。您也可以選擇在內嵌框架中呈現元件。

例如，若您的元件對頁面進行非全面的更新，每當特性變更需要重新建立頁面時，您可以選擇在內嵌框架中呈現元件。此外，遠端元件則一律是在內嵌框架中呈現。

本節中的範例，來自於您建立本機元件時選擇**建立可在 iframe 中呈現的元件**選項時，系統為您建立的檔案。然而，您可以將檔案集代管在遠端伺服器上，讓他們可以平均套用至遠端元件。

### 內嵌框架和非內嵌框架元件之間的相似性

#### 設定值面板

因為「設定值」面板一律會置於內嵌框架的頁面中，因此不論元件是否使用內嵌框架，「設定值」面板的程式碼將不會變更。您將為這兩個使用案例建立相同的「設定值」面板程式碼。

#### Sites SDK API

這兩個使用案例的 SDK API 相同。您將會使用相同的程式碼來引發觸發程式、監聽動作以及取得並設定特性值。雖然某些特性可能無法這兩個案例皆適用 (例如，您無法對未使用內嵌框架的元件設定 "height" 特性)，但 API 不會變更。因此，您可以複製這兩種類型元件之間的程式碼，本教學課程中討論的範例程式碼將會適用於這兩個案例。

### 內嵌框架和非內嵌框架元件之間的差異

#### 檔案結構與相依項

當您在建立本機元件時選取**建立可在 iframe 中呈現的元件**，您將會看到下列為您建立的檔案：

```
<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
```

建立這些檔案是為了讓您在頁面上的內嵌框架中立即執行元件。此結構與標準本機元件結構的主要差異是：

- JavaScript 相依項：

- 您正在取得這些檔案的完整複本，以便能執行元件。需要這些檔案，才能執行範例內嵌框架元件。您可以視需要新增和移除此目錄的內容。
- 因為元件發布時，元件之 `assets` 目錄底下的所有項目都會發布至公用網站，所以在「網站產生器」中和程式實際執行時都可以使用 `js` 目錄中的所有項目。
- 注意：建立這些檔案是為了易於使用。您應在主題或其他公用位置中合併這些檔案，而不是為每個內嵌框架元件個別建立這些檔案的版本。

· `render.html`：

- 此為完整 HTML 文件，而非標準元件的 `render.js` 檔案，其為 AMD 模組。

### 元件「高度」管理

使用內嵌框架的其中一個問題是內嵌框架本身的高度管理。如果設定錯誤，您將會在頁面上看到可能不是您想要的元件捲軸。

為了管理內嵌框架的高度，元件必須告知頁面想要的內嵌框架高度。透過遠端元件，您可以處理跨網域的問題，因此當元件於頁面上呈現之後，您必須使用 **Sites SDK** 訊息傳遞來要求頁面將內嵌框架設為所需的高度。這可以透過 `SitesSDK.setProperty('height', {value})` API 來完成。(請參閱 **Oracle Content and Experience SDK**。)

例如，建立 `setHeight` 函數和自訂連結處理程式，以便元件於頁面上呈現時進行呼叫。

· 更新高度函數：

```
// 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');
};
```

· 每當元件於頁面上呈現或特性變更時，**Knockout** 自訂連結處理程式可呼叫 `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();
    }
};
```

- 更新樣板以呼叫連結處理程式：

```
<div data-bind="sampleAppSetAppHeight: true"></div>
```

### 觸發程式和動作註冊

雖然不在內嵌框架中之元件的觸發程式/動作註冊是位於 `appinfo.json` 檔案中，對內嵌框架元件而言，元件本身會負責提供此資訊。這可以使用以下兩個 API 來完成：

```
SitesSDK.subscribe('GET_ACTIONS', self.getAppActions);  
SitesSDK.subscribe('GET_TRIGGERS', self.getAppTriggers);
```

以下為使用這些 API 的範例。

```
// 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;  
};
```

### 存取主題樣式

因為元件是在內嵌框架中呈現，所以其無法存取主題中可用的樣式。Sites SDK 提供可擷取這些樣式的 API，讓這些樣式可以套用至內嵌框架中的元素。

[步驟 14：在內嵌框架中呈現元件時使用自訂樣式](#)中會進一步探索此主題。

## 混合 HTTPS 和 HTTP 協定

因為 Oracle Content Management 使用 HTTPS 協定，所有頁面內參照的資源也必須使用 HTTPS。資源包含將在內嵌框架中呈現的基本 .html 檔案，以及其參照的所有檔案。

此資源要求主要適用於遠端元件，但是您需注意此限制。使用內嵌框架的本機元件資源由 Oracle Content Management 伺服器提供，因此這些元件已使用對應的協定。

繼續進行 [步驟 14：在內嵌框架中呈現元件時使用自訂樣式](#)。

# 步驟 14：在內嵌框架中呈現元件時使用自訂樣式

在內嵌框架中呈現的元件無法直接存取 design.css 檔案。而是在元件中有一個額外步驟來取得 design.css 的 URL，然後將其新增至頁面中。您必須更新元件以反映使用者選取的樣式。

若要在元件中包含並使用 design.css 檔案，需要在 render.html 檔案中進行變更：

1. 找出並包含 design.css 檔案的 URL
2. 每當所選樣式類別變更時便取得其值
3. 更新樣板以反映選取的 styleClass
4. 在元件中反映對所選樣式類別的變更
5. 確定內嵌框架會在樣式變更時調整大小

以下是編輯 render.html 檔案的詳細指示：

1. 找出並包含 design.css 檔案的 URL。

以動態方式將 design.css 檔案新增至頁面的 <head> 區段。載入之後，請設定內嵌框架的高度，因為高度可能已透過套用樣式進行更改。

將下列程式碼新增至 viewModel 物件中：

```
// 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();
                    }
                }
            }
        };
    styleSheetDeferred.poll(pollFunction, interval);
};
```

```

        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. 每當所選樣式類別變更時便取得其值。

建立可監測項目以追蹤 styleClass 特性的值何時變更：

```
self.selectedStyleClass = ko.observable();
```

請注意，擁有樣式類別後才能夠呈現。請將以下程式碼：

```
self.customSettingsDataInitialized = ko.observable(false);
self.initialized = ko.computed(function () {
    return self.customSettingsDataInitialized();
}, self);
```

改為使用以下程式碼：

```
self.customSettingsDataInitialized = ko.observable(false);
self.styleClassInitialized = ko.observable(false);
self.initialized = ko.computed(function () {
```

```

    return self.customSettingsDataInitialized() &&
self.styleClassInitialized();
}, self);

```

透過新增下列程式碼取得所選樣式類別的最初值：

```

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. 更新樣板以反映 styleClass。請將以下程式碼：

```
<p data-bind="attr: {id: 'titleId'}, text: titleText"></p>
```

改為使用以下程式碼：

```
<p data-bind="attr: {id: 'titleId'}, text: titleText, css:
selectedStyleClass"></p>
```

4. 在元件中反映對所選樣式類別的變更。請將以下程式碼：

```

if (settings.property === 'customSettingsData') {
    self.updateCustomSettingsData(settings.value);
}

```

改為使用以下程式碼：

```

if (settings.property === 'customSettingsData') {
    self.updateCustomSettingsData(settings.value);
}
if (settings.property === 'styleClass') {
    self.updateStyleClass(settings.value);
}

```

5. 確定內嵌框架會在樣式變更時調整大小。請將以下程式碼：

```

// 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();

```

改為使用以下程式碼：

```

// 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. 儲存您的檔案並將其同步至 Oracle Content Management 執行處理伺服器。

#### 檢查步驟 14 的結果

1. 重新整理網站中的頁面，讓網站產生器能夠取得元件的變更。
2. 將頁面切換為「編輯」模式
3. 將您的元件拖放到頁面中。
4. 顯示您元件的「設定值」面板。
5. 前往「樣式」頁籤。
6. 切換您 design.json 檔案中定義的 **Gothic** 和 **Plain** 樣式。

您會注意到元件切換每次選取所套用的 CSS 類別時，元件中的字型大小會有所調整以反映變更。

繼續進行步驟 15：與頁面還原和重做行為的整合。

## 步驟 15：與頁面還原和重做行為的整合

由於 Oracle Content Management 會儲存自訂元件的特性，因此這些特性的變更都會自動歸屬於頁面的**還原與重做**行為。

為了確保能清楚的知道使用者在按**還原**或**重做**時發生了什麼情況，這些「還原事件」只應在使用者實際對頁面執行動作時才發生。例如，在使用者實際對特性做變更之前，顯示自訂元件「設定值」面板不應該更新頁面內的特性。只在「設定值」面板中起始特性不應導致更新事件。

如果不注意確保此行為，可能會發生非預期的行為。頁面還是會執行，但會損害使用者體驗。例如，可能會生以下行為：

- 只要顯示「設定值」面板，**儲存**按鈕即變為作用中。
- 使用者必須按**還原**數次，才能見到效果。
- Redo 堆疊被移除，因為元件寫回非預期的變更並以新值更新 Redo 堆疊。

此「設定值」面板的教學課程所提供的範例程式碼，提供您準備好實際呼叫 saveData (而非於初始化) 時，只進行寫回的範例。除非與使用者互動有關，否則元件本身也要注意不要更新 customSettingsData，但這通常不是什麼大問題。

繼續進行步驟 16：資產管理。

## 步驟 16：資產管理

此步驟描述並說明如何管理元件所使用的資產。

資產包括 Oracle Content Management 已知並能管理資產生命週期的元件與自訂元件。

## Oracle Content Management 內容資料夾

您在 Oracle Content Management 建立的每個網站都隨附於自己的 content 資料夾。您通常看不到這個隱藏資料夾。當網站發布時，所有 content 資料夾中的檔案也會發布至檔案系統中。

例如，當您使用「影像」元件選取影像時，Oracle Content Management 會複製選取的影像，並將其放在內容資料夾。您的 URL 會一律指向此影像的複製版本，因此即便您刪除原始影像，網站也不會中斷。這同樣適用於 Oracle Content Management 提供的其他元件：圖庫、圖庫格線、文件、社群列、檔案下載，以及槽位和 componentGroups 的背景影像。

若要將自訂元件加入到此資產生命週期中，自訂元件必須告知 Oracle Content Management 希望服務代表管理的任何資產。因為這涉及複製資產，因此自訂元件還必須使用 Oracle Content Management API 選取資產，我們才知道如何進行管理。

### 管理 URL

資產的 URL 會根據一些條件而變更。

- 元件的程式實際執行 URL 與該元件的網站產生器 URL 不同
- 如果您複製一個頁面，Oracle Content Management 也會複製內容資料夾中所有參照的資產，這樣就不會有兩個元件指向內容資料夾中相同的資產
- 將 componentGroup 拖放到頁面上會為 componentGroup 中元件所參照的任何資產產生新的複本

此外，雖然相對 URL 可能適用於本機元件，但是遠端元件仍需要您想要 Oracle Content Management 代表管理的任何資產完整 URL，這樣遠端元件才能使用完整 URL 呈現其內嵌框架內容。

因為無法倚賴 URL 保持靜態，因此必須在程式碼中保留資產的 ID 參照，並在要呈現資產時擷取資產的 URL。

### 管理資產

這些 Sites SDK API 適用於管理資產。

```
SitesSDK.getProperty('componentAssets', callback);
```

- 這會取得目前資產的陣列
- 每個資產項目包含：
  - **id**：資產的唯一 ID。
  - **title**：Oracle Content Management 標題描述資料。
  - **description**：Oracle Content Management 描述性描述資料。
  - **fileName**：所選檔案的原始名稱。在自訂元件的「設定值」面板中顯示時很有用，讓使用者知道他們選取的是哪一個檔案。這不是複製到內容資料夾的檔案名稱。
  - **source**：啟用巨集的資產 URL。這個值會隨時間變更，因此您的元件不應該參考這個值，但必須儲存為資產的一部分。
  - **url**：根據呼叫 getProperty() 相關資訊環境的資產完整 URL。

```
SitesSDK.setProperty('componentAssets', [assets]);
```

- 呼叫此函數以儲存所有想要讓 Oracle Content Management 代替您管理的資產。
- 如果您未呼叫此函數，則不會儲存任何資產。
- 當網站發布後，任何不在此陣列的資產都將被刪除。
- 資產參數是一個資產陣列，其格式與 `getProperty` 傳回的相同，`filePicker` 也會傳回同樣的格式。

 **備註：**

未儲存任何 url 值。該值會在您要求資產時動態建立。

```
SitesSDK.filePicker(options, callback);
```

- 會啟動檔案選擇器以選取資產清單的 API。
- 它會在成功選擇傳入所選資產陣列的資產時呼叫回呼。
- 此時不會儲存任何項目，並由元件呼叫 `setProperty('componentAssets', [assets]);` 以儲存與任何其他要儲存之資產結合的選擇項目。

### 選取資產的範例

本節會顯示如何從儲存的資產選取資產、儲存其 ID，以及重新擷取實際值。

1. 編輯 `settings.html` 檔案。
2. 變更樣板物件以包含 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. 變更 `viewModel` 以新增可監測項目來儲存所選資產的 ID。

```
self.imageID = ko.observable();
```

4. 透過啟動檔案選擇器及顯示所選資產的名稱，變更 `viewModel` 以管理資產的選擇。

```
//
// 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**，以便在取得 `customSettingsData` 之前，先擷取資產。此程式碼也會在 `self.ImageID()` 可監測項目變更時，呼叫 `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. 最後，更新 `save` 函數以儲存 `imageID`，並確定使用參照資產清單更新 `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);
```

### 檢查所選資產的結果

1. 重新整理網站中的頁面，讓網站產生器能夠取得元件的變更。
2. 將頁面切換為「編輯」模式
3. 將您的元件拖放到頁面上。
4. 啟動「設定值」面板。
5. 按一下**選取影像**按鈕。
6. 瀏覽 (或上傳) 並選取影像。  
請注意，影像名稱已儲存，會顯示所選影像。
7. 關閉「設定值」面板。
8. 再次啟動「設定值」面板。  
請注意，將再次反映影像的名稱。

### 呈現資產的範例

本節會顯示如何擷取資產並在元件中呈現，以及每當設定值面板中的值發生變更時，元件也能動態更新。

#### 備註：

雖然這裡顯示的是在頁面上內嵌框架中的本機元件範例，類似的程式碼也適用於在頁面中以內嵌方式呈現的元件。

1. 編輯 render.html 檔案。
2. 更新樣板以包含您的資產。

```
<!-- 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` 建立兩個可監測項目以從 `customSettingsData` 取得 `imageID`，並儲存從儲存的資產清單擷取的 `imageURL`。

```
self.imageID = ko.observable();
self.imageURL = ko.observable();
```

4. 更新 `viewModel`，以便在每次 `imageID` 發生變更時，都會取得對應的影像資產 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` 以從 `customSettingsData` 擷取 ID。

#### 檢查呈現資產的結果

1. 重新整理網站中的頁面，讓網站產生器能夠取得元件的變更。
2. 將頁面切換為「編輯」模式
3. 將您的元件拖放到頁面上。
4. 啟動「設定值」面板。
5. 按一下**選取影像**按鈕。
6. 瀏覽 (或上傳) 並選取影像。  
請注意，影像名稱已儲存，會顯示所選影像。
7. 關閉「設定值」面板。  
此時您應該會看到所選影像呈現在元件中。

繼續進行[教學課程複習](#)。

## 教學課程複習

此教學課程提供如何使用 `Knockout` 元件處理站建立自訂元件的總覽。

此教學課程的主要目的是運用此模式，只要透過更新 `SampleComponentViewModel` 和 `sampleComponentTemplate` `JavaScript` 物件便可建立任何自訂元件。`sampleComponentFactory` 和 `SampleComponentImpl` 物件在您進行教學課程時不會有所變更。您無須處理與頁面的通訊即可實行這些變更，並且能夠執行以下作業：

- 從「設定值」面板傳送變更至您的元件，並且保存這些變更。
- 執行觸發程式和動作，並且與頁面上的其他元件互動。
- 建立版面配置及運用巢狀元件。

- 定義元件特定樣式。

雖然此範例將自訂元件分割為多個檔案，但這是為了讓此教學課程更簡單明瞭。為了獲得最佳結果，您應該考慮適度封裝您的檔案以避免多次下載。

最後，儘管此教學課程適用於以 **Knockout** 為基礎的元件，但如果您想要使用其他 **JavaScript** 技術堆疊 (例如 **AngularJS**) 建立自訂元件，您必須重新實行 `SampleComponentImpl` 物件以建立與框架對應的通訊，以及特定於實際元件實行的技術。這項工作超出此教學課程的範圍。

## Sites SDK 參考資料

Oracle Content Management JavaScript SDK (Sites SDK) 提供一組函數，讓元件與 Oracle Content Management 的整合性更佳。

您可以從 Oracle Content Management 伺服器下載 Sites SDK：

```
http://{server}/_sitesclouddelivery/renderer/app/sdk/js/sites.min.js
```

Sites 應用程式設計介面 (API) 定義涵蓋 SitesSDK 全域物件的所有已定義命名空間：

- [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 的主要目的是讓使用者可以建置及管理應用程式。它提供數項功能：

- 跨內嵌框架和網域界限進行通訊
  - 無論代管位置在哪裡，元件都能與頁面進行通訊
  - 含括頁面模型生命週期：
    - \* 建立元件
    - \* 以儲存的特性呈現元件
    - \* 更新元件並儲存所做的變更
    - \* 刪除頁面中的元件
  - 儲存及擷取用來設定頁面中所用元件的自訂設定值
  - 使用元件間通訊來參與觸發程式和動作
- 頁面的幾何提示 (不適用於自訂元件)
  - 頁面可依照元件指示，動態調整內嵌框架的大小
  - 元件顯示為頁面的一部分，頁面可移動以回應元件調整事件 (無捲軸)

- 元件可感知 Oracle Content Management 的呼叫
  - 元件與頁面之間的通訊會有一個以私密金鑰簽署的權杖。
  - Sites SDK 可讓元件解碼權杖並加速呼叫者驗證。

若要包含 Sites SDK 函數，請提供已知的命令檔路徑。例如：

```
<script type="text/javascript"
src="<sdk_install_dir_path_prefix>/sites.min.js">
</script>
```

Sites SDK 包含以下的全域物件：

- SitesSDK
- SitesSDK.Utils

### SitesSDK

SitesSDK 全域物件可在呈現和設定值端點中使用。

命令	工作
<code>SitesSDK.getProperty(propertyName, callback)</code>	從主機網站取得具名特性的值。
<code>SitesSDK.setProperty(propertyName, propertyValue)</code>	將具名特性的值設為 <i>propertyValue</i> 。
<code>SitesSDK.getSiteProperty(propertyName, callback)</code>	從主機網站取得具名的網站特性值。例如，可以擷取主機網站目前所使用的主題設計。
<code>SitesSDK.publish(messageType, payload)</code>	傳送訊息到伺服器，並接受 <i>messageType</i> 和 JSON 物件作為有效負載。
<code>SitesSDK.subscribe(messageType, callback)</code>	從主機網站分配之訊息的訊息監聽器。這是非同步呼叫。這些是 JavaScript 回呼，所以函數應使用 JavaScript 閉包或適當地連結函數，以確保能夠存取適當的內容。

### SitesSDK.Utils

SitesSDK.Utils 命名空間包含 Oracle Content Management 元件之所有端點都可呼叫的公用程式函數。遠端元件的「設定值」面板提供此命名空間，內嵌本機元件的則沒有。

#### ▲ 注意：

Oracle 並未正式支援這些函數；它們僅作為範例實行使用。若使用這些函數，需自行承擔風險。

命令	工作
<a href="#">SitesSDK.Utills.Logger</a>	傳回 Sites SDK 目前使用的日誌記錄器物件。日誌層級可設為下列其中之一： <ul style="list-style-type: none"> <li>· 除錯</li> <li>· 日誌</li> <li>· 資訊</li> <li>· 錯誤</li> <li>· 警告</li> </ul>
<a href="#">SitesSDK.Utills.addSiteThemeDesign(cssUrl)</a>	將目前的網站主題設計新增至元件。
<a href="#">Utills.getStyle(elem,styleProp)</a>	計算內嵌框架高度的公用程式方法。

## SitesSDK.getProperty('componentAssets', callback)

此函數會傳回網站內由元件代為儲存的資產清單。

### 參數

名稱	類型	描述
componentAssets	字串	呼叫清單建立者。
callback	函數	回呼函數

### 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)

此函數會從主機網站取得指定的特性值。

### 參數

名稱	類型	描述
propertyName	字串	特性名稱。
callback	函數	回呼函數。

### 相關特性

特性名稱	描述	設定值	呈現
height	框架的高度	NA	是
width	框架的寬度	NA	是
customSettingsData	設定值資料	是	是
styleClass	選取的樣式類別	是	是
任何其他樣式屬性，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
    }
  );
});
```

### 從呈現 URL 擷取的額外相關特性

除了使用 getProperty() 擷取特性值之外，可從呈現 URL 擷取的其他相關特性還包括這些。

特性名稱	描述
id	元件 ID

特性名稱	描述
instance	加密的權杖
width	iframe 的寬度
height	iframe 的高度
viewMode	呈現模式值
locale	網站或瀏覽器中設定的語言
settingsId	目前的設定值 ID

### 從設定值 URL 擷取的額外特性

除了使用 getProperty() 擷取特性值之外，可從設定值 URL 擷取的其他特性還包括這些。

特性名稱	描述
currCompId	目前編輯的元件 ID
instance	加密的權杖
width	設定值 iframe 的寬度
locale	網站或瀏覽器中設定的語言
OrigSettingsId	元件的原始設定值 ID
settingsId	新設定值 ID

## SitesSDK.getSiteProperty(propertyName, callback)

此函數會從主機網站取得指定的網站特性值。其中一個範例就是擷取主機網站正在使用的目前主題設計。

### 參數

名稱	類型	描述
propertyName	字串	網站特性的名稱
callback	函數	用於讀取特性值的回呼函數

### 網站特性

特性名稱	描述	設定值	呈現
theme	目前主題設計的 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])

此函數可更新自訂元件所儲存的網站資產清單。

### 參數

名稱	類型	描述
componentAssets	字串	建立元件資產清單。
assets	JSON 物件	傳回所選的資產清單。

### 引數

特性名稱	描述
filetypes	支援的檔案類型清單
multiSelect	選取單一或多個檔案。

### 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:
//   fileType - list of supported file types
//   multiSelect - single or multiple file select
//
// returns a list of selected assets. If user cancels out of the
// picker,
// the callback is never called
SitesSDK.filePicker({options}, callback);

//
// example 'componentAssets' JSON returned:
//
'componentAssets': [{
  'name': <nameOfAssetInContentFolder>, // this is used to uniquely
and persistently identify the asset. It is typically prefixed by the
&originalName; followed by a 16 digit string
  'originalName': <originalName>, // name of the asset as

```

```

selected from DOCS
  'description': <description>,           // description/other attributes
that are available from DOCS
  'url': <fully qualified url to the asset>
}]

```

## SitesSDK.setProperty(propertyName, propertyValue)

此函數可將具名特性的值設為 *propertyValue*。

### 參數

名稱	類型	描述
propertyName	字串	特性名稱。
propertyValue	JSON 物件	特性的值。

### 相關特性

此表格列出可使用 `setProperty()` 擷取的一組相關特性。

特性名稱	描述	設定值	呈現
height	框架的高度	NA	是
width	框架的寬度	NA	是
customSettingsData	設定值資料	是	是
styleClass	選取的樣式類別	是	是
description	用於頁面上本機元件執行處理的橫幅中		
layout	傳回目前針對本機元件選取的版面配置。可根據 <code>appinfo.json</code> 檔案中的值，從「設定值」面板進行更新。		
renderStatus	允許自訂元件回報呈現完成。	<b>true</b>	
任何其他樣式屬性，例如 <code>borderColor</code> 、 <code>borderRadius</code> 、 <code>alignment</code> 、 <code>borderStyle</code> 、 <code>margin</code> 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)

此函數會傳回選取的檔案清單。

### 參數

名稱	類型	描述
options	字串	用以指定要傳回之所選檔案的選項。
callback	函數	回呼函數。如果使用者從選擇器取消，便一律不會呼叫回呼。

### 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)

此函數會傳回選取的檔案清單。

SitesSDK.openDocumentPicker() 函數會取用單一引數，並傳回解析至所選文件的承諾。可用的選項符合 *Integrating and Extending Oracle Content Management* 中找到之 Documents View Embed UI (documentsView Component) API V2 的 documentsView 元件選項。

### 用法

```
SitesSDK.openDocumentPicker({
  selectable: "any",
  layout: "grid"
}).then(function (selection) {
  console.log(selection);
});
```

擷取文件之後，必須使用 [SitesSDK.getProperty\('componentAssets', callback\)](#) 和 [SitesSDK.setProperty\('componentAssets', \[assets\]\)](#) 特性告知 Oracle Content Management 將這些文件的 ID 儲存至網站。否則，它們將不會與網站一起發布，而且可能遭到刪除。

## SitesSDK.openAssetPicker(options)

此函數會傳回選取的檔案清單。

SitesSDK.openAssetPicker() 函數會取用單一引數，並傳回解析至所選取資產的承諾。可用的選項符合 *Integrating and Extending Oracle Content Management* 中找到之 資產檢視內嵌 UI (assetsView 元件) API V2 的 assetsView 元件選項。

### 用法

```
SitesSDK.openAssetPicker({
  select: "single"
}).then(function (selection) {
  console.log(selection);
});
```

擷取資產之後，必須使用 [SitesSDK.getProperty\('componentAssets', callback\)](#) 和 [SitesSDK.setProperty\('componentAssets', \[assets\]\)](#) 特性告知 Oracle Content Management 將這些資產的 ID 儲存至網站。否則，它們將不會與網站一起發布，而且可能遭到刪除。

## SitesSDK.publish(messageType, payload)

此函數會傳送訊息到伺服器。它接受 messageType 和 JSON 物件作為有效負載。訊息類型會在頁面中進行辨識及處理。如果沒有處理程式，則會忽略傳送的訊息類型。

## 參數

名稱	類型	描述
messageType	字串	訊息類型
payload	JSON 物件	訊息有效負載

## 訊息類型

此表格列出本機元件 (使用內嵌框架) 或遠端元件可辨識且可利用 SitesSDK.publish() 函數傳送的訊息類型。

訊息類型	描述
SETTINGS_UPDATED	用於「設定值」面板中，將設定值已更新的訊息傳送至呈現端點。
TRIGGER_ACTIONS	用於元件呈現端點，以在主機網站中觸發動作。

## 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)

此函數實際上是訊息監聽器，用來監聽從主機網站發送的訊息。這是非同步呼叫。

當 SDK 收到特定類型的訊息時，就會呼叫已註冊的回呼。如果回呼傳回值，則該值會傳回給頁面。如果沒有對此函數傳送任何回呼，系統就會刪除該特定訊息類型的任何已註冊監聽器。

這些是 JavaScript 回呼，所以函數應使用 JavaScript 閉包或適當地連結函數，以確保能夠存取適當的內容。

## 參數

名稱	類型	描述
messageType	字串	訊息類型： <ul style="list-style-type: none"> <li>· SETTINGS_UPDATED</li> <li>· TRIGGER_ACTIONS</li> <li>· EXECUTE_ACTION</li> <li>· GET_ACTIONS</li> <li>· GET_TRIGGERS</li> <li>· COPY_CUSTOM_DATA</li> <li>· PASTE_CUSTOM_DATA</li> </ul>
callback	函數	從主機網站接收訊息時的函數。

## 訊息類型

訊息類型	描述
SETTINGS_UPDATED	訂閱此訊息即可在「設定值面板」中的任何值發生變更時收到通知。元件通常會藉由更新其在使用者介面中的轉譯來回應。
TRIGGER_ACTIONS	發布此訊息會引發觸發程式並傳送有效負載。
EXECUTE_ACTION	訂閱此訊息即可處理自訂動作。監聽器通常會藉由執行指定的動作來處理此訊息。
GET_ACTIONS	訂閱此訊息即可宣告元件可執行的動作陣列。傳回動作陣列。
GET_TRIGGERS	訂閱此訊息即可宣告元件可引發的觸發程式陣列。傳回觸發程式陣列。
COPY_CUSTOM_DATA	訂閱此訊息即可處理將「自訂設定值資料」複製到剪貼簿。傳回代表要放到剪貼簿的資料物件。
PASTE_CUSTOM_DATA	訂閱此訊息即可處理從剪貼簿貼上「自訂設定值資料」。

## 用法

有些監聽器會預期傳送資料，有些則預期要傳回資料。

### 範例 1：

註冊 EXECUTE\_ACTION 訊息類型的訊息監聽器：

```
SitesSDK.subscribe(SitesSDK.MESSAGE_TYPES.EXECUTE_ACTION,
self.executeAction,self);
```

已註冊的 callback 函數應接受一個引數，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
}
```

### 範例 2：

通常會使用如以下的程式碼來實行 COPY\_CUSTOM\_DATA 監聽器：

```
// listen for COPY_CUSTOM_DATA request
SitesSDK.subscribe(SitesSDK.MESSAGE_TYPES.COPY_CUSTOM_DATA,
```

copyCustomDataListener() 方法會傳回代表要複製到剪貼簿的資料物件，例如：

```
// Handle Copy Style (save customSettingsData to the clipboard)
self.copyCustomDataListener = function() {
    return {
```

```

        prop1: this.prop1(),
        prop2: this.prop2()
    };
};

```

### 範例 3：

PASTE\_CUSTOM\_DATA 要求會接受資料，但不一定要傳回任何項目：

```

// listen for PASTE_CUSTOM_DATA request
SitesSDK.subscribe(SitesSDK.MESSAGE_TYPES.PASTE_CUSTOM_DATA,
pasteCustomDataListener);

```

此外，還有：

```

// Handle Paste Style (apply customSettingsData from the clipboard)
self.pasteCustomDataListener = function(data) {
    ...
};

```

## SitesSDK.Utils.addSiteThemeDesign(cssUrl)

此函數會在目前的頁面 HTML <head> 標記中建立一個 link 元素。來源會設為 cssUrl 的路徑。

### ▲ 注意：

Oracle 並未正式支援 Utils 命名空間中的函數。它們主要作為範例實行使用。若使用這些函數，需自行承擔風險。

### 參數

名稱	類型	描述
cssUrl	字串	目前主題設計的 URL 路徑。只能搭配遠端元件使用。 本機元件 (顯現於 iframe 中) 必須從頁面 (而非 URL) 擷取特性。

### 用法

此函數通常用於從主機網站擷取目前的主題設計時，如本範例程式碼所示：

```

// 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

此函數可傳回 Sites SDK 所使用的日誌記錄器物件。

### ▲ 注意：

Oracle 並未正式支援 `utils` 命名空間中的函數。它們主要作為範例實行使用。如果使用 `utils` 函數，需自行承擔風險。

日誌層級可設為 `debug`、`log`、`info`、`error` 或 `warn`。

## 疑難排解

如果您在建立網站時遇到下列問題，請參閱這裡的解答。

- [我嘗試建立一個網站，但是沒有任何樣板可用](#)
- [我無法刪除網站](#)
- [我無法開啟網站樹狀結構或編輯頁面](#)
- [頁面中未顯示已新增的元件](#)
- [我的資料夾、檔案和對話元件無法運作](#)
- [頁面中未顯示已上傳的新版影像](#)
- [部分內容在頁面版面配置變更後消失不見](#)
- [我使用其他服務新增了元件，但該元件無法運作](#)
- [我的企業網站顯示警告](#)

### 我嘗試建立一個網站，但是沒有任何樣板可用

樣板必須先安裝並共用後，您才能夠建立網站。

Oracle Content Management 所提供的樣板必須先由管理員安裝並共用後，您才能夠使用這些樣板。請洽詢您的管理員。

同樣地，您在建立樣板時 (不論是透過匯入、複製或從網站建立)，建立的樣板都要等到您明確共用後，才可供其他人使用。

請參閱 *Administering Oracle Content Management* 中的設定網站設定值。

### 我無法刪除網站

以下是您無法刪除網站的原因：

- 網站處於線上狀態。  
先將網站離線，然後刪除網站。您必須具備網站的管理者角色，才能讓網站離線。
- 您沒有必要權限。  
如果是您建立的網站 (您是網站擁有者)，或者，如果有人已與您共用網站並將提供者或管理者角色指定給您，則可以刪除網站。

請參閱[管理網站和網站設定值](#)瞭解詳細資訊。

### 我無法開啟網站樹狀結構或編輯頁面

「網站產生器」預設會以預覽模式開啟，可讓您檢視網站，但是無法進行編輯。編輯器處於編輯模式時，您才能編輯網站。

確定  設為編輯。

## 頁面中未顯示已新增的元件

有許多可能的原因。

- 請確認遠端元件 (應用程式) 的 URL 有效。請參閱[註冊遠端元件](#)。
- 編輯器會將遠端元件包含在稱為內嵌框架 (iframe 標記) 的 HTML 元素中。並非所有遠端元件都能夠被包含在內嵌框架中。  
請與元件提供者確認，以瞭解是否可以將元件包含在內嵌框架中。

## 我的資料夾、檔案和對話元件無法運作

與 Oracle Content Management 通訊的元件需要使用某些資源和設定值。

下列元件需要存取 Oracle Content Management 的資源：

- 資料夾清單
- 檔案清單
- 文件管理程式
- 對話

資料夾元件和檔案元件都需要存取 REST 應用程式設計介面 (API)。

請檢查下列各項：

- 元件必須能夠存取 API 端點，才能存取資料夾和檔案 REST 作業。  
根據預設，若您使用為網站提供的標準 URL，則可使用 Oracle Content Management REST API 端點。若您使用自訂 URL，可能必須明確提供 / documents REST API 端點的存取權。  
請參閱[如何對應網站 URL](#)？

## 部分內容在頁面版面配置變更後消失不見

如果所選擇之版面配置的指定槽位較少或不同，則其他槽位中的現有內容將不會顯示在新的版面配置中。

除非您選擇的版面配置具有同名的槽位，否則內容不會被刪除，只是無法顯示而已。

請務必選擇槽位數目相同的版面配置。若您選擇槽位數目相同的版面配置，但部分內容仍舊未顯示，那麼很有可能是槽位的名稱不一樣。請洽詢主題設計者以解決相異處。

## 頁面中未顯示已上傳的新版影像

當您選取一個要搭配元件使用的影像時，系統會建立一個唯一複本並儲存在網站。

當您上傳檔案時，它們會儲存在網站儲存區域中。如果您上傳的檔案與現有的檔案同名，系統就會建立一個新的檔案版本。

您在頁面中見到的影像檔是從儲存區域中選取之檔案的複本，而且不會自動更新。這是有意設計的，目的在於防止現有頁面因儲存區域中的影像檔更新而意外變更。

若要更新搭配元件使用的影像，您必須明確地從元件特性頁面選取該影像。請參閱[影像](#)瞭解詳細資訊。

## 我使用其他服務新增了元件，但該元件無法運作

您可以將網站與幾個不同的處理作業和服務整合。

如果您是使用與 **Process Cloud Service**、**Oracle Intelligent Advisor** (之前稱為 **Oracle Policy Automation**) 及其他服務的整合而建立元件，那麼兩項服務都必須使用相同的識別網域。請向設定整合的人員確認兩項服務使用相同的識別網域。

## 我的企業網站顯示警告

企業網站若使用 **v1.0** 內容版面配置，在檢視網站時，您將會在「網站產生器」中見到警告，原因是版面配置無法呈現。這類版面配置需要更新成 **v1.1** 內容版面配置。

以下是會發生此情況的案例：

- 若您將現有網站設為可翻譯，該網站將會自動轉換成企業網站，而且會有指定的預設語言和本地化規定。網站樣板若使用 **v1.0** 內容版面配置，這類內容版面配置將無法呈現。
- 若您使用 **v1.0** 內容版面配置的舊版樣板建立企業網站，這類內容版面配置還是無法呈現。

若要更正此問題，需將內容版面配置更新成 **v1.1** 內容版面配置。您可以在元件目錄中建立新的內容版面配置，以瞭解差異。您需要新增一行來註冊內容版面配置的版本，並且將 `data.fieldname` 變更為 `fields.fieldname`。