

# Oracle® Cloud

## Building Sites with Oracle Content Management



F26844-21  
2021 June



Oracle Cloud Building Sites with Oracle Content ManagementNOT\_SUPPORTED

F26844-21

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

주요 작성자: Bruce Silver

기고 작성자: Jean WilsonNOT\_SUPPORTEDSarah BernauNOT\_SUPPORTEDKalpana  
NNOT\_SUPPORTEDRon van de Crommert

기고자: 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.

# 목차

## 머리말

---

대상	xvi
설명서 접근성	xvi
다양성과 포용성	xvi
관련 리소스	xvi
표기법	xvii

## I부 소개

---

### 1 Oracle Content Management 개요

---

Oracle Content Management 액세스	1-1
롤 이해	1-1
자산 관리	1-2
콘텐츠 관리	1-2
콘텐츠에 대한 공동 작업	1-2
사이트 구축	1-3
Oracle Content Management 통합 및 확장	1-3
시작하기	1-4
Oracle Cloud Infrastructure로 이전	1-4

### 2 사이트 구축 시작하기

---

사이트 시작하기	2-1
사이트 생성 프로세스 이해	2-2
사이트 개발에 대해 알아보기	2-3
사이트 제어 이해	2-5

### 3 첫번째 웹 사이트 생성

---

시작하기 전에	3-1
단계 1: 환경 설정	3-2

Minimal-NavMenu 사용자정의 구성요소 импорт	3-2
Minimal-NavMenu 사용자정의 구성요소 게시	3-3
샘플 이미지 추가	3-4
단계 2: 웹 사이트 설정	3-6
웹 사이트 생성	3-6
웹 사이트 편집	3-7
단계 3: 웹 사이트 게시	3-27
추가 작업	3-29

## II부 사이트 생성 및 편집

---

### 4 사이트 생성

---

사이트 생성	4-1
사이트 복사	4-3
사이트 요청 관리	4-4
사이트 요청 세부정보 변경	4-5
사이트 요청 정책 보기	4-6

### 5 사이트 편집

---

사이트 작성기 페이지 알아두기	5-1
사이트 업데이트 이해	5-3
업데이트 사용	5-5
팁과 힌트 편집	5-5
스타일 및 서식 사용	5-10
사이트 설정 사용자정의	5-10
사용자정의 사이트 속성 추가	5-11
테이블 작업	5-13
사이트 파일 업로드	5-15

### 6 사이트에서 템플릿 및 테마 사용

---

템플릿 이해	6-1
사이트에서 템플릿 생성	6-4
템플릿 세부정보 변경	6-5
템플릿 정책 변경	6-6
템플릿 상태 또는 대상 변경	6-7
템플릿 관리	6-8
템플릿 익스포트 및 импорт	6-11
테마 이해	6-12

테마 관리	6-14
테마 게시	6-16

## 7 사용자정의 구성요소 및 레이아웃 관리

---

사용자정의 구성요소 이해	7-1
레이아웃 이해	7-2
사용자정의 구성요소 및 레이아웃 사용	7-4
원격 구성요소 등록	7-7
로컬 구성요소, 레이아웃, 콘텐츠 필드 편집기 또는 콘텐츠 폼 생성	7-8
구성요소 또는 레이아웃 익스포트 또는 임포트	7-10

## 8 사이트 페이지 작업

---

페이지로 이동	8-1
페이지 보기	8-1
페이지 추가	8-3
페이지 이동	8-6
페이지 삭제	8-6
페이지 설정 변경	8-7
페이지 레이아웃 변경	8-9
배경 또는 테마 변경	8-10

## 9 페이지 콘텐츠 배열

---

구성요소 및 섹션 레이아웃 추가	9-1
자산 및 콘텐츠 항목 작업	9-3
트리거 및 작업 사용	9-4
가로 섹션 레이아웃 사용	9-5
2열 및 3열 레이아웃 사용	9-6
세로 섹션 레이아웃 사용	9-8
탭 섹션 레이아웃 사용	9-9
슬라이더 섹션 레이아웃 사용	9-9

## 10 내장 구성요소 사용

---

기본 구성요소	10-2
제목	10-2
단락	10-4
일반 텍스트	10-5
단추	10-6

구조 구성요소	10-8
구분선	10-8
스페이서	10-8
미디어 구성요소	10-9
이미지	10-9
갤러리	10-11
갤러리 그리드	10-14
YouTube 비디오	10-17
비디오	10-18
문서 구성요소	10-20
문서	10-20
폴더 목록	10-21
파일 목록	10-22
문서 관리자	10-23
프로젝트 라이브러리	10-25
소셜 구성요소	10-27
소셜 표시줄	10-27
Facebook 좋아요 및 추천	10-28
Twitter 공유 및 팔로우	10-28
대화 구성요소	10-29
대화 목록	10-31
프로세스 구성요소	10-32
프로세스 시작 폼	10-32
프로세스 작업 목록	10-34
작업 세부정보 폼	10-35
콘텐츠 항목	10-37
콘텐츠 항목 구성요소	10-37
콘텐츠 위치 표시자	10-39
콘텐츠 목록	10-39
콘텐츠 검색	10-44
권장사항	10-45
기타 구성요소	10-47
맵	10-47
헤드라인	10-48
기사	10-49
텍스트가 있는 이미지	10-51
구성요소 그룹	10-52
페이지에서 Cobrowse 사용	10-53
Oracle Intelligent Advisor	10-55
Oracle Visual Builder	10-55

## III부 사이트 게시 및 관리

---

### 11 사이트 관리

---

사이트 페이지 알아두기	11-1
사이트 및 사이트 설정 관리	11-2
사이트 설명, 로고 또는 포함 속성 변경	11-6
검색 엔진 속성 설정	11-7
검색 엔진 최적화에 대한 사전 렌더링 서비스 사용	11-8
정적 사이트 전달 옵션 설정	11-10
게시할 때 자동 컴파일 사용	11-10
컴파일된 사이트에 대한 기본 캐시 제어 헤더 무효화	11-11
컴파일된 적응형 레이아웃을 지원하도록 모바일 사용자-에이전트 지정	11-11
Vanity URL 지정 및 구성	11-12
Cobrowse 통합 사용	11-14
Analytics 추적 추가	11-15

### 12 사이트 게시

---

사이트를 온라인으로 전환 또는 오프라인으로 설정	12-1
사이트 변경사항 게시	12-2

### 13 사이트 보안

---

사이트 보안 이해	13-1
사이트 보안 변경	13-3

### 14 다국어 사이트 작업

---

다국어 사이트 개요	14-1
사이트 번역	14-2
사이트 번역 작업 관리	14-4
번역용 로케일	14-4
번역용 사용자정의 로케일	14-5
URL 재지정용 로케일 별칭 설정	14-5

### 15 사이트 재지정 또는 URL 매핑 사용

---

재지정 계획	15-1
단순 문자열-대-문자열 일치	15-1
간소화된 와일드 카드 문자 일치	15-2

사이트 재지정 추가	15-2
JSON 파일에 재지정 규칙 지정	15-2
사이트에 재지정 규칙 파일 업로드	15-8
사이트 URL 매핑	15-8

## 16 사이트 성능 향상

---

캐싱을 활용하여 성능 향상	16-1
런타임 캐싱	16-2
사이트 작성기 캐싱	16-3
ATF(Above the Fold) 렌더링	16-3

## IV부 사이트 개발

---

### 17 디자인 및 스타일 사용자정의

---

디자인 정보	17-1
디자인 파일	17-1
반응형 테이블 디자인	17-4
대화 목록 스타일 사용자정의	17-7
폴더 목록 및 파일 목록 스타일 사용자정의	17-9
소셜 표시줄 아이콘 사용자정의	17-10
Oracle Intelligent Advisor에 대한 인터뷰 스타일링 확장 구성	17-11

### 18 배경 사용 이해

---

배경 및 테마 정보	18-1
배경이 구현되는 방법	18-1
설정이 저장되는 위치	18-2

### 19 트리거 및 작업 설정

---

트리거 및 작업 정보	19-1
트리거 설정	19-1
작업 설정	19-3

### 20 템플릿 개발

---

템플릿 정보	20-1
기본 템플릿 구조	20-2
템플릿 생성	20-4



템플릿 익스포트	20-5
템플릿 임포트	20-5
스타터 템플릿 작업	20-6
Bootstrap 또는 웹 사이트 디자인 템플릿에서 사이트 템플릿 생성	20-9
Developer Cloud Service로 템플릿 개발	20-15
Oracle Content Management용 Developer Cloud Service 콘솔에 사인인	20-15
Developer Cloud Service에서 프로젝트 생성	20-16
Developer Cloud Service에서 템플릿 생성	20-16
Developer Cloud Service에서 템플릿 복사	20-17
Developer Cloud Service로 템플릿 임포트	20-17
변경사항 병합	20-18
Developer Cloud Service에서 템플릿 익스포트	20-18

## 21 테마 개발

---

테마 정보	21-1
기본 테마 구조	21-3
사이트 탐색	21-4
테마 생성	21-7
테마에 대한 구성요소 및 섹션 레이아웃 숨기기	21-11
테마에 대해 구성요소 맞춤, 너비 또는 간격 옵션 숨기기	21-13
테마와 구성요소 연관	21-14
사이트 렌더링 API	21-16

## 22 레이아웃 개발

---

레이아웃 정보	22-1
검색 엔진 최적화(SEO)	22-3
components.json 파일 및 형식 이해	22-5
사이트 작성기에서 도구모음 그룹 사용자정의	22-8
슬롯의 구성요소 제한	22-11
레이아웃 콘텐츠를 편집 가능으로 만들기	22-14
섹션 레이아웃 생성	22-15
지연 로드를 지원하는 섹션 레이아웃 생성	22-16
API를 사용한 사용자정의 섹션 레이아웃 개발	22-17
콘텐츠 레이아웃 개발	22-21
Oracle Content Management로 콘텐츠 레이아웃 생성	22-23
콘텐츠 레이아웃에 레이아웃 뷰 전달	22-29
API를 사용한 사이트 세부정보 페이지 URL 생성	22-29
Developer Cloud Service로 로컬에서 콘텐츠 레이아웃 개발	22-30
Developer Cloud Service로 콘텐츠 레이아웃 생성	22-31

RequireJS 모듈 정의	22-32
생성자 함수 매개변수 구성	22-33
콘텐츠 레이아웃 렌더링	22-34
Mustache 템플릿에서 콘텐츠 레이아웃 편집	22-34
동적 DOM 조작 추가	22-34
design.css 파일에서 스타일 정의	22-35
참조 항목 가져오기	22-35
미디어 URL 가져오기	22-35
트리거 발생	22-35
검색 질의를 사용하여 검색 페이지로 이동	22-36
매크로 확장 및 서식 있는 텍스트 렌더링	22-37
세부정보 페이지에 링크	22-37
콘텐츠 목록 질의에서 매크로 확장	22-37
강력한 콘텐츠 레이아웃 개발	22-41
콘텐츠 항목 렌더링	22-41
콘텐츠 레이아웃을 위한 데이터 구조 표준화	22-42
샘플 블로그 템플릿 생성	22-46
콘텐츠 레이아웃 매핑을 템플릿에 추가	22-46
로컬 테스트 하네스로 콘텐츠 레이아웃 테스트	22-47
로컬 테스트 하네스로 테스트	22-47
콘텐츠 레이아웃이 있는 템플릿을 Oracle Content Management로 импорт	22-47

## 23 구성요소 개발

구성요소 정보	23-1
구성요소 개발 정보	23-2
구성요소 생성	23-5
Developer Cloud Service로 사용자정의 구성요소 개발	23-6
Oracle Content Management용 사용자정의 구성요소 개발	23-7
사용자정의 구성요소 개발	23-7
단위 테스트 쓰기 및 실행	23-9
성능 향상을 위해 구성요소 최적화(축소)	23-9
연속 통합 작업 실행	23-10
다국어 사이트를 위한 번역 가능한 구성요소 개발	23-10
설정 패널이 있는 H1 구성요소 빌드	23-11
새 로컬 구성요소 생성	23-11
기본 H1 구성요소 빌드	23-12
구성요소에 대한 CSS 추가	23-15
설정 패널을 추가하여 머리글 텍스트 변경	23-16
다른 사람이 H1 구성요소 스타일을 선택하도록 테마 업데이트	23-18
HTML 템플릿을 사용하여 간단한 구성요소 생성	23-19

HTML 구성요소 생성	23-20
HTML 구성요소에 사용자 제공 데이터 추가	23-23
로컬 구성요소와 원격 구성요소 비교	23-27
구성요소 설정 렌더링	23-28
로컬 구성요소 구현	23-29
구성요소에 대한 스타일 클래스	23-30
내장 구성요소 스타일링 방법	23-32
구성요소 스타일링 기본사항	23-33
구성요소별 스타일링	23-34
구성요소 속성 설정	23-41
인라인 프레임으로 렌더링된 구성요소	23-42
인라인 프레임으로 렌더링된 구성요소에 대한 인스턴스 ID 및 구조 정보	23-43
원격 구성요소에 대한 보안	23-45
원격 구성요소 등록	23-46
구성요소 삭제	23-47
Sites SDK	23-47

## 24 컨트롤러 파일 사용자정의

---

컨트롤러 파일 정보	24-1
기본 컨트롤러 파일	24-1
기본 controller.html 파일 수정	24-2
SCS 객체 정보	24-3
SCS.sitePrefix	24-3
SCS.preInitRendering	24-3
SCS.getDeviceInfo	24-4
사용자정의하지 않아야 하는 컨트롤러 파일 섹션	24-4
사용자정의 컨트롤러 파일 이식성 허용을 위한 토큰 사용	24-5
사용자정의 컨트롤러 파일 샘플	24-6
사이트 접두어 변경	24-6
대기 그래픽 사용자정의	24-7
즐거찾기 아이콘 사용자정의	24-8
비JavaScript Crawler에 대한 <noscript> 및 <meta> 태그 사용자정의	24-9
JavaScript 파일 사전 인출	24-9
추가 마크업으로 사이트 소유권 확인	24-10
증분적인 장치 감지	24-10
controller.htm 이식성 향상을 위한 토큰 사용	24-11

## V부 기타 툴로 사이트용 개발

---

## 25 OCE Toolkit으로 개발

---

로컬 시스템에서 OCE Toolkit 설정	25-1
npm을 통해 종속성 설치	25-2
cec 명령행 유틸리티 사용	25-2
로컬 테스트 하네스로 테스트	25-67
jQuery 3.5.x로 업그레이드	25-67
Developer Cloud Service로 Oracle Content Management용 개발	25-68
Developer Cloud Service 사용 정보	25-69
Oracle Content Management용 Developer Cloud Service 콘솔에 사인인	25-69
Developer Cloud Service에서 프로젝트 생성	25-70
Oracle Content Management 템플릿으로 Developer Cloud Service 프로젝트 생성	25-70
Oracle Content Management에서 다운로드한 OCE Toolkit으로 Developer Cloud Service에서 프로젝트 생성	25-70
새 Git 저장소에서 프로젝트 코드에 OCE Toolkit 추가	25-70
로컬 테스트 하네스에서 사용자정의 구성요소, 템플릿 및 콘텐츠 레이아웃 테스트	25-71
변경사항 병합	25-71
OCE Toolkit을 사용하여 테스트에서 운용으로 변경사항 전달	25-71
비밀번호 암호화	25-77
서버 등록	25-77
사이트의 사용 및 권한 보고서 생성	25-78
문서 및 폴더 다운로드 및 업로드	25-78
템플릿에서 사이트 생성 및 콘텐츠에 동일한 GUID 유지	25-79
표준 사이트에서 엔터프라이즈 템플릿 생성	25-79
분류법 임포트 및 익스포트	25-79
권장사항 임포트 및 익스포트	25-81
모음 콘텐츠 추가 또는 제거	25-82
OCE Toolkit을 사용하여 사용자정의 필드 편집기 개발	25-82
한 서버에서 다른 서버로 사이트 전송 또는 업데이트	25-87
콘텐츠 항목이 없는 사이트 전송	25-87
그룹으로 사이트의 콘텐츠 항목 다운로드 또는 업로드	25-88
OCE Toolkit으로 사이트 페이지 인덱스화	25-88
사이트 페이지 텍스트에 대한 콘텐츠 유형 생성	25-89
OCE Toolkit으로 페이지 인덱스 콘텐츠 항목 생성	25-90
Oracle Content Management의 사이트에 콘텐츠 검색 추가	25-91
사이트에 검색 페이지 추가	25-91
테마에 검색 필드 추가	25-91
OCE Toolkit으로 다국어 사이트 인덱스화	25-92
쉬운 구성요소 개발을 위해 단순화된 구성요소 생성	25-94
사이트 컴파일 서비스 설정	25-95
사이트 컴파일로 사이트 페이지의 런타임 성능 향상	25-96

사이트 컴파일 개요	25-96
사전 렌더링과 상호작용	25-96
컨트롤러 사이트 페이지 렌더링	25-96
컴파일된 사이트 페이지 렌더링	25-97
템플리트 컴파일	25-97
설정	25-97
템플리트 컴파일	25-98
사이트 컴파일	25-103
사이트 컴파일을 위한 필요 조건	25-104
사이트 컴파일	25-104
사용자정의 컴파일러	25-105
제약 조건	25-106
사용자정의 컴파일러 디버그	25-106
페이지 레이아웃 컴파일러	25-106
구성요소 컴파일러	25-108
콘텐츠 레이아웃 컴파일러	25-109
SCSCompileAPI	25-111
구성요소 하이드레이션	25-111
게시	25-114
정적 사이트 전달 우선순위	25-114
캐싱 헤더	25-115
세부정보 페이지	25-115
콘텐츠 항목을 채널에 추가	25-116
모바일 장치에 대한 사이트 컴파일	25-117
사이트 수명 주기 및 컴파일된 페이지	25-118
Oracle Content Management 서버에서 새 사이트 또는 자산 번역 작업 생성	25-118
언어 서비스 제공자를 사용하여 사이트 번역	25-120
OCE Toolkit으로 번역 작업 생성	25-122
번역 작업 나열	25-122
번역 커넥터 생성	25-122
다국어 사이트의 사이트 맵 생성	25-123
언어 서비스 제공자에게 번역 작업 제출	25-124
서버에 번역 작업 업로드	25-125

## VI부   부록

---

### 26   자습서: Knockout으로 구성요소 개발

---

Knockout으로 구성요소 개발 소개 및 필요 조건	26-1
단계 1: 구성요소 생성	26-2

단계 2: 로컬 구성요소 렌더링의 구조 검토	26-3
단계 3: 로컬 구성요소 설정의 구조 검토	26-7
단계 4: 구성요소에 새 속성 표시	26-10
단계 5: 트리거 등록	26-11
단계 6: 트리거 발생	26-12
단계 7: 작업 등록	26-15
단계 8: 작업 실행	26-16
단계 9: 각 구성요소 인스턴스에 대한 고유 제목 생성	26-17
단계 10: 인라인 편집과 함께 중첩 구성요소 사용	26-18
단계 11: 다양한 레이아웃 지원	26-20
단계 12: 사용자정의 스타일 정의	26-22
단계 13: 인라인 프레임으로 구성요소 렌더링	26-23
단계 14: 구성요소가 인라인 프레임으로 렌더링될 때 사용자정의 스타일 사용	26-26
단계 15: 페이지 실행 취소 및 재실행 동작과 통합	26-29
단계 16: 자산 관리	26-30
자습서 검토	26-35

## 27 Sites SDK 참조

---

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)	27-10
SitesSDK.subscribe(messageType, callback)	27-10
SitesSDK.Utils.addSiteThemeDesign(cssUrl)	27-12
SitesSDK.Utils.Logger	27-13

## 28 문제 해결

---

사이트를 생성하려고 시도하는 중이지만 템플릿이 없음	28-1
사이트를 삭제할 수 없는 경우	28-1
사이트 트리를 열거나 페이지를 편집할 수 없는 경우	28-2
구성요소를 추가했지만 페이지에 노출되지 않는 경우	28-2
내 폴더, 파일 및 대화 구성요소가 작동하지 않음	28-2
페이지 레이아웃 변경 후 내 콘텐츠 일부가 사라지는 경우	28-2
업로드한 새 버전의 이미지가 페이지에 표시되지 않는 경우	28-3

다른 서비스를 기반으로 추가한 구성요소가 작동하지 않음  
내 엔터프라이즈 사이트에 경고가 표시됨

28-3  
28-3

# 머리말

이 문서에서는 Oracle Content Management를 사용하여 환경(사이트, 자산 및 연관된 모든 구조와 정책 등)을 생성하고 관리하는 방법에 대해 설명합니다.

## 대상

이 설명서는 웹 브라우저, 데스크톱 동기화 애플리케이션 또는 모바일 장치를 사용해서 파일 작업을 수행하려는 Oracle Cloud 사용자를 위해 제작되었습니다.

## 설명서 접근성

오라클의 접근성 개선 노력에 대한 자세한 내용은 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>에서 Oracle Accessibility Program 웹 사이트를 방문하십시오.

### 오라클 고객지원센터 액세스

지원 서비스를 구매한 오라클 고객은 My Oracle Support를 통해 온라인 지원에 액세스할 수 있습니다. 자세한 내용은 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info>를 참조하거나, 청각 장애가 있는 경우 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs>를 방문하십시오.

## 다양성과 포용성

오라클은 다양성과 포용성을 위해 최선을 다하고 있습니다. 오라클은 사고적 리더십과 혁신을 주도하는 다양한 인적 자원을 보유하는 데에 가치를 두고 있습니다. 직원과 고객 및 파트너에게 긍정적인 영향을 주는 보다 포용적인 문화를 조성하기 위한 이니셔티브의 일환으로 제품 및 문서에서 부적절한 용어를 제거하기 위해 노력하고 있습니다. 또한 고객의 기존 기술과의 호환성을 유지하고 오라클 제품 및 산업 표준의 발전에 따라 서비스의 연속성을 보장해야 할 필요성을 인식하고 있습니다. 이러한 기술적 제약으로 인해 부적절한 용어를 제거하려는 당사의 노력에는 시간이 소요되며 외부 협력이 필요합니다.

## 관련 리소스

자세한 내용은 다음 Oracle 리소스를 참조하십시오.

- 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*

## 표기법

다음은 이 문서에서 사용된 표기법입니다.

표기법	의미
굵은체	굵은체는 작업과 연관된 그래픽 사용자 인터페이스 요소 또는 텍스트나 용어집에 정의된 용어를 나타냅니다.
기울임꼴	기울임꼴은 책 제목, 강조 항목 또는 사용자가 특정 값을 제공하는 위치 표시자 변수를 나타냅니다.
고정 폭	고정 폭은 단락 내의 명령, URL, 예제에 있는 코드, 화면에 나타나는 텍스트 또는 사용자가 입력하는 텍스트를 나타냅니다.

# 1부 소개

이 부분에서는 Oracle Content Management로 사이트 구축에 대한 개념적 정보를 제공하고 첫번째 사이트 생성을 위한 간편한 자습서를 포함합니다. 다음 장이 포함됩니다.

- [사이트 구축 시작하기](#)
- [첫번째 웹 사이트 생성](#)

# 1

## Oracle Content Management 개요

Oracle Content Management는 클라우드 기반 콘텐츠 허브로, 옴니채널 콘텐츠 관리를 지원하고 경험 제공을 가속화합니다. 강력한 협업 및 워크플로우 관리 기능을 제공하여 콘텐츠의 생성과 제공을 간소화하고 고객과 직원의 참여를 강화합니다.

Oracle Content Management를 사용하면 모든 장치에서 내외부적으로 신속한 협업을 이루어 콘텐츠를 승인하고 상황별 경험을 생성할 수 있습니다. 내장된 기업 친화적 도구를 사용하여 뛰어난 콘텐츠로 새로운 웹 경험을 구축할 수 있습니다. 동일한 콘텐츠 플랫폼과 동일한 프로세스를 사용하여 모든 관계자의 디지털 참여를 유도할 수 있습니다. 기술적, 조직적 병목현상이 사라져 매력적인 경험 생성에 더 이상의 장벽은 없습니다.



주:

Oracle Content Management Starter Edition의 기능 집합은 제한적입니다. 전체 기능 집합을 활용하려면 Premium Edition으로 업그레이드하십시오.

## Oracle Content Management 액세스

Oracle Content Management에 대한 액세스 권한이 부여되면 인스턴스 URL 및 사용자 이름에 대한 세부정보가 포함된 시작 전자메일을 받게 됩니다. 이 정보는 서비스에 로그인할 때 필요하므로 나중에 참조할 수 있도록 보관하는 것이 좋습니다.

Oracle Content Management와 상호작용하는 다양한 방법이 있습니다.

- 웹 인터페이스를 사용하면 즐겨찾는 웹 브라우저에서 쉽게 액세스할 수 있습니다. 클라우드에서 콘텐츠를 관리하고, 다른 사람과 파일 및 폴더를 공유하고, 대화를 시작하고 참여하며, (허용된 경우) 웹 사이트를 생성할 수 있습니다.
- 데스크톱 앱을 사용하면 클라우드와 컴퓨터 간에 파일 및 폴더를 동기화된 상태로 유지할 수 있습니다. 자신의 파일과 공유 파일을 동기화할 수 있으므로 항상 최신 버전에 액세스할 수 있습니다.
- Microsoft Office 추가 기능을 사용하면 Microsoft Word, Excel, PowerPoint 및 Outlook에서 직접 Oracle Content Management 기능에 액세스할 수 있습니다.
- Android 및 iOS용 모바일 앱을 사용하면 휴대폰이나 다른 모바일 장치에서 쉽게 액세스할 수 있습니다. 모바일 앱은 웹 브라우저의 서비스와 모양 및 작동이 비슷하므로 바로 익숙해집니다. 클라우드 콘텐츠에 액세스하고, 파일 및 폴더를 검색/정렬하고, 콘텐츠를 공유하고, 대화를 진행할 수 있습니다.
- REST API 및 SDK는 개발자에게 Oracle Content Management 기능을 웹 애플리케이션 및 모바일 앱에 프로그래밍 방식으로 통합할 수 있는 강력한 도구를 제공합니다.

## 롤 이해

액세스 가능한 Oracle Content Management 기능은 지정된 롤에 따라 다릅니다. 사용자 롤에 따라 다른 옵션이 표시됩니다. 표준 사용자는 문서, 대화 및 사이트 작업을 수행할 수 있습니다.

엔터프라이즈 사용자는 자산에 액세스할 수도 있습니다. 개발자에게는 템플릿, 테마, 구성요소 및 레이아웃과 같은 웹 사이트 조각을 구축하고 사용자정의할 수 있는 옵션이 표시됩니다. 관리자에게는 서비스를 구성하고, 다른 비즈니스 애플리케이션과 서비스를 통합하고, 자산 저장소를 설정할 수 있는 옵션이 표시됩니다.

Oracle Content Management에는 다양한 유형의 롤이 있습니다.

- **조직 롤** — 조직에서 맡은 롤에 따라 수행할 작업과 기능 사용 방법이 결정됩니다.
- **사용자 롤** — 사용자 롤은 Oracle Content Management에서 표시되는 기능을 제어합니다.
- **리소스 롤(권한)** — 문서, 콘텐츠 항목, 사이트 또는 템플릿 등의 리소스로 확인 및 수행할 수 있는 작업은 리소스가 공유될 때 지정된 롤에 따라 다릅니다.

자세히 알아보기...

## 자산 관리

Oracle Content Management는 웹 사이트, 마케팅 자료, 전자메일 캠페인, 온라인 상점, 블로그를 포함한 다양한 채널에서 사용할 수 있도록 모든 자산을 관리할 수 있는 강력한 기능을 제공합니다. 모든 자산의 중앙 콘텐츠 허브로서, 저장소 및 모음으로 자산을 구성하고 자산 사용 방법 및 위치를 정의하는 규칙을 생성할 수 있습니다.

또한 자산 생성 및 승인 프로세스를 안내하고 인증된 버전만 사용할 수 있도록 광범위한 관리 및 워크플로우 기능을 제공합니다.

자산 태그 및 필터가 쉬우므로 필요한 자산을 빠르게 찾아서 허용된 곳에 사용할 수 있습니다. 또한 스마트 콘텐츠 기능은 자산을 사용할 때 자동으로 태그를 지정하고 추천합니다!

자산 유형을 생성하여 사용자가 자산을 생성할 때 어떤 정보를 수집해야 하는지 정의합니다. *디지털 자산 유형*은 디지털 자산(파일, 이미지 및 비디오)에 필요한 사용자정의 속성을 정의합니다. *콘텐츠 유형*은 다양한 콘텐츠 조각을 재사용 가능한 단위로 그룹화합니다. 그러면 사용자는 해당 자산 유형을 기반으로 디지털 자산 및 콘텐츠 항목을 생성하여 채널 간에 일관되게 사용하고 표현할 수 있습니다.

자세히 알아보기...

## 콘텐츠 관리

Oracle Content Management를 사용하여 클라우드에서 콘텐츠 관리를 시작하면 한곳에서 모두 관리하고 어디서나 액세스할 수 있습니다.

로컬 컴퓨터와 동일한 방식으로 폴더에 파일을 그룹화하고 일반적인 파일 관리 작업(복사, 이동, 삭제 등)을 수행할 수 있습니다. 또한 모든 파일이 클라우드에 있으므로 언제 어디서나 모바일 장치에서 액세스할 수 있습니다. 데스크톱 앱을 설치하면 모든 콘텐츠를 로컬 컴퓨터와 자동으로 동기화할 수 있으므로 항상 가장 최근 버전을 쉽게 사용할 수 있습니다.

자세히 알아보기...

## 콘텐츠에 대한 공동 작업

클라우드의 모든 콘텐츠에 대해 파일 또는 폴더를 쉽게 공유하여 조직 내외의 다른 사람과 공동 작업할 수 있습니다. 콘텐츠를 공유하는 모든 사람은 언제 어디서나 필요할 때마다 최신 정보에 액세스할 수 있습니다. 전체 폴더에 액세스 권한을 부여하거나 특정 항목에 대한

링크를 제공할 수 있습니다. 공유 항목에 대한 모든 액세스가 기록되므로 각 공유 항목에 액세스한 방법과 시간을 모니터링할 수 있습니다.

Oracle Content Management에서 대화를 통해 주제에 대해 토론하고 실시간으로 의견을 게시하여 다른 사람과 공동 작업할 수 있습니다. 필요에 따라 파일을 추가하여 모든 주제에 대해 독립형 대화를 시작할 수 있습니다. 또는 특정 파일, 폴더, 자산 또는 사이트에 대한 대화를 시작하여 빠르고 쉬운 피드백을 얻을 수 있습니다.

대화와 연관된 모든 메시지, 파일 및 주석이 보존되므로 쉽게 토론을 추적하고 검토할 수 있습니다. 또한 대화는 클라우드에서 실시간으로 이루어지므로 모바일 장치에서 대화를 보고 이동 중에도 참여할 수 있습니다.

자세히 알아보기...

## 사이트 구축

Oracle Content Management를 사용하면 마케팅 및 커뮤니티 웹 사이트를 개념부터 런칭까지 신속히 구축하고 게시함으로써 매력적인 온라인 경험을 제공할 수 있습니다. 프로세스가 완전히 통합되어 콘텐츠와 공동 작업, 창의성이 하나의 저작 및 게시 환경으로 결합되었습니다.

빠르게 시작하려면 미리 정의된 템플릿, 끌어 놓기 구성요소, 샘플 페이지 레이아웃 및 사이트 테마를 사용하여 미리 정의된 빌딩 블록에서 사이트를 조립하십시오. 또는 개발자가 사용자정의 템플릿, 사용자정의 테마 또는 사용자정의 구성요소를 생성하여 고유한 온라인 경험을 생성할 수 있습니다.

페이지에 지정된 슬롯으로 구성요소를 끌어 놓기만 하면 YouTube 비디오, 스트리밍 비디오, 이미지, 헤드라인, 단락, 소셜 미디어 링크 및 기타 사이트 객체를 추가할 수 있습니다. 단추 하나만 누르면 테마를 전환하고 사이트를 리브랜딩하여 조직 전체에 최적화된 일관된 모양을 제공할 수 있습니다.

업데이트를 하나 이상 작업하고 사이트에서 업데이트를 미리본 다음, 준비가 되었을 때 한 번만 누르면 업데이트를 게시할 수 있습니다.

사이트 작성기에서 사이트를 생성하고 게시하는 것 외에도 Oracle Content Management는 REST API, React JS, Node JS 및 기타 웹 기술을 사용하여 '헤드리스' 사이트 개발을 지원합니다.

자세히 알아보기...

## Oracle Content Management 통합 및 확장

Oracle PaaS(Platform-as-a-Service)가 제공됨에 따라 Oracle Content Management는 다른 Oracle Cloud 서비스와 효율적으로 작동합니다.

웹 애플리케이션에 웹 UI 포함을 통해 사용자가 콘텐츠와 직접 상호작용할 수 있습니다. AIF(Application Integration Framework)를 사용하여 사용자정의 작업을 통해 타사 서비스 및 애플리케이션을 Oracle Content Management 인터페이스로 통합할 수 있습니다. 또는 콘텐츠 커넥터를 개발하여 이미 다른 곳에서 생성한 콘텐츠를 Oracle Content Management로 가져와서 중앙에서 관리하고 여러 채널의 새로운 경험에 사용할 수 있습니다.

콘텐츠 및 사이트 관리, 전달 및 공동 작업을 위한 풍부한 REST API 및 SDK를 사용하여 Oracle Content Management 기능을 웹 애플리케이션에 통합할 수 있습니다.

클라우드에서 콘텐츠 SDK 및 자산과 상호작용하는 클라이언트 애플리케이션을 생성합니다. 공동 작업 객체와의 사용자정의 통합을 개발하거나 필요한 곳에 사용할 자산을 검색합니다. 웹 사이트, CDN(콘텐츠 전달 네트워크) 또는 모바일 앱을 통해 각 채널에 최적화된 모든 콘텐츠 및 자산에 액세스하고 전달할 수 있습니다.

자세히 알아보기...

## 시작하기

Oracle Content Management를 시작하려면 [설명서](#), [비디오](#), [둘러보기](#), [개발자 정보](#)를 포함한 많은 리소스가 있는 [Oracle 도움말 센터](#)를 방문하십시오.

또한 필요한 경우 도움을 주는 [지원](#) 및 [커뮤니티](#)가 있습니다.

## Oracle Cloud Infrastructure로 이전

Infrastructure 콘솔을 사용한 OCI(Oracle Cloud Infrastructure)에서 Oracle Content Management 구독이 아직 실행되지 않는 경우 오라클은 고유 OCI 환경으로 이전할 것을 권장합니다. 그러면 향후 Oracle Cloud 플랫폼의 이점과 발전을 누릴 수 있습니다.

이전은 자동이 아니므로 프로세스를 시작하려면 서비스 요청을 제출해야 합니다.

자세히 알아보기...

## 2

# 사이트 구축 시작하기

Oracle Content Management는 클라우드 기반 콘텐츠 허브로, 옴니채널 콘텐츠 관리를 지원하고 경험 제공을 가속화합니다. Oracle Content Management를 사용하면 모든 장치에서 내외부적으로 신속한 협업을 이루어 콘텐츠를 승인하고, 디지털 자산을 관리하며, 내장된 기업 친화적 도구를 사용하여 상황별 경험을 생성할 수 있습니다.

- [Oracle Content Management 개요](#)
- [사이트 시작하기](#)
- [사이트 생성 프로세스 이해](#)
- [사이트 개발에 대해 알아보기](#)
- [사이트 제어 이해](#)
- [첫번째 웹 사이트 생성](#)

 [비디오](#)

## 사이트 시작하기

적절한 권한을 가진 경우 누구나 Oracle Content Management를 사용하여 웹 사이트를 구축할 수 있습니다. 독점적 도구, 코드나 소프트웨어를 사용할 필요가 없습니다. 사용자 인터페이스는 직관적이고 친화적인 그래픽 인터페이스입니다.

 **주:**

Oracle Content Management Starter Edition에서는 단일 사이트로 제한되며 사이트 제어가 지원되지 않습니다. 전체 기능 집합이 필요하다면 Oracle Content Management Premium Edition으로 업그레이드하십시오.

사이트를 구축할 수 있는 사람은 여러 요인에 따라 결정됩니다.

- 서비스 관리자가 사이트 생성을 사용으로 설정했는지 여부
- 서비스 관리자가 사이트 제어를 사용으로 설정했는지 여부. [사이트 제어 이해](#)를 참조하십시오.
- 사이트 생성이 사이트 관리자로 제한되는지 여부(사이트 제어가 사용 안함으로 설정된 경우 사용 가능)
- 사이트 관리자가 템플리트를 사용 가능하도록 설정했는지 여부

사이트를 생성할 때 템플릿으로 시작합니다. 템플릿은 사이트 코드 프레임워크, 샘플 페이지와 콘텐츠로 구성된 기본 사이트, 스타일이 적용된 테마, 리소스(예: 이미지)는 물론 사용자정의 구성요소까지 사이트를 시작할 때 필요한 모든 것을 포함합니다. [사이트 생성 프로세스 이해](#)를 참조하십시오.

엔터프라이즈 사이트를 생성하는 경우 저장소와 연결되며, 기본 언어를 지정하는 지역화 정책을 정의해야 합니다. 사이트를 생성하기 전에 저장소 및 지역화 정책을 모두 생성해야 합니다.

사이트에 필요한 자산과 문서를 저장소에 저장하면 저장소 정책에 따라 자산으로 수행할 수 있는 작업이 결정됩니다. 저장소를 사용하는 방법에 대한 자세한 내용은 자산 저장소 이해를 참조하십시오.

사이트를 편집할 때 새 업데이트를 생성하거나 기존 업데이트를 사용합니다. 한 업데이트 내에서 콘텐츠를 추가/편집하고, 스타일 설정을 조정하고, 페이지를 추가/삭제하고, 페이지 레이아웃을 변경하고, 페이지를 구성할 수 있습니다. 편집기로 수행할 수 있는 작업을 알아보려면 [사이트 작성기 페이지 알아두기](#)를 참조하십시오.

업데이트를 한꺼번에 모두 준비할 필요는 없습니다. 당신과 팀 멤버는 여러 업데이트를 동시에 독립적으로 작업할 수 있습니다. 예를 들어, 내가 주간 뉴스 업데이트 작업을 하는 동안, 다른 팀 멤버는 예정된 영업 컨퍼런스 페이지를 추가할 수 있습니다. 필요할 때마다 업데이트의 변경사항을 편집, 검토, 저장할 수 있고 언제든지 기본 사이트에 업데이트를 병합할 수 있습니다.

준비가 되면 사이트를 런칭합니다. 클릭 한번으로 웹에 사이트를 게시합니다. 개념부터 런칭까지 아주 간단합니다.

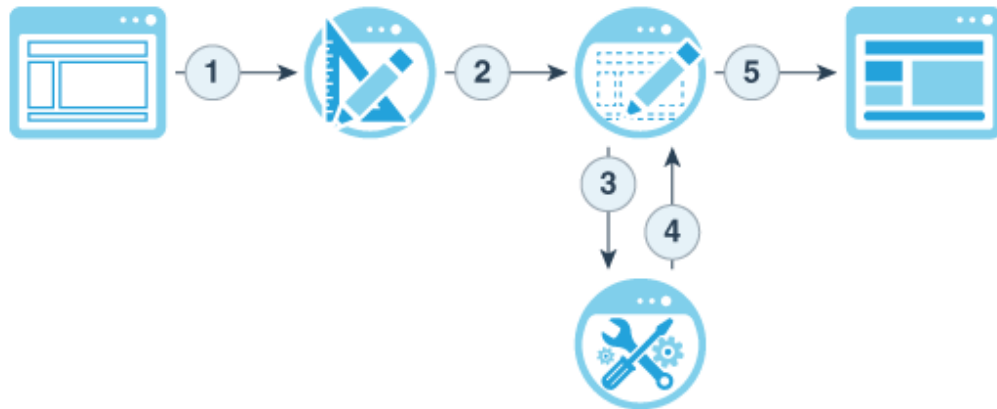
## 사이트 생성 프로세스 이해

웹 사이트를 생성하고 게시하는 전체 프로세스를 살펴보겠습니다.

서비스 관리자가 사이트 생성을 사용으로 설정하고 사이트를 생성할 수 있는 사용자를 구성하고 사이트 생성에 사용될 템플릿을 설치 및 제공한 경우에만 시작할 수 있습니다. 서비스 관리자가 최소한 템플릿에 대한 다운로드 권한을 지정해야 합니다. 사용자 룰에 대해 알아보기, [사이트 시작하기](#) 및 [사이트 제어 이해](#)를 참조하십시오.

자산 및 다국어 사이트의 사용을 가능하게 해주는 엔터프라이즈 사이트를 생성하면 저장소와 연관됩니다. 사이트에 필요한 자산과 문서를 저장소에 저장하면 저장소 정책에 따라 자산으로 수행할 수 있는 작업이 결정됩니다. 저장소는 엔터프라이즈 사이트를 생성하기 전에 생성되어 있어야 합니다. 자산 저장소 이해를 참조하십시오.

다음은 사이트 생성 프로세스의 개요이며 자세한 세부정보에 대한 링크를 제공합니다.



- 1. 템플릿 선택 및 사이트 이름 지정:** 사이트 제어가 사용으로 설정되고 사이트에 승인이 필요한 경우 승인이 완료될 때까지 기다렸다가 다음 단계를 완료할 수 있습니다. 게시할 준비가 될 때까지 사이트는 오프라인으로 남아 있습니다.
- 2. 사이트 생성:** 사용자정의할 수 있는 빈 프레임워크를 생성합니다.



3. **새 업데이트 생성 또는 기존 업데이트 사용:** 업데이트는 현재 기본 사이트에 대한 변경사항의 명명된 모음입니다. 편집기에서 사이트를 보거나 편집할 때마다 업데이트를 사용합니다. [사이트 업데이트 이해](#)를 참조하십시오.
4. **편집기에서 사이트 열기:** 편집기에서 끌어 놓기를 사용하여 사이트에 페이지와 콘텐츠를 추가합니다. [사이트 업데이트 이해](#)를 참조하십시오.
  - **사이트에 페이지 추가:** 페이지 유형을 빠르게 정의하려면 미리 정의된 레이아웃을 선택합니다. [페이지 추가](#)를 참조하십시오.
  - **페이지에 구성요소 추가:** 텍스트, 이미지, 문서 등을 페이지에 끌어 놓습니다. [내장 구성요소 사용](#)을 참조하십시오.
  - **구성요소의 콘텐츠나 속성 변경:** 구성요소의 간격, 맞춤 및 기타 속성을 미세 조정합니다. [스타일 및 서식 사용](#)을 참조하십시오.
  - **선택적으로 페이지에 사용된 레이아웃 전환:** 즉시 페이지 디자인을 변경합니다. [페이지 레이아웃 변경](#)을 참조하십시오.
  - **선택적으로 사이트에 사용된 테마 전환:** 전체 사이트의 모양을 빠르게 변경합니다. [사이트에서 템플릿 및 테마 사용](#)을 참조하십시오.
  - **검색 엔진 속성 설정:** 검색 엔진이 트래픽 증가 사이트를 찾도록 도와줍니다. [검색 엔진 속성 설정](#)을 참조하십시오.
5. **사이트 업데이트 적용:** 업데이트의 변경사항으로 사이트를 업데이트합니다. [사이트 변경사항 게시](#)를 참조하십시오.
  - **다른 팀 멤버와 사이트 공유:** 지정된 사용자와 사이트를 공유하고 각자 사이트에서 수행할 수 있는 작업을 결정하는 룰을 지정합니다. 공유에 대한 자세한 내용은 [사이트 보안 이해](#)를 참조하십시오.
  - **사이트 보안 설정:** 게시된 사이트에 액세스할 수 있는 사용자를 선택합니다. [사이트 보안 변경](#)을 참조하십시오.
  - **사이트 게시:** 사이트 변경사항을 게시합니다. [사이트 변경사항 게시](#)를 참조하십시오.
  - **사이트를 온라인으로 전환:** 사이트 보안에 따라 사용자에게 사이트를 제공합니다. [사이트를 온라인으로 전환 또는 오프라인으로 설정](#)을 참조하십시오.

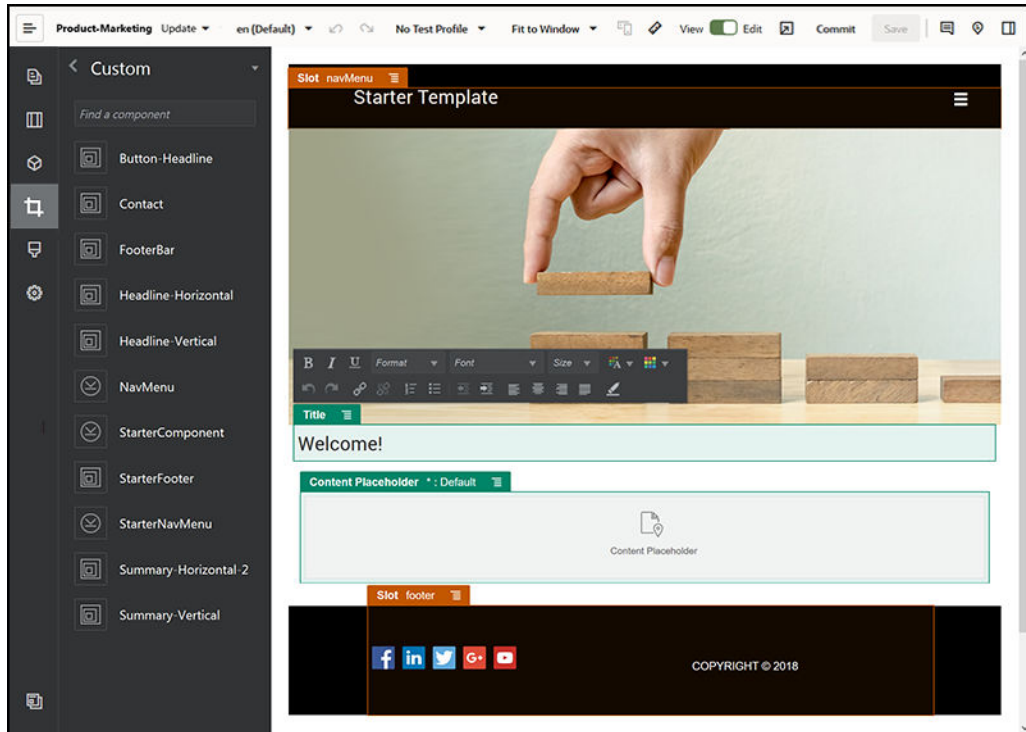
## 사이트 개발에 대해 알아보기

사이트 개발자는 다음과 같이 사이트 생성자가 사이트 구축에 사용할 프레임워크를 정의합니다.

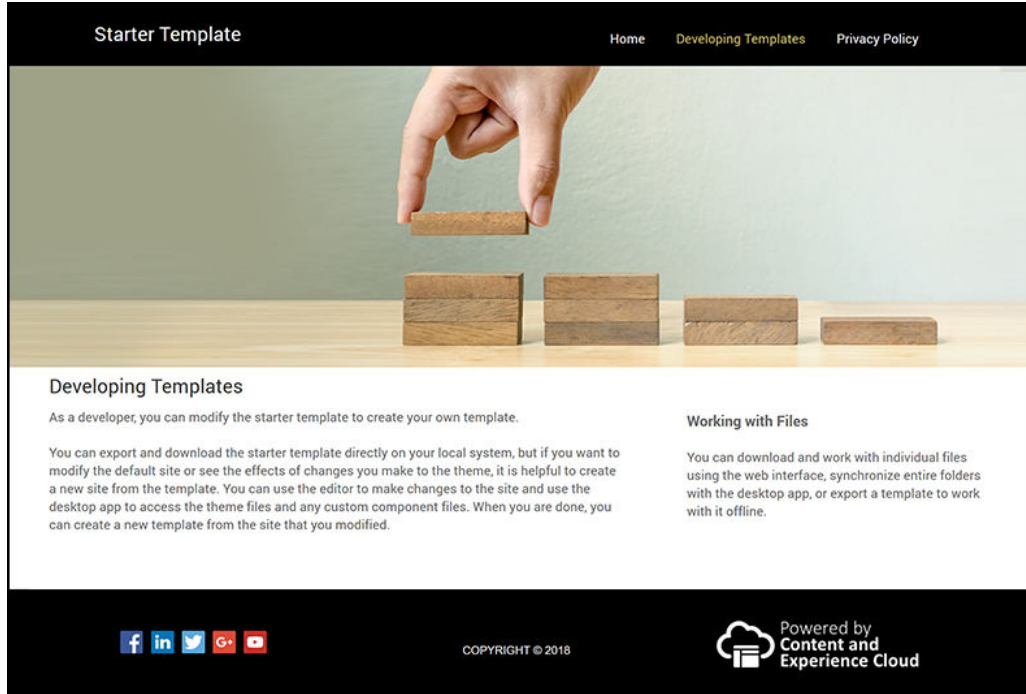
- **표준 템플릿** - 샘플 페이지 및 콘텐츠, 사용자정의 구성요소 및 기타 사이트 구축을 시작할 때 필요한 리소스와 함께 테마를 기반으로 사이트의 프레임워크로 사용됩니다.
- **테마** - 전체 사이트의 모양을 정의하는 데 사용되며 로고, 스타일시트, 구성 파일 및 사이트 탐색을 정의하는 배경 코드로 구성됩니다.
- **사용자정의 구성요소** - 특정 유형의 콘텐츠를 사이트 페이지에 추가하는 데 사용됩니다. 그러면 원하는 페이지 기술을 사용하여 사이트 페이지 안에 포함할 수 있는 복합 요소를 개발할 수 있습니다.

사이트 콘텐츠와 사이트 표현을 분리하면 모든 생성된 사이트에서 조직의 표준 및 브랜딩이 유지되고 콘텐츠 생성자는 콘텐츠에만 집중하여 개발 및 사이트 생성을 더욱 효율적이고 효과적으로 만들 수 있습니다.

다음은 콘텐츠 생성자가 사이트에 콘텐츠를 추가할 때 사용할 수 있는 간단한 템플릿의 예입니다. 일반 위치 표시자가 페이지에 레이아웃됩니다.



다음은 간단한 템플릿을 사용하여 구축된 사이트에 콘텐츠를 추가한 후 미리보기할 때의 모습을 보여주는 예입니다. 위치 표시자가 특정 콘텐츠 및 탐색 요소로 사용자정의된 모습을 살펴보십시오.



또한 사이트 개발자는 **스타일시트**를 생성 및 유지보수하고, 웹 인터페이스를 통해 또는 OCE Toolkit을 사용하여 **사이트 구축**을 수행하며, Oracle Content Management와 다른 서비스 간의 통합을 구성합니다. 그리고 다른 직원과 마찬가지로 콘텐츠를 공유하거나 대화를 시작/참여하거나 데스크톱 또는 모바일 앱을 사용하여 다른 사람과 협업하기도 합니다.

개발자가 Oracle Content Management를 사용할 수 있도록 하려면 개발자에게 표준 사용자 또는 엔터프라이즈 사용자 롤이 지정되어야 합니다. 표준 사용자 롤을 가진 개발자는 구성요소, 테마 및 표준 템플리트를 생성할 수 있습니다. 엔터프라이즈 사용자 롤을 가진 개발자는 레이아웃을 생성하고 사이트를 표준 또는 엔터프라이즈 템플리트로 저장할 수 있습니다.

사이트 개발을 넘어 Oracle Content Management는 헤드리스 환경에서 클라우드의 강력하고 유연한 백엔드 CMS(콘텐츠 관리 시스템)로 사용될 수 있습니다.

## 사이트 제어 이해

사이트 제어를 통해 비즈니스 사용자는 회사 정책을 준수하는 사이트를 손쉽게 생성할 수 있으며 사이트 관리자는 중앙 위치에서 사이트를 간편하게 제어 및 추적할 수 있습니다.

### 주:

Oracle Content Management Starter Edition에서는 단일 사이트로 제한되며 사이트 제어가 지원되지 않습니다. 전체 기능 집합이 필요하다면 Oracle Content Management Premium Edition으로 업그레이드하십시오.

### 문제

오늘날 업계에는 한 가지 문제가 있습니다. 바로 폭발적으로 증가하는 다양한 사용자 환경입니다. 이러한 환경을 빠르게 생성 및 배치해야 하며, 일관적인 회사 브랜드 및 메시지 전달을 위해 중앙에서 관리할 수 있어야 합니다. 기업의 측면에서 볼 때 새로운 환경을 개발하고 제공하는 것은 어쩌면 영원히 지속되는 과제로 보일 수 있습니다. 기업의 성공을 위해서는 여러 환경을 일관적인 방식으로 생성 및 관리하고, 제품 또는 브랜드 시책을 빠르게 추진할 수 있게 해주는 강력한 적응형 기술이 필요합니다. 새로운 환경을 빠르고 효율적으로 제공할 수 없다면, 시장 기회를 놓치거나, 고객 및 잠재 고객의 기대에 제대로 부응하지 못할 수 있습니다.

사용자 환경이 폭발적으로 증가하게 된 원인은 수년간에 걸쳐서 개별적으로 진행된 시책들 때문일 수도 있고, 중앙에서의 관리 없이 각각의 부서, 브랜드, 업무 팀 등이 WCM(웹 콘텐츠 관리) 시스템을 개별적으로 사용했기 때문일 수도 있습니다. 그 결과 시장에 빠르게 반응하기 어렵고, 보안 위협에 취약하며, 반복적이며, 예측하기 어려운 비용이 발생하기 쉬운 복잡한 환경이 만들어졌습니다.

여러 WCM 기술을 개별적으로 사용하고, 일관성 없이 사용자 환경을 구축함으로써 조직은 엄청난 보안 위협에 직면하게 됩니다. 웹 사이트에 있는 각각의 문제들이 여러 이해당사자들에게 걸려 있는 상황에서는, 보안과 같은 핵심 의무도 각 사용자들로 분산되어, 전체 조직이 위협에 취약하게 됩니다. 이렇게 서로 다른 웹 애플리케이션 및 IT 인프라 감독이 사용되는 서로 다른 WCM 시스템으로부터 발생한 이 모든 환경은, 불가능까진 아니더라도, 모든 것에 대한 적절한 보호가 어렵도록 만듭니다.

또한 고가의 여러 상업용 WCM 시스템을 유지함에 따라, 매년 라이선스 갱신 비용과 내부 또는 외부 파트너의 상시 지원 비용 등의 중복 비용이 발생합니다.

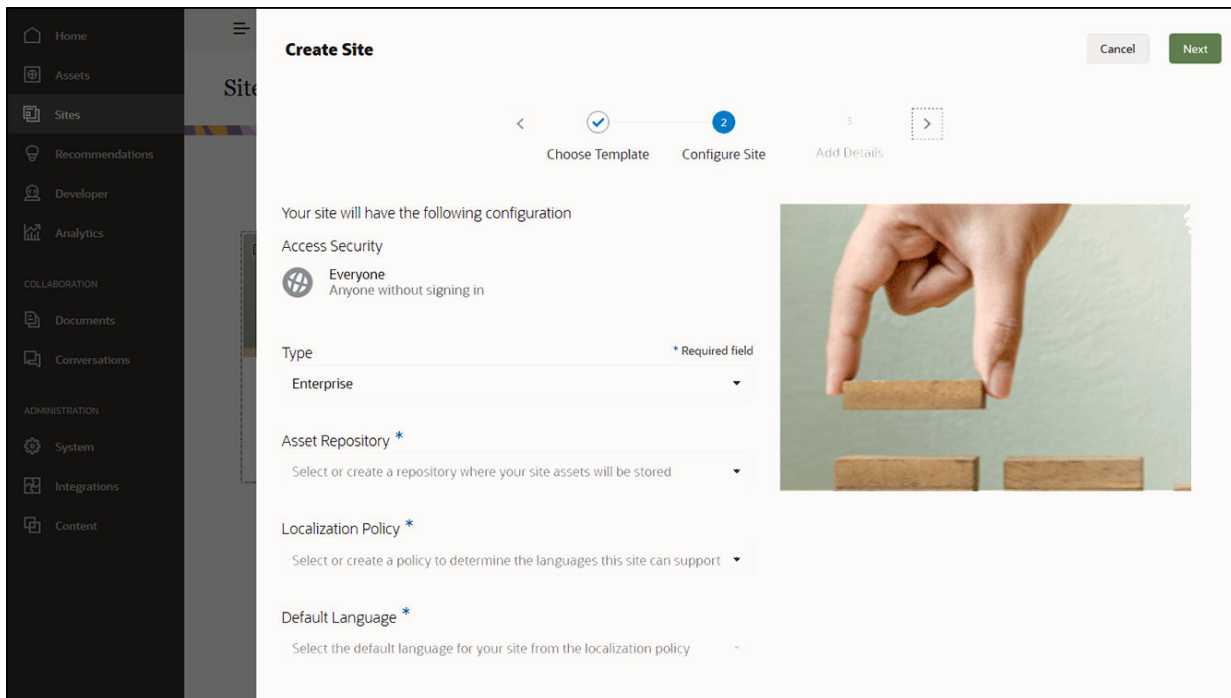
결과적으로 다음과 같이 모든 사람들의 불만으로 이어질 수 있습니다.

- 경영진의 불만:
  - 셀프 서비스 불가능, IT 부서 또는 외주 업체 의존에 따른 비용 증가
  - 기술 지원 없이는 불가능한 업데이트
  - 관리하기 까다롭고 비즈니스에 친화적이지 않은 도구
  - 모든 사용 환경 및 작업에 대한 중앙 관리 기능 부재

- IT 부서의 불만:
  - 모든 사용 환경에 IT 부서가 관여해야 함
  - 모니터링되지 않는 외주 환경을 포함하여 비즈니스 부서에서 생성되는 사용 환경에 대한 제어 기능 부족
  - 사용 환경 구축을 위해 사용된 단편화된 스택
  - 수백 가지 환경 간에 콘텐츠 및 레이아웃 변경사항을 관리하고 배치해야 함
- 사용자의 불만:
  - 채널 간 일관적이지 않은 메시징
  - 오래된 정보
  - 낮은 성능으로 인한 채널 방치

### 해결 방법

Oracle Content Management의 제어 기능은 비즈니스 사용자를 위한 사용 환경 제공을 단순화하고 가속화합니다. 또한 IT 부서는 전체 사용 환경의 수명 주기를 완전히 관리하고, 중앙 위치에서 사용 환경을 쉽게 제어 및 추적할 수 있습니다. 그 결과 회사에 필요한 새로운 사용 환경을 생성하고 유지하는 데 필요한 비용을 줄일 수 있습니다. 이러한 제어 기능은 Oracle Content Management에 이미 포함되어 있으므로, 관리자가 이를 사용으로 설정하기만 하면 됩니다. *Administering Oracle Content Management*의 사이트 및 자산 설정 구성을 참조하십시오.



### 특징

- **승인된 템플릿 카탈로그** - 여러 비즈니스 라인의 요구에 맞게 IT 개발자가 일련의 사이트 템플릿으로 템플릿 카탈로그를 채울 수 있습니다. 어떤 템플릿을 사용할 수 있고 누구에게 제공되는지 제한할 수 있습니다. 템플릿 정책을 적용하여 새 사이트가 준수해야 하는 보안 유형과 새 사이트에 승인이 필요한지 여부를 지정할 수 있습니다. 그

결과 새 사이트를 쉽고 빠르게 요청할 수 있고, 비즈니스 사용자가 브랜드 및 보안 지침을 따르도록 보장할 수 있습니다. [템플릿 상태 또는 대상 변경](#) 및 [템플릿 정책 변경](#)을 참조하십시오.

- **효율적으로 진행되는 요청, 승인 및 프로비저닝** - 비즈니스 사용자가 필수 승인 및 자동 프로비저닝에 따라 새 사이트를 빠르게 요청할 수 있습니다. [사이트 생성](#) 및 [사이트 요청 관리](#)를 참조하십시오.
- **사이트 관리** - 사이트 생성자 및 배치자가 누군지 관계없이 IT 부서가 한 곳에서 모든 사이트를 관리할 수 있습니다. IT 사용자가 사이트 상태를 모니터링하고 배치된 사이트의 상태를 변경할 수 있습니다. [사이트 및 사이트 설정 관리](#)를 참조하십시오.

### 혜택

제어 기능을 사용하면 새로운 환경 생성 및 유지에 필요한 비용을 줄일 수 있습니다.

- 경영진에 대한 혜택:
  - IT에 의존하지 않는 신속한 프로비저닝
  - 콘텐츠와 디자인의 분리, 재사용 가능
  - 환경, 사용자 및 권한을 모두 하나의 콘솔에서 전역으로 관리
  - 보고서를 통해 환경 작업에 대한 통찰력 확보
- IT 부서에 대한 혜택:
  - 복잡하지 않은 구현(클라우드 고유 솔루션)
  - 성장 및 계절적 요구 증가를 처리할 수 있는 자동 크기 조정
  - 환경에 대한 제어 기능을 통해 보안과 기업 브랜딩 및 준수 표준 충족
- 사용자에게 대한 혜택:
  - 채널 간 일관적인 메시지
  - 최신 정보
  - 최적 성능
  - 보안 환경

### 프로세스

사이트 제어를 통한 사이트 생성 및 관리에서는 다음 단계가 수행됩니다.

1. 관리자가 제어를 사용으로 설정합니다. *Administering Oracle Content Management*의 사이트 및 자산 설정 구성을 참조하십시오.
2. 사이트 관리자가 사용자에게 사이트 생성용으로 승인된 템플릿을 제공합니다. [템플릿 상태 또는 대상 변경](#) 및 [템플릿 정책 변경](#)을 참조하십시오.
3. 사용자가 승인된 템플릿에서 사이트 요청을 생성합니다. [사이트 생성](#)을 참조하십시오.
4. 사이트 관리자가 사이트 요청을 승인하면 사이트가 자동으로 생성됩니다. 또한 사이트 관리자가 요청이 거부된 이유를 설명하는 메모를 사용하여 사이트 요청을 거부할 수 있으며, 사이트 생성자가 문제를 해결하고 요청을 다시 제출할 수 있습니다. [사이트 요청 관리](#)를 참조하십시오.
5. 사이트 생성자가 사이트 생성 프로세스를 계속하여 사이트를 편집 및 게시합니다. [사이트 생성 프로세스 이해](#)를 참조하십시오.

6. 상시 사이트 관리 및 업데이트를 사이트 페이지에서 수행할 수 있습니다. 사이트 관리자는 사이트 페이지의 *모든* 사이트를 확인할 수 있지만, 다른 사용자는 직접 생성하거나 속해 있는 사이트만 확인할 수 있습니다. [사이트 및 사이트 설정 관리](#)를 참조하십시오.

# 3

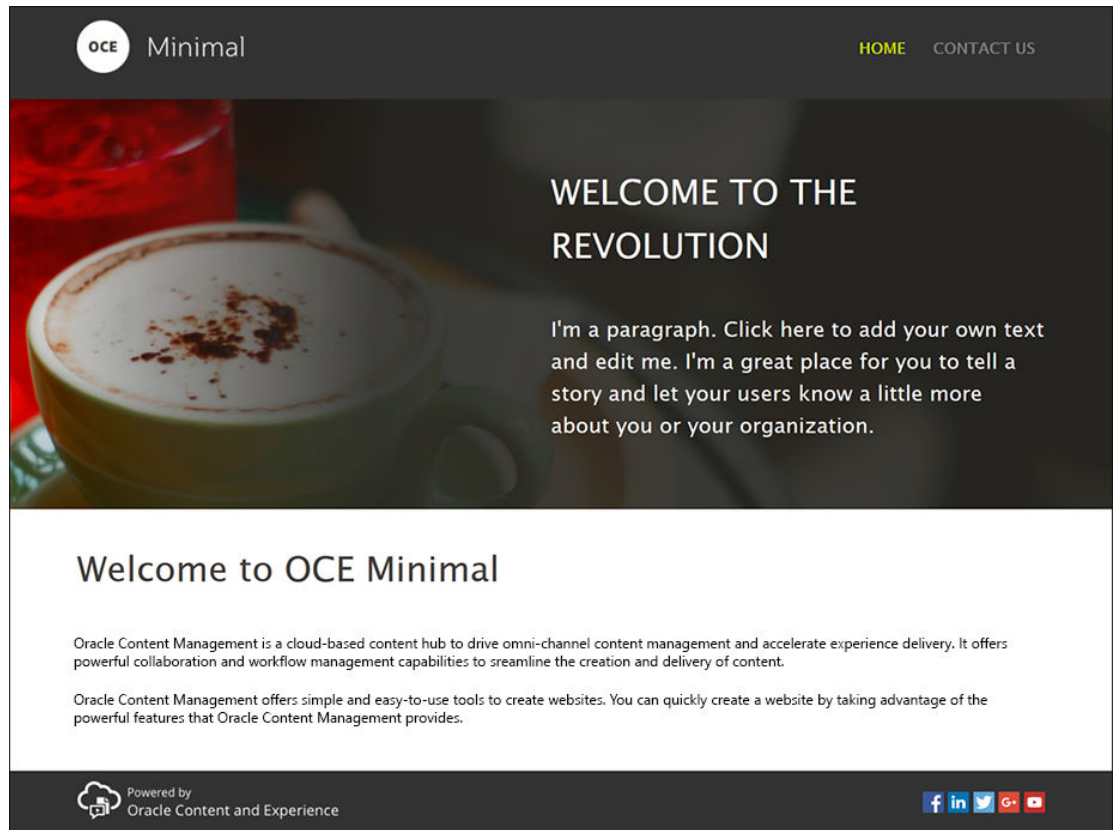
## 첫번째 웹 사이트 생성

Oracle Content Management를 사용하여 쉽고 빠르게 첫번째 웹 사이트를 구축하십시오.

이 자습서에서 Oracle Content Management에 제공된 미리 정의된 템플릿인 **Blank-Template**를 사용하여 웹 사이트를 설정하고 게시하는 방법을 배웁니다. 추가 코딩할 필요 없이 사이트 작성기를 사용하여 첫번째 웹 사이트가 얼마나 쉽게 구축되는지 확인하십시오.

이 자습서는 세 가지 기본 단계로 구성됩니다.

1. 환경 설정
2. 웹 사이트 설정
3. 웹 사이트 게시



그러나 시작하기 전에 먼저 몇 가지 사항을 확인하겠습니다.

## 시작하기 전에

시작하려면 다음 애플리케이션 **롤**을 사용하여 Oracle Content Management 인스턴스에 액세스해야 합니다.

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

위의 롤이 없으면 서비스 관리자에게 롤 지정을 요청하십시오.

Oracle Content Management 인스턴스에 액세스할 수 있으면 콘텐츠 관리자로 사인인하십시오.

시작하겠습니다.

## 단계 1: 환경 설정

먼저 Minimal-NavMenu 사용자정의 구성요소를 임포트하고 샘플 이미지를 업로드하여 환경을 설정해야 합니다.

따라서 자산 팩 **OCESamplesAssetPack.zip**을 다운로드합니다. <https://www.oracle.com/middleware/technologies/content-experience-downloads.html>에서 제공됩니다. 다운로드한 자산 팩 내에서 **OCECreateYourFirstSite\_data.zip** 파일을 찾습니다. 다음 두 개의 zip 파일이 들어 있습니다.

- **Minimal-NavMenu.zip**: Minimal-NavMenu 사용자정의 구성요소를 포함합니다.
- **Minimal-Images.zip**: 웹 사이트에 사용되는 샘플 이미지를 포함합니다.

환경 설정을 시작하겠습니다.

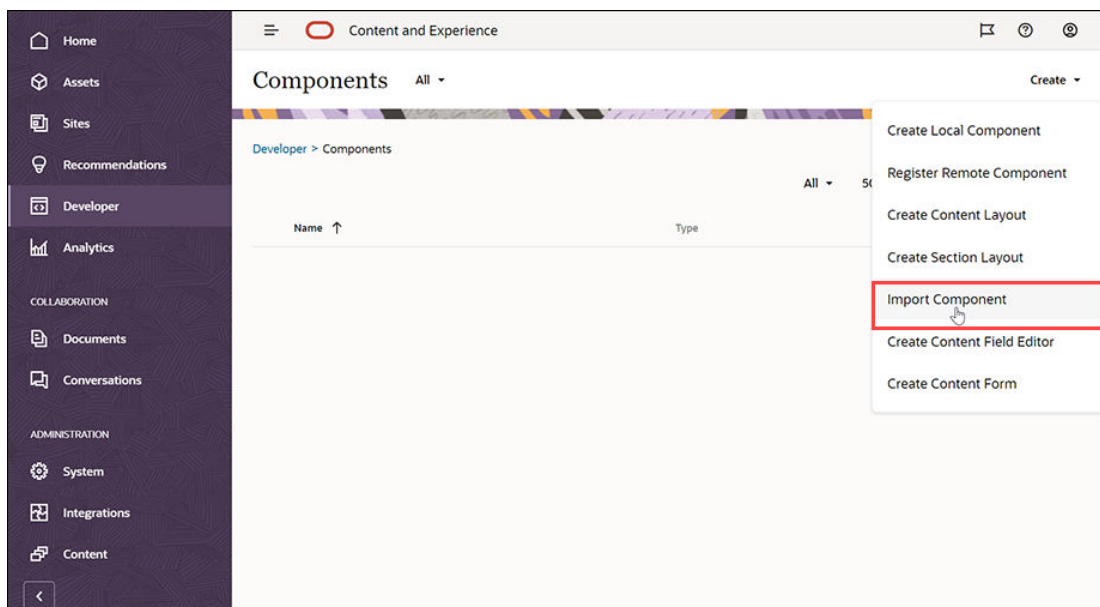
1. [Minimal-NavMenu 사용자정의 구성요소 임포트](#)
2. [Minimal-NavMenu 사용자정의 구성요소 게시](#)
3. [샘플 이미지 추가](#)

## Minimal-NavMenu 사용자정의 구성요소 임포트

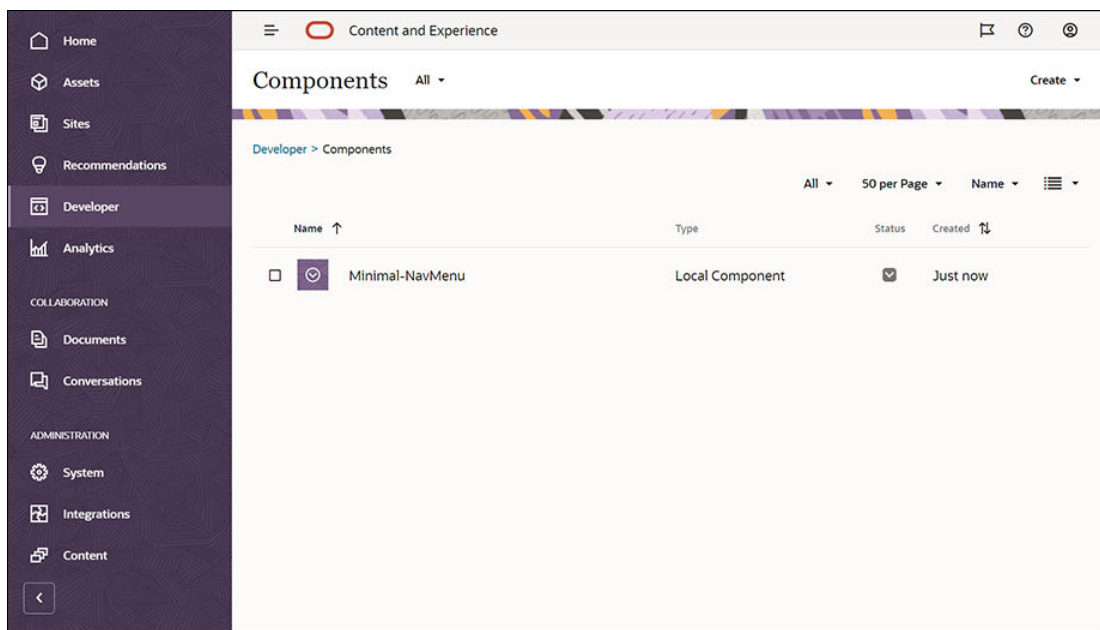
Oracle Content Management 웹 인터페이스에 로그인한 후 왼쪽 탐색 메뉴에서 **개발자**를 누르고 **모든 구성요소 보기**를 누릅니다. **개발자** 옵션이 보이지 않으면 필요한 사용자 **롤**이 없는 것입니다.

[구성요소] 페이지에서 **생성**을 누르고 **구성요소 임포트**를 선택합니다. **Minimal-NavMenu.zip** 파일(**OCECreateYourFirstSite\_data.zip** 파일 내에서 **사용 가능**)을 업로드하고 선택합니다.





이제 Minimal-NavMenu 구성요소가 [구성요소] 페이지에 나열됩니다.

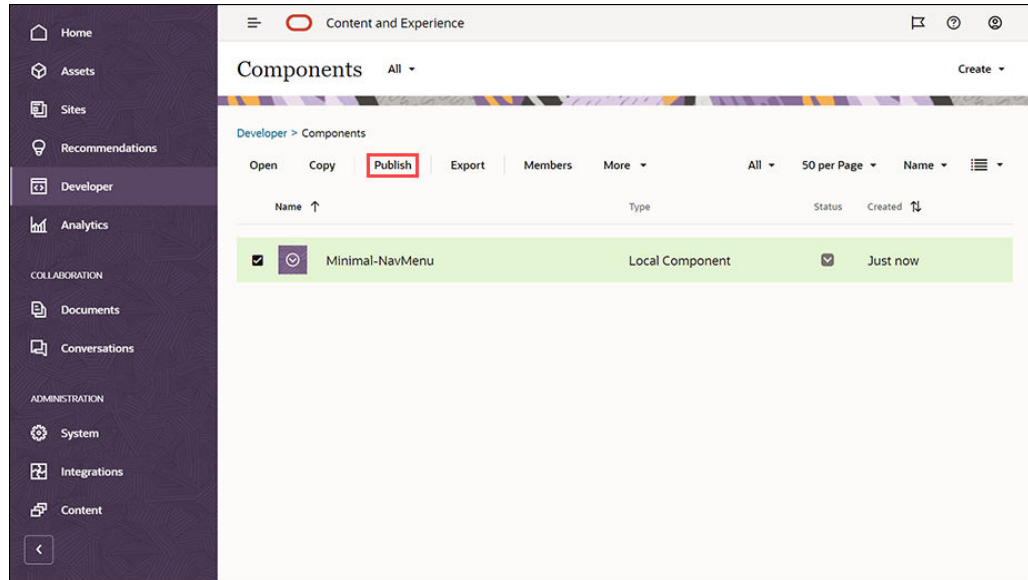


다음 단계: [Minimal-NavMenu 사용자정의 구성요소 게시](#)

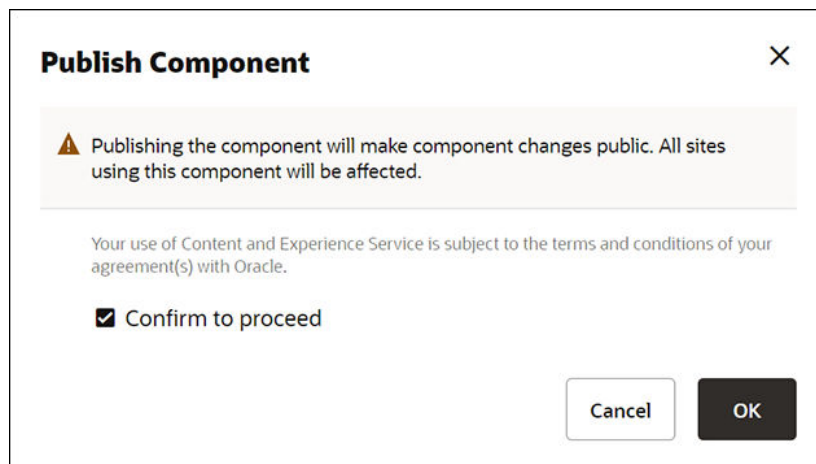
## Minimal-NavMenu 사용자정의 구성요소 게시

이제 임포트한 Minimal-NavMenu 구성요소를 게시해야 합니다.

[구성요소] 페이지에서 Minimal-NavMenu 구성요소를 선택하고 메뉴 모음 또는 마우스 오른쪽 단추 누르기 메뉴에서 **게시**를 누릅니다.



[구성요소 게시] 대화상자에서 **진행 확인**을 선택하고 **확인**을 누릅니다.



Minimal-NavMenu 구성요소가 게시되면 페이지 상단에 통지가 표시되어 구성요소가 게시되었음을 알립니다.

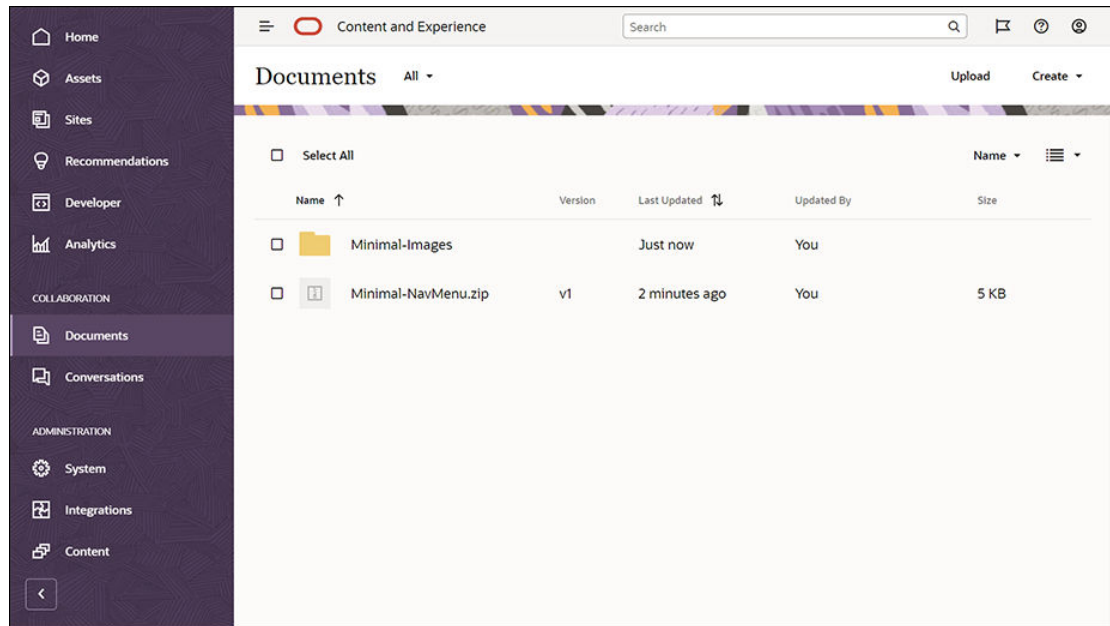
다음 단계: [샘플 이미지 추가](#)

## 샘플 이미지 추가

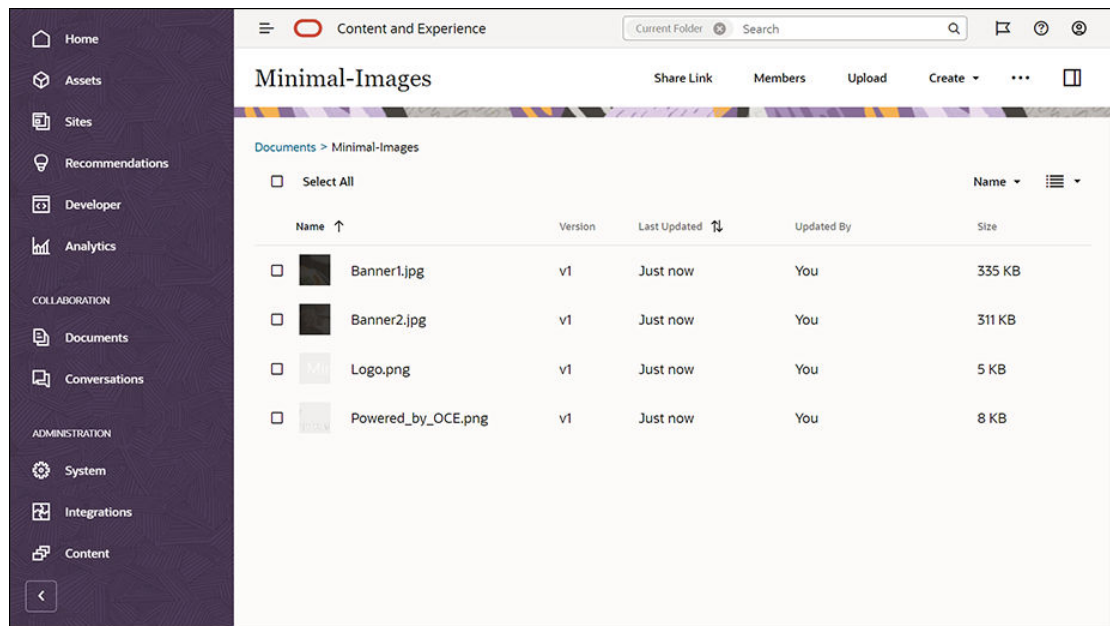
이제 샘플 이미지(생성할 웹 사이트에서 사용)를 문서에 추가해야 합니다.

**Minimal-Images.zip** 파일([OCECreateYourFirstSite\\_data.zip](#) 파일 내에서 사용 가능)의 내용을 로컬 컴퓨터의 **Minimal-Images**라는 폴더에 추출합니다.

왼쪽 탐색 메뉴에서 문서를 누르고 **생성**을 눌러 **Minimal-Images**라는 폴더를 생성합니다.



로컬 컴퓨터의 **Minimal-Images** 폴더에 있는 모든 이미지를 Oracle Content Management에 새로 생성된 **Minimal-Images** 폴더로 업로드합니다.



**주:**

템플릿과 웹 사이트에 사용되는 이미지는 Shutterstock 이미지입니다. 이 이미지를 재사용할 경우 Shutterstock를 통해 라이선스를 받아야 합니다. 이 이미지를 자신의 이미지로 바꿀 수도 있습니다.

다음 단계: 웹 사이트 설정

## 단계 2: 웹 사이트 설정

이제 웹 사이트 생성에 필요한 모든 것이 준비되었으므로 웹 사이트 설정을 시작할 수 있습니다.

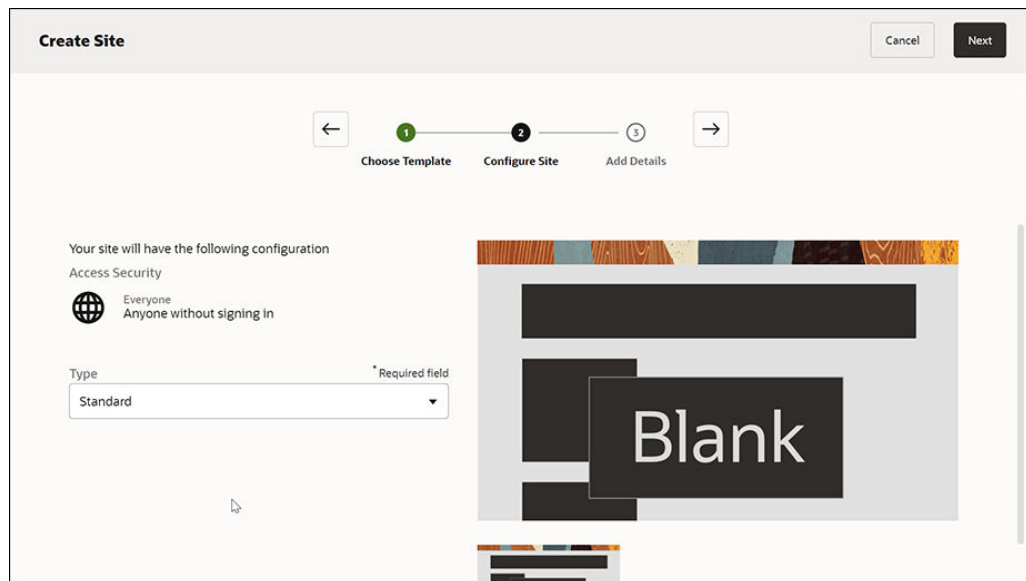
1. 웹 사이트 생성
2. 웹 사이트 편집

### 웹 사이트 생성

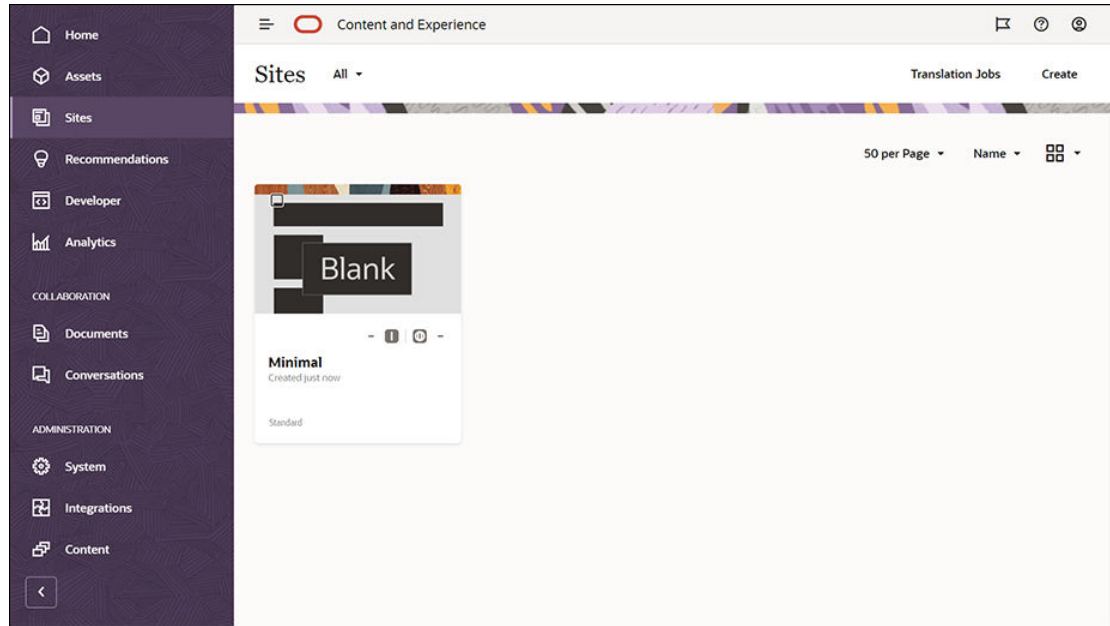
왼쪽 탐색 메뉴에서 **사이트**를 누르고 **생성**을 누릅니다. **Blank-Template**를 선택하고 다음을 누릅니다.

**Blank-Template**가 보이지 않으면 서비스 관리자에게 문의하십시오. 서비스 관리자는 미리 정의된 템플릿을 사용 가능하게 해야 합니다. 서비스 관리자는 일반적으로 서비스를 프로비전할 때 Oracle Content Management가 제공하는 **미리 정의된 템플릿**을 설치합니다.

표준 웹 사이트를 생성하도록 선택했는지 확인하고 다음을 누릅니다. 웹 사이트의 이름(예: **Minimal**)을 입력합니다. **완료**를 누릅니다.



새 웹 사이트가 생성되고 [사이트] 페이지에 나열됩니다.

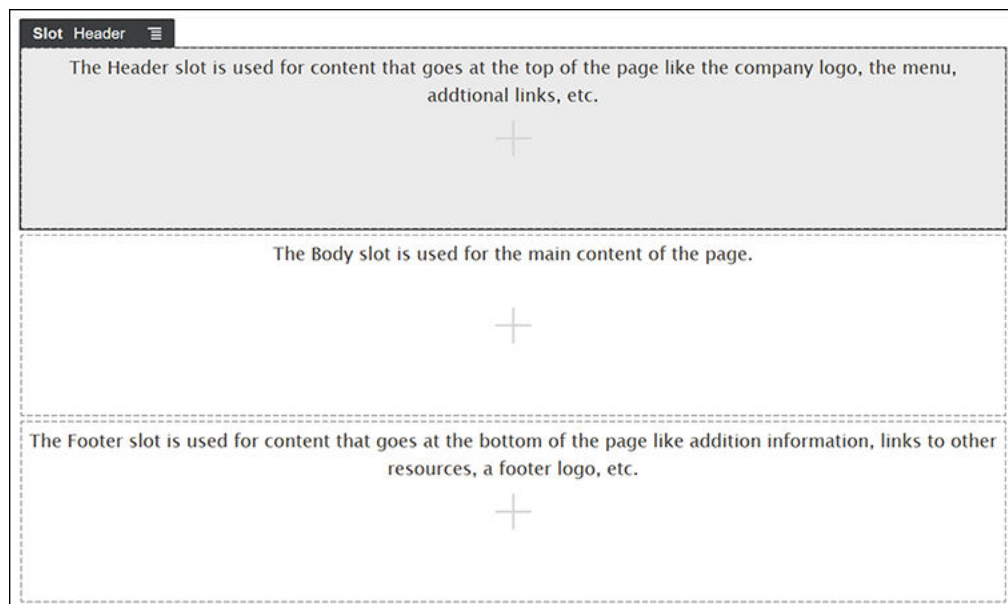


다음 단계: 웹 사이트 편집

## 웹 사이트 편집

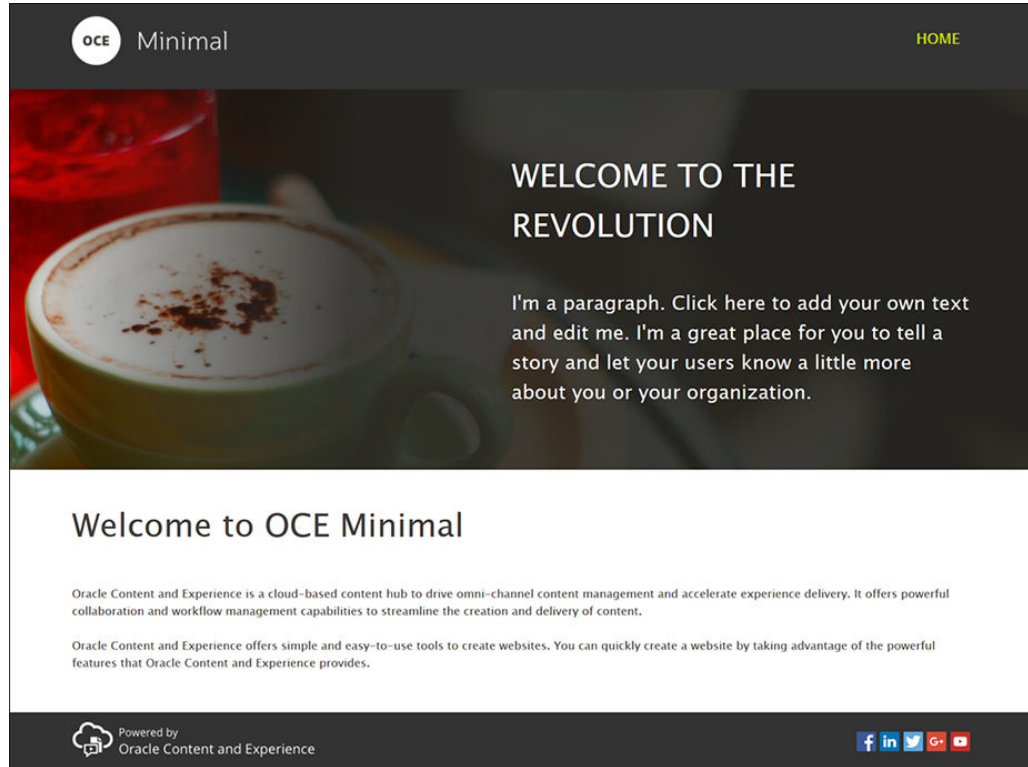
사이트 작성기에서 새로 생성된 웹 사이트를 선택하고 메뉴 모음 또는 마우스 오른쪽 단추 누르기 메뉴에서 열기를 선택하여 사이트를 엽니다. 사이트 작성기에서  스위치를 편집 모드로 설정합니다. 업데이트 이름을 입력하고 확인을 누릅니다.

편집 모드에서 웹 사이트에는 3개의 슬롯(페이지 레이아웃에 따라 페이지에서 사용 가능한 영역)이 있습니다. 페이지의 각 + 위로 커서를 가져가면 머리글, 본문, 바닥글과 같은 슬롯이 표시됩니다.




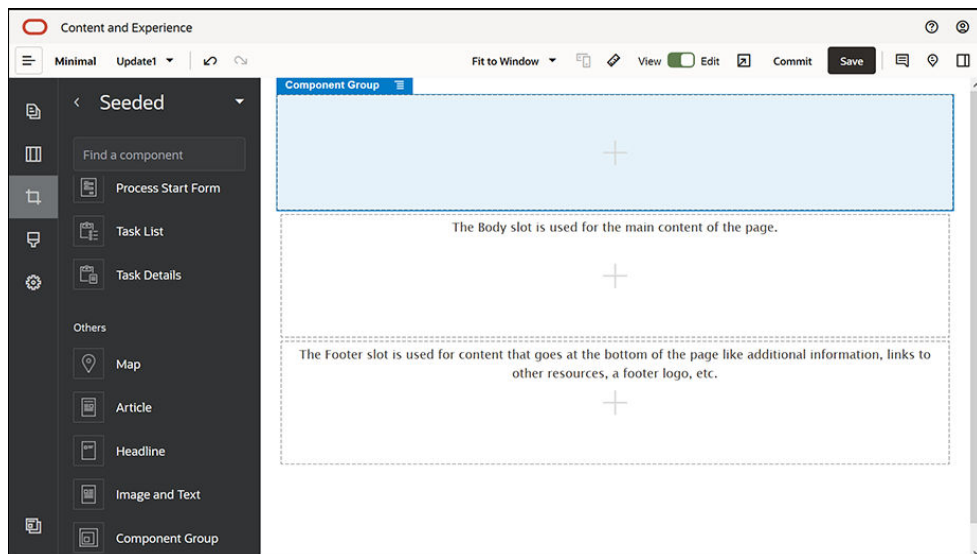
일반적으로 머리글 슬롯을 사용하여 회사 로고, 탐색 메뉴 등을 표시합니다. 본문 슬롯은 페이지의 주 콘텐츠에 사용하고 바닥글 슬롯은 저작권 정보, 소셜 미디어 링크 및 추가 정보에 사용합니다.


먼저 홈 페이지를 구축하겠습니다. 홈 페이지가 완성되면 다음과 비슷한 모양이 됩니다.

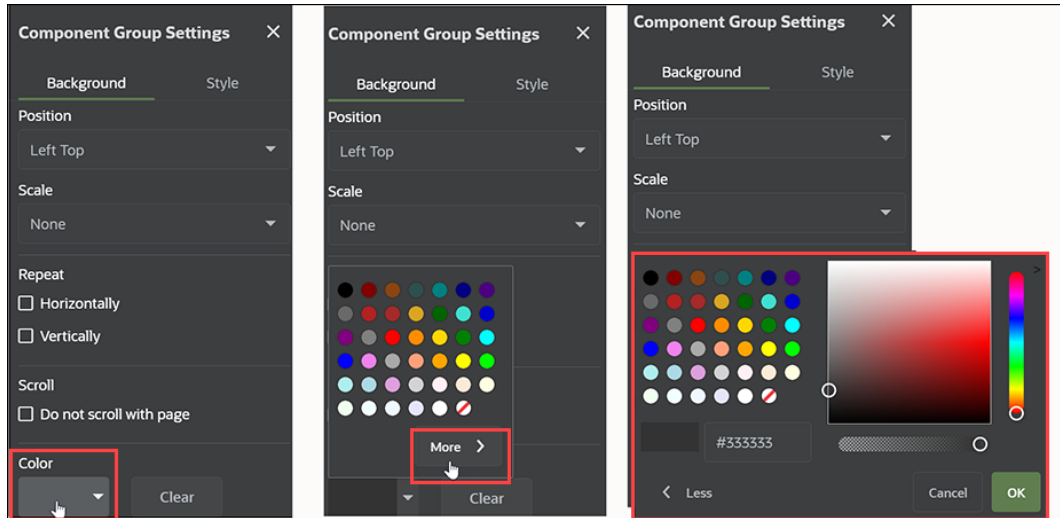


미리 정의된 구성요소를 사용하여 머리글 슬롯을 채우겠습니다.

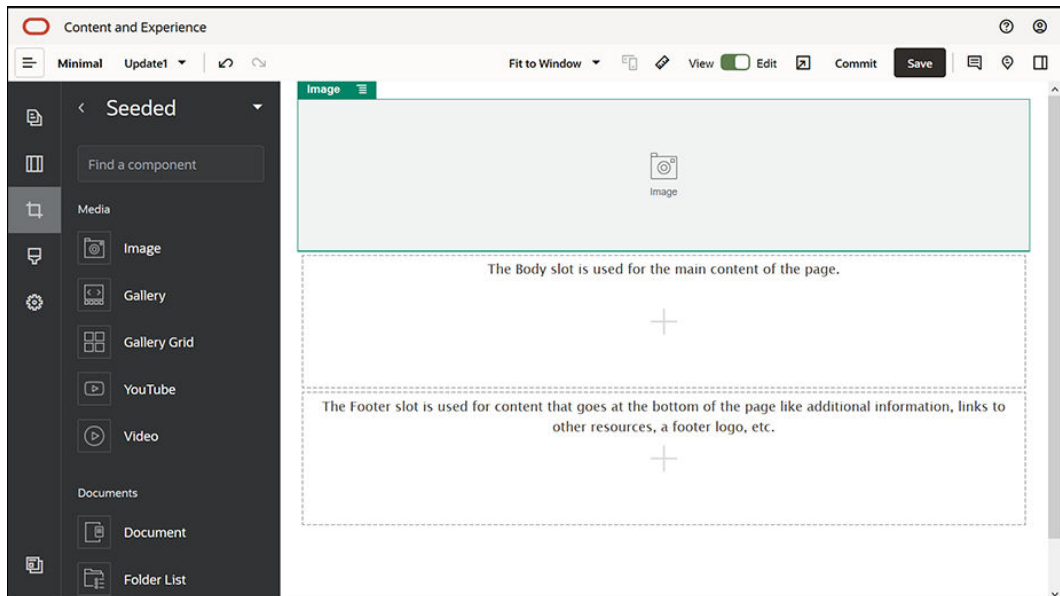
1. 왼쪽 사이드바에서  을 누르고 시드를 눌러 Oracle Content Management와 함께 제공된 미리 정의된 구성요소 목록을 표시합니다.
2. 왼쪽 사이드바에서 '구성요소 그룹'이라는 미리 정의된 구성요소를 찾습니다. 머리글 슬롯으로 끌어 놓습니다.




3. 구성요소 그룹의 메뉴 아이콘  을 누르고 설정을 누릅니다. 설정에서 색상 드롭다운 목록 (설정 목록 아래쪽에서 사용 가능)을 누르고 자세히를 누릅니다. #333333을 입력하고 확인을 누릅니다.



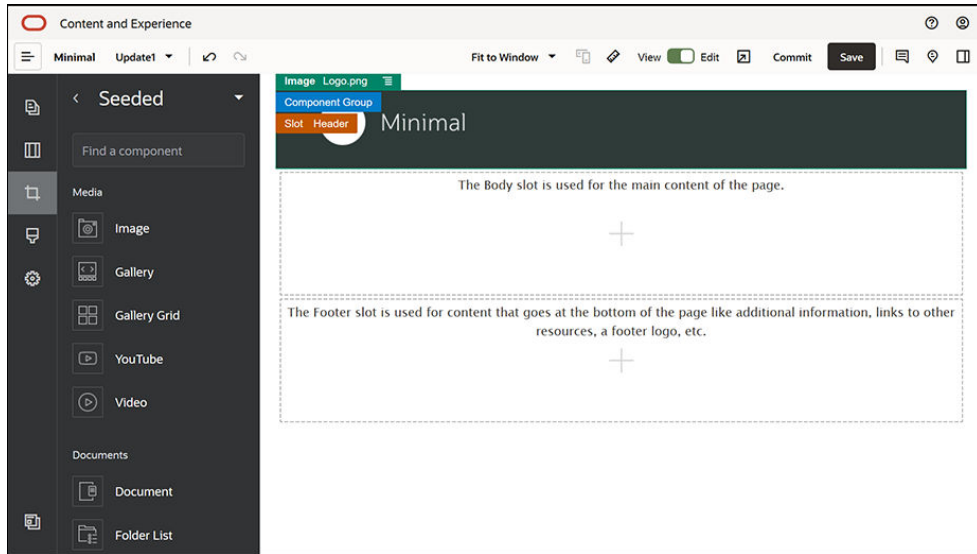
4. 이미지 구성요소를 구성요소 그룹으로 끌어 놓습니다.



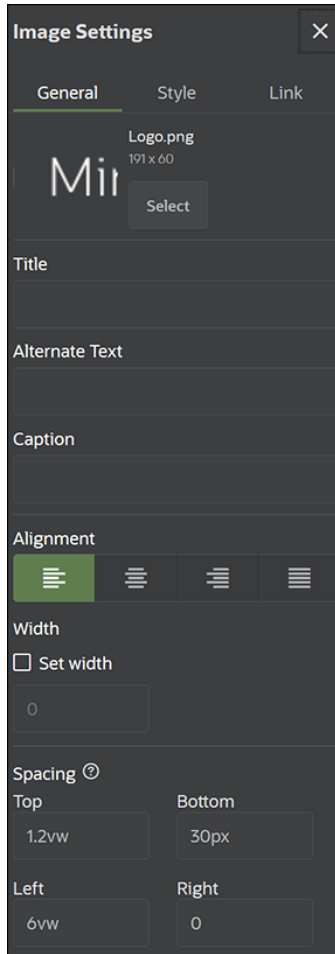
5. 이미지 구성요소의 메뉴 아이콘  을 누르고 설정을 누릅니다. 일반 사항 탭에서 설정을 완료합니다.

속성	값
선택	Minimal-Images 폴더의 Logo.png
맞춤	왼쪽
너비	너비 설정 선택 해제

속성	값
위쪽	1.2vw
아래쪽	30px
왼쪽	6vw
오른쪽	0




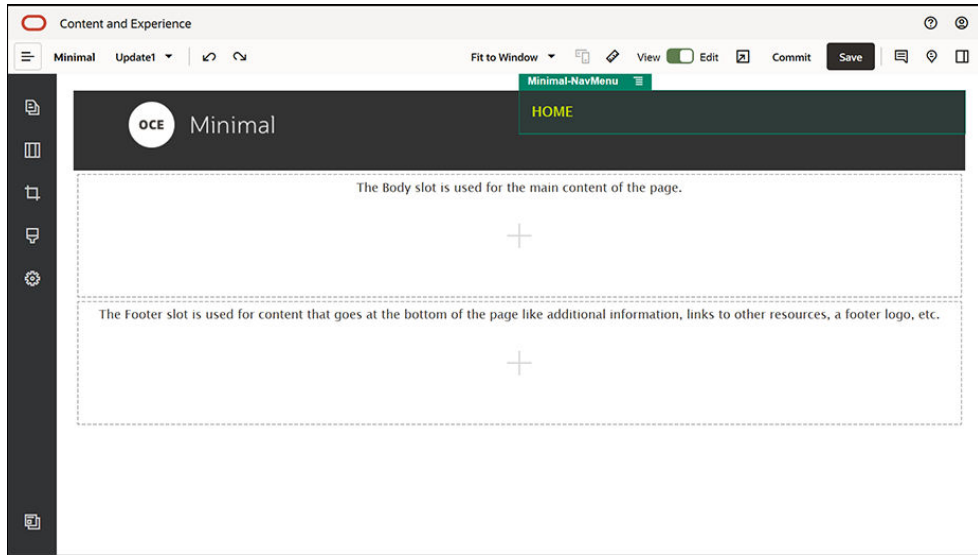




6. 이 로고 이미지를 홈 페이지에 링크하겠습니다. 링크 탭에서 이미지 구성요소의 설정을 완료합니다.

속성	값
링크 유형 선택	사이트 페이지
페이지	HOME
대상	동일한 창에서 열기

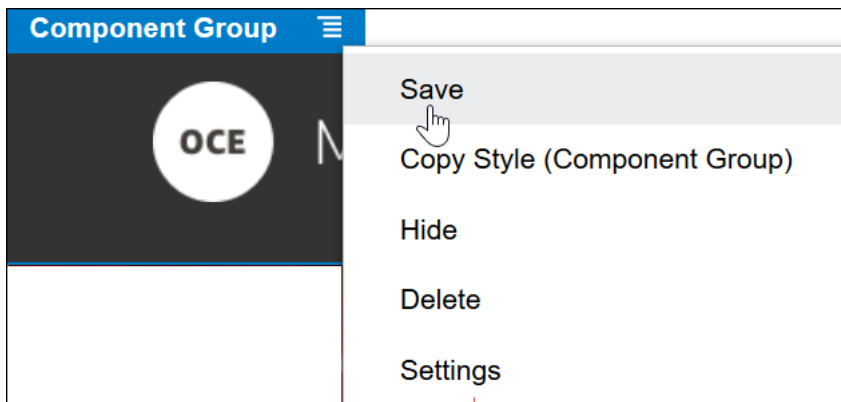
7. 왼쪽 사이드바에서  을 누르고 **사용자정의**를 눌러 사용자정의 구성요소 목록을 표시합니다.
8. 이제 Minimal-NavMenu 사용자정의 구성요소를 사용하여 홈 페이지에 탐색 메뉴를 추가하겠습니다. Minimal-NavMenu 구성요소를 이미지 구성요소의 오른쪽에 구성요소 그룹으로 끌어 놓습니다. Minimal-NavMenu 구성요소의 제목을 눌러 그 순위가 이전에 추가한 구성요소 그룹인지 확인합니다. 이것은 웹 페이지 구조 내에서 구성요소가 어디에 있는지 확인하는 유용한 방법입니다.



일반 사항 탭에서 설정을 완료합니다.

속성	값
맞춤	오른쪽
위쪽	1.2vw
아래쪽	0
왼쪽	0
오른쪽	6vw

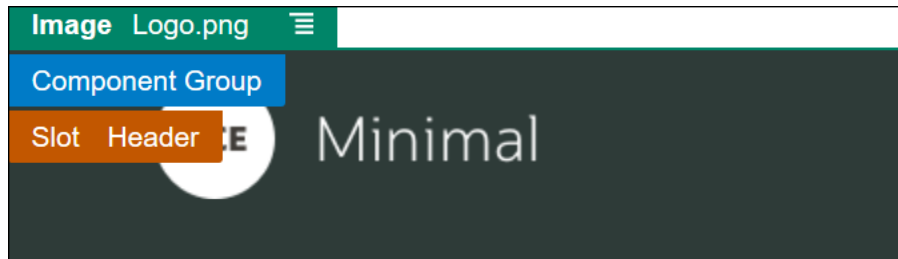
9. 이제 머리글이 준비되었습니다. 이 구성요소 그룹을 사용자정의 구성요소 그룹으로 저장하여 나중에 다른 웹 사이트 페이지에서 사용할 수 있도록 하겠습니까. 구성요소 그룹의 제목을 누르고 메뉴 아이콘 을 누른 다음 **저장**을 누릅니다. [구성요소 그룹 저장] 대화상자에서 **이름** 필드에 "Minimal-Header"를 입력하고 **저장**을 누릅니다.



**팁:**

구성요소 그룹을 누를 때 구성요소 그룹 제목이 아닌 이미지 또는 Minimal-NavMenu 구성요소 제목이 표시될 경우, 이미지 또는 Minimal-NavMenu 구성요소 제목을 다시 누르면 구성요소 그룹의 제목이 표시됩니다. 이제 구성요소 그룹의 제목을 누르고 위의 단계를 수행할 수 있습니다.



다음 이미지는 머리글 슬롯에서 이미지 구성요소의 상위 구조를 보여줍니다.

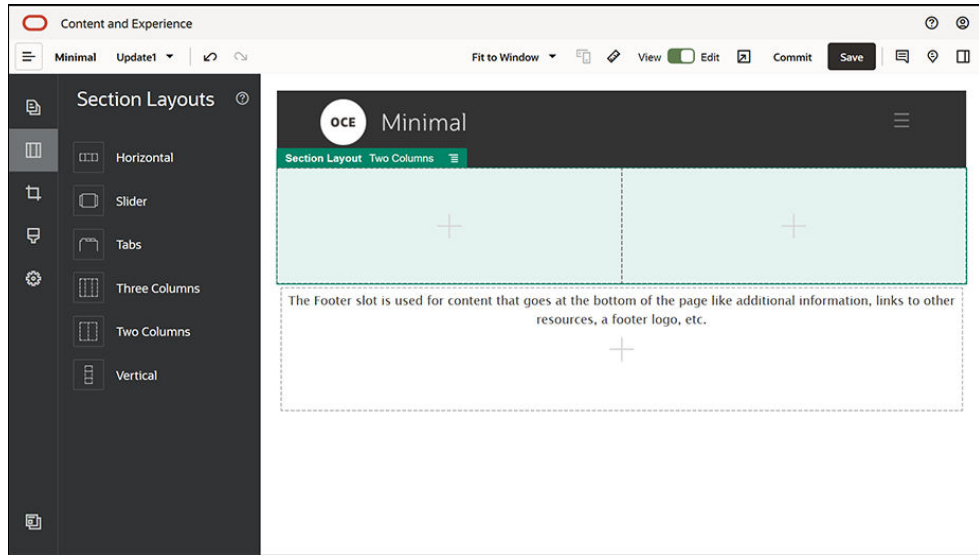


10. 사이트 작성기의 오른쪽 상단에서 **저장**을 눌러 변경사항을 저장합니다. 이제 머리글 슬롯은 다음 이미지처럼 보입니다.



본문 슬롯으로 넘어가겠습니다.

1. 왼쪽 사이드바에서 을 누르고 **시드**를 누릅니다.
2. 왼쪽 사이드바에서 구성요소 그룹을 본문 슬롯으로 끌어 놓습니다. 이 구성요소 그룹과 (그룹에 추가할) 구성요소를 사용하여 배너를 생성할 것입니다.
3. 왼쪽 사이드바에서 을 누릅니다.
4. 2열 섹션 레이아웃을 구성요소 그룹으로 끌어 놓습니다.




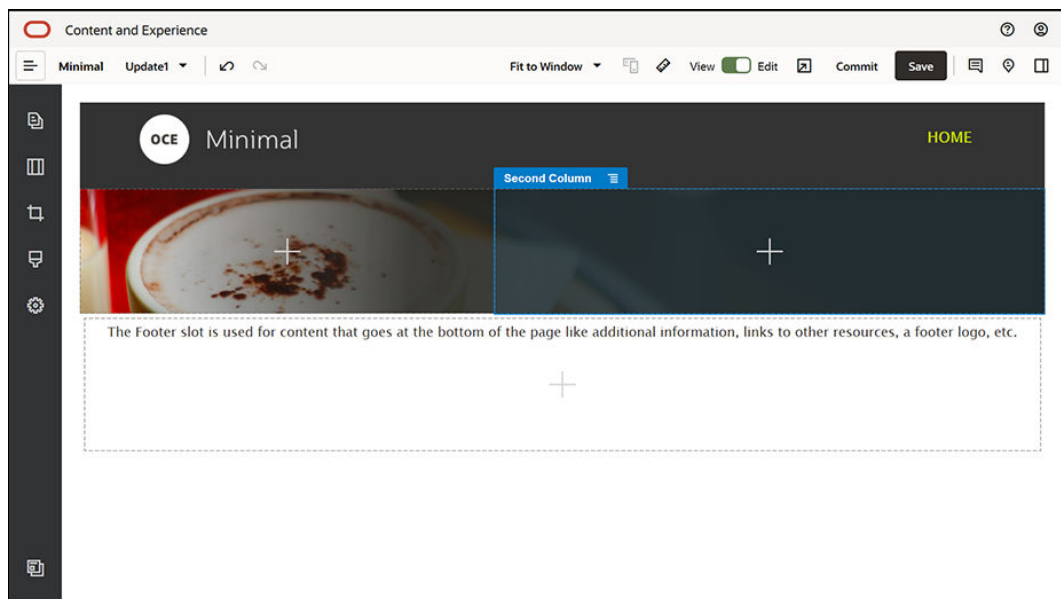
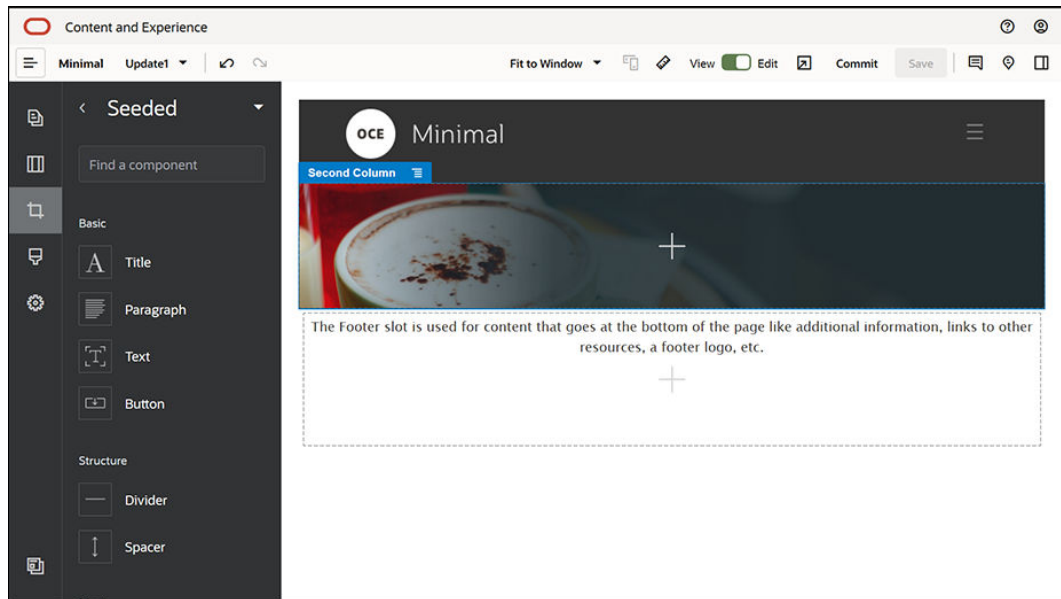
5. 일반 사항 탭에서 섹션 레이아웃의 설정을 완료합니다. 사용자정의 설정을 눌러 다음 설정을 지정합니다.

속성	값
첫번째 열 너비(%)	43
두번째 열 너비(%)	57
반응형 중단점(픽셀)	1,023
반응형 동작	첫번째 열 숨기기

배경 탭에서 설정을 완료합니다.

속성	값
이미지	Minimal-Images 폴더의 Banner1.jpg
위치	가운데 중간
배율	늘이기

6. 왼쪽 사이드바에서  을 눌러 시드된 구성요소 목록을 확인합니다.
7. 시드된 구성요소 목록에서 제목 구성요소를 2열 레이아웃의 두번째 열로 끌어 놓습니다.



- 제목 구성요소 안을 누르고 "WELCOME TO THE REVOLUTION"을 입력합니다. 텍스트를 선택하고 텍스트 편집기에서 글꼴 색상을 흰색으로 설정합니다. 일반 사항 탭에서 제목 구성요소의 설정을 완료합니다.

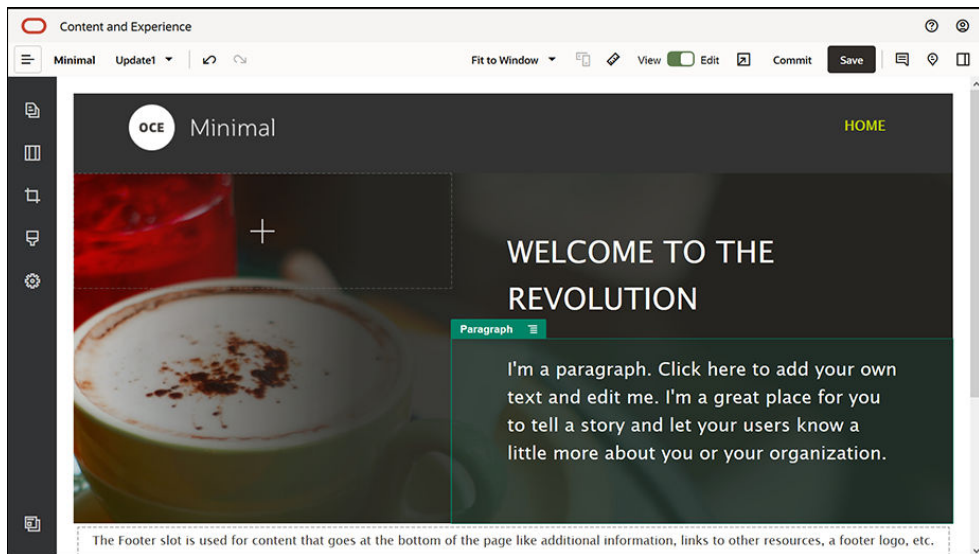
속성	값
위쪽	6vw
아래쪽	1.8vw
왼쪽	6vw
오른쪽	6vw


- 왼쪽 사이드바의 시드된 구성요소 목록에서 단락 구성요소를 2열 레이아웃의 두번째 열 안의 제목 구성요소 아래에 끌어 놓습니다. 일반 사항 탭에서 설정을 완료합니다.

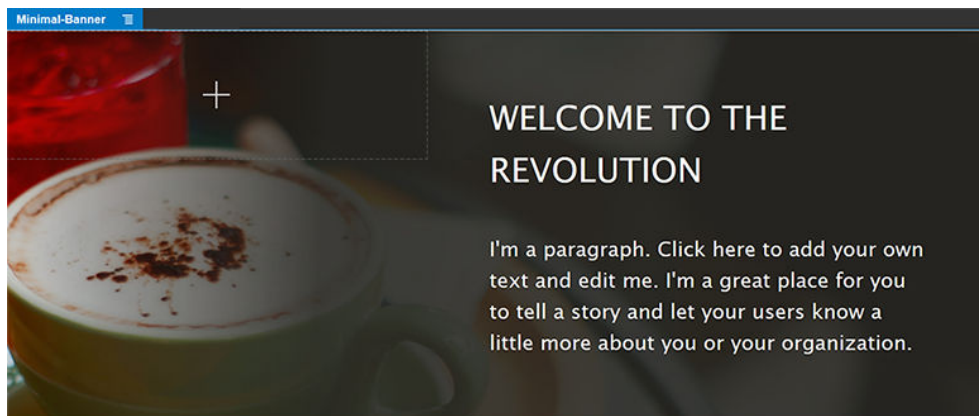
속성	값
위쪽	1.8vw
아래쪽	6vw
왼쪽	6vw
오른쪽	6vw

10. 단락 구성요소 안을 눌러 다음 텍스트를 입력합니다.  
"I'm a paragraph. Click here to add your own text and edit me. I'm a great place for you to tell a story and let your users know a little more about you or your organization."

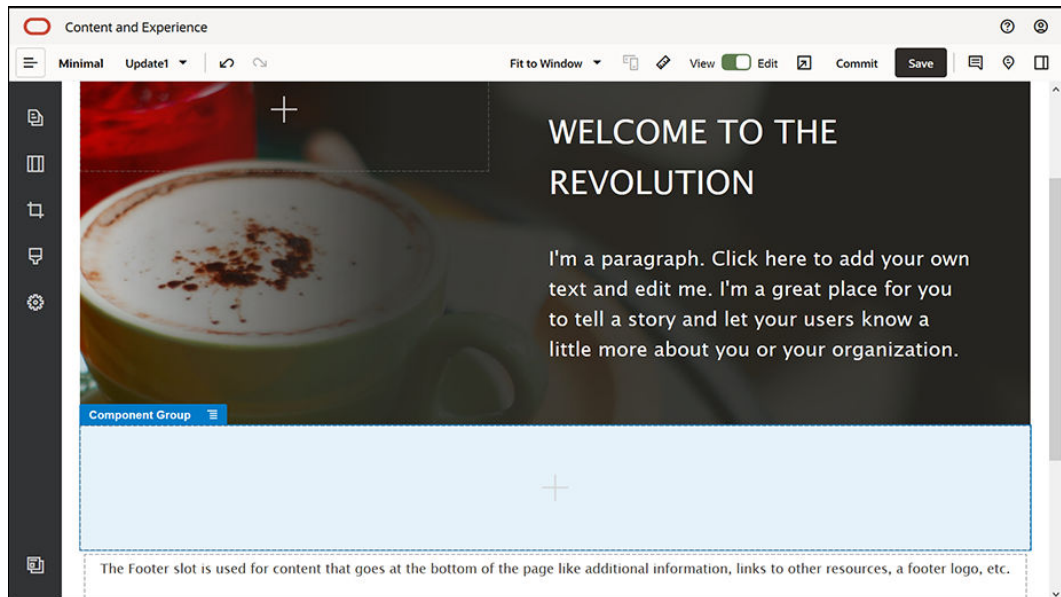
텍스트를 선택하고 텍스트 편집기에서 크기를 24로 설정합니다. 또한 텍스트 편집기에서 글꼴 색상을 흰색으로 설정합니다.



11. 이제 배너가 준비되었습니다. 이 구성요소 그룹을 사용자정의 구성요소 그룹으로 저장하여 나중에 다른 웹 사이트 페이지에서 사용할 수 있도록 하겠습니까. 구성요소 그룹의 메뉴 아이콘  을 누르고 **저장**을 누릅니다. [구성요소 그룹 저장] 대화상자에서 이름 필드에 "Minimal-Banner"를 입력하고 **저장**을 누릅니다. 이제 구성요소 그룹에 이름(**Minimal-Banner**)이 나타나는 것을 알 수 있습니다.

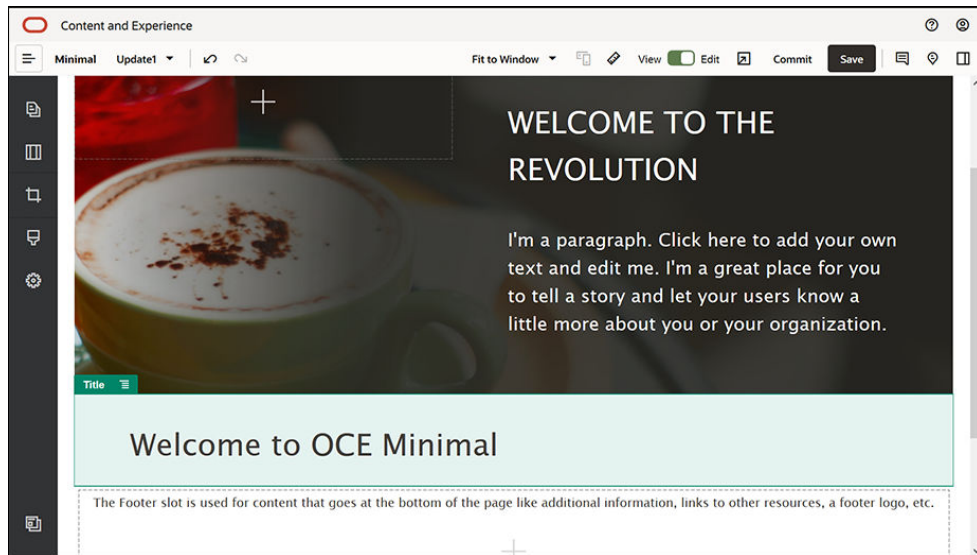


12. 왼쪽 사이드바에서 또 다른 구성요소 그룹을 이미 추가한 Minimal-Banner 구성요소 그룹 아래에 본문 슬롯으로 끌어 놓습니다.



13. 제목 구성요소를 구성요소 그룹으로 끌어 놓습니다.
14. 제목 구성요소 안을 누르고 "Welcome to OCE Minimal"을 입력합니다.
15. 일반 사항 탭에서 제목 구성요소의 설정을 완료합니다.

속성	값
위쪽	3vw
아래쪽	1.8vw
왼쪽	6vw
오른쪽	6vw



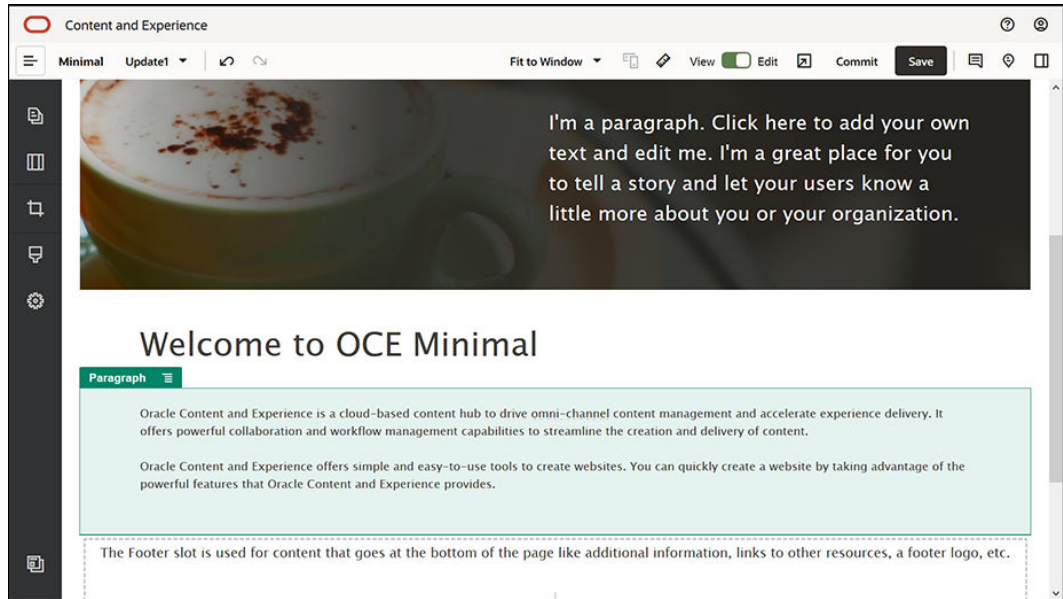
16. 왼쪽 사이드바에서 단락 구성요소를 제목 구성요소 아래에 구성요소 그룹으로 끌어 놓습니다. 단락 구성요소 안을 눌러 다음 텍스트를 입력합니다.  
"Oracle Content Management is a cloud-based content hub to drive omni-channel content management and accelerate experience delivery. It offers powerful collaboration and workflow management capabilities to streamline the creation and delivery of content."


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

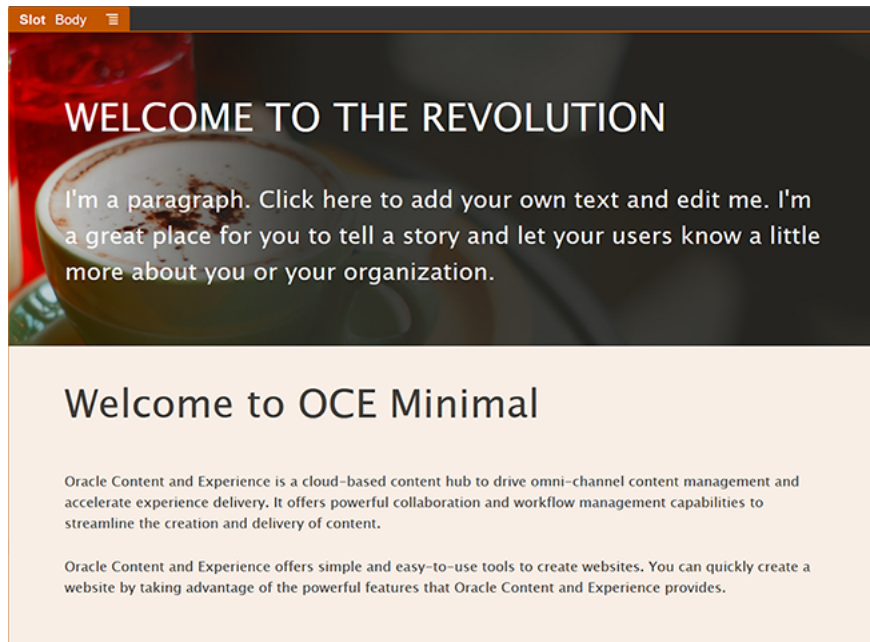
17. 일반 사항 탭에서 단락 구성요소의 설정을 완료합니다.

속성	값
위쪽	20px
아래쪽	50px
왼쪽	6vw
오른쪽	6vw



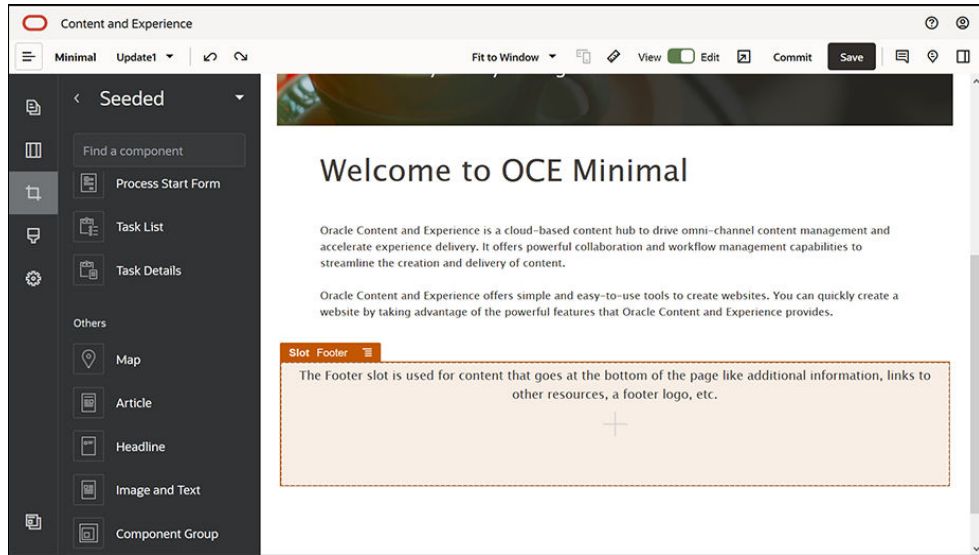


18. 본문 슬롯을 완성했습니다. 구성요소 그룹을 사용자정의 구성요소 그룹으로 저장하여 나중에 다른 웹 사이트 페이지에서 사용할 수 있도록 하겠습니다. 구성요소 그룹의 메뉴 아이콘  을 누르고 **저장**을 누릅니다. [구성요소 그룹 저장] 대화상자에서 **이름** 필드에 "Minimal-Body"를 입력하고 **저장**을 누릅니다.
19. 사이트 작성기의 오른쪽 상단에서 **저장**을 눌러 변경사항을 저장합니다. 이제 본문 슬롯은 다음 이미지처럼 보입니다.



바닥글 슬롯을 완성하겠습니다.

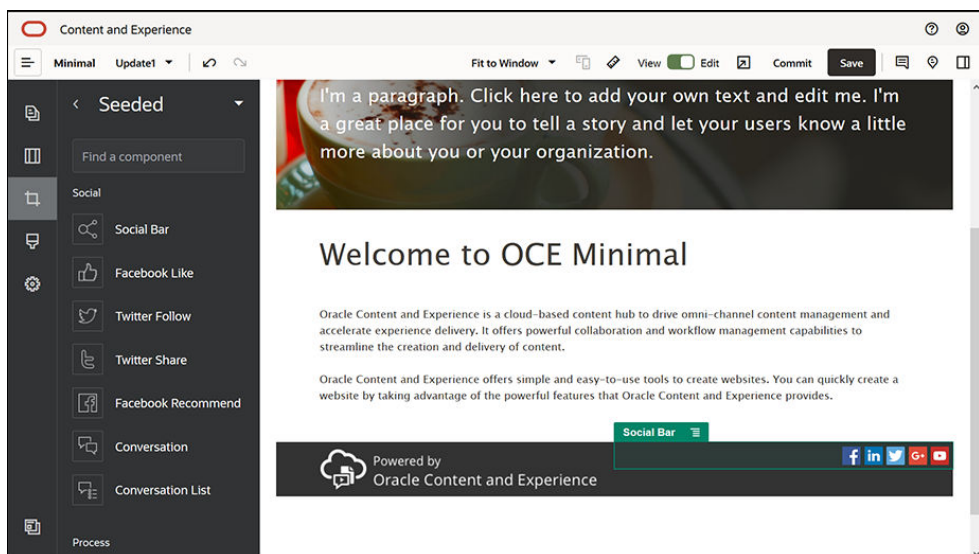
1. 왼쪽 사이드바에서 구성요소 그룹을 바닥글 슬롯으로 끌어 놓습니다. 구성요소 그룹의 설정에서 **색상** 필드를 #333333으로 설정합니다.



2. 이미지 구성요소를 구성요소 그룹으로 끌어 놓고 **일반 사항** 탭에서 설정을 완료합니다.

속성	값
선택	Minimal-Images 폴더의 Powered_by_OCE.png
맞춤	왼쪽
너비	너비 설정 선택 해제
위쪽	0.9vw
아래쪽	0.9vw
왼쪽	6vw
오른쪽	0

3. 왼쪽 사이드바에서 소셜 표시줄 구성요소를 이미지 구성요소의 오른쪽에 구성요소 그룹으로 끌어 놓습니다.




**일반 사항** 탭에서 소셜 표시줄 구성요소의 설정을 완료합니다.

속성	값
위쪽	1.8vw
아래쪽	1.8vw
왼쪽	0.3vw
오른쪽	6vw


일반 사항 탭에서 **아이콘**을 누르고 아이콘 이름을 눌러 설정을 완료합니다.

속성	값
URL	<ul style="list-style-type: none"> <li>• <a href="https://www.facebook.com/Oracle/">https://www.facebook.com/Oracle/</a> (Facebook용)</li> <li>• <a href="https://www.linkedin.com/company/oracle/">https://www.linkedin.com/company/oracle/</a> (LinkedIn용)</li> <li>• <a href="https://twitter.com/Oracle">https://twitter.com/Oracle</a> (Twitter용)</li> <li>• <a href="https://www.youtube.com/oracle/">https://www.youtube.com/oracle/</a> (YouTube용)</li> </ul>
대상	새 창에서 열기

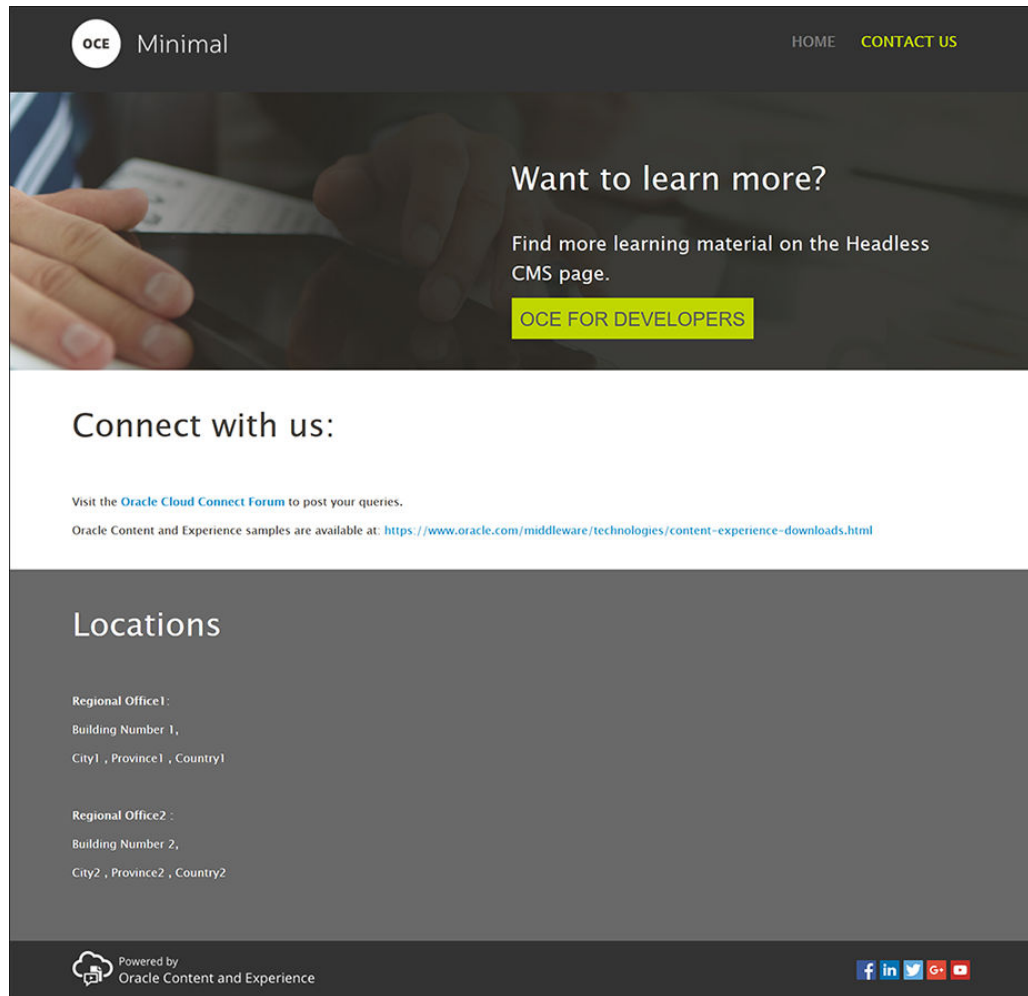
- 이제 바닥글이 준비되었습니다. 이 구성요소 그룹을 사용자정의 구성요소 그룹으로 저장하여 나중에 다른 웹 사이트 페이지에서 사용할 수 있도록 하겠습니까. 구성요소 그룹의 메뉴 아이콘 을 누르고 **저장**을 누릅니다. [구성요소 그룹 저장] 대화상자에서 **이름** 필드에 "Minimal-Footer"를 입력하고 **저장**을 누릅니다. 구성요소 그룹은 다음처럼 보입니다.





- 사이트 작성기의 오른쪽 상단에서 **저장**을 눌러 변경사항을 저장합니다.

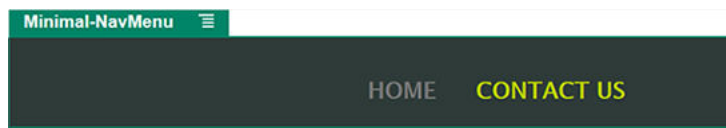
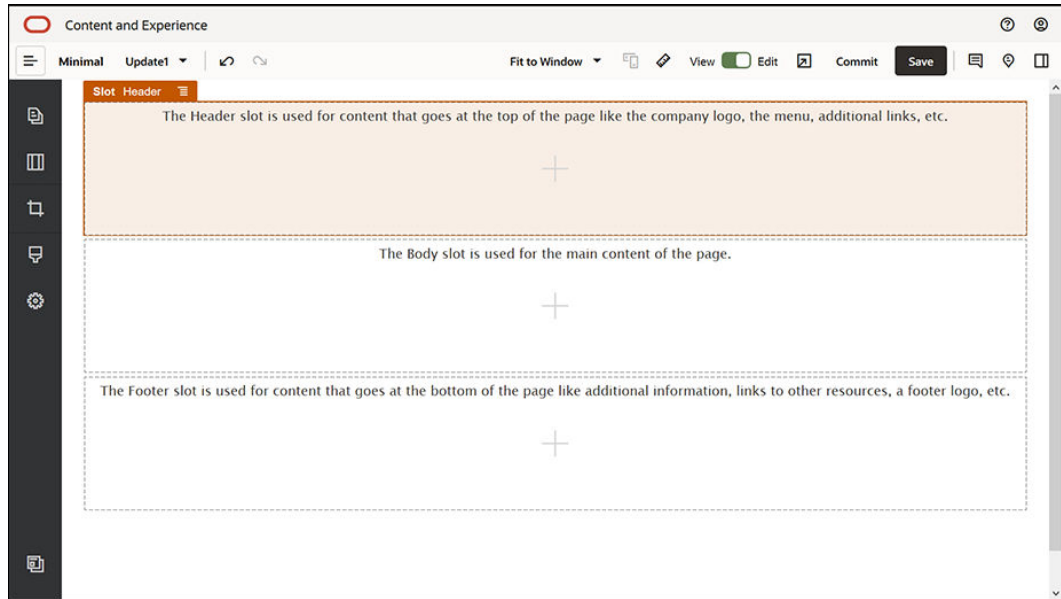
사이트 작성기의 오른쪽 상단에서 을 눌러 웹 사이트의 첫번째 페이지를 미리 봅니다. 아직 웹 사이트가 게시되지 않아서 다른 사람은 지금 볼 수 없습니다.

HOME 페이지 생성을 완성했습니다. CONTACT US 페이지를 구축하겠습니다. 연락처 페이지가 완성되면 다음과 비슷한 모양이 됩니다.

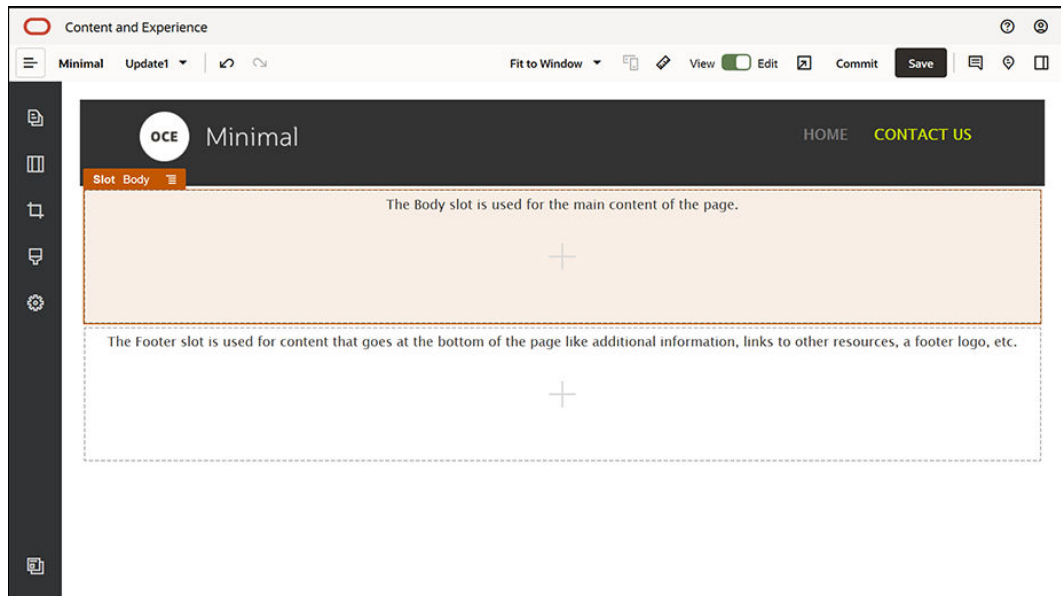


다양한 슬롯에 구성요소를 추가하겠습니다.

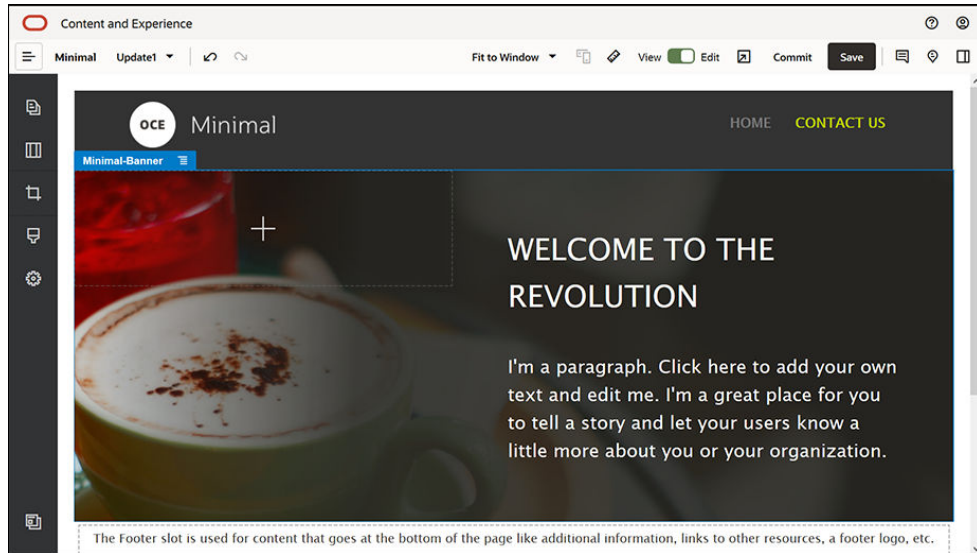
1. 왼쪽 사이드바에서  을 누르고 **페이지 추가**를 누릅니다.
2. **페이지 이름** 필드에 "CONTACT US"를 입력하고 **닫기**를 누릅니다. 새 페이지를 웹 사이트에 추가했습니다.
3. 왼쪽 사이드바에서  을 누르고 **사용자정의**를 누릅니다.
4. Minimal-Header 구성요소(이전에 생성하고 저장함)를 머리글 슬롯으로 끌어 놓습니다.



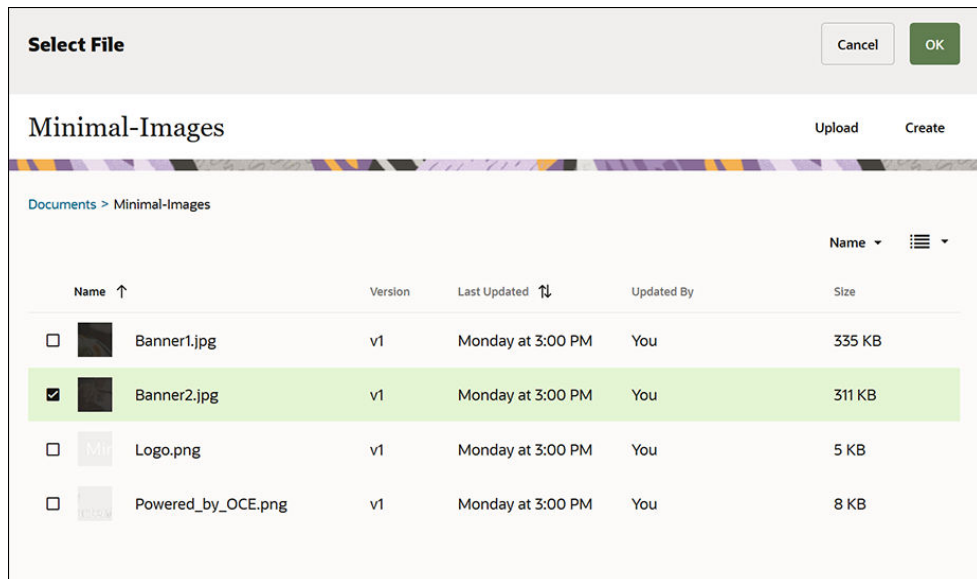
Minimal-NavMenu 구성요소가 방금 생성한 새 CONTACT US 페이지를 자동으로 선택한 것을 알 수 있습니다.




5. 왼쪽 사이드바에서 Minimal-Banner 구성요소를 본문 슬롯으로 끌어 놓습니다.



6. HOME 페이지의 배너와 다르게 CONTACT US 페이지에 맞게 배너를 수정하겠습니다. 구성요소 그룹 내에서 2열 섹션 레이아웃의 설정을 수정합니다. 배경 탭의 이미지 필드에서 이미지 선택을 누르고 문서에 이전에 생성한 **Minimal-Images** 폴더에서 **Banner2.jpg**를 선택합니다.



7. 2열 섹션 레이아웃 내에서 제목 및 단락 구성요소의 텍스트를 수정합니다.
  - 제목 구성요소에 "Want to learn more?"를 입력합니다.
  - 단락 구성요소에 "Find more learning material on the Headless CMS page."를 입력합니다.
8. 왼쪽 사이드바에서  을 누르고 시드를 누릅니다.
9. CONTACT US 페이지의 이미지는 배너의 일부로 **OCE FOR DEVELOPERS**라는 단추가 있으므로 이 단추를 배너에 추가하겠습니다. Minimal-Banner 구성요소 그룹

안의 단추 구성요소를 2열 섹션 레이아웃의 두번째 열(최근 추가된 단락 구성요소 아래)로 끌어 놓습니다. **일반 사항** 탭에서 단추 구성요소의 설정을 완료합니다.

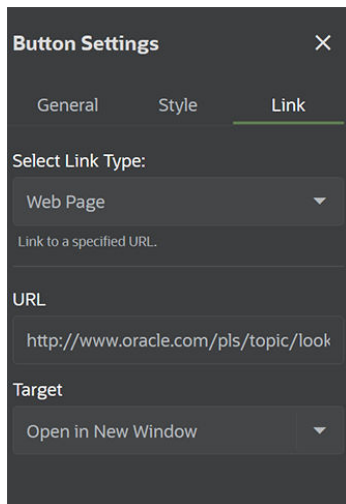
속성	값
레이블	OCE FOR DEVELOPERS
위쪽	0.3vw
아래쪽	3vw
왼쪽	6vw
오른쪽	0.3vw

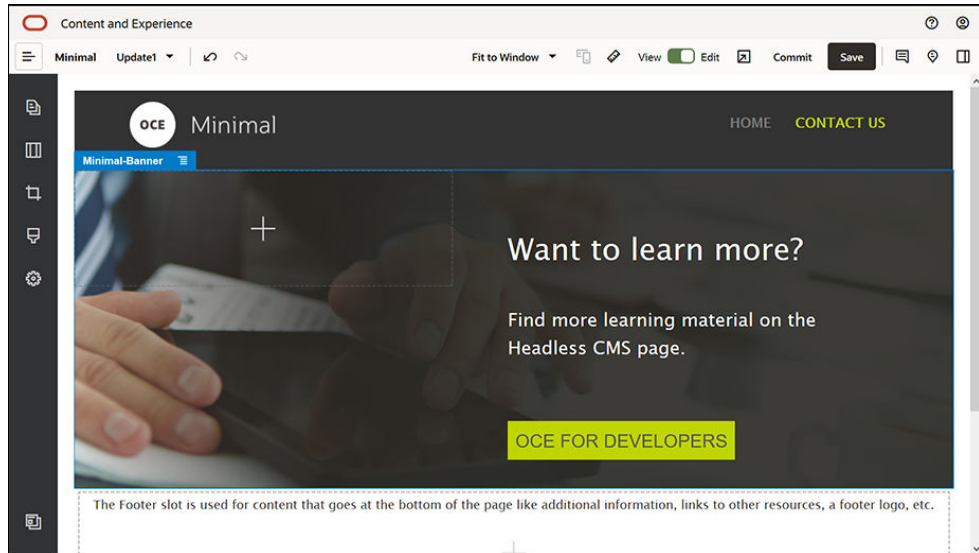
**스타일** 탭에서 **사용자정의**를 선택하고 설정을 완료합니다.


속성	값
배경색	#c0d600
글꼴	<ul style="list-style-type: none"> <li>크기로 24를 입력합니다.</li> <li>#58595b 색상을 입력합니다.</li> </ul>
테두리	없음
커서를 위로 가져갈 때 표시되는 색상	<ul style="list-style-type: none"> <li><b>BACKGROUND</b>를 #e1fa00으로 설정합니다.</li> <li><b>FONT</b>를 #58595b로 설정합니다.</li> <li><b>BORDER</b>를 #2222dd로 설정합니다.</li> </ul>
모서리	0

**링크** 탭에서 설정을 완료합니다.

속성	값
링크 유형 선택	웹 페이지
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>
대상	새 창에서 열기






10. 왼쪽 사이드바에서  을 누르고 사용자정의를 누릅니다.
11. Minimal-Body 구성요소를 Minimal-Banner 구성요소 아래에 본문 슬롯으로 끌어 놓습니다.
12. Minimal-Body 구성요소 내에서 제목 및 단락 구성요소의 텍스트를 수정합니다.
  - a. 제목 구성요소에 "Connect with us:"를 입력합니다.
  - b. 단락 구성요소에 전자메일 주소 및 기타 지원 관련 링크와 같은 세부정보를 입력합니다.  
 "Visit the [Oracle Cloud Connect Forum](#) to post your queries."  
 "Oracle Content Management samples are available at: <https://www.oracle.com/middleware/technologies/content-experience-downloads.html>"
13. 왼쪽 사이드바에서 또 다른 Minimal-Body 구성요소를 이전에 추가한 Minimal-Body 구성요소 아래에 본문 슬롯으로 끌어 놓습니다.
14. 구성요소 그룹의 설정에서 배경 탭에서 색상 필드를 #696969로 설정합니다.
15. Minimal-Body 구성요소 내에서 제목 및 단락 구성요소를 수정합니다.
  - a. 제목 구성요소에 "Locations"를 입력합니다. 텍스트를 선택하고 텍스트 편집기에서 글꼴 색상을 흰색으로 설정합니다.
  - b. 단락 구성요소에 다음 텍스트를 입력합니다.  
 "Regional Office1:  
 Building Number 1,  
 City1, Province1, Country1  
 Regional Office2:  
 Building Number 2,  
 City2, Province2, Country2"  
 텍스트를 선택하고 텍스트 편집기에서 글꼴 색상을 흰색으로 설정합니다.
16. 이제 CONTACT US 페이지에 바닥글을 추가하겠습니다. Minimal-Footer 구성요소를 바닥글 슬롯으로 끌어 놓습니다.



17. 사이트 작성기의 오른쪽 상단에서 **저장**을 눌러 변경사항을 저장합니다.

사이트 작성기의 오른쪽 상단에서 을 눌러 웹 사이트를 미리 보고 모든 사항이 괜찮은지 확인합니다. CONTACT US 페이지의 로고 이미지를 누르면 홈 페이지로 돌아가는지 확인합니다. 웹 사이트 페이지 간의 탐색이 올바르게 작동하는지 메뉴를 테스트합니다.

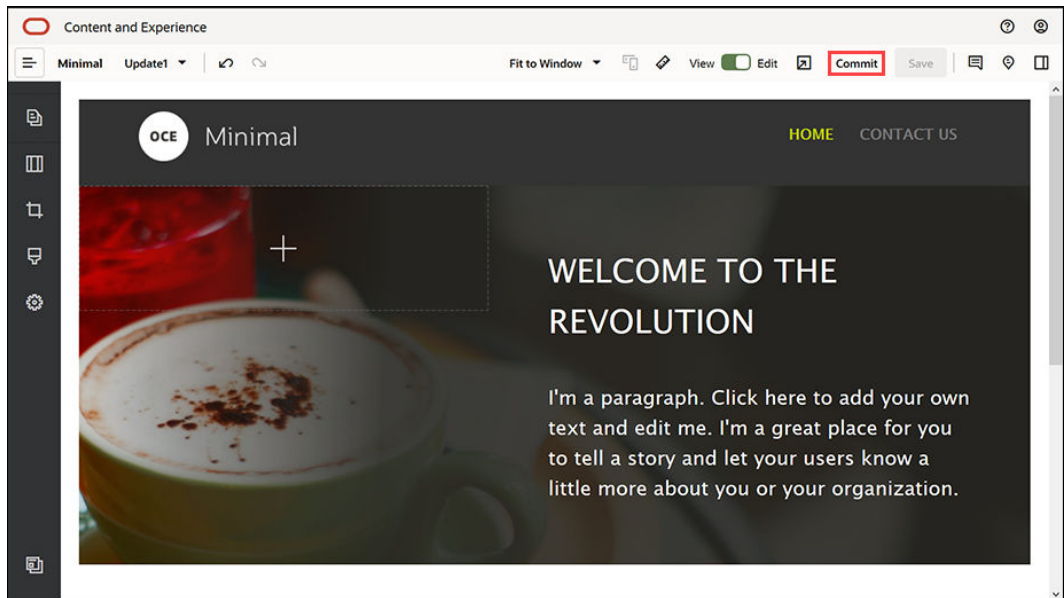
웹 사이트를 게시할 준비가 되었습니다.

다음 단계: 웹 사이트 게시

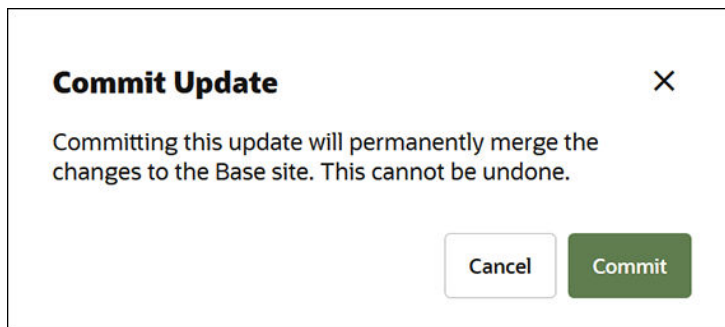
## 단계 3: 웹 사이트 게시

이제 웹 사이트를 성공적으로 생성했으므로 사이트를 게시하여 사용자가 볼 수 있도록 온라인으로 제공할 수 있습니다.

1. 모든 사항이 괜찮으면 사이트 작성기의 오른쪽 상단에서 **커밋**을 눌러 기본 웹 사이트에 변경사항을 커밋할 수 있습니다.

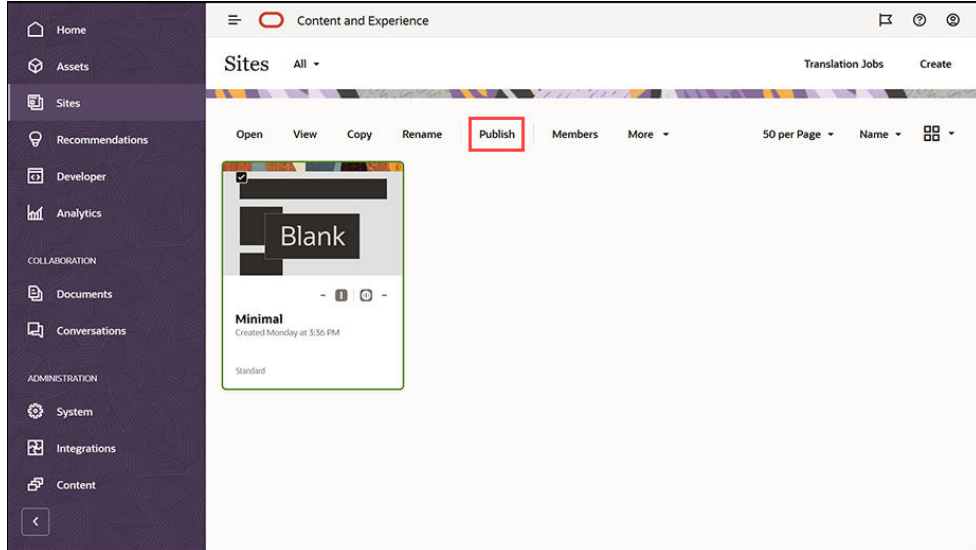


2. [업데이트 커밋] 대화상자에서 커밋을 누릅니다.

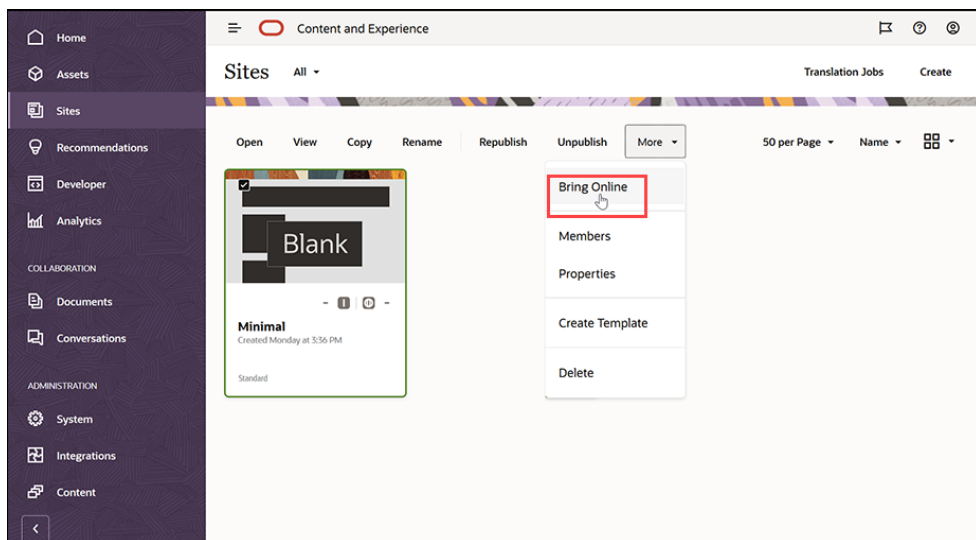


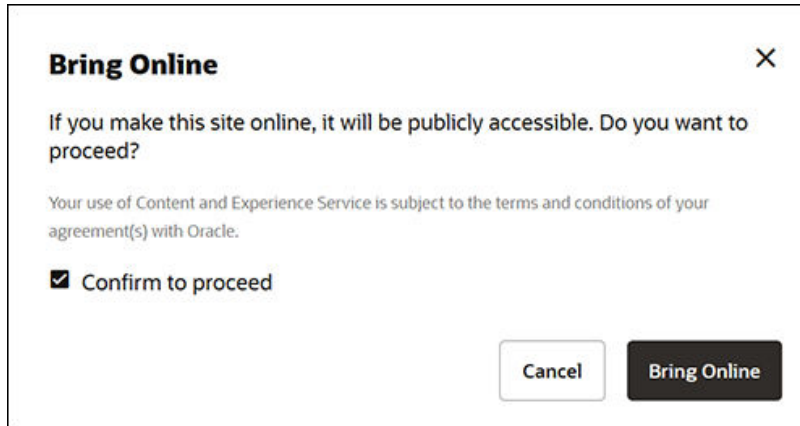
3. 변경사항이 커밋되면 웹 사이트를 게시할 준비가 되었습니다.

4. 사이트 작성기를 닫습니다.
5. Oracle Content Management 웹 인터페이스의 왼쪽 탐색 메뉴에서 **사이트**를 누르고 웹 사이트를 선택합니다.
6. 메뉴 모음 또는 마우스 오른쪽 단추 누르기 메뉴에서 **게시**를 선택합니다.



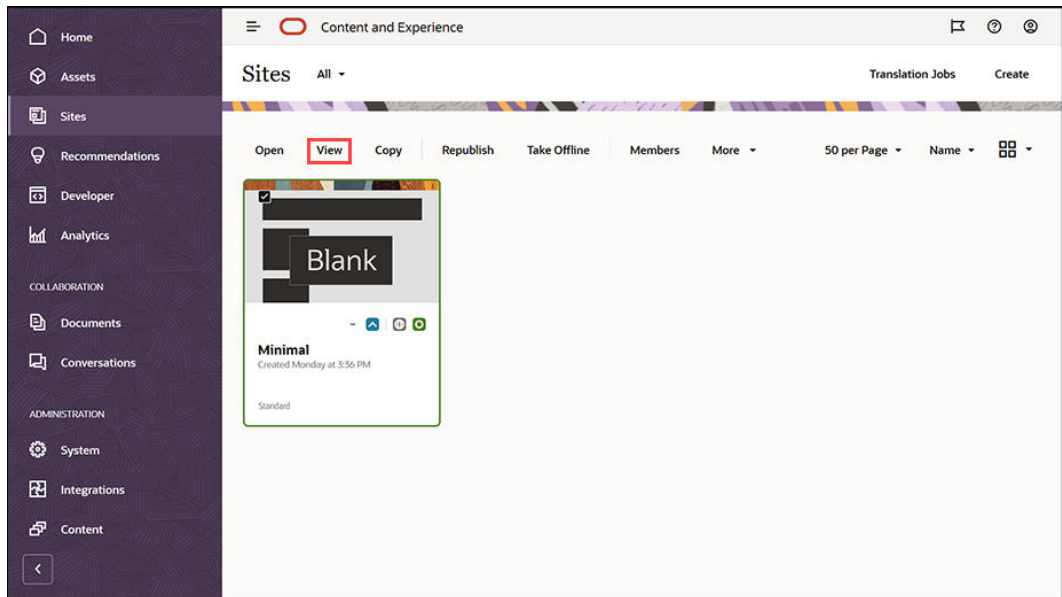
7. 웹 사이트가 게시되면 페이지 상단에 통지가 표시되어 웹 사이트가 게시되었음을 알립니다.
8. [사이트] 페이지에서 웹 사이트를 다시 선택하고 메뉴 모음 또는 마우스 오른쪽 단추 누르기 메뉴에서 **온라인으로 전환**을 선택합니다. [온라인으로 전환] 대화상자에서 **진행 확인**을 선택하고 **온라인으로 전환**을 누릅니다.





9. 이러면 끝입니다. 웹 사이트가 온라인이고 다른 사람이 볼 수 있습니다.

공용 웹 사이트를 보려면 메뉴 모음 또는 마우스 오른쪽 단추 누르기 메뉴에서 **보기**를 선택합니다.



## 추가 작업

조직의 요구사항에 맞게 웹 사이트를 사용자정의할 수 있습니다. 다음은 시작을 도와주는 몇 가지 유용한 링크입니다.

- [스타일 및 서식 사용](#)
- [팁과 힌트 편집](#)
- [사이트 및 사이트 설정 관리](#)
- [사이트 파일 업로드](#)
- [페이지 추가](#)
- [페이지 이동](#)

- 페이지 콘텐츠 배열
- 페이지 설정 변경
- 페이지 레이아웃 변경
- 테이블 작업
- 검색 엔진 속성 설정

## II부

# 사이트 생성 및 편집

이 부분에서는 템플릿 및 테마를 사용하여 사이트 생성을 시작하고, 구성요소 및 레이아웃을 사용하여 사이트를 편집하여 콘텐츠를 구성/추가하는 방법을 자세히 설명합니다. 다음 장이 포함됩니다.

- [사이트 생성](#)
- [사이트 편집](#)
- [사이트에서 템플릿 및 테마 사용](#)
- [사용자정의 구성요소 및 레이아웃 관리](#)
- [사이트 페이지 작업](#)
- [페이지 콘텐츠 배열](#)
- [내장 구성요소 사용](#)

# 4

## 사이트 생성

적절한 권한을 가진 경우 누구나 Oracle Content Management를 사용하여 웹 사이트를 구축할 수 있습니다. 독점적 도구, 코드나 소프트웨어를 사용할 필요가 없습니다. 사용자 인터페이스는 직관적이고 친화적인 그래픽 인터페이스입니다.

사이트를 생성할 때 템플릿으로 시작합니다. 템플릿은 사이트 코드 프레임워크, 샘플 페이지와 콘텐츠로 구성된 기본 사이트, 스타일이 적용된 테마, 리소스(예: 이미지)는 물론 사용자정의 구성요소까지 사이트를 시작할 때 필요한 모든 것을 포함합니다. [사이트 생성 프로세스 이해](#)를 참조하십시오.

Oracle Content Management의 사이트 제어 기능은 비즈니스 사용자를 위한 사용 환경 제공을 단순화하고 가속화합니다. 또한 IT 부서는 전체 사용 환경의 수명 주기를 완전히 관리하고, 중앙 위치에서 사용 환경을 쉽게 제어 및 추적할 수 있습니다. 그 결과 회사에 필요한 새로운 사용 환경을 생성하고 유지하는 데 필요한 비용을 줄일 수 있습니다. 이러한 제어 기능은 Oracle Content Management에 이미 포함되어 있으므로, 관리자가 이를 사용으로 설정하기만 하면 됩니다. [Administering Oracle Content Management](#)의 사이트 및 자산 설정 구성을 참조하십시오.

- [사이트 생성 프로세스 이해](#)
- [사이트 생성](#)
- [사이트 복사](#)
- [사이트 요청 관리](#)
- [사이트 요청 세부정보 변경](#)
- [사이트 요청 정책 보기](#)

## 사이트 생성

사이트를 생성하려면 템플릿을 선택하고 사이트 이름을 지정하고 콘텐츠를 추가합니다. 제어가 사용으로 설정된 경우 콘텐츠를 추가하기 전에 사이트 요청을 제출한 다음 승인이 완료되면 콘텐츠를 추가해야 합니다.

사이트를 생성하려면 먼저 관리자가 사이트 생성을 사용으로 설정하고 템플릿을 하나 이상 제공해야 합니다. 사이트 페이지에 **생성** 옵션이 표시되지 않거나 사용 가능한 템플릿이 없을 경우 관리자에게 문의하십시오. [사이트 시작하기](#) 및 [사이트 제어 이해](#)를 참조하십시오.

자산 및 다국어 사이트의 사용을 가능하게 해주는 엔터프라이즈 사이트를 생성하는 경우 저장소, 지역화 정책 및 기본 언어와 사이트가 연관됩니다. 저장소 및 지역화 정책을 사용할 수 없는 경우 직접 생성하거나 콘텐츠 관리자에게 저장소 생성을 요청하십시오.

### 주:

Oracle Content Management Starter Edition을 통해서는 번역을 사용할 수 없으며 제어를 사용으로 설정할 수 없고 하나의 사이트만 생성할 수 있습니다. 전체 기능과 무제한 사이트가 필요하면 Oracle Content Management Premium Edition으로 업그레이드하십시오.

1. 사이트를 누릅니다.
2. 생성을 누릅니다.
3. 템플릿 선택 페이지에서 사이트에 대한 템플릿을 선택합니다.
  - 제어가 사용으로 설정된 경우 생성할 사이트가 표준 사이트(자산 저장소 또는 지역화 정책 사용 안함) 또는 엔터프라이즈 사이트(자산 저장소 및 지역화 정책 사용)인지 템플릿이 결정합니다.
4. 사이트 구성 페이지에서 사이트 생성 전 필요한 승인, 사이트에 필요한 최소 보안 레벨 및 사이트 페이지 이미지를 확인합니다. 특정 사람으로 승인이 제한된 경우 **승인자 표시**를 눌러 사이트 요청을 승인할 수 있는 사람을 확인합니다.
  - 제어가 사용으로 설정되지 않은 경우 생성할 사이트 유형에 대해 **표준** 또는 **엔터프라이즈**를 선택합니다.
  - 표준 사이트를 생성하는 경우 **다음**을 눌러 다음 페이지로 이동합니다.
  - 엔터프라이즈 사이트를 생성하는 경우 다음 단계를 수행하십시오.
    - a. 사이트의 콘텐츠 관리에 사용되는 기본 저장소를 선택합니다. 저장소 관리자인 경우 사용 가능한 저장소가 없거나 새 저장소를 사용하려면 **새 저장소 생성**을 선택하고 저장소 생성 단계를 따르십시오. 사이트에서 여러 저장소 사용에 대한 자세한 내용은 *Managing Assets with Oracle Content Management*의 사이트에 여러 저장소에 대한 액세스 권한 부여를 참조하십시오.
    - b. 사이트에 대한 지역화 정책을 선택합니다. 저장소 관리자인 경우 사용 가능한 정책이 없거나 새 정책을 사용하려면 **새 지역화 정책 생성**을 선택하고 정책 생성 단계를 따르십시오.
    - c. 템플릿의 지역화 정책에 설정된 기본 언어가 없으면 지금 하나를 선택합니다. 지역화 정책에 필요한 언어만 표시됩니다.
    - d. **다음**을 눌러 다음 페이지로 이동합니다.
5. 세부정보 추가 페이지에서 다음 정보를 입력합니다.
  - 사이트에 대한 이름을 입력합니다. 이 이름은 사이트 URL에 사용됩니다. 문자, 숫자, 밑줄(\_) 및 하이픈(-)을 사용할 수 있습니다. URL은 대소문자를 구분합니다. 공백을 입력하면 자동으로 하이픈으로 바뀝니다. 기존 사이트 이름을 대소문자가 다른 형태로 사용하지 마십시오. 예를 들어 ABC라는 사이트가 존재할 경우, Abc라는 다른 사이트를 생성하지 마십시오.

 **주:**

템플릿, 테마, 구성요소, 사이트 또는 사이트 페이지에 다음 이름을 사용하지 마십시오: authsite, content, pages, scstemplate\_\*, \_comps, \_components, \_compsdelivery, \_idcservice, \_sitescloud, \_sitesclouddelivery, \_themes, \_themesdelivery. 사이트 페이지에는 다음 이름을 사용할 수 있지만 템플릿, 테마, 구성요소 또는 사이트에는 사용하지 마십시오: documents, sites.

- 엔터프라이즈 사이트를 생성하는 경우 템플릿 정책에 따라 친화적 URL 값을 위해 기본 사이트 접두어 편집할 수 있습니다. 이 접두어는 콘텐츠 항목 슬러그 값 (페이지나 자산에 특정한 URL 부분)에 추가됩니다.
- 선택적으로 사이트에 대한 설명을 입력합니다.

- 선택적으로 이 사이트 요청에 대한 근거를 입력합니다. 그러면 사이트 관리자가 요청을 승인해야 할지 여부를 보다 쉽게 결정할 수 있습니다.

#### 6. 완료

사이트 요청에 승인이 필요한 경우 승인해야 할 사람에게 전자메일이 전송됩니다. 승인되면 승인이 완료되었으며 사이트가 생성될 것임을 알리는 전자메일이 전송됩니다. 요청이 거부되면 거부되었음을 알리는 전자메일이 전송됩니다. 필터 메뉴에서 **요청**을 선택하여 사이트 페이지에서 보류 중인 요청을 확인할 수 있습니다. 요청이 거부된 경우 문제를 해결한 후 요청을 다시 제출할 수 있도록 거부 이유에 대한 메시지가 표시됩니다.

사이트 요청이 자동으로 승인되도록 설정된 경우 사이트가 자동으로 생성됩니다. 진행률 표시줄에 새 사이트 이름 및 생성 상태가 표시됩니다. 사이트가 생성되면 사이트 목록에 이름이 나타나고 상태가 오프라인입니다.

엔터프라이즈 사이트를 생성한 경우 해당 사이트 모음이 선택된 저장소에 생성됩니다. 사이트를 사용자와 공유하는 경우 해당 사용자는 연관된 모음에 대한 동일한 권한을 갖습니다.


사이트가 생성된 후에는 사이트를 공유하고, 보안을 변경하고, 콘텐츠를 추가 및 편집하고, 사이트를 게시하고, 온라인으로 전환할 수 있습니다. [사이트 및 사이트 설정 관리](#) 및 [사이트 편집](#)을 참조하십시오.

## 사이트 복사

빠른 시작을 위해서는 사이트를 복사할 수 있습니다. 테마, 모든 미해결 업데이트, 페이지, 페이지 콘텐츠, 권장사항, 기타 모든 이미지 자산 및 정책을 비롯한 원본 사이트의 모든 내용이 제공한 새 이름으로 새 사이트에 복사됩니다.

### 주:

여러 저장소의 콘텐츠를 사용하는 사이트를 복사하는 경우 OCE 툴킷을 사용하여 작업을 수행해야 합니다. [OCE Toolkit으로 개발](#) 및 [cec 명령행 유틸리티 사용](#)을 참조하십시오.

1. **사이트**를 누릅니다.
2. 복사할 사이트를 선택하고 마우스 오른쪽 단추 누르기 메뉴에서 **복사**를 선택하거나 작업 모음에서  을 누릅니다.
3. 사이트 구성 페이지에서 사이트 생성 전 필요한 승인, 사이트에 필요한 최소 보안 레벨 및 사이트 페이지 이미지를 확인합니다. 다음 단계를 수행하십시오.
  - a. **복사** 드롭다운 목록에서 기본 사이트만 복사하거나 업데이트를 포함하도록 선택할 수 있습니다. 업데이트를 포함할 경우, 복사된 사이트에는 원래 사이트의 모든 미해결 업데이트가 포함됩니다. 표준 사이트를 복사하는 경우 화살표를 눌러서 다음 페이지로 이동하고 단계 4로 건너뛩니다.
  - b. 엔터프라이즈 사이트를 복사하는 경우에는 기본 자산 저장소, 지역화 정책, 필수 언어 및 기본 언어가 원래 사이트를 기준으로 설정됩니다. 선택적으로 사이트 콘텐츠 관리를 위해 다른 저장소를 선택할 수 있습니다.
  - c. **다음**을 눌러 다음 페이지로 이동합니다.
4. 세부정보 추가 페이지에서 다음 정보를 입력합니다.
  - 사이트에 대한 이름을 입력합니다. 이 이름은 사이트 URL에 사용됩니다. 문자, 숫자, 밑줄 ( ) 및 하이픈(-)을 사용할 수 있습니다. URL은 대소문자를 구분합니다. 공백을 입력하면 자동으로 하이픈으로 바뀝니다. 기존 사이트 이름을 대소문자가 다른 형태로 사용하지



마십시오. 예를 들어 ABC라는 사이트가 존재할 경우, Abc라는 다른 사이트를 생성하지 마십시오.

#### 주:

템플릿, 테마, 구성요소, 사이트 또는 사이트 페이지에 다음 이름을 사용하지 마십시오: authsite, content, pages, scstemplate\_\*, \_comps, \_components, \_compsdelivery, \_idcservice, \_sitescloud, \_sitesclouddelivery, \_themes, \_themesdelivery. 사이트 페이지에는 다음 이름을 사용할 수 있지만 템플릿, 테마, 구성요소 또는 사이트에는 사용하지 마십시오: documents, sites.

- 엔터프라이즈 사이트를 복사하는 경우 선택적으로 친화적 URL 값을 위해 기본 사이트 접두어를 편집할 수 있습니다. 이 접두어는 콘텐츠 항목 슬러그 값(페이지나 자산에 특정한 URL 부분)에 추가됩니다.
- 사이트에 대한 선택적 설명을 입력합니다.
- 이 사이트 요청에 대한 근거(선택사항)를 입력합니다. 그러면 사이트 관리자가 요청을 승인해야 할지 여부를 보다 쉽게 결정할 수 있습니다.

#### 5. 완료를 누릅니다.

사이트 요청에 승인이 필요한 경우 사이트 관리자의 승인 후 사이트가 생성됩니다. 필터 메뉴에서 **요청**을 선택하여 사이트 페이지에서 보류 중인 요청을 확인할 수 있습니다. 요청이 거부되면 문제를 해결한 후 요청을 다시 제출할 수 있도록 거부 이유에 대한 메시지가 표시됩니다.

사이트 요청이 자동으로 승인되도록 설정된 경우 사이트가 자동으로 생성됩니다. 진행률 표시줄에 새 사이트 이름 및 생성 상태가 표시됩니다. 사이트가 생성되면 사이트 목록에 이름이 나타나고 상태가 오프라인입니다.

엔터프라이즈 사이트를 복사한 경우 선택한 저장소에 해당 사이트 모음이 생성됩니다. 사이트를 사용자와 공유하는 경우 해당 사용자는 연관된 모음에 대한 동일한 권한을 갖습니다.

사이트가 생성된 후에는 사이트를 공유하고, 보안을 변경하고, 콘텐츠를 추가 및 편집하고, 사이트를 게시하고, 온라인으로 전환할 수 있습니다. 여러 저장소에 액세스 권한이 있는 사이트를 공유하는 경우 기본 저장소만 공유됩니다. 추가 저장소는 개별적으로 공유되어야 합니다. [사이트 및 사이트 설정 관리](#) 및 [사이트 편집](#)을 참조하십시오.

## 사이트 요청 관리

사이트 제어가 사용으로 설정된 경우 사이트가 생성 전 승인되어야 할 수도 있습니다. [사이트] 페이지에서 [요청] 필터를 사용하여 보류 중인 사이트 요청을 볼 수 있습니다.

#### 주:

Oracle Content Management Starter Edition을 통해서 제어 사용으로 설정할 수 없습니다. 전체 기능과 무제한 사이트가 필요하다면 Oracle Content Management Premium Edition으로 업그레이드하십시오.

사이트 제어 이해를 참조하십시오.

보류 중인 사이트 요청을 보려면 사이트 페이지의 필터 메뉴에서 **요청**을 선택합니다. *사이트 관리자*인 경우 다른 사람의 승인이 필요한 요청을 포함해 시스템의 모든 사이트 요청이 표시됩니다. *승인자*인 경우 본인이 승인자로 있는 모든 사이트 요청이 표시됩니다. *모든 사용자*는 자신이 제출한 사이트 요청이 표시됩니다.

요청 목록에서 사이트의 작은 그림, 사이트 이름, 요청자, 요청 시기, 사이트 설명, 요청 상태 및 사이트 사용을 위해 로그인에 필요한지 여부를 보여주는 아이콘을 확인할 수 있습니다.

roles에 따라 다음 작업을 수행할 수 있습니다.

- 요청의 추가 세부정보(예: 필요한 최소 보안, 사용된 템플릿, 사이트에 대한 선택적 근거 및 작은 그림)를 보려면 사이트 이름을 누르거나 요청을 선택한 다음 **보기**를 누릅니다.
- 사이트 관리자 또는 승인자인 경우 선택 후 **승인**을 눌러 요청을 승인할 수 있습니다. 사이트는 승인된 후 자동으로 생성됩니다.
- 사이트 관리자 또는 승인자인 경우 세부정보를 확인한 다음 **거부**를 눌러 요청을 거부할 수 있습니다. 요청 거부 사유를 입력한 다음 **거부**를 누릅니다. 요청 목록에서 요청을 선택한 다음 **거부**를 눌러 거부할 수도 있지만 이 경우 거부 사유를 추가하지 못할 수 있습니다.
- 요청이 실패하거나 거부된 경우 세부정보를 확인하고 필요에 따라 요청을 편집한 다음 요청을 **다시 제출**할 수 있습니다.
- 사이트 생성자인 경우 선택 후 **삭제**를 눌러 요청을 삭제할 수 있습니다.

#### 주:

사이트 요청자가 삭제된 경우 사이트 관리자가 사이트 요청을 삭제할 수 있습니다.

사이트가 생성된 후에는 사이트를 공유하고, 보안을 변경하고, 콘텐츠를 추가 및 편집하고, 사이트를 게시하고, 온라인으로 전환할 수 있습니다. [사이트 및 사이트 설정 관리](#) 및 [사이트 편집](#)을 참조하십시오.

## 사이트 요청 세부정보 변경

사이트 요청 세부정보 탭에는 사이트 이름, 설명, 사용된 템플릿, 근거, 소유자(요청 생성자) 및 사이트 이미지가 표시됩니다.

요청을 생성한 경우(요청 소유자인 경우) 사이트 요청 세부정보를 업데이트할 수 있습니다.

사이트 요청 세부정보를 보거나 변경하려면 다음과 같이 하십시오.

1. **사이트**를 누른 다음 사이트 메뉴에서 **요청**을 누릅니다.
2. 확인 또는 편집할 요청을 누릅니다.
3. 승인자인 경우 요청을 **승인** 또는 **거부**할 수 있습니다. 요청자인 경우 필요에 따라 정보를 편집한 다음 요청을 **다시 제출**할 수 있습니다.

## 사이트 요청 정책 보기

사이트 요청 정책 탭에는 사이트 유형(표준 또는 엔터프라이즈), 게시된 사이트에 액세스할 수 있는 사용자 및 사이트 생성 전 필요한 승인이 표시됩니다. 엔터프라이즈 사이트의 경우 사이트용 자산 저장소, 지역화 정책, 필요한 언어 및 기본 언어도 표시됩니다.

사이트 요청 정책을 보려면 다음과 같이 하십시오.

1. **사이트**를 누른 다음 사이트 메뉴에서 **요청**을 누릅니다.
2. 확인 또는 편집할 요청을 누른 다음 **정책** 탭을 누릅니다.
3. 승인자인 경우 요청을 **승인** 또는 **거부**할 수 있습니다. 요청자인 경우 필요에 따라 자산 저장소를 변경한 다음 요청을 **다시 제출**할 수 있습니다.

# 5

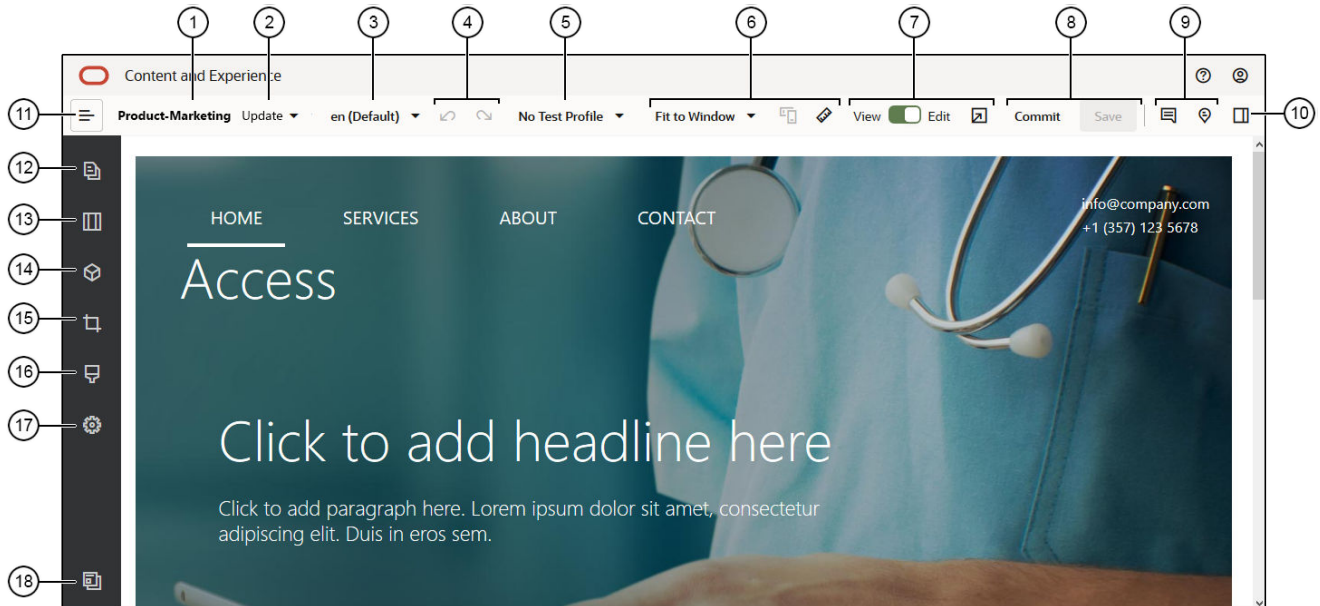
## 사이트 편집

페이지 편집기에 대해 간략히 알아보고 수행할 수 있는 작업을 살펴보겠습니다.










- 사이트 작성기 페이지 알아보기
- 사이트 업데이트 이해
- 업데이트 사용
- 팁과 힌트 편집
- 스타일 및 서식 사용
- 사용자정의 사이트 속성 추가
- 사이트 설정 사용자정의
- 테이블 작업
- 사이트 파일 업로드


### 사이트 작성기 페이지 알아보기


기존 업데이트를 편집하거나 사이트 업데이트를 생성할 때 사이트 작성기에서 업데이트가 열립니다. 페이지 레이아웃과 사용 가능한 도구에 익숙해지도록 잠시 둘러보십시오.




몇 가지 주의 사항이 있습니다.








콜아웃	설명
1	사이트의 이름이 나열됩니다.
2	업데이트에서 현재 작업 중인 업데이트의 이름을 확인합니다. 사용 가능한 업데이트가 두 개 이상인 경우 다른 업데이트로 전환할 수 있습니다.
3	언어에서 현재 작업 중인 버전을 확인합니다.
4	<p><b>실행 취소</b>  를 사용하여 편집기에서 수행한 최종 편집 또는 변경사항을 되돌립니다.</p> <p><b>재실행</b>  을 사용하여 실행 취소를 통해 되돌린 가장 최근의 변경사항을 재적용합니다. 실행 취소를 여러 번 사용하여 콘텐츠, 스타일 및 페이지 구성에 대한 변경사항을 비롯하여 현재 업데이트의 일련의 변경사항을 되돌릴 수 있습니다.</p>
5	<b>테스트 프로파일</b> 을 사용하면 여러 대상 속성으로 프로파일을 생성하고, 프로파일을 사용해서 여러 사이트 방문자에게 테스트 권장사항을 제공할 수 있습니다. 테스트 프로파일은 사이트 관리자가 권장사항을 생성할 때 설정됩니다.
6	<p><b>미리보기 옵션</b>을 사용하여 여러 가지 상황에서 콘텐츠가 표시되는 방식을 확인합니다. 원하는 페이지로 이동하여 현재 업데이트 변경사항이 적용된 기본 사이트를 확인할 수 있습니다. <b>창에 맞춤</b>을 누르고 치수를 선택하여 해당 화면 크기로 장치에 표시되는 대로 페이지를 봅니다. 여러 크기가 제공되며, 자신의 고유 장치 크기를 생성할 수 있습니다.</p> <p> 을 눌러서 표시를 나타냅니다. 눈금자 간격을 눌러서 여러 크기에서 사이트가 표시되는 방식을 빠르게 확인합니다. 또한  을 선택해서 방향에 따라 모바일 장치에 사이트가 표시되는 방식을 볼 수 있습니다. 반응형 디자인 테마는 선택한 화면 크기에 최적 사용되도록 페이지 콘텐츠를 자동으로 배열합니다.</p>
<div style="border: 1px solid #0070C0; padding: 10px; background-color: #E6F2FF;"> <p> <b>주:</b></p> <p>미리보기 모드에서는 사이트 페이지에 대한 링크가 작동하지 않습니다.</p> </div>	
7	페이지 상단에 있는 <b>스위치</b> 를 사용하여 보기 또는 편집으로 전환합니다. 사이트를 미리 보려면  을 누릅니다. 편집하는 동안 사용되는 슬롯/구성요소 테두리 및 기타 시각적 도구 없이 사이트 방문자에 보이는 대로 페이지를 표시합니다. 페이지의 링크를 사용하여 탐색 메뉴의 링크, 텍스트의 링크 등 사이트를 탐색할 수 있습니다. 다른 사이트에 대한 링크는 보안상 새 창에서 열립니다. 현재 사이트 내 페이지에 대한 링크는 지정된 링크 대상에 따라 동일한 창 또는 새 창에서 열립니다.
8	사용 가능한 <b>옵션</b> 을 사용하여 변경사항을 처리합니다. <b>커밋</b> 을 눌러 <b>기본 사이트</b> 에 변경사항을 병합하거나 <b>저장</b> 을 눌러 <b>업데이트</b> 에 대한 변경사항을 저장합니다.
9	<p><b>주석 컨트롤</b>을 사용하여 주석을 추가하거나 기존 주석을 봅니다.</p> <ul style="list-style-type: none"> <li>주석을 추가하려면  을 누르고, 주석을 추가할 구성요소를 누른 후, 텍스트 상자에 주석을 입력하고 <b>게시</b>를 누릅니다. <i>Collaborating on Documents with Oracle Content Management</i>의 주석 추가를 참조하십시오.</li> <li>주석을 보려면  을 누릅니다. 주석을 누르면 대화에서 주석 메모가 포커스됩니다.</li> </ul>
10	 을 눌러 이 사이트에 대한 토론을 생성하거나 참여할 수 있는 <b>대화 패널</b> 을 표시하거나 숨깁니다. <i>Collaborating on Documents with Oracle Content Management</i> 의 대화 사용을 참조하십시오.

콜아웃	설명
11	 을 눌러 사이드바에서 도구를 표시하거나 숨깁니다.

 **주:**

사이드바에서 도구를 사용하기 전에  를 편집으로 설정해야 합니다.

사이드바의 아이콘 (12-18)을 눌러 페이지를 관리하고, 구성요소를 추가하고, 테마를 변경합니다.

콜아웃	누르기...	목적...
12		페이지를 추가/편집하고, 중첩 페이지 구조를 관리하고, 페이지를 재구성하고, 페이지 설정을 변경합니다.
13		가로 또는 세로 레이아웃과 같은 섹션 레이아웃을 추가합니다.
14		사이트에 자산을 추가합니다. 사이트를 생성할 때 선택된 저장소에 있는 모든 자산을 선택할 수 있습니다.
15		사이트에 구성요소를 삽입합니다. 테마, 사용자정의, 시드와 같은 여러 유형의 구성요소 중에서 선택하거나 사용 가능한 모든 구성요소를 표시할 수 있습니다.
16		사이트에 대해 다른 테마를 선택하거나 페이지 또는 슬롯에 대한 배경을 변경합니다.
17		SEO(검색 엔진 최적화) 및 사이트 재지정을 위한 키워드와 같은 사이트 설정을 조정합니다.
18		사이트 관리를 위한 [사이트] 페이지로 돌아갑니다.

## 사이트 업데이트 이해

업데이트는 현재 기본 사이트에 대한 변경사항의 명명된 모음입니다. 변경사항은 커밋 후 영구적으로 기본 사이트를 업데이트할 때까지 업데이트에 유지됩니다.

편집기에서 사이트를 보거나 편집할 때마다 업데이트를 사용합니다. 편집기에서 수행한 변경사항은 해당 업데이트의 일부가 됩니다. 하나 이상의 업데이트를 가질 수 있고 시간이 지나도 계속 업데이트에 변경사항을 추가할 수 있습니다. 업데이트는 사이트 편집 관리 방법에 유연성을 제공합니다. 예를 들어, 여러 사람이 각자 사이트의 다른 부분에 업데이트 작업을 진행할 수 있습니다. 개별 업데이트를 검토 및 수정할 수 있으며, 준비된 경우 기본 사이트에 업데이트를 커밋할 수 있습니다. 단, 업데이트를 온라인에서 사용할 수 있도록 하려면 사이트를 게시해야 합니다.

여러 방법으로 업데이트를 구성할 수 있습니다.

- 페이지 이름을 따서 업데이트에 페이지별 변경








- 특정 사용자가 사용자 이름으로 업데이트에 수행한 변경
- 특정일이나 특정 프로젝트 마일스톤에 수행한 변경

### 주:

업데이트는 기본 사이트 대비 해당 업데이트의 변경사항을 보여줍니다. 편집기 내에서 사용 가능한 업데이트 사이를 쉽게 전환할 수 있지만, 한 번에 하나의 업데이트만 볼 수 있습니다. 어떤 페이지에 여러 업데이트가 있는 경우, 동일 콘텐츠 영역에 여러 변경사항이 있는지 모를 수 있습니다. 페이지에 변경사항 충돌을 막으려면 특정 페이지나 사이트 영역에 개별 업데이트를 지정하십시오.

현재 업데이트의 변경사항을 커밋하면 변경사항이 기본 사이트에 적용되고 업데이트가 삭제됩니다. 해당 변경사항이 사이트 액세스 권한을 가진 모든 사용자에게 표시되도록 하려면 사이트를 온라인으로 전환하거나 재게시하는 방법으로 게시해야 합니다.

사이트를 편집하려면 다음과 같이 하십시오.

1. 목록에서 사이트를 선택하고 마우스 오른쪽 단추 누르기 메뉴에서 **열기**를 선택하거나 작업 모음에서  을 누릅니다.
2. 미리보기 모드로 편집기가 열립니다. 내용을 변경하거나 사이드바의 탐색 옵션을 사용하려면 편집 스위치  가 **편집**으로 설정되어 있는지 확인합니다.
3. 사이트에 대한 첫번째 업데이트인 경우 업데이트에 대한 이름 및 선택적 설명을 입력한 다음 **확인**을 누릅니다. 문자, 숫자, 밑줄(\_), 하이픈(-)을 이름에 사용할 수 있습니다. 공백을 입력하면 자동으로 하이픈으로 바뀝니다. 사이트에 대한 업데이트가 이미 있을 경우 목록에서 업데이트를 선택하고  을 누릅니다.
4. 특정 페이지를 편집하려면 왼쪽 사이드바의 사이트 트리를 사용하거나 페이지를 검색하거나 사이트의 고유 탐색을 사용하여 페이지를 찾습니다. 사이트 트리를 표시하려면 왼쪽의  을 누른 다음  을 누릅니다.
5. 페이지 콘텐츠를 추가하고 필요에 따라 변경합니다. **창에 맞춤**을 선택하고 레이아웃 옵션을 사용하여 여러 장치 및 여러 크기에서 페이지가 표시되는 방식을 확인합니다. 여러 크기가 제공되며, 자신의 고유 장치 크기를 생성할 수 있습니다.  을 눌러서 표시를 나타냅니다. 눈금자 간격을 눌러서 여러 크기에서 사이트가 표시되는 방식을 빠르게 확인합니다. 또한  을 선택해서 방향에 따라 모바일 장치에 사이트가 표시되는 방법을 볼 수 있습니다.
6. 사이트 편집이 완료되면 다음 방법 중 하나로 변경사항을 저장합니다.
  - 현재 업데이트에 변경사항을 저장하려면 **저장**을 누릅니다. 현재 업데이트에서 작업을 계속할 수도 있고, 나중에 업데이트로 돌아갈 수도 있습니다.
  - **커밋**을 눌러 기본 사이트에 변경사항을 적용합니다.

웹 사이트 사용자에게 변경사항이 표시되기 전 사이트를 게시해야 합니다. **사이트를 온라인으로 전환 또는 오프라인으로 설정** 또는 **사이트 변경사항 게시**를 참조하십시오.

현재 업데이트의 변경사항을 커밋하면 변경사항이 기본 사이트에 적용되고 업데이트가 삭제됩니다. 편집기에서 사이트를 보거나 편집할 때마다 활성 업데이트를 사용해야 하므로 새 업데이트를 생성할 수 있는 사이트 목록으로 돌아옵니다.

업데이트는 선형 프로세스입니다.

- 원본 사이트(기본 사이트)가 존재합니다. 이것을 버전 1로 부르겠습니다.
- 업데이트를 생성합니다. 업데이트를 병합하면(커밋 사용) 기존 사이트에 업데이트가 영구적으로 기록됩니다. 이제 기본 사이트의 새 버전(버전 2)이 생깁니다.
- 또 다른 업데이트를 생성하고 병합하면 기존 사이트에 업데이트가 영구적으로 기록됩니다. 이제 기본 사이트의 새 버전(버전 3)이 생깁니다.



업데이트에서 다음을 기억하십시오.

현재 기본 사이트 + 병합된 업데이트 = 새 버전의 기본 사이트

## 업데이트 사용

편집기에서 사이트를 편집할 때마다 업데이트를 사용합니다. 편집기에서 수행한 변경사항은 해당 업데이트의 일부가 됩니다.

업데이트를 생성 및 사용하려면 다음과 같이 하십시오.

1. [사이트] 페이지에서 사이트를 선택하고 마우스 오른쪽 단추 누르기 메뉴에서 **열기**를 선택하거나 작업 모음에서  을 누릅니다.
2. 편집 스위치  를 편집으로 설정합니다.
3. 사이트에 대한 첫번째 업데이트인 경우 업데이트에 대한 이름 및 선택적 설명을 입력한 다음 **확인**을 누릅니다. 문자, 숫자, 밑줄(\_) 및 하이픈(-)을 사용할 수 있습니다. 공백을 입력하면 자동으로 하이픈으로 바뀝니다. 기존 업데이트를 사용하려면 목록에서 업데이트를 누릅니다.
4. 사이트 이름 및 현재 업데이트가 편집기의 상단 표시줄에 나타납니다. 업데이트가 여러 개인 경우 업데이트 목록에서 다른 업데이트를 선택하여 업데이트를 전환할 수 있습니다.
5. 페이지 콘텐츠를 추가하고 필요에 따라 변경합니다. 다른 표시 크기 옵션을 사용해서 페이지가 다른 장치에 표시되는 방식을 확인합니다.
6. 편집이 완료되면 **저장**을 누릅니다. 변경사항을 기본 사이트와 병합할 준비가 된 경우 업데이트를 커밋할 수 있습니다. 웹 사이트 사용자에게 변경사항이 표시되기 전 사이트를 게시해야 합니다. [사이트 변경사항 게시](#)를 참조하십시오.


## 팁과 힌트 편집



편집기 시작을 도와주는 몇 가지 사항이 있습니다.

- [사이드바 숨기기 또는 표시](#)
- [둘러보기](#)
- [페이지 관리](#)
- [끌어 놓기 편집](#)
- [구성요소](#)
- [구성요소의 크기 및 간격 조정](#)
- [편집 가능 레이아웃 콘텐츠](#)
- [스타일 및 서식](#)
- [변경사항 취소](#)




## 사이드바 숨기기 또는 표시

페이지 및 페이지 콘텐츠를 관리하고 편집하는 옵션을 표시하려면  을 누릅니다.













사이드바에 액세스하려면  를 **편집**으로 설정합니다. 사이드바를 숨기고 페이지 미리보기 시 보이는 영역을 늘리려면  을 누릅니다.

## 둘러보기


사이트 트리를 사용하여 다른 페이지를 선택하려면 사이드바에서  을 누릅니다. 페이지 목록 상단의 검색 상자를 사용하여 페이지를 검색합니다. 페이지 자체에서 사이트 탐색 또는 링크를 사용할 수도 있습니다.

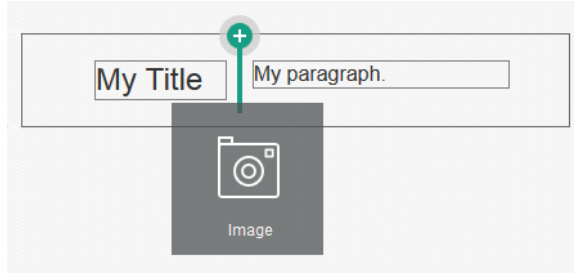
페이지에 수행한 변경사항은 다른 페이지로 전환할 때 저장됩니다. **저장**을 눌러 현재 업데이트에 변경사항을 저장할 수도 있습니다.

## 페이지 관리

- 페이지를 추가하려면 페이지를 추가할 레벨 또는 분기를 선택하고 **페이지 추가**를 누릅니다. 또는 하위 페이지를 추가하려면  을 누르고  을 누릅니다. “새 페이지”가 사이트 트리의 맨 아래에 추가되고, 페이지 이름과 다른 설정을 지정하라는 메시지가 표시됩니다.
- 페이지를 삭제하려면 선택 후  을 누릅니다.
- 페이지를 이동하려면 선택 후 새 위치로 끌어옵니다. 또는 페이지를 잘라서 새 위치에 붙여넣을 수도 있습니다. 페이지를 잘라내려면  을 누르고  을 누릅니다. 페이지를 붙여넣으려면 붙여넣을 분기를 선택하고  을 누르고  을 누릅니다.
- 페이지를 복사하려면 선택 후  을 누르고  을 누릅니다. 페이지를 붙여넣으려면 붙여넣을 분기를 선택하고  을 누르고  을 누릅니다.
- 메타데이터, 머리글, 바닥글 및 기타 옵션 등의 페이지 설정을 변경하려면  을 누릅니다. **페이지 설정 변경**을 참조하십시오.

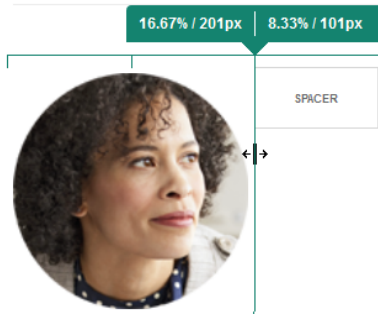
## 끌어 놓기 편집

사이드바에서 구성요소를 추가하거나 페이지에 구성요소를 이동하려면 항목을 눌러 페이지의 위치로 끌어 놓습니다. 항목을 페이지로 끌어오면 사용 가능한 슬롯과 기존 항목의 경계가 표시됩니다. 배치 막대  는 새 콘텐츠의 이동 가능한 위치(위쪽, 아래쪽, 왼쪽, 오른쪽)를 나타냅니다.




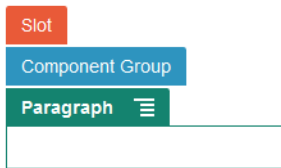
슬롯에 여러 항목을 가질 수 있고 페이지의 새 위치로 항목을 끌어서 이동할 수 있습니다. 두 구성요소 사이의 경계를 눌러 끌어오는 방식으로 슬롯에서 두 구성요소의 상대 너비를 조정할 수도 있습니다. 구성요소는 구성요소 위에 표시되는 "눈금자"에 나와 있는 다음 격자선에 맞춰집니다.

각 구성요소의 크기는 픽셀 단위와 슬롯에서 사용 가능한 공간의 퍼센트로 표시됩니다. 격자를 통해 정의된 값 이외의 다른 값에 맞게 너비를 조정하려면 **Ctrl** 키를 누른 상태에서 구성요소 경계를 눌러 끌어옵니다.




## 구성요소

페이지에 구성요소를 놓은 후 항목을 선택하고 항목의 메뉴 아이콘  을 누르고 **설정**을 선택하여 맞춤, 간격 및 기타 속성을 조정할 수 있습니다. 메뉴 아이콘 대신 구성요소 이름을 누르면 슬롯과 구성요소 그룹(현재 구성요소가 구성요소 그룹에 속한 경우)의 메뉴 아이콘을 보고 선택할 수 있습니다. 다른 탭 중 하나를 선택하면 메뉴 아이콘을 보고 해당 요소의 속성을 설정할 수 있습니다.



테마 디자이너는 테마에 포함되는 구성요소를 지정할 수 있습니다. 따라서 특정 사이트에서 사용 가능한 구성요소를 다른 사이트에서는 사용하지 못할 수도 있습니다. 또한 테마 디자이너는 제공된 페이지 레이아웃의 제공된 슬롯에서 허용되는 구성요소를 지정할 수 있습니다. 특정 슬롯에서

구성요소가 허용되지 않을 경우 배치 막대의 색상과 기호(빼기)  가 변경되고 다음과 유사한 메시지가 표시됩니다.



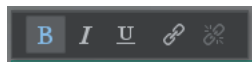
### 구성요소의 크기 및 간격 조정


갤러리 또는 이미지와 같은 여러 구성요소의 크기를 다른 CSS 단위로 조정하고 구성요소 주위 간격을 조정할 수 있습니다. 예를 들어, 다음과 같습니다.

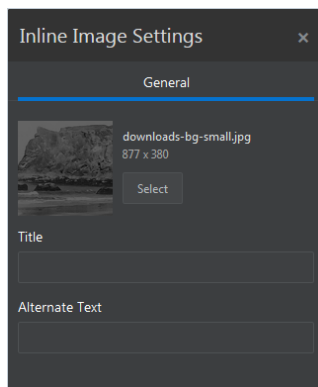
- px(픽셀): 기본값입니다. 숫자 값만 지정된 경우 픽셀로 간주됩니다.
- %(백분율): 항목을 해당 항목의 상위 HTML 요소 크기의 백분율로 설정합니다. 예: 25%
- em: 구성요소 크기를 em 간격으로 설정합니다. 예: 20em
- vw: 크기를 보기 영역 너비의 백분율로 설정하고 보기 영역 크기에 대응합니다. 예: 10vw

### 편집 가능 레이아웃 콘텐츠

테마 디자이너는 편집기에서 변경할 수 없는 "내장" 콘텐츠(예: 저작권 공지)를 페이지 레이아웃에 추가할 수 있습니다. 또한 테마 디자이너는 디지털 자산을 포함해서 간단한 텍스트 및 이미지 콘텐츠를 편집 가능 요소로 지정할 수 있습니다. 그러면 제공자가 텍스트 또는 이미지 콘텐츠는 변경하되 위치 또는 기타 레이아웃 속성은 변경하지 못합니다. 편집 가능 텍스트 요소는 굵은체, 기울임꼴 및 밑줄 텍스트 지정과 링크 변경 또는 제거 옵션이 포함된 메뉴를 제공합니다.



편집 가능 그래픽 요소는 설정 아이콘  과 이미지, 제목 및 대체 텍스트를 지정할 수 있는 패널을 제공합니다.



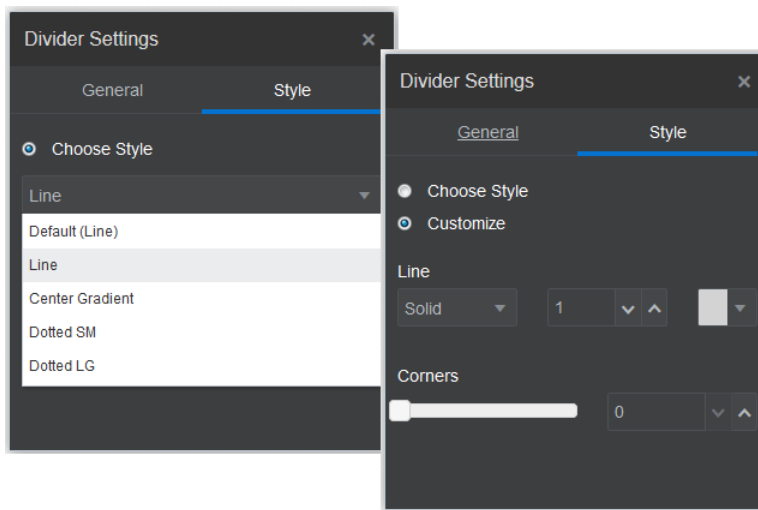
텍스트 또는 이미지를 둘러싸는 프레임은 텍스트 길이 또는 이미지 크기에 맞게 조정됩니다.


**주:**


변경사항은 현재 페이지에만 적용됩니다. 원본 콘텐츠는 레이아웃과 함께 테마에 저장되며 레이아웃을 처음 페이지에 적용할 때 기본값이 됩니다.

**스타일 및 서식**

대부분의 구성요소에는 구성요소의 모양 측면을 지정하는 테마에 의해 정의된 하나 이상의 기본 스타일이 있습니다. 스타일을 쉽게 전환하거나 스타일을 무효화할 수 있습니다. 사용 가능한 스타일 중 선택하려면 구성요소의 설정 패널을 열고, **스타일** 탭을 누르고, **스타일 선택**을 누르고, 메뉴에서 스타일을 선택합니다. 스타일 정의에 지정된 속성의 고유 값을 지정하려면 **사용자정의**를 누르고 서식 옵션을 지정합니다.

**변경사항 취소**

 을 사용하여 편집기에서 수행한 최종 편집 또는 변경사항을 되돌립니다. 실행 취소를 여러 번 사용하여 일련의 변경사항을 되돌릴 수 있습니다.






 을 사용하여 실행 취소를 통해 되돌린 가장 최근의 변경사항을 재적용합니다. 실행 취소를 연속으로 여러 번 사용했을 경우 재실행을 여러 번 사용할 수 있습니다.

현재 업데이트의 콘텐츠, 스타일 및 페이지 구성 변경사항을 취소할 수 있습니다. 일부 작업은 실행 취소 체인에 포함되지 않습니다.

- 다른 업데이트로 전환하면 실행 취소 체인이 재설정되고, 이전에 작업한 업데이트에서 수행한 변경사항을 취소할 수 없습니다. 한 업데이트 내에서는 변경사항을 저장한 후에도 취소할 수 있습니다.
- 페이지 전환 또는 제공된 페이지 크기 변경 등 편집기에서 뷰를 변경할 경우 해당 변경사항은 수동으로 되돌려야 합니다.
- 제목 또는 단락과 같은 텍스트 구성요소를 편집하는 경우 텍스트 편집기는 자체 실행 취소 체인을 갖습니다. 텍스트 편집기에서 나가면 해당 변경사항을 더 이상 취소할 수 없습니다.

## 스타일 및 서식 사용

대부분의 구성요소에는 구성요소의 모양 측면을 지정하는 테마에 의해 정의된 하나 이상의 기본 스타일이 있습니다. 스타일을 쉽게 전환하거나 선택한 옵션으로 스타일을 무효화할 수 있습니다.



1. 편집할 페이지로 이동하고 가 **편집**으로 설정되어 있는지 확인합니다.
2. 구성요소의 기본 스타일을 설정하려면 구성요소의 메뉴 아이콘 을 누르고 **설정**을 선택합니다. **스타일** 탭을 누릅니다.
  - 사이트의 테마에서 스타일을 사용하려면 **스타일 선택**을 누르고 메뉴에서 스타일을 선택합니다. 개별 구성요소에 대해 스타일이 정의되므로 스타일 목록이 다를 수 있습니다. 예를 들어, 이미지 스타일은 단락 스타일과 다릅니다.
  - 고유의 기본 서식 옵션을 지정하려면 **사용자정의**를 누르고 서식 옵션을 지정합니다.
3. 하나 이상의 유사 구성요소에 기본 스타일을 복사하여 붙여넣으려면 구성요소의 메뉴 아이콘 을 누르고 **component 스타일 복사**를 선택합니다. 유사 구성요소의 메뉴 아이콘 을 누르고 **component 스타일 붙여넣기**를 선택합니다.
4. 제목 또는 단락 구성요소 내의 텍스트 서식을 지정하려면 텍스트 구성요소를 누릅니다. 서식 지정 옵션과 함께 도구모음이 표시됩니다. 서식을 지정할 텍스트를 선택하고 글꼴, 색상, 맞춤과 같은 옵션을 선택합니다. 변경사항이 즉시 적용됩니다. 서식을 제거하려면 텍스트를 선택하고 을 누릅니다. 서식 변경사항은 기본 스타일보다 우선 적용됩니다. 기본 스타일을 변경해도 우선 적용은 그대로 남습니다.

## 사이트 설정 사용자정의

사이트 설정을 사용자정의하여 사이트 아이콘, 컨트롤러 파일, 사이트맵, 로봇 파일, 보조 파일을 추가하고, 맵 제공자를 지정할 수 있습니다.

이러한 설정은 업데이트를 커밋할 때까지 업데이트에 저장됩니다. 이 파일은 게시 후에 테마의 루트 폴더에 저장되어 이 테마를 사용하는 모든 사이트에서 사용하게 됩니다.

아이콘 및 전체 사이트에 대한 링크 동작을 변경하려면 다음을 수행합니다.

1. 편집을 위해 사이트를 엽니다.
2. 사이드바에서 을 누르고  **사이트**를 누릅니다.
3. **즐거찾기 아이콘** 섹션에서 브라우저 또는 모바일 장치와 같은 다른 플랫폼에서 사이트가 최소화되는 경우 사용할 이미지를 선택합니다. 아이콘은 액세스할 수 있는 디지털 자산으로 저장해야 합니다. **업로드할 파일 선택**을 누르고 아이콘으로 이동하여 선택한 다음 **확인**을 누릅니다. 즐겨찾기 아이콘이 모든 브라우저에서 작동하려면 사용자정의가 필요합니다.
  - **Chrome 및 Safari의 경우** 다음 예에서와 같이 컨트롤러 파일에 사이트 접두어 등 즐겨찾기 아이콘에 대한 참조가 있어야 합니다.

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

**Internet Explorer 11 및 Firefox의 경우 유사한 항목이 페이지 템플릿에 포함되어야 합니다.** 하지만 페이지 템플릿은 테마의 일부이므로 여러 사이트에서 사용될 수 있습니다. 따라서 고정 사이트 접두어를 사용할 수 없으며 대신 토큰을 사용해야 합니다. 다음 예를 참조하십시오.

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

토큰은 페이지가 제공될 때 사이트 접두어로 교체됩니다.

Internet Explorer 및 Firefox는 컨트롤러와 페이지 템플릿에서 즐겨찾기 아이콘을 로드합니다. 따라서 컨트롤러와 페이지 템플릿에서 동일한 항목이 참조되지 않을 경우 아이콘이 깜박입니다. Chrome 및 Safari는 컨트롤러에서 참조되는 아이콘만 로드합니다.

4. **컨트롤러 파일** 섹션에서는 브라우저가 링크 요청을 처리하는 방식을 변경하는 파일을 추가할 수 있습니다. 즐겨찾기 아이콘을 참조하는 것과 더불어 OpenGraph 태그나 메타데이터 태그를 추가하여 웹 마스터 사이트 확인 또는 소셜 매체에서의 사이트 공유에 사용할 수 있습니다. 기본 컨트롤러 파일을 다운로드하여 편집하거나 자신의 파일을 업로드합니다. 파일은 액세스할 수 있는 디지털 자산으로 저장해야 합니다. **업로드할 파일 선택**을 누르고 파일로 이동하여 선택한 다음 **확인**을 누릅니다.

예를 들어, 사이트에 Facebook Share 단추가 있을 경우 다음 예에서와 같이 Facebook이 사이트에 대한 세부정보를 Facebook에 표시하는 데 사용할 수 있는 메타데이터를 제공할 수 있습니다.

```
<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"/>
```

다음 예와 유사하게 Google 웹 마스터 확인 태그를 추가할 수 있습니다.

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



5. **사이트맵과 로봇 파일** 섹션에서 사용자정의 사이트맵과 로봇 파일을 업로드할 수 있습니다. *사이트맵*은 웹 사이트에 대한 URL 및 각 URL에 대한 정보(예: 최종 업데이트 날짜)를 나열하는 데 사용할 수 있는 XML 파일입니다. *로봇 파일*은 검색 엔진 로봇이 웹 사이트에서 페이지에 인덱스화하는 방식을 지시하는 데 사용할 수 있는 텍스트 파일입니다. 파일은 액세스할 수 있는 디지털 자산으로 저장되어야 합니다. **업로드할 파일 선택**을 누르고 파일로 이동하여 선택한 다음 **확인**을 누릅니다.
6. **보조 파일** 섹션에서는 필요에 따라 사이트 소유권 확인을 위해 필요한 것과 같은 추가 파일을 필요에 따라 업로드할 수 있습니다. 파일은 액세스할 수 있는 디지털 자산으로 저장해야 합니다. **업로드할 파일 선택**을 누르고 파일로 이동하여 선택한 다음 **확인**을 누릅니다.
7. 맵 구성요소의 제공자와 사용된 링크를 선택할 수 있습니다. **Oracle Maps** 또는 **Google Maps**를 선택합니다.
8. 업데이트를 게시하는 경우 변경사항이 게시되어 사용할 수 있게 됩니다.

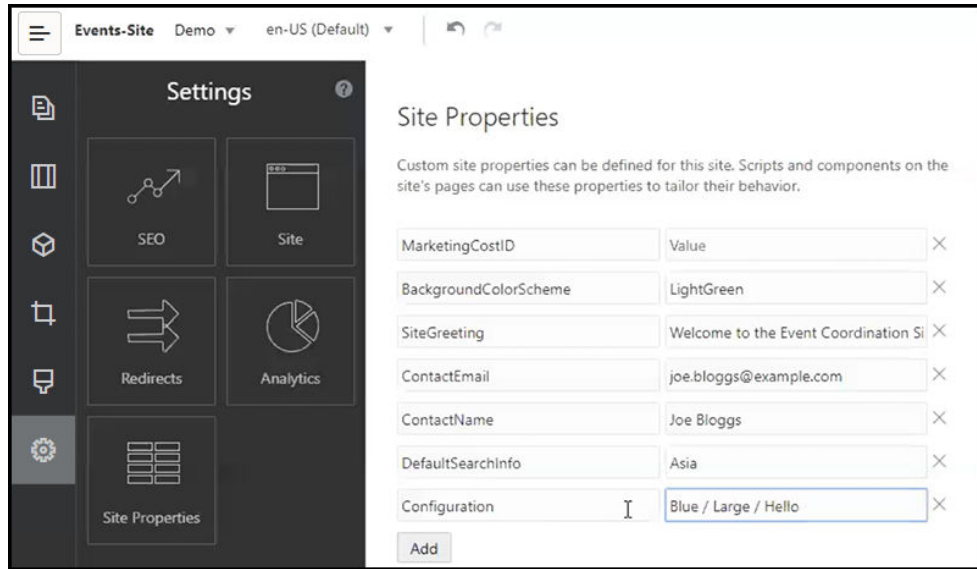
## 사용자정의 사이트 속성 추가

이름/값 쌍 형태로 사용자정의 속성을 사이트에 추가할 수 있습니다. 이 속성은 사이트와 함께 저장되며 사이트 페이지의 스크립트 및 구성요소에서 사용할 수 있습니다. 이렇게 하면 기본 스크립트 및 구성요소 코드를 변경하지 않고도 사이트를 매개변수화하거나 사용자정의할 수 있습니다.

예를 들어, 사용자정의 속성을 사용하여 페이지 배경색을 변경하고, 검색 결과를 세분화하고, 목록을 채우고, 일반적으로 사이트 종속 변수를 제어할 수 있습니다.

사용자정의 사이트 속성은 사이트를 편집할 때 설정 패널을 사용하여 추가됩니다.

1. 편집을 위해 사이트를 엽니다.
2. 사이드바에서  을 누르고  사이트 속성을 누릅니다.
3. 추가를 누릅니다.
4. 사용자정의 사이트 속성에 대한 이름 및 값을 입력합니다. 최대 50개까지 사용자정의 사이트 속성을 추가할 수 있습니다. 이름 필드에는 200자 제한이 있고 값 필드에는 2000자 제한이 있습니다.



5. 삭제하려면 이름/값 쌍 옆에 있는 X를 누릅니다.
6. 사용자정의 사이트 속성 추가/제거를 마쳤으면 커밋을 누릅니다.

 주:

변경사항은 기존 사용자정의 사이트 속성과 병합되지 않습니다. 사용자정의 사이트 속성에 대한 변경사항을 커밋하면 기본 사이트의 기존 사용자정의 사이트 속성을 겹쳐씹니다.

일단 정의되면 사용자정의 사이트 속성은 사이트 및 사이트 구성요소 전체에서 스크립팅에 사용될 수 있습니다. 예를 들어, 바닥글이나 SCSSMacro 확장을 지원하는 구성요소(예: 콘텐츠 목록)의 추가 질의 문자열 필드에서, 또는 제목 및 단락 구성요소의 토큰을 통해 사용할 수 있습니다.

이 스크립팅은 런타임에 디자인 동안 SCSRenderAPI와 함께 작동합니다. 또한 사용자정의 사이트 속성 값은 SCSSCompileAPI를 통해 템플릿 컴파일러에서 레이아웃 및 구성요소 코드에 사용할 수 있습니다. 이 API에는 SCSRenderAPI, getCustomSiteProperty와 비슷한 새로운 함수가 있어서 레이아웃 및 구성요소 코드가 사용자정의 섹션 속성의 값을 읽을 수 있습니다.

예를 들어, SCSRenderAPI를 사용하여 머리글, 연락처 이름 및 연락처 전자메일에 대한 사용자정의 사이트 속성을 호출하고 읽어들이는 사용자정의 섹션 레이아웃을 정의해 보겠습니다. 다음 스크립트는 DOM에 추가되는 사용자정의 사이트 속성을 사용하여 HTML 문자열을 작성합니다.

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

사용자정의 사이트 속성이 추가된 사이트에서 템플릿을 생성하거나 사용자정의 사이트 속성이 있는 템플릿에서 사이트를 생성할 때 사용자정의 속성이 보존됩니다.

## 테이블 작업

단락 구성요소 내에서 생성하거나 기존 HTML 소스에서 붙여 넣은 테이블을 포함할 수 있습니다.

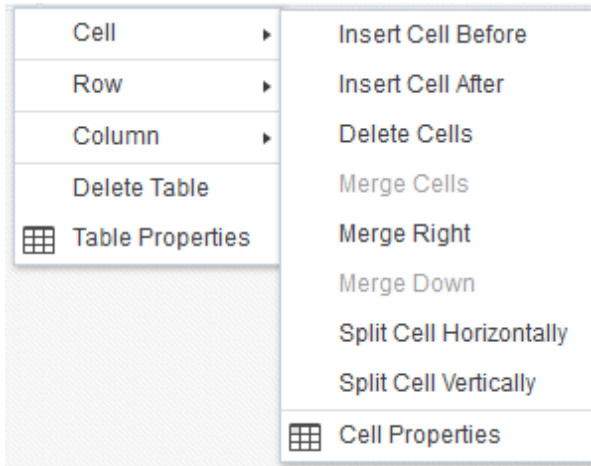
1. 편집할 페이지로 이동하고 가 **편집**으로 설정되어 있는지 확인합니다.
2. 페이지에 구성요소를 추가하거나 기존 단락 구성요소를 누릅니다.
3. 다른 HTML 페이지의 테이블을 추가하려면 테이블을 누른 후 끌어와서 선택하고 복사하여 단락 구성요소로 붙여 넣습니다. 현재 커서 위치에서 새 테이블을 생성하려면 을 누릅니다. 행/열 수, 너비 및 기타 서식 옵션을 선택합니다. 테이블 위의 중앙에 선택적 **표제**와 선택적 **요약**을 추가할 수 있습니다. 그러면 자동화된 스크린 리더와 같은 보조 기술에 추가 컨텍스트를 제공할 수 있습니다.
4. 완료되면 **확인**을 눌러 창을 닫습니다. 나중에 해당 테이블 설정을 변경하려면 테이블에서 마우스 오른쪽 단추를 누르고 **테이블 속성**을 선택합니다. 테이블을 삭제하려면 테이블에서 마우스 오른쪽 단추를 누르고 **테이블 삭제**를 선택합니다.



**주:**

여러 표시 크기(예: 모바일 화면)를 개발하는 경우 StarterTheme design.css 파일에서 발견된 코드를 사용하여 **반응형 테이블 생성**을 수행하면 모바일 장치에서 표시될 때 행 데이터가 누적됩니다.

- 특정 행이나 열, 셀을 추가, 제거 또는 수정하려면 행이나 열, 셀에서 마우스 오른쪽 단추를 누르고 옵션 메뉴 중에서 선택합니다. 예를 들어, 다음과 같습니다.

**주:**

한 번에 하나의 셀에 대해서만 셀 속성을 변경할 수 있습니다.

- 셀 병합 및 분할을 통해 복잡한 테이블 레이아웃을 생성할 수 있습니다. 예를 들어, 다음과 같습니다.

My First Table			
		Split Cells	
Merged cells			

- 한 셀을 두 개의 셀로 분할하려면 셀에서 마우스 오른쪽 단추를 누르고 **셀**을 선택한 다음 **가로로 셀 분할** 또는 **세로로 셀 분할**을 선택합니다.
  - 두 개의 가로 셀을 병합하려면 왼쪽 셀에서 마우스 오른쪽 단추를 누르고 **셀**을 선택한 다음 **오른쪽으로 병합**을 선택합니다.
  - 두 개의 세로 셀을 병합하려면 맨 위 셀에서 마우스 오른쪽 단추를 누르고 **셀**을 선택한 다음 **아래쪽으로 병합**을 선택합니다.
- 일반 서식 메뉴 옵션을 사용하여 셀 내 텍스트의 서식 및 맞춤을 변경합니다.

## 사이트 파일 업로드

Oracle Content Management 인터페이스를 사용하여 언제든지 사이트에 사용할 이미지와 문서를 업로드할 수 있습니다. 또한 배경 이미지나 이미지, 갤러리, 문서 구성요소로 작업할 때 사이트 작성기 내에서 파일을 업로드할 수 있습니다.

사이트에 이미지와 문서 파일을 사용할 때 사이트나 액세스 가능한 다른 위치에 저장된 이미지를 사용할 수 있습니다. 또한 나와 공유된 이미지를 사용하거나 로컬 또는 네트워크 파일 위치에서 업로드한 이미지를 사용할 수 있습니다.

### 파일 업로드


로컬 또는 네트워크 위치에서 하나 이상의 파일을 업로드하려면 다음과 같이 하십시오.

1. **문서**를 누르고 파일을 저장할 위치로 이동합니다. **생성**을 눌러 현재 위치에 새 폴더를 추가합니다.
2. **업로드**를 누릅니다.
3. 하나 이상의 파일을 찾아 선택하고 **열기**를 누릅니다.

### 편집기 내에서 파일 업로드

배경 이미지나 구성요소(예: 문서 또는 이미지)로 작업할 때 사용할 파일이 Oracle Content Management 위치에 없는 경우 파일을 직접 업로드할 수 있습니다.

예를 들어, 갤러리 구성요소에 사용하기 위해 로컬 또는 네트워크 위치에서 하나 이상의 파일을 업로드합니다.

1. 갤러리에 이미지를 추가하려면 메뉴 아이콘 을 누르고, **설정**을 선택하고, **일반 사항** 탭에서 **이미지**를 누릅니다.
2. **이미지 추가**를 누릅니다.
3. 파일을 저장할 위치로 이동하거나 **생성**을 눌러 현재 위치에 새 폴더를 추가합니다.
4. **업로드**를 누릅니다.
5. 하나 이상의 파일을 찾아 선택하고 **열기**를 누릅니다.
6. 저장소에서 하나 이상의 이미지를 선택하고 **확인**을 누릅니다.

#### 주:

모든 사용 가능한 파일이 창에 표시됩니다. 컨텍스트에 적합한 파일 유형을 선택합니다. 예를 들어, 이미지 파일을 선택하는 경우 적합한 이미지 형식의 파일 (GIF, JPG, JPEG, PNG, SVG)을 선택합니다. 파일에 링크하려면 **사이트에 파일을 복사하는 대신 원본 파일의 참조 사용**을 선택합니다. 이 옵션을 선택하지 않을 경우 파일의 복사본이 사이트와 함께 저장되며 사이트에서 참조됩니다.

# 6

## 사이트에서 템플리트 및 테마 사용

템플리트는 사이트 코드 프레임워크, 샘플 페이지와 콘텐츠로 구성된 기본 사이트, 스타일이 적용된 테마, 리소스(예: 이미지)는 물론 사용자정의 구성요소까지 사이트를 시작할 때 필요한 모든 것을 포함합니다. *테마*는 색상표, 글꼴 크기, 글꼴 유형, 페이지 배경 등 일반적인 사이트 모양(전체적 스타일)을 정의합니다.

### 템플리트

- [템플리트 이해](#)
- [사이트에서 템플리트 생성](#)
- [템플리트 세부정보 변경](#)
- [템플리트 정책 변경](#)
- [템플리트 상태 또는 대상 변경](#)
- [템플리트 관리](#)
- [템플리트 익스포트 및 임포트](#)

### 테마

- [테마 이해](#)
- [테마 관리](#)
- [테마 게시](#)

## 템플리트 이해

템플리트는 사이트 코드 프레임워크, 샘플 페이지와 콘텐츠로 구성된 기본 사이트, 스타일이 적용된 테마, 리소스(예: 이미지)는 물론 사용자정의 구성요소까지 사이트를 시작할 때 필요한 모든 것을 포함합니다.

### 기본 템플리트

Oracle Content Management는 사이트 생성에 사용할 수 있는 수많은 템플리트를 제공합니다. 템플리트를 선택하고 사이트 이름을 지정하면 바로 콘텐츠 추가를 시작할 수 있습니다. 이 템플리트는 일반적으로 서비스를 초기화할 때 서비스 관리자에 의해 설치됩니다.

템플리트	설명
빈 템플리트	빈 템플리트는 머리글, 본문, 바닥글 슬롯이 있는 단일 페이지로, 완전히 자유롭게 자신만의 디자인을 사용할 수 있습니다.
참고 자료 시연	참고 자료 시연 템플리트는 오른쪽 맞춤된 가로 메뉴 스타일이며 간단한 페이지 레이아웃, 관련 비디오 자산을 나열하는 최상위 레벨 페이지와 그 뒤로 특정 비디오를 보여주는 중첩된 세부정보 페이지를 제공합니다. 표준 메뉴 구성요소를 제공하며 현재 로그인한 사용자를 보여줍니다.

템플릿	설명
신규 제품 출시	신규 제품 출시 템플릿에는 사례 연구에 대한 가로 오른쪽 맞춤 메뉴와 레이아웃, 기능에 대한 세부정보, 가격, 회사 및 연락처 정보가 포함됩니다. 홈 페이지에는 회전 배너 이미지와 텍스트가 있습니다.
제품 및 서비스 개요	제품 및 서비스 개요 템플릿에는 사례 연구에 대한 레이아웃, 제품 오퍼링에 대한 세부정보, 회사, 개인정보 보호정책 등이 포함됩니다. 홈 페이지에는 회전 배너 이미지와 텍스트가 있습니다. 템플릿은 완전한 반응형입니다.
시작 템플릿	시작 템플릿을 사용하여 고유의 기본 제공 사이트 솔루션을 생성할 수 있습니다. 시작 템플릿은 구성요소 및 상호작용을 탐색하고 확장할 수 있는 간단하고 완전한 기능의 예를 제공합니다. 사이트 코드 프레임워크, 샘플 페이지와 콘텐츠로 구성된 기본 사이트, 스타일이 적용된 테마, 리소스(예: 이미지), 트리거 및 작업 기능을 갖춘 사용자정의 구성요소가 포함됩니다. 샘플 페이지에는 템플릿 생성에 대한 정보와 세부정보를 제공하는 리소스 링크가 포함됩니다.
알아보기 프리젠테이션 쇼	이 템플릿은 서로 색으로 구분됩니다. 반응형이며 그룹 내 사용자정의 JavaScript 메뉴 구성요소를 제공합니다. 다양한 표준 구성요소를 제공하는 사용자정의 구성요소 그룹이 있습니다. 로그인한 사용자도 표시합니다.
액세스 지식 관련 항목	이 템플릿은 서로 색으로 구분됩니다. 반응형이며 그룹 내 사용자정의 JavaScript 메뉴 구성요소를 제공합니다. 다양한 표준 구성요소를 제공하는 사용자정의 구성요소 그룹이 있습니다.
공유	이 템플릿은 하나의 긴 페이지를 제공하는 사이트입니다. 반응형이며 별도의 페이지가 아니라 페이지의 위치를 탐색하는 사용자정의 JavaScript 메뉴를 제공합니다.
JET 시작 템플릿	Oracle JET(JavaScript 확장 툴킷)은 오픈 소스 JavaScript 라이브러리 모음을 사용하는 모듈식 오픈 소스 툴킷입니다. 자세한 내용은 <a href="#">Oracle Jet</a> 를 참조하십시오.

또한 기존 사이트에서 템플릿을 생성할 수도 있고, 기존 템플릿을 익스포트하고 오프라인에서 수정한 후 새 템플릿으로 임포트할 수도 있습니다. 조직에서 멤버가 사용할 템플릿을 생성했을 수 있습니다.

### 템플릿 작동 방식

사이트를 생성할 때 다음과 같이 템플릿이 사용됩니다.

- 사이트 제어가 사용으로 설정된 경우 사이트 생성 전 승인이 필요할 수 있습니다. [사이트 제어 이해](#)를 참조하십시오.
- 템플릿의 기본 사이트가 새 사이트로 복사되어 페이지 시작점을 제공합니다.
- 모든 필요한 지원 파일이 새 사이트로 복사됩니다.
- 템플릿 테마가 테마 폴더에 존재하지 않을 경우 테마 폴더로 복사됩니다. 사이트는 테마 폴더의 해당 위치에서 테마를 참조합니다. 테마가 있는 경우 새 사이트는 단순히 기존 테마를 참조합니다.
- 사용자정의 구성요소가 구성요소 폴더에 존재하지 않을 경우 구성요소 폴더로 복사됩니다. 사이트는 구성요소 폴더의 해당 위치에서 구성요소를 참조합니다. 구성요소가 존재하는 경우 새 사이트는 기존 구성요소를 참조합니다.

## 사용자정의 템플릿 생성

사용자가 웹 개발자인 경우 템플릿이 사이트, 레이아웃, 탐색, 샘플 콘텐츠 등 웹 사이트 구축에 필요한 모든 조각을 수집합니다. 사이트에 구성요소와 상호작용을 추가하여 비즈니스 요구에 맞는 기본 제공 사이트 솔루션을 제공할 수 있습니다.

템플릿은 폴더 구조로 표현되어 다른 폴더와 마찬가지로 작업할 수 있습니다. 테마 및 사용자정의 구성요소와 같은 일부 템플릿 요소는 Oracle Content Management의 연관된 위치에서 참조됩니다. 예를 들어, 사이트가 테마를 참조하는 것과 같이 템플릿은 사용 가능한 테마 목록에서 연관된 테마를 참조합니다.

기존 사이트에서 템플릿을 생성하는 경우 새 템플릿은 사이트 복사본을 기본 사이트로 사용합니다. 템플릿은 사이트에 사용된 테마와 사이트 페이지에 사용된 사용자정의 구성요소를 참조합니다. 테마와 사용자정의 구성요소는 템플릿에 복사되지 않지만, 사이트와 동일한 방법으로 참조됩니다.

### 주:

템플릿에는 템플릿 생성 당시 사용된 사이트가 반영됩니다. 템플릿 생성에 사용된 사이트에 추가 변경사항이 발생할 경우 템플릿과 함께 저장된 사이트에 반영되지 않습니다.

콘텐츠 모델(예: 샘플 콘텐츠) 지원에 필요한 콘텐츠 레이아웃, 콘텐츠 항목 및 디지털 자산을 포함하며 콘텐츠 모델 공유에 사용 가능한 콘텐츠 템플릿을 생성할 수 있습니다. 콘텐츠 템플릿은 게시된 콘텐츠 항목 및 디지털 자산을 사용하여 사이트에서 생성해야 합니다. 콘텐츠가 포함된 패키지에서 사이트 템플릿을 임포트하는 경우 콘텐츠 유형이 생성되지 않습니다. 콘텐츠 유형은 콘텐츠 템플릿을 기반으로 사이트가 생성되는 경우 생성됩니다. 따라서 콘텐츠 유형의 소유자는 사이트 템플릿을 임포트한 사용자가 아닙니다. 콘텐츠 유형의 소유자는 해당 템플릿을 기반으로 첫번째 사이트를 생성한 사용자입니다. 해당 사용자는 필요에 따라 다른 사용자와 템플릿을 공유할 수 있습니다. Content Management Cloud 구독 유형 및 서비스 시작 날짜에 따라 이 기능을 사용하지 못할 수도 있습니다.

템플릿을 익스포트할 때 테마 복사본과 구성요소를 포함한 모든 템플릿 요소가 템플릿 패키지에 수집되고, 이를 다운로드하여 오프라인으로 작업할 수 있습니다.

오프라인에서 수정한 템플릿을 임포트할 경우 템플릿, 테마 또는 사용자정의 구성요소 이름/ID가 존재하면 충돌을 해결하라는 메시지가 표시됩니다. 새 템플릿, 테마 또는 사용자정의 구성요소를 생성하는 옵션이 제시되거나, 어떤 경우 기존 템플릿, 테마 또는 사용자정의 구성요소를 임포트 버전으로 겹쳐쓸 수 있습니다. [템플릿 개발](#)을 참조하십시오.

특정 저장소로 템플릿을 임포트할 수도 있습니다. 이때 기존 자산을 업데이트하거나 복제할 기회가 주어집니다. [특정 저장소로 템플릿 임포트](#)를 참조하십시오.

## 템플릿 공유

내가 생성한 템플릿은 공유하기 전까지 다른 사람이 사이트에서 임포트 또는 복사하거나 생성 기반으로 활용하는 등의 방식으로 사용할 수 없습니다.

 주:

Oracle Content Management와 함께 제공되어 관리자가 설치한 템플릿도 마찬가지로입니다. 템플릿이 표시되지 않을 경우 관리자에게 문의하십시오. 템플릿이 공유되지 않았을 수 있습니다.

처음으로 사용자와 템플릿을 공유하는 경우 관련 테마 및 연관된 사용자정의 구성요소가 자동으로 사용자와 공유되고 다운로드 롤이 부여되어 사용자가 템플릿에서 사이트를 생성할 때 해당 요소를 사용할 수 있습니다. 템플릿에서 해당 사용자의 롤에 후속 변경사항이 발생할 경우 관련 테마 또는 사용자정의 구성요소에 대한 공유 정보가 업데이트되지 않습니다.

사이트 제어가 사용으로 설정된 경우 템플릿 정책을 통해 템플릿을 사용 가능하도록 설정합니다. [사이트 제어 이해](#) 및 [템플릿 정책 변경](#)을 참조하십시오.

## 사이트에서 템플릿 생성

다른 사이트의 시작점으로 사용할 사이트가 있는 경우 해당 사이트에서 템플릿을 생성할 수 있습니다.

 주:


사이트에서 템플릿을 생성하려면 사이트의 다운로드, 제공자 또는 관리자 롤을 보유해야 합니다. 또한 관리자가 **생성** 메뉴의 옵션을 사용으로 설정해야 합니다. 템플릿 페이지에 **생성** 메뉴가 보이지 않으면 관리자에게 문의하십시오.

다음 단계는 사이트에서 템플릿을 생성하는 방법을 보여줍니다. 직접 생성하거나 오프라인에서 수정한 템플릿 패키지를 импорт할 수도 있습니다. [템플릿 익스포트 및 임포트](#)를 참조하십시오.

 주:

여러 저장소의 콘텐츠를 사용하는 사이트에서 템플릿을 생성하는 경우 OCE 툴킷을 사용하여 작업을 수행해야 합니다. [OCE Toolkit으로 개발](#) 및 [cec 명령행 유틸리티 사용](#)을 참조하십시오.

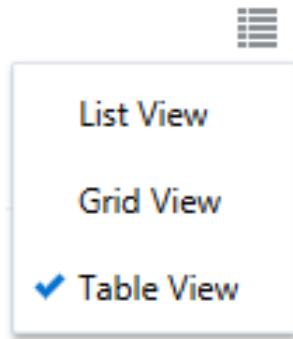
Oracle Content Management 웹 인터페이스를 사용하여 사이트에서 템플릿을 생성하려면 다음과 같이 하십시오.



1. **사이트**를 누르고 사용할 사이트를 선택합니다.
2. 마우스 오른쪽 단추 누르기 메뉴에서 **템플릿 생성**을 선택하거나 작업 모음에서  을 누릅니다.
3. 템플릿의 이름을 입력합니다. 문자, 숫자, 밑줄(\_) 및 하이픈(-)을 사용할 수 있습니다. 공백을 입력하면 자동으로 하이픈으로 바꿉니다.

템플릿, 테마, 구성요소, 사이트 또는 사이트 페이지에 다음 이름을 사용하지 마십시오: authsite, content, pages, scsTEMPLATE\_\*, \_comps, \_components, \_compsdelivery, \_idcSERVICE, \_sitescloud, \_sitesclouddelivery, \_themes, \_themesdelivery. 사이트 페이지에는 다음 이름을 사용할 수 있지만 템플릿, 테마, 구성요소 또는 사이트에는 사용하지 마십시오: documents, sites.

4. 템플릿에 대한 선택적 설명을 입력합니다.
5. 템플릿에 게시되지 않은 콘텐츠 항목 및 디지털 자산을 포함할지 여부를 선택합니다.
6. 준비되면 **템플릿 생성**을 누릅니다.

템플릿이 생성되면 템플릿 목록에 이름이 나타납니다. 모든 템플릿을 보려면 **개발자**, **모든 템플릿 보기**를 차례로 누릅니다. 보기 아이콘을 누르고 목록에서 옵션을 선택하여 템플릿이 표시되는 방법을 제어할 수 있습니다.



템플릿과 연관된 폴더 및 파일을 보려면 마우스 오른쪽 단추 누르기 메뉴에서 **열기**를 선택하거나 작업 모음에서 을 누릅니다. 템플릿에 대한 이름, 설명 및 기타 세부정보를 보거나 변경하려면 템플릿 이름을 누르거나 마우스 오른쪽 단추 누르기 메뉴에서 **세부정보**를 선택하거나 작업 모음에서 을 누릅니다.

새 템플릿은 사이트 복사본을 기본 사이트로 사용합니다. 템플릿은 사이트에 사용된 테마와 사이트 페이지에 사용된 사용자정의 구성요소를 참조합니다. 테마와 사용자정의 구성요소는 템플릿에 복사되지 않지만, 사이트와 동일한 방법으로 참조됩니다.

템플릿이 엔터프라이즈 사이트에서 생성된 경우 기본적으로 템플릿에서 생성된 새 사이트에 대해 원래 사이트에서 사용된 지역화 정책 및 기본 언어가 선택됩니다.

템플릿에는 템플릿 생성 당시 사용된 사이트가 반영됩니다. 템플릿 생성에 사용된 사이트에 추가 변경사항이 발생할 경우 템플릿과 함께 저장된 사이트에 반영되지 않습니다.

## 템플릿 세부정보 변경




템플릿 세부정보 탭에는 템플릿 이름, 작성자, 설명, 템플릿 테마, 템플릿에 포함된 사용자정의 구성요소 및 템플릿 미리보기 이미지가 표시됩니다. 사이트 제어가 사용으로 설정된 경우 템플릿 상태(사이트 생성 시 템플릿을 사용할 수 있는지 여부)도 표시됩니다.

적절한 권한이 있으면 이름 및 설명과 같은 템플릿 속성을 변경하거나 업데이트할 수 있습니다. 템플릿 미리보기 이미지를 추가하거나 제거할 수도 있습니다.

템플릿을 직접 생성했거나(템플릿 소유자), 사이트 관리자이거나, 누군가 나와 템플릿을 공유하고 내게 제공자 또는 관리자 역할을 부여한 경우 템플릿 세부정보를 업데이트할 수 있습니다.

미리보기 이미지를 추가하면 템플릿의 자산 폴더에 파일이 저장됩니다. 해당 파일은 연관된 웹 사이트에 나타나지 않지만, 템플릿 소유자에게 템플릿 자체에 대한 정보를 제공할 수 있습니다.

템플릿 세부정보를 보거나 변경하려면 다음과 같이 하십시오.


1. 개발자를 누르고 모든 템플릿 보기를 누릅니다.
2. 템플릿을 선택하고 마우스 오른쪽 단추 누르기 메뉴에서 세부정보를 선택하거나 작업 모음에서  을 누릅니다.
3. 필요에 따라 정보를 편집합니다. 정보를 편집할 수 없을 경우 제공자 또는 관리자 롤이 없는 것입니다.
4. 미리보기 이미지를 추가하려면  을 누릅니다. 사용할 이미지를 찾거나 업로드를 눌러 로컬에 저장된 이미지를 업로드합니다. 이미지는 최소한 180x135 픽셀(가로 세로 비율 4:3)이어야 합니다. 더 작은 이미지는 최적으로 표시되지 않을 수 있고, 훨씬 큰 이미지는 성능에 영향을 미칠 수 있습니다.  
모든 미리보기 이미지 위치가 채워지면 새로 추가하기 전에 이미지를 삭제해야 합니다.  
미리보기 이미지를 삭제하려면 이미지의  을 누릅니다.
5. 저장을 누릅니다.

## 템플릿 정책 변경

템플릿 정책 탭에는 템플릿 유형(표준 또는 엔터프라이즈), 필요한 승인, 템플릿에서 생성된 사이트에 대한 최소 보안이 표시됩니다. 템플릿 상태(사이트 생성 시 템플릿을 사용할 수 있는지 여부)도 표시됩니다.

사이트 제어가 사용으로 설정되고 사이트 관리자인 경우에만 템플릿 정책 탭이 표시됩니다. [사이트 제어 이해](#)를 참조하십시오.

템플릿 정책을 보거나 변경하려면 다음과 같이 하십시오.

1. 개발자를 누르고 모든 템플릿 보기를 누릅니다.
2. 템플릿을 선택하고 마우스 오른쪽 단추 누르기 메뉴에서 세부정보를 선택하거나 작업 모음에서  을 누릅니다.
3. 정책 탭을 선택합니다.
4. 이 템플릿에서 생성된 사이트 요청을 사이트 관리자가 승인해야 하는지, 자동으로 승인되는지, 또는 특정 사람의 승인이 필요한지 선택합니다. 특정 사람으로 승인을 제한하려면 승인자로 추가할 개인이나 그룹의 이름 또는 전자메일을 입력하기 시작하고 검색 결과에서 개인이나 그룹을 선택합니다. 승인자를 제거하려면 이름 옆에 있는 X를 누릅니다.
5. 이 템플릿에서 생성된 사이트에 필요한 최소 보안을 선택합니다. 사이트 생성자는 원하는 경우 사이트에 대해 보다 높은 보안 레벨을 선택할 수 있습니다.
  - **특정 서비스 사용자** - 이 Oracle Content Management 인스턴스에 사인인할 수 있는 선택된 사용자만 사이트에 액세스할 수 있습니다. 사이트가 생성된 후 사이트 생성자가 Oracle Content Management 사용자를 선택합니다. [사이트 보안 변경](#)을 참조하십시오.
  - **특정 클라우드 사용자** - 도메인에 사인인할 수 있는 선택된 사용자만 사이트에 액세스할 수 있습니다. 사이트가 생성된 후 사이트 생성자가 클라우드 사용자를 선택합니다. [사이트 보안 변경](#)을 참조하십시오.
  - **서비스 사용자** - 이 Oracle Content Management 인스턴스에 사인인할 수 있는 모든 사용자가 사이트에 액세스할 수 있습니다.



- **클라우드 사용자** - 도메인에 사인인할 수 있는 모든 사용자가 사이트에 액세스할 수 있습니다.
  - **모든 사람** - 사인인하지 않고도 누구나 사이트에 액세스할 수 있습니다.
6. 만료 정책을 선택하여 템플릿을 사용하는 사이트가 만료될 시기를 결정합니다. 사이트 관리자는 만료된 사이트를 오프라인으로 전환하거나 삭제할지 여부를 지정할 수 있습니다. 사이트 소유자 및 관리자는 사이트 만료 전에 전자메일로 통지를 받고 만료 시간을 연장할 기회가 주어집니다. 만료를 연장하지 않으면 사이트가 오프라인으로 전환되거나 삭제되었을 때 전자메일로 통지를 받습니다. 이 때 만료를 연장하고 사이트를 다시 온라인으로 전환하거나 필요한 경우 휴지통에서 복원할 수 있습니다. 만료 정책 옵션은 다음과 같습니다.
- **안함**
  - **1년**
  - **2년**
  - **사용자정의**  
사용자정의를 선택하면 1년보다 짧거나 2년보다 길게 만료 기간을 설정할 수 있습니다.

 **주:**

사이트가 생성된 후 사이트 관리자는 **사이트 속성** 대화상자에서 사이트의 만료 정책을 변경할 수 있습니다.


7. 엔터프라이즈 템플릿을 편집하는 경우, 친화적 URL 값을 위해 사이트 접두어 생성 방법을 선택할 수 있습니다. 이 접두어는 콘텐츠 항목 슬러그 값(페이지나 자산에 특정한 URL 부분)에 추가됩니다. 사이트 이름을 기준으로 접두어가 자동으로 생성되도록 하거나 사용자가 접두어를 입력하도록 허용할 수 있습니다.
8. 엔터프라이즈 템플릿을 편집하는 경우, 사이트에 필요한 언어를 결정하는 데 사용할 지역화 정책을 선택합니다.
9. 완료되면 **저장**을 누릅니다.

## 템플릿 상태 또는 대상 변경

템플릿 대상 탭에는 템플릿 상태(사이트 생성 시 템플릿을 사용할 수 있는지 여부)와 템플릿을 사용하여 사이트를 생성할 수 있는 사람이 표시됩니다.

사이트 제어가 사용으로 설정되고 사이트 관리자인 경우에만 템플릿 대상 탭이 표시됩니다. [사이트 제어 이해](#)를 참조하십시오.

템플릿 대상을 보거나 변경하려면 다음과 같이 하십시오.

1. **개발자**를 누르고 **모든 템플릿 보기**를 누릅니다.
2. 템플릿을 선택하고 마우스 오른쪽 단추 누르기 메뉴에서 **세부정보**를 선택하거나 작업 모음에서  을 누릅니다.
3. **대상자** 탭을 선택합니다.
4. 템플릿 상태를 변경하려면 상태 스위치를 누릅니다. 사용자가 템플릿을 기반으로 사이트를 생성하려면 템플릿을 활성화해야 합니다. 기본적으로 템플릿을 활성화하면 모든 사이트 생성자가 사용할 수 있습니다. 템플릿을 사용할 수 있는 사람을 변경하려면 이 탭의 설정을 변경합니다.
5. 이 템플릿을 사용하여 사이트를 생성할 수 있는 사람을 선택합니다.

- **특정 사람**—선택한 사람만 이 템플릿을 사용하여 사이트를 생성할 수 있습니다. 이 템플릿을 사용할 수 있는 개인이나 그룹의 이름 또는 전자메일을 입력하기 시작하고 검색 결과에서 개인이나 그룹을 선택합니다. 개인이나 그룹을 제거하려면 이름 옆에 있는 **X**를 누릅니다.
  - **모든 사람**—사이트를 생성할 수 있는 누구나 이 템플릿을 사용할 수 있습니다.
6. 완료되면 **저장**을 누릅니다.

## 템플릿 관리

다른 폴더와 마찬가지로 템플릿 폴더를 복사하고 이름을 바꾸고 삭제할 수 있지만, 템플릿을 임포트하거나 공유할 때 특별한 고려 사항이 있습니다.



**개발자, 모든 템플릿 보기**를 차례로 선택하여 템플릿 페이지를 엽니다. 템플릿 페이지에서 **작업 모음** 또는 마우스 오른쪽 단추 누르기 메뉴의 옵션을 사용하여 다음 작업을 수행할 수 있습니다.


작업	설명
템플릿 생성	다른 사이트의 시작점으로 사용할 사이트가 있는 경우 해당 사이트에서 템플릿을 생성할 수 있습니다.

### 주:

사이트의 다운로드, 제공자 또는 관리자 롤을 보유해야 하고 관리자가 옵션을 사용하여 설정해야 합니다. 템플릿 페이지에 **생성** 메뉴가 보이지 않으면 관리자에게 문의하십시오. 또한 여러 저장소의 콘텐츠를 사용하는 사이트에서 템플릿을 생성하는 경우 OCE 툴킷을 사용하여 작업을 수행해야 합니다. [OCE Toolkit으로 개발](#) 및 [cec 명령행 유틸리티 사용](#)을 참조하십시오.


템플릿 페이지에서 **생성**을 누르고 **기존 사이트에서 생성**을 선택합니다. 사이트를 선택하고 템플릿 이름을 지정하고 **생성**을 누릅니다. [사이트에서 템플릿 생성](#)을 참조하십시오.

템플릿 파일 보기 및 편집	<p>템플릿을 구성하는 폴더 및 파일을 탐색할 수 있습니다. 템플릿을 선택하고 마우스 오른쪽 단추 누르기 메뉴에서 <b>열기</b>를 선택하거나 작업 모음에서  을 누르면 됩니다.</p> <p>웹 개발자의 경우 개별 템플릿 파일을 다운로드하여 수정하거나, 데스크톱 앱을 사용하여 로컬 시스템에서 수행한 변경사항을 동기화할 수 있습니다. 또한 템플릿 패키지(관련 테마와 사용자정의 구성요소 포함)를 익스포트하고, 원하는 개발 환경에서 오프라인으로 작업할 수 있습니다.</p> <p>템플릿은 폴더 구조로 표현되어 다른 폴더와 마찬가지로 작업할 수 있습니다. 템플릿을 익스포트할 때 테마 복사본과 구성요소를 포함한 모든 템플릿 요소가 템플릿 패키지에 수감되고, 이를 다운로드하여 오프라인으로 작업할 수 있습니다. <a href="#">템플릿 개발</a>을 참조하십시오.</p>
사이트 생성	<p>템플릿에서 사이트를 생성하려면 템플릿을 선택하고 마우스 오른쪽 단추 누르기 메뉴에서 <b>사이트 생성</b>을 선택하거나 작업 모음에서  을 누릅니다. <a href="#">사이트 생성</a>을 참조하십시오.</p>

작업	설명
템플릿 이름 바꾸기	이름을 바꿀 템플릿을 선택하고 마우스 오른쪽 단추 누르기 메뉴에서 <b>이름 바꾸기</b> 를 선택하거나 작업 모음에서  을 누릅니다. 동일 서버의 다른 템플릿과 다르게 템플릿의 이름을 입력합니다. 이름 지정 지침은 <a href="#">사이트에서 템플릿 생성</a> 을 참조하십시오.
템플릿 복사	기존 템플릿을 복사하고 사본을 변경하여 템플릿을 생성할 수 있습니다.

 주:

템플릿을 복사할 때 템플릿과 연관된 공유 정보는 복사되지 않습니다.

복사할 템플릿을 선택하고 마우스 오른쪽 단추 누르기 메뉴에서 **복사**를 선택하거나 작업 모음에서  을 누릅니다. 동일 서버의 다른 템플릿과 다르게 템플릿의 이름을 입력합니다. 기타 이름 지정 지침은 [사이트에서 템플릿 생성](#)을 참조하십시오.


진행을 표시줄에 새 템플릿 이름 및 복사 상태가 표시됩니다.

템플릿 삭제 또는 복원	적절한 권한이 있으면 템플릿 폴더와 콘텐츠를 삭제할 수 있습니다. 템플릿을 삭제하면 템플릿 폴더와 모든 관련 폴더 및 파일이 휴지통으로 이동합니다.
--------------	--

템플릿을 직접 생성했거나(템플릿 소유자) 누군가 나와 템플릿을 공유하고 내게 제공자 또는 관리자 롤을 부여한 경우 템플릿을 삭제하거나 복원할 수 있습니다.



 주:



템플릿을 삭제할 때 관련 테마와 사용자정의 구성요소는 삭제되지 않습니다.

템플릿을 삭제하려면 삭제할 템플릿을 선택하고 마우스 오른쪽 단추 누르기 메뉴에서 **삭제**를 선택하거나 작업 모음에서  을 누릅니다. 템플릿을 휴지통으로 이동할지 묻는 메시지가 표시됩니다. 다음 이벤트 전까지 삭제된 템플릿이 휴지통에 보존됩니다.

- 템플릿을 복원합니다.
- 템플릿을 영구적으로 삭제합니다.
- 휴지통 할당량에 도달합니다.
- 서비스 관리자가 설정한 간격에 따라 자동으로 휴지통이 비워집니다. 기본값은 90일입니다.

템플릿을 복원하려면 [템플릿] 드롭다운 목록에서 **휴지통**을 선택한 다음 템플릿을 선택하고 메뉴 모음 또는 마우스 오른쪽 단추 누르기 메뉴에서 **복원**을 선택합니다.

작업	설명
템플릿 익스포트 또는 임포트	<p>템플릿을 익스포트하고 오프라인에서 수정한 후 새 템플릿으로 임포트하거나 기존 템플릿을 바꿀 수 있습니다. 또한 템플릿을 익스포트하여 다른 인스턴스로 이동하고 거기서 임포트할 수도 있습니다.</p> <p>템플릿을 익스포트할 때 폴더에 단일 .zip 파일로 템플릿을 복사합니다. 폴더에서 직접 템플릿 패키지를 다운로드하여 압축을 풀고 개별 파일로 작업할 수 있습니다. 파일 작업이 완료되면 템플릿 패키지가 포함된 .zip 파일을 생성하여 임포트하고 원본 템플릿을 겹쳐쓰거나 새로 생성합니다.</p> <p><a href="#">템플릿 익스포트 및 임포트</a>을 참조하십시오.</p>
템플릿 공유 및 멤버 롤 관리	<p>템플릿 소유자나 관리자인 경우 다른 사용자와 템플릿을 공유하고 사용자가 템플릿으로 수행할 수 있는 작업을 정의하는 롤을 지정할 수 있습니다.</p> <p>제어가 사용으로 설정되지 않은 경우 사용자에게 롤을 지정하면 템플릿에서 사이트를 생성할 권한이 부여됩니다. 제어가 사용으로 <i>설정된</i> 경우 템플릿을 활성화하고 사용자가 템플릿 대상에 포함되었는지 확인하여 템플릿에서 사이트를 생성할 권한을 부여해야 합니다. <a href="#">템플릿 상태 또는 대상 변경</a>을 참조하십시오.</p> <p>템플릿의 테마와 사용자정의 구성요소는 별도의 객체로 저장되며 템플릿에서 참조될 뿐입니다. 따라서 테마와 사용자정의 구성요소 멤버십은 별도로 관리됩니다. 사용자와 템플릿을 공유하는 경우 템플릿에 부여된 롤에 관계없이 사용자에게 관련 테마와 사용자정의 구성요소에 대한 다운로드 롤이 자동으로 부여되어 사용자가 템플릿에서 사이트를 생성할 때 해당 객체를 사용할 수 있습니다.</p> <p>공유할 템플릿을 선택하고 마우스 오른쪽 단추 누르기 메뉴 또는 작업 모음에서 <b>멤버</b>를 선택합니다.</p> <p>사용자 이름 또는 전자메일 주소를 하나 이상 입력하고 다음 롤 중 하나를 지정합니다.</p> <ul style="list-style-type: none"> <li>• <b>열람자:</b> 열람자는 템플릿의 폴더 및 파일을 볼 수 있지만 어떤 내용도 변경할 수 없습니다.</li> <li>• <b>다운로더:</b> 다운로더는 파일을 다운로드하여 자신의 컴퓨터에 저장할 수도 있습니다.</li> <li>• <b>작성자:</b> 작성자는 템플릿 세부정보와 템플릿 파일을 편집하고 새 파일을 업로드하며 파일을 삭제하고 템플릿 자체를 삭제할 수도 있습니다.</li> <li>• <b>관리자:</b> 관리자는 사용자를 추가하고 사용자 롤을 지정할 수 있습니다. 템플릿 소유자(생성자)에게는 자동으로 관리자 롤이 지정됩니다.</li> </ul>
템플릿 세부정보 확인 또는 편집	<p>템플릿 세부정보를 보려면 템플릿을 선택하고 마우스 오른쪽 단추 누르기 메뉴에서 <b>세부정보</b>를 선택하거나 작업 모음에서  을 누릅니다. 템플릿 세부정보에는 템플릿 이름, 작성자, 지원 파일, 설명, 테마, 사용자정의 구성요소 및 작은 그림이 포함됩니다. 적절한 템플릿 롤이 있으면 세부정보를 편집할 수 있습니다. <a href="#">템플릿 세부정보 변경</a>을 참조하십시오.</p>
템플릿 정책 보기 또는 편집 (제어가 사용으로 설정된 경우)	<p>제어가 사용으로 설정되고 사이트 관리자인 경우 필요한 승인, 최소 보안, 사이트 만료 및 지역화와 같은 템플릿 정책을 볼 수 있습니다. 템플릿을 선택하고 마우스 오른쪽 단추 누르기 메뉴에서 <b>세부정보</b>를 선택하거나 작업 모음에서  을 누른 다음 <b>정책</b> 탭을 선택합니다. <a href="#">템플릿 정책 변경</a>을 참조하십시오.</p>

작업	설명
사이트 생성용으로 템플릿 제공 및 사용하여 사이트를 생성할 수 있는 사람 관리(제어가 사용으로 설정된 경우)	제어가 사용으로 설정되고 사이트 관리자인 경우 사이트 생성용으로 템플릿을 제공하고 템플릿을 사용하여 사이트를 생성할 수 있는 사람을 관리할 수 있습니다. 활성화(사이트 생성용으로 제공) 템플릿은 옆에  이 표시됩니다. 상태 또는 대상자를 변경하려면 템플릿을 선택하고 마우스 오른쪽 단추 누르기 메뉴에서 세부정보를 선택하거나 작업 모음에서  을 누릅니다. <ul style="list-style-type: none"> <li>• 사이트 생성용으로 템플릿을 제공하려면 상태 스위치를 눌러 활성화합니다.</li> <li>• 템플릿을 사용하여 사이트를 생성할 수 있는 사람을 변경하려면 대상 탭을 누릅니다. <a href="#">템플릿 상태 또는 대상 변경</a>을 참조하십시오.</li> </ul>

## 템플릿 익스포트 및 임포트

템플릿을 익스포트하고 오프라인에서 수정한 후 새 템플릿으로 임포트하거나 기존 템플릿을 바꿀 수 있습니다. 또한 템플릿을 익스포트하여 다른 Oracle Content Management 인스턴스로 이동하고 거기서 임포트할 수도 있습니다.

### 템플릿 익스포트

템플릿을 익스포트할 때 반드시 Oracle Content Management의 폴더에 단일 .zip 파일로 템플릿을 복사해야 합니다. 폴더에서 직접 템플릿 패키지를 다운로드하여 압축을 풀고 개별 파일로 작업할 수 있습니다. 템플릿 파일 작업이 완료되면 템플릿 패키지가 포함된 .zip 파일을 생성하여 사이트로 임포트하고 원본 템플릿을 겹쳐쓰거나 새로 생성합니다.

#### 주:

템플릿을 익스포트할 때 템플릿 공유 정보는 포함되지 않습니다.

템플릿을 익스포트하려면 다음과 같이 하십시오.

1. **개발자**를 누르고 **모든 템플릿 보기**를 누릅니다.
2. 템플릿을 선택하고 메뉴 모음 또는 마우스 오른쪽 단추 누르기 메뉴에서 **익스포트**를 선택합니다.
3. 기존 폴더로 이동하거나, **생성**을 누르고 이름과 선택적 설명을 제공하여 새 폴더를 생성합니다.
4. 체크박스를 선택하고 **확인**을 눌러 폴더를 선택합니다.

## 템플리트 импорт

### 주:

다른 서버로 템플리트를 импорт하는 경우 기본 사이트의 일부 링크가 새 서버 컨텍스트에서 부적합할 수 있습니다. 사이트에서 콘텐츠를 사이트로 직접 복사하지 않고 이미지 또는 기타 콘텐츠에 대한 참조 링크를 사용하는 경우 해당 콘텐츠는 새 서버에서 사용할 수 없습니다. 콘텐츠를 새 서버로 복사하는 경우에도 콘텐츠는 다른 내부 ID를 갖게 되어 이 링크가 부적합하게 됩니다. 템플리트를 импорт하는 경우 부적합한 참조 링크가 포함되어 있는 페이지에 대한 통지를 받게 됩니다.

템플리트 패키지를 импорт하려면 다음과 같이 하십시오.

1. 개발자를 누르고 **모든 템플리트 보기**를 누릅니다.
2. **생성**을 누르고 **템플리트 패키지 импорт**를 선택합니다.
3. 템플리트 패키지를 업로드한 경우 템플리트 패키지가 포함된 폴더로 이동하여 폴더를 엽니다. 템플리트 패키지를 아직 업로드하지 않은 경우 업로드할 폴더로 이동하거나 새 폴더를 생성합니다. **업로드**를 누르고 템플리트 패키지를 찾아 선택하고 **열기**를 누릅니다.
4. 템플리트를 사용하려면 템플리트 파일 이름 옆에 있는 체크박스를 누르고 **확인**을 누릅니다. 템플리트, 관련 테마와 사용자정의 구성요소에 대한 새 폴더가 생성됩니다. 템플리트, 테마 또는 사용자정의 구성요소 이름/ID가 존재하면 충돌을 해결하라는 메시지가 표시됩니다. 새 템플리트, 테마 또는 구성요소를 생성해야 할 수 있습니다. 또는 기존 항목을 импорт 버전으로 겹쳐쓸 수 있습니다.

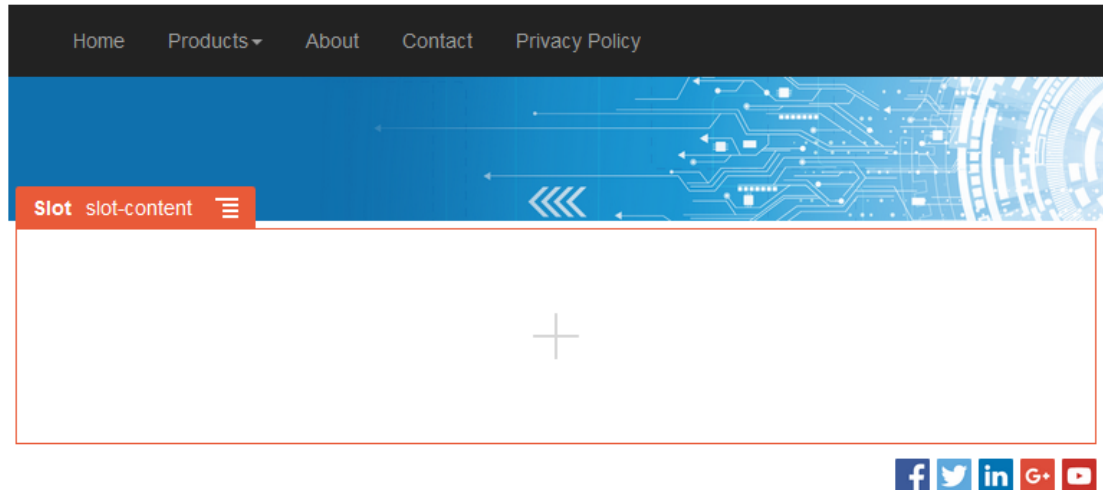
### 특정 저장소로 템플리트 импорт

저장소를 선택하고 특별히 템플리트 패키지를 импорт할 수 있습니다.

1. **콘텐츠**를 누르고 메뉴에서 **저장소**를 선택합니다.
2. 템플리트 패키지를 импорт할 저장소를 선택하고 **콘텐츠 импорт**를 누릅니다.
3. 템플리트 패키지를 업로드한 경우 템플리트 패키지가 포함된 폴더로 이동하여 폴더를 엽니다. 템플리트 패키지를 아직 업로드하지 않은 경우 업로드할 폴더로 이동하거나 새 폴더를 생성합니다. **업로드**를 누르고 템플리트 패키지를 찾아 선택하고 **확인**을 누릅니다.
4. 저장소에 이미 있는 자산을 새 개정으로 업데이트할지 또는 자산의 복제본을 만들지 선택하고 **импорт**를 누릅니다.
5. **세부정보**를 눌러 импорт된 자산 및 콘텐츠 유형 목록을 확인합니다. 완료되면 **확인**을 누릅니다.

## 테마 이해

*테마*는 색상표, 글꼴 크기, 글꼴 유형, 페이지 배경 등 일반적인 사이트 모양(전체적 스타일)을 정의합니다. 테마는 사이트 내 페이지 간에 시각적 일관성을 제공합니다. 디자인을 조정하고 콘텐츠를 추가하여 나만의 스타일, 브랜드, 비전이 담긴 사이트를 생성할 수 있습니다.



테마는 다음을 포함합니다.

- 페이지 레이아웃
- CSS(중속형 스타일시트)
- 스타일시트의 변형
- 구성 파일
- 사이트 탐색을 정의하는 배경 코드

각 웹 사이트는 테마를 사용합니다. 템플릿에서 사이트를 생성할 때 템플릿에서 테마를 상속합니다. 언제든지 사이트의 테마를 변경할 수 있습니다.

Oracle Content Management는 시작할 때 사용할 수 있는 수많은 템플릿과 테마를 제공합니다. 새 테마를 생성하려면 기존 테마를 복사합니다. 테마 파일을 다운로드하여 수정할 수도 있고, 데스크톱 앱을 사용하여 로컬 시스템에서 만든 변경사항을 동기화할 수도 있습니다. 테마를 생성하는 다른 방법에 대한 자세한 내용은 [테마 개발](#)을 참조하십시오.

또한 개발자가 특정 구성요소 부분 집합을 사용하는 테마를 생성할 수 있습니다. 이러한 테마 중 하나를 사용하여 사이트를 생성할 때는 해당 테마에서 허용되는 구성요소만 표시됩니다. 따라서 선택한 테마에 구성요소가 적합한지 여부를 걱정할 필요가 없습니다. 테마 구성요소와 함께 시드된 구성요소를 계속 사용할 수 있습니다.

사이트에 게시되지 않은 새로운 테마를 사용하는 경우, 처음 사이트를 온라인에 놓을 때 테마가 자동으로 사이트에 게시됩니다. 테마를 변경하고 온라인 사이트를 업데이트하여 변경사항을 표시하려면 테마를 명시적으로 게시해야 합니다. 테마 소유자나 관리자 권한을 가진 사용자만 테마를 명시적으로 게시할 수 있습니다.

#### 주:

테마에 변경사항을 게시하는 경우 테마를 사용하는 모든 온라인 사이트에 변경사항이 반영됩니다. 예를 들어, 테마의 기본 글꼴을 변경하고 테마를 게시하는 경우 테마를 사용하는 모든 사이트는 새로운 기본 글꼴을 사용합니다.

## 테마 관리

다른 폴더와 마찬가지로 테마 폴더를 복사 및 삭제하고 이름을 바꿀 수 있지만, 테마는 둘 이상의 사이트에서 공유될 수 있으므로 테마의 경우 특수한 고려사항이 제안됩니다.

**개발자**를 누르고 **모든 테마 보기**를 눌러 모든 테마를 볼 수 있습니다. [테마] 페이지의 메뉴 모음 또는 마우스 오른쪽 단추 누르기 메뉴의 옵션을 사용하여 다음 작업을 수행할 수 있습니다.

작업	설명
새 테마 생성	새 테마를 생성하려면 기존 테마를 복사합니다. 다른 테마 생성 방법에 대한 자세한 내용은 <a href="#">테마 개발</a> 을 참조하십시오.
테마 복사	기존 테마를 복사하고 사본을 변경하여 새 테마를 생성할 수 있습니다.



### 주:

테마를 복사할 때 테마와 관련된 공유 정보는 복사되지 않습니다.

복사할 테마를 선택하고 메뉴 모음 또는 마우스 오른쪽 단추 누르기 메뉴에서 **복사**를 선택합니다. 다른 테마에 사용되지 않는 이름을 입력합니다. 문자, 숫자, 밑줄(\_) 및 하이픈(-)을 사용할 수 있습니다. 공백을 입력하면 자동으로 하이픈으로 바뀝니다. **복사**를 누릅니다.

템플릿, 테마, 구성요소, 사이트 또는 사이트 페이지에 다음 이름을 사용하지 마십시오: authsite, content, pages, scstemplate\_\*, \_comps, \_components, \_compsdelivery, \_idcservice, \_sitescloud, \_sitesclouddelivery, themes, \_themesdelivery. 사이트 페이지에는 다음 이름을 사용할 수 있지만 템플릿, 테마, 구성요소 또는 사이트에는 사용하지 마십시오: documents, sites.



작업	설명
테마 공유	<p>관리자가 공유를 사용으로 설정했고 테마 소유자나 관리자인 경우 다른 사용자와 테마를 공유할 수 있습니다. 테마를 공유하는 경우 사용자가 테마로 수행할 수 있는 작업을 정의하는 룰을 지정합니다.</p> <div style="border: 1px solid #0070C0; padding: 10px; margin: 10px 0;"> <p> <b>주:</b></p> <p>누군가 사용자와 템플리트를 공유하는 경우 템플릿에 부여된 룰에 관계없이 사용자에게 관련 테마에 대한 다운로드 룰이 자동으로 부여되어 사용자가 템플릿에서 사이트를 생성할 때 테마를 사용할 수 있습니다.</p> </div> <p>공유할 테마를 마우스 오른쪽 단추로 누르고 <b>공유</b>를 선택한 다음 <b>멤버 추가</b>를 누릅니다. 사용자 이름 또는 전자메일 주소를 하나 이상 입력하고 다음 룰 중 하나를 지정합니다.</p> <ul style="list-style-type: none"> <li>• <b>열람자:</b> 열람자는 테마의 폴더 및 파일을 볼 수 있지만 어떤 내용도 편집할 수 없습니다.</li> <li>• <b>다운로더:</b> 다운로더는 파일을 다운로드하여 자신의 컴퓨터에 저장할 수도 있습니다.</li> <li>• <b>작성자:</b> 작성자는 테마를 편집하고 새 파일을 업로드하며 파일을 삭제하고 테마 자체를 삭제할 수 있습니다.</li> <li>• <b>관리자:</b> 관리자는 사용자를 추가하고 사용자 룰을 지정할 수 있습니다. 테마 생성자(소유자)에게는 자동으로 관리자 룰이 지정됩니다.</li> </ul>
테마 편집	<p>웹 개발자인 경우 개별 테마 파일을 다운로드 및 수정할 수도 있고, 데스크톱 앱을 사용하여 로컬 시스템에서의 변경사항을 동기화할 수도 있습니다.</p> <p>연관된 테마 및 사용자정의 구성요소를 포함하는 템플릿 패키지를 익스포트하고 선호 개발 환경에서 오프라인으로 작업할 수도 있습니다. <b>테마 개발</b>을 참조하십시오.</p> <p>테마를 변경할 경우 테마를 사용하는 사이트에서 변경사항이 반영되도록 테마를 게시해야 합니다.</p>
테마 게시	<p>사이트에 게시되지 않은 새로운 테마를 사용하는 경우, 처음 사이트를 온라인에 놓을 때 테마가 자동으로 사이트에 게시됩니다. 테마를 변경하고 온라인 사이트를 업데이트하여 변경사항을 표시하려면 테마를 명시적으로 게시해야 합니다. 내가 소유자이거나 관리자 룰이 지정된 경우 테마를 게시할 수 있습니다. 목록에서 테마를 선택하고 메뉴 모음에서 <b>게시</b>(또는 이전에 게시된 테마의 경우 <b>재게시</b>)를 누릅니다. 목록에서 테마 옆에 게시됨 아이콘  이 추가됩니다. <b>테마 게시</b>를 참조하십시오.</p>

작업	설명
테마 삭제 또는 복원	적절한 권한이 있으면 테마 폴더와 콘텐츠를 삭제할 수 있습니다. 테마를 삭제하면 테마 폴더와 모든 관련 폴더 및 파일이 휴지통으로 이동합니다. 테마를 직접 생성했거나(테마 소유자) 누군가 나와 테마를 공유하고 내게 제공자 또는 관리자 롤을 부여한 경우 테마를 삭제하거나 복원할 수 있습니다.

 주:

사이트에서 사용 중인 테마는 삭제할 수 없습니다.

테마를 삭제하려면 삭제할 테마를 마우스 오른쪽 단추로 누르고 **삭제**를 선택합니다. 휴지통으로 테마를 이동할지 묻는 메시지가 표시됩니다. 다음 전까지 삭제된 테마 폴더가 휴지통에 보존됩니다.

- 테마를 복원합니다.
- 테마를 영구적으로 삭제합니다.
- 휴지통 할당량에 도달합니다.
- 서비스 관리자가 설정한 간격에 따라 자동으로 휴지통이 비워집니다. 기본값은 90일입니다.

테마를 복원하려면 테마 페이지 메뉴 모음에서 **휴지통**을 누른 다음 목록에서 테마를 마우스 오른쪽 단추로 누르고 **복원**을 선택합니다.

## 테마 게시


테마는 일반적인 사이트 모양을 정의합니다. 테마를 업데이트하여 테마를 사용하는 사이트 모양을 변경할 수 있습니다.

사이트에 게시되지 않은 새로운 테마를 사용하는 경우, 처음 사이트를 온라인에 놓을 때 테마가 자동으로 사이트에 게시됩니다. 테마를 변경하고 온라인 사이트를 업데이트하여 변경사항을 표시하려면 테마를 명시적으로 게시해야 합니다. 테마에 변경사항을 게시하려면 테마 소유자이거나 관리자 롤을 보유해야 합니다.

 주:

테마에 변경사항을 게시하는 경우 테마를 사용하는 모든 온라인 사이트에 변경사항이 반영됩니다. 테마에 업데이트를 게시하기 전에 오프라인으로 변경사항을 테스트했고 연관된 사이트에 미치는 영향을 이해했는지 확인합니다.

테마를 게시하려면 다음과 같이 하십시오.

1. **개발자, 모든 테마 보기를** 차례로 누릅니다.
2. 테마 목록에서 기존 테마를 선택합니다.
3. 메뉴 모음 또는 마우스 오른쪽 단추 누르기 메뉴에서 **게시**를 누릅니다.
4. **진행 확인**을 누르고 **확인**을 누릅니다. 목록에서 테마 옆에 게시됨 아이콘 이 추가됩니다.

# 7

## 사용자정의 구성요소 및 레이아웃 관리

개발자는 사용자정의 구성요소 및 레이아웃을 생성하고 관리할 수 있습니다. 사이트 제공자는 타사 구성요소(앱) 및 구성요소 그룹을 등록할 수 있습니다.

사용자정의 구성요소에는 편집기에서 생성한 구성요소 그룹과, 이 절에 설명된 옵션을 사용하여 생성한 로컬 및 원격 구성요소가 포함됩니다. 사용자정의 레이아웃에는 페이지의 슬롯에 구성요소를 배열하기 위한 *섹션 레이아웃*과 콘텐츠 항목에 필드를 배열하기 위한 *콘텐츠 레이아웃*이 포함됩니다.

- [사용자정의 구성요소 이해](#)
- [레이아웃 이해](#)
- [사용자정의 구성요소 및 레이아웃 사용](#)
- [원격 구성요소 등록](#)
- [로컬 구성요소, 레이아웃, 콘텐츠 필드 편집기 또는 콘텐츠 폼 생성](#)
- [구성요소 또는 레이아웃 익스포트 또는 임포트](#)



사이트에서 구성요소를 사용하는 방법을 알아보려면 [페이지 콘텐츠 배열](#)을 참조하십시오.

개별 구성요소 사용에 대한 자세한 내용은 [내장 구성요소 사용](#)을 참조하십시오.

고유 구성요소 생성 방법에 대한 자세한 내용은 [구성요소 개발](#)을 참조하십시오.

## 사용자정의 구성요소 이해

*구성요소*는 웹 페이지의 개별 부품입니다. 웹 페이지를 볼 때 무엇이 보입니까? 몇몇 제목, 일부 단락 텍스트, 사이트의 다른 페이지로 가는 여러 링크가 보일 것입니다. 이미지, 단추, 구분선, 맵, 갤러리가 보일 수도 있습니다. 해당 각 항목이 구성요소입니다.

구성요소를 페이지에 추가하려면 가 **편집**으로 설정되었는지 확인하고 을 누르고 사용하려는 구성요소 유형을 선택합니다.

패널에서 구성요소를 끌어서 페이지상 슬롯에 놓습니다. 이러면 끝입니다. 제목, 단락, 이미지 및 기타 구성요소를 페이지의 원하는 위치로 끌어 놓으십시오.

원격 구성요소(앱)를 쉽게 등록/통합하고 구성요소 관리자의 옵션을 사용하여 고유의 구성요소를 생성할 수 있습니다. **개발자**를 누르고 **모든 구성요소 보기**를 누릅니다.

**생성**을 누르고 새 로컬 구성요소를 생성하거나 원격 구성요소를 등록하는 관련 옵션을 선택합니다. 이 방법으로 내가 생성하고 공유한 구성요소는 편집기의 사용자정의 구성요소 패널에 나열됩니다.


고유 구성요소 생성 방법에 대한 자세한 내용은 [구성요소 개발](#)을 참조하십시오.

### 원격 구성요소

타사 구성요소(앱)를 사용하려면 그냥 등록하고 사이트에서 사용하면 됩니다. 아주 쉽습니다.

원격 구성요소를 등록할 때 원격 구성요소 자체의 URL과 웹 저작자가 원격 구성요소에 지정할 수 있는 설정을 위한 두번째 URL을 지정합니다.

원격 구성요소를 등록한 후에 다른 사용자와 공유할 수 있습니다. 내가 소유하거나 공유한 등록된 원격 구성요소는 사이트 작성기의 [사용자정의 구성요소] 패널에 나열됩니다.

다른 구성요소에 대해 수행하는 것과 동일한 방식으로 편집기 내에서 원격 구성요소에 대한 속성을 변경할 수 있습니다. 메뉴 아이콘  을 누르고 **설정**을 선택하면 됩니다. **사용자정의 단추**를 누르면 원격 구성요소를 등록할 때 지정한 설정 URL이 열립니다.

#### 주:

원격 구성요소는 Oracle Content Management 호스트 서버가 아닌 다른 서버에서 호스트되기 때문에 보안상 인라인 프레임(`iframe` 요소 사용)으로 묶입니다. 모든 원격 구성요소가 인라인 프레임으로 묶일 수 있는 것은 아닙니다. 제공자에게 문의하여 묶을 수 있는지 알아보십시오.

#### 로컬 구성요소

개발자는 Oracle Content Management에서 제공한 것과 동일한 기능에 액세스할 수 있는 구성요소를 생성할 수 있습니다. 페이지에 직접 구성요소를 삽입하거나 `iframe` 요소를 사용하여 인라인 프레임으로 묶을 수 있습니다.

생성을 눌러 로컬 구성요소를 생성하면 전기능의 샘플 구성요소가 지정된 이름과 고유 식별자로 구성요소 목록에 추가됩니다. 개발자는 샘플을 수정하여 고유의 솔루션을 생성할 수 있습니다.

고유 구성요소 생성 방법에 대한 자세한 내용은 [구성요소 개발](#)을 참조하십시오.

## 레이아웃 이해

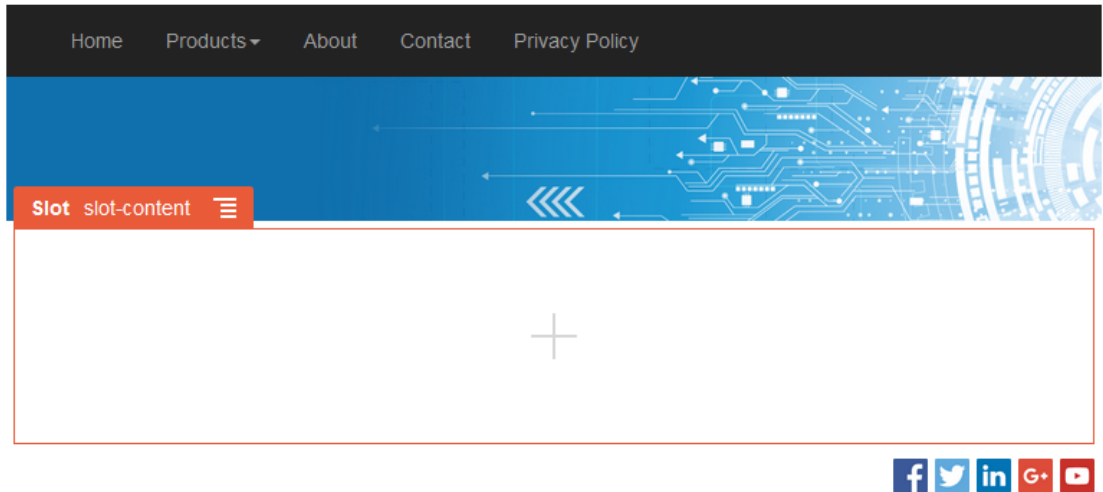
페이지 레이아웃은 페이지에 슬롯 및 콘텐츠를 배열합니다. 섹션 레이아웃은 슬롯 내에 콘텐츠를 배열합니다. 콘텐츠 레이아웃은 콘텐츠 항목에 필드를 배열합니다.

일반적으로 레이아웃은 콘텐츠 표현을 지정하는 것이지 콘텐츠 자체를 지정하지 않습니다. 프리젠테이션에서 콘텐츠를 분리하면 동일한 콘텐츠를 다양한 방법으로 손쉽게 표현할 수 있으며 콘텐츠를 건드리지 않고도 프리젠테이션을 손쉽게 변경할 수 있습니다.

#### 페이지 레이아웃

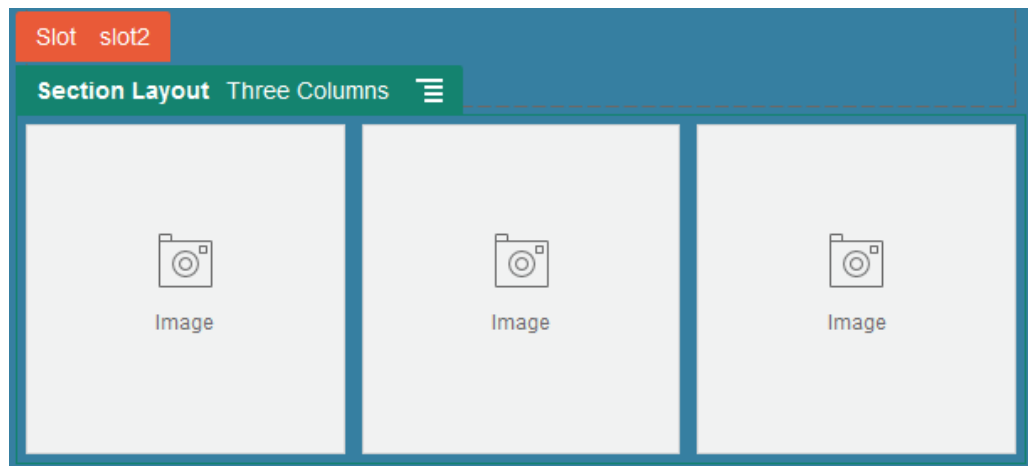
사이트에 페이지를 추가할 때 해당 페이지에 사용할 레이아웃을 선택합니다. 각 레이아웃에는 제공자가 콘텐츠를 끌어 놓을 수 있는 슬롯이라는 페이지의 영역이 있습니다. 페이지 레이아웃은 페이지의 슬롯 수와 위치를 정의합니다. 레이아웃에는 페이지에서 미리 정의되고 위치가 지정된 콘텐츠도 포함될 수 있습니다. 이는 회사 로고와 같이 정적이며 편집할 수 없거나 제공자가 변경할 수 있는 머리글 텍스트(위치나 모양은 변경할 수 없음)와 같이 최소한으로 편집할 수 있는 콘텐츠입니다.

페이지 레이아웃은 테마에 저장됩니다. 테마에는 페이지 레이아웃이 하나 이상 있을 수 있습니다. 개발자는 기존 테마를 복사 및 수정하여 새 테마를 생성할 수 있습니다. [레이아웃 개발](#)을 참조하십시오.



### 섹션 레이아웃

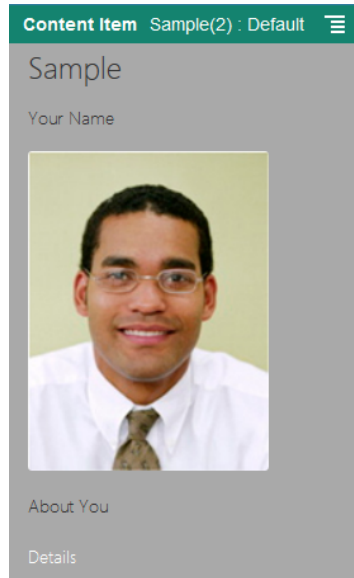
섹션 레이아웃은 여기에 추가되는 콘텐츠를 자동으로 구성하여 제공자가 페이지의 형식을 지정하는데 시간을 들일 필요 없이 콘텐츠를 손쉽게 추가할 수 있게 해 줍니다. 예를 들어 섹션 레이아웃은 콘텐츠를 여러 열 또는 세로 목록으로 자동으로 구성할 수 있습니다. 사이트 제공자는 슬롯에 섹션 레이아웃을 하나 이상 추가하여 콘텐츠를 구성할 수 있습니다.



개발자는 제공된 레이아웃 외에도 추가 섹션 레이아웃을 생성하여 특정 레이아웃 문제를 해결하거나 제공자를 위해 작성을 간소화할 수 있습니다. 테마 디자이너는 페이지 레이아웃의 슬롯에 섹션 레이아웃을 구성할 수도 있습니다. [레이아웃 개발](#)을 참조하십시오.

### 콘텐츠 레이아웃

엔터프라이즈 사용자인 경우 콘텐츠 유형 및 제공된 레이아웃에 따라 콘텐츠 항목을 생성하고 사용할 수 있습니다. 이 방식으로 구조화된 콘텐츠는 제공자가 사이트 작성기 외부의 콘텐츠 항목에 대한 콘텐츠를 조합할 수 있게 해줍니다. 콘텐츠 유형과 연관된 여러 콘텐츠 레이아웃을 사용하면 사이트 디자이너가 조합된 콘텐츠를 건드리지 않고도 콘텐츠 항목을 다양한 컨텍스트로 표시할 수 있습니다. 개발자는 제공된 기본 레이아웃에서 새 레이아웃을 생성할 수 있습니다. [레이아웃 개발](#)을 참조하십시오.



## 사용자정의 구성요소 및 레이아웃 사용


개발자는 사용자정의 구성요소 및 레이아웃을 생성하고 관리할 수 있습니다. 사이트 제공자는 타사 구성요소(앱) 및 구성요소 그룹을 등록할 수 있습니다.

사용자정의 구성요소에는 편집기에서 생성하는 구성요소 그룹과, 아래 설명된 옵션을 사용하여 생성하는 로컬 및 원격 구성요소가 포함됩니다. 타사 앱(원격 구성요소)을 사용하려면 그냥 등록하고 사이트에서 사용하면 됩니다. 개발자는 Oracle Content Management에서 제공한 것과 동일한 기능에 액세스할 수 있는 로컬 구성요소를 생성할 수도 있습니다.

사용자정의 레이아웃에는 페이지의 슬롯에 구성요소를 배열하기 위한 **섹션 레이아웃**과 콘텐츠 항목에 필드를 배열하기 위한 **콘텐츠 레이아웃**이 포함됩니다.


구성요소 페이지에서 메뉴 모음 또는 마우스 오른쪽 단추 누르기 메뉴의 옵션을 사용하여 다음 작업을 수행할 수 있습니다.


작업	설명
원격 구성요소 등록	원격 구성요소를 등록할 때 원격 구성요소 자체의 URL과 웹 저작자가 원격 구성요소에 지정할 수 있는 설정을 위한 두번째 URL을 지정합니다. 구성요소를 생성하기 전에 관리자가 <b>생성</b> 메뉴의 옵션을 사용하여 설정해야 합니다. 구성요소 페이지에 <b>생성</b> 메뉴가 보이지 않으면 관리자에게 문의하십시오. <b>생성</b> 을 누르고 원격 구성요소를 등록하는 관련 옵션을 선택합니다. <b>원격 구성요소 등록</b> 을 참조하십시오.

작업	설명
로컬 구성요소 또는 레이아웃 생성	<p>로컬 구성요소 또는 레이아웃을 생성할 때 자신만의 구성요소 또는 레이아웃을 생성하기 위한 기초로 사용하는 모든 기능이 들어 있는 샘플 구성요소 또는 레이아웃이 제공됩니다.</p> <p>구성요소 또는 레이아웃을 생성하려면 관리자가 <b>생성</b> 메뉴의 옵션을 사용으로 설정해야 합니다. 구성요소 페이지에 <b>생성</b> 메뉴가 보이지 않으면 관리자에게 문의하십시오.</p> <p><b>생성</b>을 누르고 새 로컬 구성요소 또는 레이아웃을 생성하기 위한 관련 옵션을 선택합니다. <b>로컬 구성요소, 레이아웃, 콘텐츠 필드 편집기 또는 콘텐츠 품 생성</b>을 참조하십시오.</p> <p>구성요소를 생성하는 다른 방법에 대한 자세한 내용은 <b>구성요소 개발</b>을 참조하십시오.</p>
구성요소 또는 레이아웃 복사	<p>기존 구성요소 또는 레이아웃을 복사하고 사본을 변경하여 새 구성요소 또는 레이아웃을 생성할 수 있습니다.</p>
<p> <b>주:</b></p> <p>구성요소 또는 레이아웃을 생성하거나 등록한 후에는 이름을 변경할 수 없습니다. 구성요소 또는 레이아웃을 복사하고 복사본에 대해 다른 이름을 지정할 수 있습니다. 원격 구성요소의 키 값을 비롯하여 기타 모든 등록 정보는 유지됩니다. 공유 정보는 등록 정보와 별개이며 복사되지 않습니다.</p>	
<p>복사할 구성요소 또는 레이아웃을 마우스 오른쪽 단추로 누르고 <b>복사</b>를 선택합니다. 이름을 입력하고 <b>복사</b>를 누릅니다. 문자, 숫자, 밑줄(_) 및 하이픈(-)을 사용할 수 있습니다. 공백을 입력하면 자동으로 하이픈으로 바뀝니다.</p>	
콘텐츠 필드 편집기 생성	<p>콘텐츠 필드 편집기를 <b>생성</b>하고 승격한 후 콘텐츠 유형을 생성할 때 데이터 필드의 모양을 제어하는 데 사용할 수 있습니다. <i>미디어</i> 및 <i>참조</i>를 제외한 모든 데이터 유형이 지원됩니다. 먼저 구성요소를 승격해야 콘텐츠 유형을 생성할 때 사용할 수 있습니다.</p>

작업	설명
구성요소 또는 레이아웃 공유	<p>다른 Oracle Content Management 사용자와 구성요소 또는 레이아웃을 공유할 수 있습니다. 내가 소유자이거나 나에게 관리자 롤이 지정된 경우 구성요소 또는 레이아웃을 공유할 수 있습니다. 구성요소 또는 레이아웃을 공유하는 경우 사용자가 구성요소 또는 레이아웃으로 수행할 수 있는 작업을 정의하는 롤을 사용자에게 지정해야 합니다.</p> <p>공유할 구성요소 또는 레이아웃을 마우스 오른쪽 단추로 누르고 <b>공유</b>를 선택하고 <b>멤버 추가</b>를 누릅니다.</p> <p>사용자 이름 또는 전자메일 주소를 하나 이상 입력하고 다음 롤 중 하나를 지정합니다.</p> <ul style="list-style-type: none"> <li>• <b>열람자:</b> 열람자는 편집기의 사용자정의 구성요소 또는 레이아웃 목록에 있는 구성요소 또는 레이아웃을 보고 사용할 수 있습니다. 또한 테마의 폴더 및 파일을 볼 수 있지만 어떤 내용도 변경할 수 없습니다.</li> <li>• <b>다운로더:</b> 다운로더는 파일을 다운로드하여 자신의 컴퓨터에 저장할 수도 있습니다.</li> <li>• <b>제공자:</b> 제공자는 구성요소 또는 레이아웃을 편집하고, 새 파일을 업로드하고, 파일을 삭제하고, 구성요소 또는 레이아웃 자체를 삭제(사이트에서 사용 중이 아닌 경우)할 수도 있습니다.</li> <li>• <b>관리자:</b> 관리자는 사용자를 추가하고 사용자 롤을 지정할 수 있습니다. 구성요소 또는 레이아웃 생성자(소유자)에게는 자동으로 관리자 롤이 지정됩니다.</li> </ul>
구성요소 또는 레이아웃 편집	<p>웹 개발자의 경우 개별 구성요소 또는 레이아웃 파일을 다운로드하여 수정하거나, 데스크톱 앱을 사용하여 로컬 시스템에서 수행한 변경사항을 동기화할 수 있습니다.</p> <p>또한 구성요소 또는 레이아웃을 개별적으로 익스포트하거나 템플릿 패키지(사용자정의 구성요소와 섹션 레이아웃 포함)의 일부로 익스포트하고, 원하는 개발 환경에서 오프라인으로 작업할 수 있습니다.</p> <p>구성요소 또는 레이아웃을 변경하면 구성요소 또는 레이아웃을 사용하는 사이트에 변경사항이 반영될 수 있도록 구성요소 또는 레이아웃을 게시해야 합니다.</p>
구성요소 또는 레이아웃 익스포트 또는 임포트	<p>구성요소 또는 레이아웃을 익스포트하고 오프라인에서 수정한 후 새 구성요소 또는 레이아웃으로 임포트하거나 기존 구성요소 또는 레이아웃을 바꿀 수 있습니다. 또한 구성요소 또는 레이아웃을 익스포트하여 다른 인스턴스로 이동하고 거기서 임포트할 수도 있습니다. 구성요소 또는 레이아웃을 개별적으로 익스포트하거나 템플릿 패키지(사용자정의 구성요소와 레이아웃 포함)의 일부로 익스포트할 수 있습니다.</p> <p>구성요소 또는 레이아웃을 익스포트할 때 폴더에 단일 .zip 파일로 구성요소 또는 레이아웃을 복사합니다. 폴더에서 구성요소 또는 레이아웃 패키지를 직접 다운로드하여 압축을 풀고 개별 파일로 작업할 수 있습니다. 구성요소 또는 레이아웃 파일 작업이 완료되면 구성요소 또는 레이아웃 패키지가 포함된 .zip 파일을 생성하여 임포트하고 원본 구성요소 또는 레이아웃을 겹쳐쓰거나 새로 생성합니다.</p> <p><a href="#">구성요소 또는 레이아웃 익스포트 또는 임포트를 참조하십시오.</a></p>



작업	설명
구성요소 또는 레이아웃 게시	<p>사이트에서 게시되지 않은 새로운 구성요소 또는 레이아웃을 사용하는 경우 사이트를 처음으로 온라인 상태로 설정할 때 구성요소 또는 레이아웃이 자동으로 사이트에 게시됩니다. 구성요소 또는 레이아웃을 변경하고 온라인 사이트를 업데이트하여 변경사항을 표시하려면 구성요소 또는 레이아웃을 명시적으로 게시해야 합니다. 내가 소유자이거나 나에게 관리자 롤이 지정된 경우 구성요소 또는 레이아웃을 게시할 수 있습니다.</p> <p>목록에서 구성요소 또는 레이아웃을 선택하고 메뉴 모음에서 <b>게시</b>(또는 이전에 게시된 구성요소 또는 레이아웃의 경우 <b>재게시</b>)를 누릅니다. 목록에서 구성요소 또는 레이아웃 옆에 게시됨 아이콘  이 표시됩니다.</p>
구성요소 또는 레이아웃 삭제 또는 복원	<p>적절한 권한이 있으면 구성요소 또는 레이아웃 폴더와 콘텐츠를 삭제할 수 있습니다. 구성요소 또는 레이아웃을 삭제하면 구성요소 또는 레이아웃 폴더와 모든 관련 폴더 및 파일이 휴지통으로 이동합니다.</p> <p>구성요소 또는 레이아웃을 직접 생성했거나(구성요소 또는 레이아웃 소유자) 누군가 나와 구성요소 또는 레이아웃을 공유하고 내게 제공자 또는 관리자 롤을 부여한 경우 구성요소 또는 레이아웃을 삭제하거나 복원할 수 있습니다.</p>

 **주:**

휴지통에 있는 사이트/업데이트를 포함해 모든 사이트/업데이트에서 사용 중인 구성요소 또는 레이아웃은 삭제할 수 없습니다.


구성요소 또는 레이아웃을 삭제하려면 삭제할 구성요소 또는 레이아웃을 마우스 오른쪽 단추로 누르고 **삭제**를 선택합니다. 구성요소 또는 레이아웃을 휴지통으로 이동할지 묻는 메시지가 표시됩니다. 다음 이벤트 전까지 삭제된 구성요소 또는 레이아웃이 휴지통에 보존됩니다.

- 구성요소 또는 레이아웃을 복원합니다.
- 구성요소 또는 레이아웃을 영구적으로 삭제합니다.
- 휴지통 할당량에 도달합니다.
- 서비스 관리자가 설정한 간격에 따라 자동으로 휴지통이 비워집니다. 기본값은 90일입니다.

구성요소 또는 레이아웃을 복원하려면 **휴지통**을 누르고 목록에서 구성요소 또는 레이아웃을 마우스 오른쪽 단추로 누르고 **복원**을 선택합니다.

## 원격 구성요소 등록

사이트에서 원격 구성요소를 사용하려면 먼저 Oracle Content Management에 등록해야 합니다. 타사 원격 구성요소 및 사용자가 개발한 원격 구성요소를 등록할 수 있습니다.

 **주:**

원격 구성요소를 등록하기 전에 관리자가 **생성** 메뉴의 옵션을 사용으로 설정해야 합니다. 구성요소 페이지에 **생성** 메뉴가 보이지 않으면 관리자에게 문의하십시오.

Oracle Content Management에서 사용하도록 원격 구성요소를 등록하려면 다음과 같이 하십시오.

1. **개발자**를 누르고 **모든 구성요소 보기**를 누릅니다.
2. **생성**을 누르고 **원격 구성요소 등록**을 선택합니다.
3. [원격 구성요소 등록] 창에서 다음 정보를 입력하거나 선택합니다.
  - **이름**: 사용자에게 표시되는 구성요소의 이름입니다.
  - **설명**: 사용자에게 표시되는 구성요소의 설명입니다.
  - **구성요소 URL**: 페이지에 구성요소 콘텐츠를 렌더링하기 위해 iframe에 사용되는 끝점입니다. HTTPS여야 합니다.
  - **설정 URL**: 페이지에 추가된 원격 구성요소 설정을 렌더링하기 위해 iframe에 사용되는 끝점입니다. HTTPS여야 합니다.
  - **설정 너비**: 구성요소 설정 패널의 기본 너비(픽셀)를 설정합니다.
  - **설정 높이**: 구성요소 설정 패널의 기본 높이(픽셀)를 설정합니다.
  - **키**: 원격 구성요소와 연관된 192비트 AES 키로, 구성요소를 프로비전할 때 서명된 해시 토큰을 생성하는 데 사용됩니다. 구성요소를 암호화하고 구성요소 설정을 안전하게 읽고 쓸 수 있도록 합니다.
4. **등록**을 누릅니다.

원격 구성요소가 생성되면 구성요소 목록에 이름이 나타납니다. 구성요소 목록에서 구성요소 이름을 눌러 구성요소 등록에 사용되는 파일을 탐색할 수 있습니다.

동일한 Oracle Content Management 인스턴스에서 생성된 사이트에서 사용되는 카탈로그에 구성요소 등록 정보가 저장되지만, 구성요소는 원격 서비스로 남아 있습니다.

구성요소 소유자의 경우 구성요소 아이콘이 편집기의 사용자정의 구성요소 패널에 구성요소에 지정된 이름으로 추가됩니다. 다른 사용자와 구성요소를 공유하면 편집기의 사용자정의 구성요소 패널에서 구성요소를 볼 수 있습니다.

## 로컬 구성요소, 레이아웃, 콘텐츠 필드 편집기 또는 콘텐츠 폼 생성

개발자인 사용자가 로컬 구성요소 또는 레이아웃을 생성할 때 자신만의 구성요소 또는 레이아웃을 수정할 수 있는 모든 기능이 들어 있는 샘플이 제공됩니다.

구성요소 또는 레이아웃을 생성하려면 관리자가 **생성** 메뉴의 옵션을 사용하여 설정해야 합니다. 구성요소 페이지에 **생성** 메뉴가 보이지 않으면 관리자에게 문의하십시오.

개발자는 Oracle Content Management에서 제공한 것과 동일한 기능에 액세스할 수 있는 구성요소를 생성할 수 있습니다.

샘플 로컬 구성요소 또는 레이아웃을 생성하려면 다음을 수행합니다.

1. **개발자**를 누르고 **모든 구성요소 보기**를 누릅니다.
2. **생성**을 누르고 생성할 구성요소 또는 레이아웃의 유형을 선택합니다.
3. 구성요소 또는 레이아웃의 이름을 입력합니다. 다른 구성요소 또는 레이아웃에 사용되는 이름은 사용할 수 없습니다.

문자, 숫자, 밑줄(\_) 및 하이픈(-)을 사용할 수 있습니다. 공백을 입력하면 자동으로 밑줄로 바뀝니다.

템플릿, 테마, 구성요소, 사이트 또는 사이트 페이지에 다음 이름을 사용하지 마십시오: authsite, content, pages, scstemplate\_\*, \_comps, \_components, \_compsdelivery, \_idcservice, \_sitescloud, \_sitesclouddelivery, \_themes,

\_themesdelivery. 사이트 페이지에는 다음 이름을 사용할 수 있지만 템플릿, 테마, 구성요소 또는 사이트에는 사용하지 마십시오: documents, sites.

4. 필요에 따라 구성요소 또는 레이아웃에 대한 설명을 입력합니다.
5. 구성요소 유형의 경우 **기본값**을 선택하여 페이지에 직접 구성요소를 삽입하거나, **Sandbox 적용**을 선택하여 인라인 프레임(iframe 요소 사용)으로 구성요소를 묶거나, **템플릿**을 선택하여 템플릿을 통해서만 렌더링합니다.

 **주:**

템플릿을 선택하면 로컬 구성요소가 Knockout을 기반으로 하지 않으므로 페이지에 배치될 때 중첩 구성요소를 사용할 수 없습니다. 예를 들어, 제목 또는 본문 텍스트를 편집하려면 페이지에 표시되는 구성요소 텍스트를 선택하는 대신, 구성요소의 설정 패널에 있는 사용자정의 설정을 통해 작업을 수행해야 합니다.

6. 생성을 누릅니다.
 

진행을 표시줄에 생성 상태가 표시됩니다. 구성요소, 레이아웃 또는 품이 생성되면 구성요소 목록에 이름이 나타납니다. 구성요소 목록에서 구성요소 또는 레이아웃 이름을 눌러 구성요소 또는 레이아웃을 구성하는 폴더 및 파일을 탐색할 수 있습니다.
7. 구성요소 또는 레이아웃에 지정된 기본 아이콘 이외의 아이콘을 선택하려면 다음을 수행합니다.
  - a. 목록에서 구성요소 또는 레이아웃을 선택합니다.
  - b. 속성을 누릅니다.
  - c. 구성요소 로고 탭을 누릅니다.
  - d. 로고 갤러리에서 로고를 누른 다음 **완료**를 누릅니다.

고유 구성요소 생성 방법에 대한 자세한 내용은 [구성요소 개발](#)을 참조하십시오.

구성요소, 레이아웃, 콘텐츠 필드 편집기 또는 콘텐츠 품을 사용자정의한 후 이를 다른 사람과 공유하거나 승격하여 다음 방법으로 사용할 수 있습니다.

- 사용자정의 구성요소: 사이트 작성기를 사용하는 경우 구성요소에 지정된 이름과 함께 구성요소 아이콘이 편집기의 [사용자정의 구성요소] 패널에 추가됩니다.
- 섹션 레이아웃: 사이트 작성기를 사용하는 경우 레이아웃에 지정된 이름과 함께 섹션 레이아웃 아이콘이 편집기의 [섹션 레이아웃] 패널에 추가됩니다.
- 콘텐츠 레이아웃: 콘텐츠 관리자는 레이아웃을 하나 이상의 콘텐츠 유형에 지정할 수 있습니다. 이때 기본 뷰로 표시하거나 사이트 디자이너가 사이트 작성기에서 선택할 수 있는 레이아웃 목록에 추가하여 해당 유형의 콘텐츠 항목이 페이지에 표시되는 방법을 지정하도록 할 수 있습니다.
- 콘텐츠 필드 편집기: 콘텐츠 유형을 생성할 때 콘텐츠 필드 편집기를 사용할 수 있게 하려면 선택 후 **승격**을 누르고 확인 후 **확인**을 누릅니다. 승격 후에는 콘텐츠 필드 편집기를 사용하여 콘텐츠 유형을 생성할 때 *미디어* 및 *참조*를 제외한 모든 데이터 필드의 모양을 제어할 수 있습니다.
- 콘텐츠 품: 콘텐츠 유형을 생성할 때 콘텐츠 품을 사용할 수 있게 하려면 선택 후 **승격**을 누르고 확인 후 **확인**을 누릅니다. 승격 후 모든 콘텐츠 유형 및 자산 유형에 콘텐츠 품을 사용할 수 있습니다.

## 구성요소 또는 레이아웃 익스포트 또는 임포트

구성요소를 익스포트하고 오프라인에서 수정한 후 새 구성요소로 임포트하거나 기존 구성요소를 바꿀 수 있습니다. 또한 구성요소를 익스포트하여 다른 Oracle Content Management 인스턴스로 이동하고 거기서 임포트할 수도 있습니다.

### 익스포트

구성요소를 익스포트할 때 반드시 Oracle Content Management 폴더에 단일 .zip 파일로 구성요소를 복사해야 합니다. 폴더에서 직접 구성요소를 다운로드하여 압축을 풀고 개별 파일로 작업할 수 있습니다. 구성요소 파일 작업이 완료되면 구성요소 폴더 및 파일이 포함된 .zip 파일을 생성하여 임포트하고 원본 구성요소를 겹쳐쓰거나 새로 생성합니다.

#### 주:

구성요소를 익스포트할 때 구성요소 공유 정보는 포함되지 않습니다.

구성요소를 익스포트하려면 다음과 같이 하십시오.

1. **개발자**를 누르고 **모든 구성요소 보기**를 누릅니다.
2. 구성요소 또는 레이아웃을 선택하고 메뉴 모음 또는 마우스 오른쪽 단추 누르기 메뉴에서 **익스포트**를 선택합니다.
3. 폴더로 이동하거나, **새로 만들기**를 누르고 이름과 선택적 설명을 제공하고 **생성**을 눌러서 새 폴더를 생성합니다. 폴더를 열려면 폴더 아이콘 또는 폴더 이름을 누릅니다.
4. 관련 폴더의 체크박스를 눌러 폴더를 선택하고 **확인**을 누릅니다.

구성요소 또는 레이아웃 패키지 파일이 구성요소 또는 레이아웃 이름에 .zip 확장자가 붙은 형태로 선택한 폴더에 생성됩니다.

### 임포트

구성요소, 레이아웃 또는 콘텐츠 필드 편집기를 임포트하거나 생성하기 전에 관리자가 **생성** 메뉴의 옵션을 사용으로 설정해야 합니다. 구성요소 페이지에 **생성** 메뉴가 보이지 않으면 관리자에게 문의하십시오.

구성요소, 레이아웃 또는 콘텐츠 필드 편집기를 임포트하려면 다음과 같이 하십시오.

1. **개발자**를 누르고 **모든 구성요소 보기**를 누릅니다.
2. **생성**을 누르고 **구성요소 임포트**를 선택합니다. 구성요소, 섹션 레이아웃 또는 콘텐츠 레이아웃을 임포트하려면 이 옵션을 선택합니다.
3. 구성요소 또는 레이아웃 패키지를 업로드한 경우 구성요소 또는 레이아웃이 포함된 폴더로 이동하여 폴더를 엽니다. 패키지를 업로드하지 않은 경우 구성요소 또는 레이아웃을 업로드할 폴더로 이동하거나 새 폴더를 생성합니다. **업로드**를 누른 다음 구성요소 또는 레이아웃 패키지를 찾아 **열기**를 누릅니다.
4. 구성요소 또는 레이아웃 파일 이름 옆에 있는 체크박스를 누르고 **확인**을 누릅니다. 새 구성요소 또는 레이아웃이 생성되어 구성요소 목록에 추가됩니다. 구성요소 또는 레이아웃 이름 또는 ID가 존재하는 경우 충돌을 해결할지 묻는 메시지가 표시됩니다. 새 구성요소 또는 레이아웃을 생성해야 할 수도 있고, 기존 구성요소 또는 레이아웃을 임포트된 버전으로 겹쳐쓸 수도 있습니다.

# 8





## 사이트 페이지 작업

페이지를 생성하고, 레이아웃을 선택하고, 사이트에서 페이지 이동 위치를 결정해 보겠습니다.

- [페이지로 이동](#)
- [페이지 보기](#)
- [페이지 추가](#)
- [페이지 이동](#)
- [페이지 삭제](#)
- [페이지 설정 변경](#)
- [페이지 레이아웃 변경](#)
- [배경 또는 테마 변경](#)

### 페이지로 이동


페이지 계층을 보고 이동하려면 다음과 같이 하십시오.


1. 편집을 위해 사이트를 엽니다. 가 편집으로 설정되어 있는지 확인합니다.
2. 페이지 및 페이지 콘텐츠를 관리하고 편집하는 옵션을 표시하려면 을 누릅니다.
3. 을 눌러 페이지의 첫번째 레벨을 나열합니다.
4. 페이지를 눌러 페이지를 표시합니다.  
화살표(>)가 있는 페이지는 중첩된 페이지 층이 있습니다. 페이지를 눌러 중첩 페이지를 표시합니다.  
페이지 목록 상단의 검색 상자를 사용하여 원하는 페이지를 찾을 수도 있습니다.
5. 특정 페이지의 설정을 보거나 변경하려면 해당 페이지의 을 누릅니다. [페이지 설정 변경](#)을 참조하십시오.

### 페이지 보기

사이트에서 페이지를 보려면 편집기에서 사이트를 열고 다른 미리보기, 크기 및 방향 옵션을 사용합니다.

편집기를 사용하는 동안 페이지 보기 방법을 변경하려면 다음과 같이 하십시오.

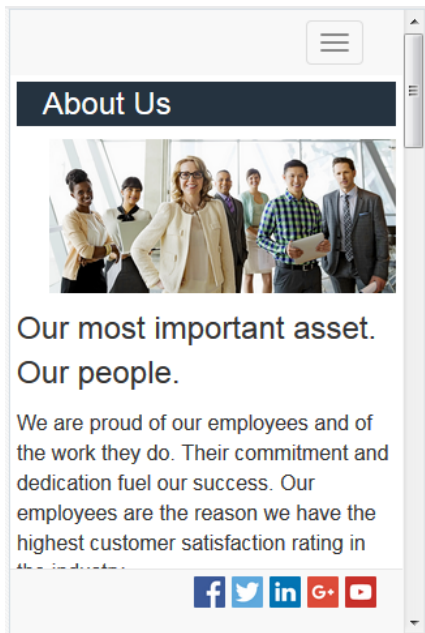
1. 편집을 위해 사이트를 엽니다.
2. 페이지로 이동합니다.
3. 편집기에서 페이지를 미리 보려면 가 미리보기로 설정되어 있는지 확인합니다. 편집하는 동안 사용되는 시각적 도구 없이 사이트 방문자에 보이는 대로 페이지를 표시합니다.




4. 별도의 브라우저 창에서 기본 사이트에 적용되는 현재 업데이트를 미리 보려면  을 누릅니다.

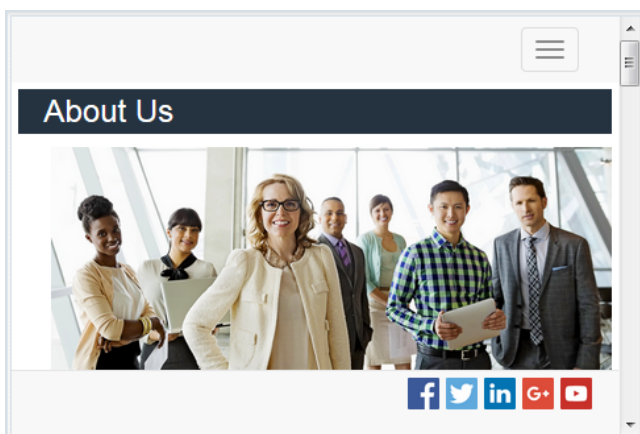
 주:

미리보기 모드에서는 사이트 페이지에 대한 링크가 작동하지 않습니다.

5. 특정 화면 크기로 장치에 표시되는 대로 페이지를 미리 보려면 상단 메뉴 모음에서 **화면에 맞춤**을 누르고 목록에서 화면 치수 집합을 선택합니다. 반응형 디자인 테마는 선택한 화면 크기에 최적 사용되도록 페이지 콘텐츠를 자동으로 배열합니다.



6. 사용자정의 크기를 생성하려면 **창에 맞춤**을 누르고 **장치 사전 설정 생성**을 선택합니다. 필요한 필드를 완성합니다. 사용자정의 크기를 삭제하려면 크기 옆에 있는  을 누릅니다. 또한 눈금자  을 활성화하고 눈금자에서 간격을 선택하여 다른 크기를 빠르게 확인할 수 있습니다.
7. 페이지 방향을 세로와 가로 사이에 교대하려면  을 누릅니다.




## 페이지 추가

사이트의 페이지는 폴더형 구조 또는 *사이트 트리*로 구성됩니다. 페이지를 추가하고, 페이지 설정을 정의하고, 사이트에서 페이지 이동 위치를 결정할 수 있습니다.



페이지를 추가하거나 외부 페이지의 링크를 추가하도록 선택할 수 있습니다.

- **웹 페이지:** 페이지와 콘텐츠가 사이트에 상주합니다. 페이지 이름을 지정하고 콘텐츠를 정의하며, 사이트에서 페이지 이동 위치를 지정하고, 현재 사이트 컨텍스트에서 페이지 작동 방법을 지정할 수 있습니다.
- **외부 링크:** URL로 지정된 다른 위치에서 페이지가 참조됩니다. 페이지 이름을 지정하고, 사이트에서 이동 위치를 지정하고, 현재 사이트 컨텍스트에서 페이지 작동 방법을 지정할 수 있습니다. 실시간 사이트에서 페이지를 사용 중이기 때문에 링크된 페이지의 내용을 변경할 수 없습니다.

사이트 트리에서 외부 URL로 링크되는 페이지는 페이지 이름 앞에 이 있습니다.

### 웹 페이지 추가

사이트에 웹 페이지를 추가하려면 다음과 같이 하십시오.

1. 편집을 위해 사이트를 엽니다.
2. 페이지를 추가할 레벨 또는 분기를 선택하고 **페이지 추가**를 누릅니다. 또는 하위 페이지를 추가하려면 을 누르고 을 누릅니다. “새 페이지”가 사이트 트리의 맨 아래에 추가되고, 페이지 이름과 다른 설정을 지정하라는 메시지가 표시됩니다.
3. 페이지 유형으로 **웹 페이지**를 선택합니다. 다른 사이트에서 페이지를 재사용하는 대신, 페이지의 콘텐츠를 관리하려면 이 옵션을 사용합니다.
4. 페이지 이름을 지정합니다. 문자, 숫자, 밑줄(\_) 및 하이픈(-)을 사용할 수 있습니다. 템플릿, 테마, 구성요소, 사이트 또는 사이트 페이지에 다음 이름을 사용하지 마십시오: authsite, content, pages, scs\_template\_\*, \_comps, \_components, \_compsdelivery, \_idc\_service, \_sitescloud, \_sitesclouddelivery, \_themes, \_themesdelivery. 사이트 페이지에는 다음 이름을 사용할 수 있지만 템플릿, 테마, 구성요소 또는 사이트에는 사용하지 마십시오: documents, sites.
5. 페이지 URL을 지정합니다. 기본적으로 페이지 이름이 페이지 URL에 사용됩니다. 공백은 자동으로 하이픈으로 바뀝니다. 기본값 이외의 URL을 사용하려면 **무효화**를 누르고 URL에 사용되는 파일 이름을 추가합니다. 문자, 숫자, 밑줄(\_) 및 하이픈(-)을 사용할 수 있습니다. 파일 확장자를 포함해야 합니다. 기본 파일 확장자는 .html입니다.

6. 페이지 레이아웃을 선택합니다. 일반적 페이지 구조를 정의하지만 콘텐츠는 없습니다. 레이아웃에는 콘텐츠를 넣을 수 있는 하나 이상의 명명된 슬롯이 있습니다. 페이지 레이아웃의 개수와 유형은 사이트와 연관된 테마에 따라 달라집니다.
7. 모바일 페이지 레이아웃을 선택합니다. 모바일 장치에서 봤을 때 일반적 페이지 구조를 정의합니다.
8. 페이지 제목을 지정합니다. 페이지 자체에 노출되지 않지만, 페이지를 표시할 때 브라우저 제목 표시줄이나 브라우저 탭에 표시됩니다. 페이지에 제목을 추가하려면 편집기를 사용하여 제목 구성요소를 추가합니다.
9. 선택적 설명을 추가합니다. 이 설명은 페이지에 노출되지 않지만, 본인이나 다른 제공자를 위해 페이지에 대한 정보를 추가할 수 있습니다.
10. 검색 엔진이 페이지 내용을 식별할 수 있도록 콤마로 구분된 선택적 키워드를 지정합니다. 키워드는 페이지 텍스트에 없거나 이미지에 나타나는 용어나 개념을 식별하는 데 유용합니다. 키워드는 페이지에 표시되지 않지만, 검색 엔진이 이를 사용하여 사이트를 찾아 식별합니다. 좋은 설명, 키워드, 동의어로 웹 사이트에 트래픽이 증가할 수 있습니다.
11. 분석/추적을 위한 선택적 페이지 머리글 스크립팅 또는 태그를 추가합니다. 머리글 또는 바닥글에 사용할 코드가 제대로 작동하는지, 사이트에 보안 위험을 일으키지 않는지 이를 검증합니다.
12. 분석/추적을 위한 선택적 페이지 바닥글 스크립팅 또는 태그를 추가합니다. 기본적으로 바닥글에는 개인정보 보호정책의 링크와 함께 사이트의 쿠키 사용과 관련해 표시되는 팝업 텍스트가 포함됩니다. 개발자의 경우 표시되는 텍스트를 편집할 수 있습니다. 머리글 또는 바닥글에 사용할 코드가 제대로 작동하는지, 사이트에 보안 위험을 일으키지 않는지 이를 검증합니다.
13. 선택사항: 사용 가능한 페이지 옵션을 사용할 수 있습니다.
  - **오류 페이지:** 오류로 인해 요청된 페이지를 표시할 수 없는 경우 기본 오류 페이지 대신 이 페이지를 표시합니다. 기본적으로 이 옵션을 선택하면 **탐색에서 페이지 숨기기**와 모든 검색 엔진 옵션도 선택됩니다. 오류 페이지를 탐색 또는 검색 엔진 결과에 포함시키려는 경우 이 옵션을 선택 해제할 수 있습니다.
  - **탐색에서 페이지 숨기기:** 자동으로 생성된 사이트 탐색에 페이지를 포함하지 않습니다.
  - **세부정보 페이지:** 이 페이지를 사용하여 다른 페이지에서 선택한 콘텐츠 항목에 대한 자세한 정보를 표시할 수 있습니다. 구조화된 콘텐츠는 엔터프라이즈 사용자만 사용할 수 있습니다. 기본적으로 이 옵션을 선택하면 **탐색에서 페이지 숨기기**도 선택됩니다. 세부정보 페이지를 탐색에 포함시키려는 경우 이 옵션을 선택 해제할 수 있습니다.  
구조화된 콘텐츠 항목이 세부정보 페이지를 사용하도록 구성된 경우 사용자가 구조화된 콘텐츠 항목에 대한 자세한 내용을 보기 위해 링크를 누르면 콘텐츠 항목에 대한 세부정보와 함께 세부정보 페이지가 표시됩니다.
  - **검색 페이지:** 이 페이지를 사용하여 검색 결과를 표시할 수 있습니다. 콘텐츠 검색 구성요소의 링크 설정에서 이 페이지를 선택할 수 있습니다. **콘텐츠 검색**을 참조하십시오. 기본적으로 이 옵션을 선택하면 **탐색에서 페이지 숨기기**도 선택됩니다. 검색 페이지를 탐색에 포함시키려는 경우 이 옵션을 선택 해제할 수 있습니다.
14. 선택사항: 검색 엔진 최적화(SEO) 옵션을 선택합니다.
  - **검색 엔진에서 페이지 숨기기:** 검색 결과에 표시되지 않게 페이지 콘텐츠를 인덱스화하지 않도록 검색 엔진에 통지합니다.
  - **검색 엔진에서 페이지 링크 숨기기:** 페이지에서 링크를 따라가서 링크 대상을 인덱스화하지 않도록 검색 엔진에 통지합니다.



- **검색 엔진 페이지 캐싱 사용 안함:** 이 페이지를 캐싱하지 않도록 검색 엔진에 통지합니다.
  - **검색 엔진에서 페이지 설명 숨기기:** 검색 결과에서 페이지 다음에 설명(위에서 지정됨)을 포함하지 않도록 검색 엔진에 통지합니다.
15. **선택사항:** 브라우징 세션에 사용할 Cobrowse 단추가 이 페이지에 포함될 경우 선택합니다. Cobrowse 세션은 Oracle Cobrowse Cloud Service를 사용하여 사이트 방문자와의 화면 공유 환경을 관리합니다. [Cobrowse 통합 사용 및 페이지에서 Cobrowse 사용](#)을 참조하십시오.
  16. **선택사항:** 사이트의 분석 추적 코드 조각을 무효화하고 이 페이지에 특정한 코드 조각을 추가합니다. Web Analytics 추적을 위해 JavaScript 추적 코드의 코드 조각을 추가하면 Google, Adobe 또는 Oracle Infinity와 같은 외부 분석 제공자와의 통합을 보다 간편하게 수행할 수 있습니다. [Analytics 추적 추가](#)를 참조하십시오.
  17. 현재 업데이트에 보류 중인 변경사항을 모두 저장하려면 **저장**을 누릅니다.

### 링크된 페이지 추가

사이트에 외부 링크 페이지를 추가하려면 다음과 같이 하십시오.

1. 편집을 위해 사이트를 엽니다.
2. 페이지를 추가할 사이트 트리의 레벨에서 페이지로 이동합니다. 현재 레벨에서 페이지가 추가되지만, 사이트 트리의 다른 위치로 쉽게 이동할 수 있습니다.
3. **페이지 추가**를 누릅니다. “새 페이지”가 사이트 트리의 맨 아래에 추가되고, 페이지 이름과 다른 설정을 지정하라는 메시지가 표시됩니다.
4. 페이지 유형으로 **외부 링크**를 선택합니다.
5. 페이지 이름을 지정합니다. 문자, 숫자, 밑줄(\_) 및 하이픈(-)을 사용할 수 있습니다. 이름은 사이트 트리에서 페이지를 식별하는 데 사용됩니다. 공백은 자동으로 하이픈으로 바뀝니다.
6. 페이지의 전체 URL을 지정합니다. 예: `https://www.example.com/sharedpage.htm`
7. 새 브라우저 창이나 탭에서 페이지를 열려면 **새 창에서 링크 열기**를 선택합니다. 이 옵션을 선택하지 않으면 현재 창에서 페이지가 열리고 사이트 페이지 및 탐색을 바꿉니다.
8. 선택적으로 사용 가능한 옵션을 선택합니다.
  - **오류 페이지:** 오류로 인해 요청된 페이지를 표시할 수 없는 경우 기본 오류 페이지 대신 이 페이지를 표시합니다. 기본적으로 이 옵션을 선택하면 **탐색에서 페이지 숨기기도** 선택됩니다. 오류 페이지를 탐색에 포함시키려는 경우 이 옵션을 선택 해제할 수 있습니다.
  - **탐색에서 페이지 숨기기:** 자동으로 생성된 사이트 탐색에 페이지를 포함하지 않습니다.
  - **세부정보 페이지:** 이 페이지를 사용하여 다른 페이지에서 선택한 콘텐츠 항목에 대한 자세한 정보를 표시할 수 있습니다. 구조화된 콘텐츠는 엔터프라이즈 사용자만 사용할 수 있습니다. 기본적으로 이 옵션을 선택하면 **탐색에서 페이지 숨기기도** 선택됩니다. 세부정보 페이지를 탐색에 포함시키려는 경우 이 옵션을 선택 해제할 수 있습니다. 구조화된 콘텐츠 항목이 세부정보 페이지를 사용하도록 구성된 경우 사용자가 구조화된 콘텐츠 항목에 대한 자세한 내용을 보기 위해 링크를 누르면 콘텐츠 항목에 대한 세부정보와 함께 세부정보 페이지가 표시됩니다.
  - **검색 페이지:** 이 페이지를 사용하여 검색 결과를 표시할 수 있습니다. 콘텐츠 검색 구성요소의 링크 설정에서 이 페이지를 선택할 수 있습니다. [콘텐츠 검색](#)을 참조하십시오. 기본적으로 이 옵션을 선택하면 **탐색에서 페이지 숨기기도** 선택됩니다. 검색 페이지를 탐색에 포함시키려는 경우 이 옵션을 선택 해제할 수 있습니다.
9. 현재 업데이트에 보류 중인 변경사항을 모두 저장하려면 **저장**을 누릅니다.

## 페이지 이동




사이트 트리의 현재 레벨에서 다른 위치로 페이지를 끌어 놓거나, 다른 레벨로 페이지를 잘라서 붙여 넣을 수 있습니다.


항상 계층의 맨 위에는 사이트 “홈” 페이지가 있고 그 아래에 다른 모든 페이지가 나열됩니다. 주어진 레벨에서 페이지를 끌어 놓아서 순서를 변경할 수 있습니다. 계층의 다른 위치로 페이지를 잘라서 붙여 넣을 수도 있습니다.


페이지는 중첩 페이지를 가질 수 있습니다. 중첩 페이지를 가진 페이지를 이동하는 경우, 중첩 페이지도 이동하고 이동된 페이지 아래에 중첩된 상태를 유지합니다. 사용 중인 테마가 탐색 자동화를 지원하는 경우 페이지 순서를 변경하면 탐색 메뉴에 나열되는 순서도 변경됩니다.

사이트 트리의 다른 위치로 페이지를 이동하려면 다음과 같이 하십시오.

1. 편집을 위해 사이트를 엽니다.
2. 이동할 페이지로 이동합니다.
3. 사이트 트리의 현재 레벨에서 다른 위치로 페이지를 이동하려면 페이지 이름을 눌러 끌어서 페이지 목록의 다른 위치에 놓습니다.
4. 사이트 트리의 다른 레벨로 페이지를 이동하려면 이동할 페이지를 누르고 해당

페이지의  을 누릅니다.  을 눌러 현재 위치에서 페이지를 잘라내거나  을

눌러 현재 위치에 페이지 사본을 남겨둡니다. 페이지를 이동할 레벨로 가서  을 누릅니다. 해당 레벨에서 페이지를 선택하면 붙여넣은 페이지가 그 아래에 중첩됩니다.

 을 눌러 현재 위치로 페이지를 붙여 넣습니다. 페이지를 재정렬하려면 사이트 트리의 현재 레벨에서 다른 위치로 페이지를 끌어 놓습니다.

## 페이지 삭제

업데이트의 일부로 페이지를 삭제할 수 있습니다.


**저장**을 누르면 삭제된 페이지가 업데이트의 일부로 기록됩니다. 업데이트를 게시해야 페이지가 기본 사이트에서 삭제됩니다. 제공자 또는 관리자 롤을 가진 누구나 페이지를 삭제할 수 있습니다.

### 주:

페이지를 삭제하는 업데이트를 게시할 경우 페이지가 영구 삭제되므로 복구할 수 없습니다. 다국어 사이트에서 페이지를 삭제하면 사이트가 게시된 경우 페이지의 번역된 버전도 삭제됩니다. 페이지 추가/삭제 관리가 우려되는 경우 사이트에서 페이지를 추가/삭제할 때 전용 업데이트를 사용하여 이 동작을 분리할 수 있습니다.


사이트에서 페이지를 삭제하려면 다음과 같이 하십시오.


1. 편집을 위해 사이트를 엽니다.

2. 삭제할 페이지로 이동하고  을 누릅니다.
3. **확인**을 눌러 삭제를 확인합니다. **저장**을 눌러 현재 업데이트에 보류 중인 변경사항을 모두 저장합니다.

## 페이지 설정 변경

이름, 제목, URL, 머리글, 바닥글 및 기타 옵션 등의 페이지 설정을 변경할 수 있습니다.

1. 편집을 위해 사이트를 엽니다.
2. 설정을 변경할 페이지를 누르고  을 누릅니다.
3. 필요에 따라 설정을 변경합니다.
  - **페이지 유형:**
    - **웹 페이지:** 페이지와 콘텐츠가 사이트에 상주합니다. 페이지 이름을 지정하고 콘텐츠를 정의하며, 사이트에서 페이지 이동 위치를 지정하고, 현재 사이트 컨텍스트에서 페이지 작동 방법을 지정할 수 있습니다.
    - **외부 링크:** URL로 지정된 다른 위치에서 페이지가 참조됩니다. 페이지 이름을 지정하고, 사이트에서 이동 위치를 지정하고, 현재 사이트 컨텍스트에서 페이지 작동 방법을 지정할 수 있습니다. 실시간 사이트에서 페이지를 사용 중이기 때문에 링크된 페이지의 내용을 변경할 수 없습니다.

사이트 트리에서 외부 URL로 링크되는 페이지는 페이지 이름 앞에  이 있습니다.

웹 페이지를 선택하면 다음 설정이 표시됩니다.

- **페이지 이름:** 문자, 숫자, 밑줄(\_) 및 하이픈(-)을 사용할 수 있습니다. 템플릿, 테마, 구성요소, 사이트 또는 사이트 페이지에 다음 이름을 사용하지 마십시오: authsite, content, pages, scstemplate\_\*, \_comps, \_components, \_compsdelivery, \_idcservice, \_sitescloud, \_sitesclouddelivery, \_themes, \_themesdelivery. 사이트 페이지에는 다음 이름을 사용할 수 있지만 템플릿, 테마, 구성요소 또는 사이트에는 사용하지 마십시오: documents, sites.
- **페이지 URL:** 기본적으로 페이지 이름이 페이지 URL에 사용됩니다. 공백은 자동으로 하이픈으로 바뀝니다. 기본값 이외의 URL을 사용하려면 **무효화**를 누르고 URL에 사용되는 파일 이름을 추가합니다. 문자, 숫자, 밑줄(\_) 및 하이픈(-)을 사용할 수 있습니다. 파일 확장자를 포함해야 합니다. 기본 파일 확장자는 .html입니다.
- **페이지 레이아웃:** 페이지 레이아웃은 일반적 페이지 구조를 정의하지만 콘텐츠는 없습니다. 레이아웃에는 콘텐츠를 넣을 수 있는 하나 이상의 명명된 슬롯이 있습니다. 페이지 레이아웃의 개수와 유형은 사이트와 연관된 테마에 따라 달라집니다.
- **모바일 페이지 레이아웃:** 모바일 페이지 레이아웃은 모바일 장치에서 봤을 때 일반적 페이지 구조를 정의합니다.
- **페이지 제목:** 페이지 제목은 페이지 자체에 노출되지 않지만, 페이지를 표시할 때 브라우저 제목 표시줄이나 브라우저 탭에 표시됩니다. 페이지에 제목을 추가하려면 편집기를 사용하여 제목 구성요소를 추가합니다.
- **페이지 설명:** 설명은 페이지에 노출되지 않지만, 본인이나 다른 제공자를 위해 페이지에 대한 정보를 추가할 수 있습니다.
- **키워드:** 검색 엔진이 페이지 내용을 식별할 수 있도록 콤마로 구분된 선택적 키워드를 지정합니다. 키워드는 페이지 텍스트에 없거나 이미지에 나타나는 용어나 개념을 식별하는 데 유용합니다. 키워드는 페이지에 표시되지 않지만, 검색 엔진이 이를 사용하여 사이트를 찾아 식별합니다. 좋은 설명, 키워드, 동의어로 웹 사이트에 트래픽이 증가할 수 있습니다.

- **페이지 머리글:** 분석/추적을 위한 선택적 페이지 머리글 스크립팅 또는 태그를 추가합니다. 머리글 또는 바닥글에 사용할 코드가 제대로 작동하는지, 사이트에 보안 위험을 일으키지 않는지 이를 검증합니다.
- **페이지 바닥글:** 분석/추적을 위한 선택적 페이지 바닥글 스크립팅 또는 태그를 추가합니다. 기본적으로 바닥글에는 개인정보 보호정책의 링크와 함께 사이트의 쿠키 사용과 관련해 표시되는 팝업 텍스트가 포함됩니다. 개발자의 경우 표시되는 텍스트를 편집할 수 있습니다. 머리글 또는 바닥글에 사용할 코드가 제대로 작동하는지, 사이트에 보안 위험을 일으키지 않는지 이를 검증합니다.
- **페이지 옵션:**
  - **오류 페이지:** 오류로 인해 요청된 페이지를 표시할 수 없는 경우 기본 오류 페이지 대신 이 페이지를 표시합니다. 기본적으로 이 옵션을 선택하면 **탐색에서 페이지 숨기기**와 모든 검색 엔진 옵션도 선택됩니다. 오류 페이지를 탐색 또는 검색 엔진 결과에 포함시키려는 경우 이 옵션을 선택 해제할 수 있습니다.
  - **탐색에서 페이지 숨기기:** 자동으로 생성된 사이트 탐색에 페이지를 포함하지 않습니다.
  - **세부정보 페이지:** 이 페이지를 사용하여 다른 페이지에서 선택한 콘텐츠 항목에 대한 자세한 정보를 표시할 수 있습니다. 구조화된 콘텐츠는 엔터프라이즈 사용자만 사용할 수 있습니다. 기본적으로 이 옵션을 선택하면 **탐색에서 페이지 숨기기도** 선택됩니다. 세부정보 페이지를 탐색에 포함시키려는 경우 이 옵션을 선택 해제할 수 있습니다.  
구조화된 콘텐츠 항목이 세부정보 페이지를 사용하도록 구성된 경우 사용자가 구조화된 콘텐츠 항목에 대한 자세한 내용을 보기 위해 링크를 누르면 콘텐츠 항목에 대한 세부정보와 함께 세부정보 페이지가 표시됩니다.
  - **검색 페이지:** 이 페이지를 사용하여 검색 결과를 표시할 수 있습니다. 콘텐츠 검색 구성요소의 링크 설정에서 이 페이지를 선택할 수 있습니다. **콘텐츠 검색**을 참조하십시오. 기본적으로 이 옵션을 선택하면 **탐색에서 페이지 숨기기도** 선택됩니다. 검색 페이지를 탐색에 포함시키려는 경우 이 옵션을 선택 해제할 수 있습니다.
- **SEO 옵션:**
  - **검색 엔진에서 페이지 숨기기:** 검색 결과에 표시되지 않게 페이지 콘텐츠를 인덱스화하지 않도록 검색 엔진에 통지합니다.
  - **검색 엔진에서 페이지 링크 숨기기:** 페이지에서 링크를 따라가서 링크 대상을 인덱스화하지 않도록 검색 엔진에 통지합니다.
  - **검색 엔진 페이지 캐싱 사용 안함:** 이 페이지를 캐싱하지 않도록 검색 엔진에 통지합니다.
  - **검색 엔진에서 페이지 설명 숨기기:** 검색 결과에서 페이지 다음에 설명(위에서 지정됨)을 포함하지 않도록 검색 엔진에 통지합니다.
- **Cobrowse:** 관리자가 시스템에서 Cobrowse를 사용으로 설정했고 이 사이트에서 Cobrowse가 사용으로 설정된 경우, 이 페이지에 브라우징 세션에 사용할 Cobrowse 단추를 포함할지 선택할 수 있는 옵션이 표시됩니다. Cobrowse 세션은 Oracle Cobrowse Cloud Service를 사용하여 사이트 방문자와의 화면 공유 환경을 관리합니다. **Cobrowse 통합 사용 및 페이지에서 Cobrowse 사용**을 참조하십시오.
- **분석 옵션:** 이 사이트에 분석 추적 코드 조각이 포함된 경우 이 페이지의 코드 조각을 무효화할 수 있습니다. 무효화 스위치를 누르고 스크립트를 편집합니다. 자세한 내용은 **Analytics 추적 추가**를 참조하십시오. 사이트 코드 조각을 편집하면 스크립트가 수정되었음을 알리는 메시지가 표시됩니다. 사용자정의를 제거하려면 **최신 사이트 스크립트로 복원**합니다.

외부 링크를 선택하면 다음 설정이 표시됩니다.

- **페이지 이름:** 문자, 숫자, 밑줄(\_) 및 하이픈(-)을 사용할 수 있습니다. 이름은 사이트 트리에서 페이지를 식별하는 데 사용됩니다. 공백은 자동으로 하이픈으로 바뀝니다.
- **링크 URL:** 페이지의 전체 URL을 지정합니다. 예: `https://www.example.com/sharedpage.htm`.
- **새 창에서 링크 열기:** 새 브라우저 창이나 탭에서 페이지를 열려면 이 옵션을 선택합니다. 이 옵션을 선택하지 않으면 현재 창에서 페이지가 열리고 사이트 페이지 및 탐색을 바꿉니다.
- **페이지 옵션:**
  - **오류 페이지:** 오류로 인해 요청된 페이지를 표시할 수 없는 경우 기본 오류 페이지 대신 이 페이지를 표시합니다. 기본적으로 이 옵션을 선택하면 **탐색에서 페이지 숨기기도** 선택됩니다. 오류 페이지를 탐색에 포함시키려는 경우 이 옵션을 선택 해제할 수 있습니다.
  - **탐색에서 페이지 숨기기:** 자동으로 생성된 사이트 탐색에 페이지를 포함하지 않습니다.
  - **세부정보 페이지:** 이 페이지를 사용하여 다른 페이지에서 선택한 콘텐츠 항목에 대한 자세한 정보를 표시할 수 있습니다. 구조화된 콘텐츠는 엔터프라이즈 사용자만 사용할 수 있습니다. 기본적으로 이 옵션을 선택하면 **탐색에서 페이지 숨기기도** 선택됩니다. 세부정보 페이지를 탐색에 포함시키려는 경우 이 옵션을 선택 해제할 수 있습니다. 구조화된 콘텐츠 항목이 세부정보 페이지를 사용하도록 구성된 경우 사용자가 구조화된 콘텐츠 항목에 대한 자세한 내용을 보기 위해 링크를 누르면 콘텐츠 항목에 대한 세부정보와 함께 세부정보 페이지가 표시됩니다.
  - **검색 페이지:** 이 페이지를 사용하여 검색 결과를 표시할 수 있습니다. 콘텐츠 검색 구성요소의 링크 설정에서 이 페이지를 선택할 수 있습니다. **콘텐츠 검색**을 참조하십시오. 기본적으로 이 옵션을 선택하면 **탐색에서 페이지 숨기기도** 선택됩니다. 검색 페이지를 탐색에 포함시키려는 경우 이 옵션을 선택 해제할 수 있습니다.

4. 현재 업데이트에 보류 중인 변경사항을 모두 저장하려면 **저장**을 누릅니다.

## 페이지 레이아웃 변경

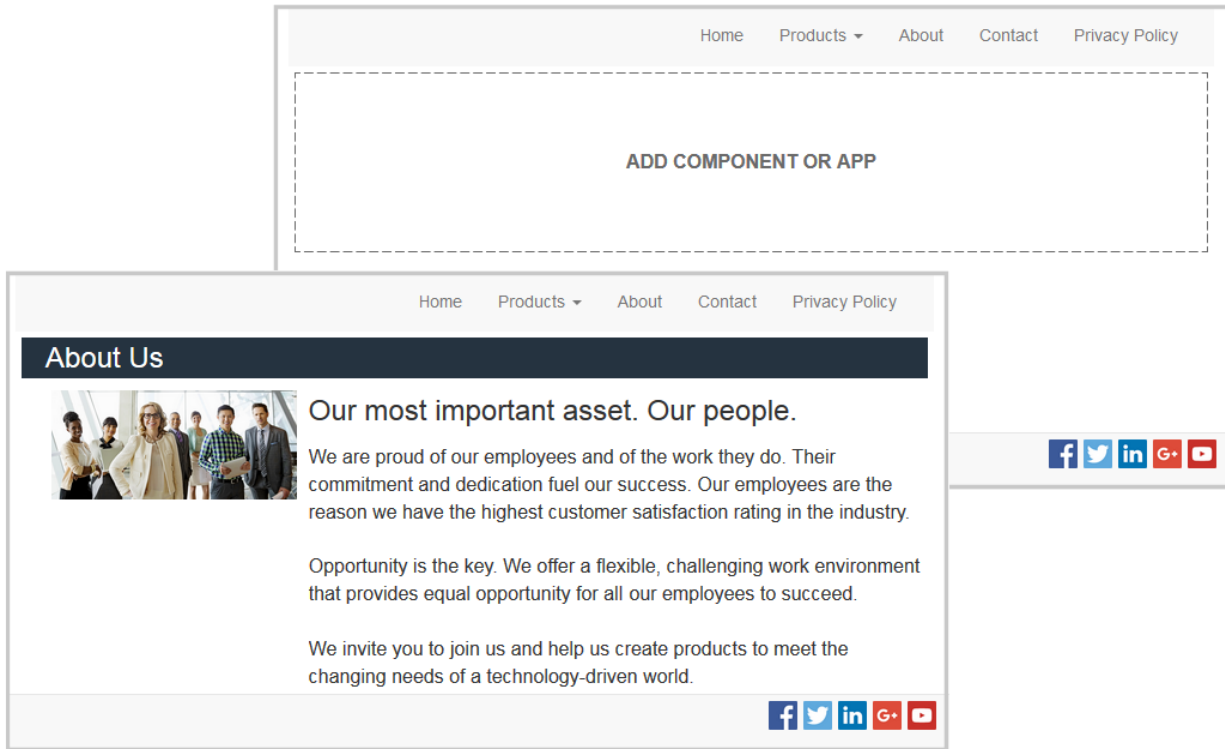
레이아웃은 페이지에 콘텐츠를 배열하는 방법을 정의합니다. 각 레이아웃은 다른 개수의 명명된 슬롯(페이지 너비에 걸친 영역)을 포함할 수 있습니다. 슬롯은 하나 이상의 콘텐츠 유형을 포함할 수 있습니다.

모든 테마는 여러 페이지 레이아웃을 갖습니다. 사이트에 페이지를 추가할 때 해당 페이지에 사용할 레이아웃을 선택합니다. 각 레이아웃에는 콘텐츠를 끌어 놓을 수 있는 슬롯이 있습니다. 이 슬롯에 어떤 콘텐츠를 담을지는 당신에게 달려 있습니다. 제목, 텍스트, 구분선부터 멀티미디어, 갤러리, 소셜 미디어에 이르기까지 어떤 것도 가능합니다. 슬롯에 콘텐츠를 배열할 수 있지만, 페이지상 슬롯의 개수나 배열은 변경할 수 없습니다. 그렇게 하려면 새 페이지 레이아웃을 사용해야 합니다.


### 주:

레이아웃을 다른 것으로 교체할 수 있습니다. 하지만 주의하십시오. 명명된 슬롯이 더 적거나 다른 레이아웃을 선택하면 다른 슬롯의 기존 콘텐츠는 새 레이아웃에 표시되지 않습니다. 콘텐츠는 삭제되지 않으며, 단지 선택한 레이아웃에 동일한 이름의 슬롯이 없는 한 콘텐츠를 표시할 수 없습니다.

다음 그림은 샘플 페이지 레이아웃을 보여줍니다. 페이지 레이아웃의 빈 슬롯과 슬롯에 제목, 이미지 및 텍스트가 추가된 완성된 페이지를 볼 수 있습니다.



페이지에 다른 레이아웃을 선택하려면 다음과 같이 하십시오.

1. 편집을 위해 사이트를 엽니다.
2. 페이지로 이동하고  을 누릅니다.
3. [페이지 레이아웃] 필드로 이동하고 메뉴에서 다른 레이아웃을 선택합니다. 페이지 레이아웃의 개수와 유형은 사이트와 연관된 테마에 따라 달라집니다.
4. 현재 업데이트에 보류 중인 변경사항을 모두 저장하려면 **저장**을 누릅니다.

## 배경 또는 테마 변경

페이지와 페이지상 개별 슬롯 및 슬롯 내 섹션 레이아웃에 대해 배경색과 이미지를 지정할 수 있습니다. 사이트의 전체 테마를 변경할 수도 있습니다.







슬롯은 페이지 너비에 걸친 영역으로, 하나 이상의 콘텐츠 유형을 포함할 수 있습니다. 슬롯에 지정한 배경은 전체 슬롯과 슬롯의 모든 구성요소에 적용됩니다.

### 배경 변경

배경은 층층이 쌓입니다. 슬롯의 배경을 지정하면 페이지에 지정된 배경 “위에” 놓입니다. 대부분의 이미지와 색상의 경우 상층이 사실상 하층을 무효화합니다. 배경색이나 이미지에 투명도를 사용하는 경우 하층의 색상이 비쳐보이거나 상층에 사용된 색상과 섞일 수 있습니다.

단락 및 제목과 같은 일부 구성요소는 미리 정의된 스타일의 일부로 또는 사용자정의된 스타일로 배경색을 제공할 수 있습니다. 구성요소에 대한 배경을 지정하는 경우 섹션 레이아웃, 슬롯 또는 페이지 배경 위로 층층이 쌓입니다. [스타일 및 서식 사용](#)을 참조하십시오.

페이지, 슬롯 및 섹션 레이아웃에 대한 배경 옵션은 액세스하는 방법만 약간 다르고 비슷합니다.

1. 편집을 위해 사이트를 엽니다.
2. 편집할 페이지로 이동하고 가 **편집**으로 설정되어 있는지 확인합니다.
3. 페이지 또는 슬롯에 대한 배경을 변경하려면 다음을 수행합니다.
  - 사이드바에서 을 누르고 을 누릅니다.
  - 전체 페이지의 배경을 지정하려면 **전체 페이지**를 누르고 을 누릅니다. 페이지상 슬롯의 배경을 지정하려면 슬롯을 선택하고 을 누릅니다.
4. 섹션 레이아웃에 대한 배경을 변경하려면 다음을 수행합니다.
  - 섹션 레이아웃에 대한 메뉴 아이콘 을 누르고 **설정**을 선택합니다.
  - **배경**을 누릅니다. 배경 옵션은 페이지, 슬롯 및 섹션 레이아웃 모두 동일합니다.
5. 배경에 이미지를 사용하려면 **이미지 선택**을 누릅니다. 액세스 가능한 모든 위치에서 이미지를 사용할 수 있습니다. 또한 나와 공유된 이미지를 사용하거나 로컬 또는 네트워크 파일 위치에서 업로드한 이미지를 사용할 수 있습니다.
6. 이미지를 선택하고 **확인**을 누릅니다. 모든 사용 가능한 파일이 창에 표시됩니다. 컨텍스트에 적합한 파일 유형을 선택합니다. 예를 들어, 이미지 파일을 선택하는 경우 적합한 이미지 형식의 파일(GIF, JPG, JPEG, PNG, SVG)을 선택합니다.
7. 이미지 설정을 조정합니다.
  - **위치**를 사용하여 페이지나 슬롯에 이미지를 놓습니다.
  - **배율** 옵션을 선택하여 이미지 표시를 조정합니다.
    - **맞춤**: 전체 이미지가 이미지 왜곡 없이 가용 공간에 맞도록 이미지가 조정됩니다.
    - **늘이기**: 두 치수(너비 또는 높이)가 더 크면 가용 공간에 맞게 조정하고, 더 작은 치수는 늘어서 가용 공간을 채웁니다.
    - **없음**: 이미지를 전체 픽셀 해상도로 사용하고 가용 공간에 맞지 않으면 균일하게 자릅니다.
  - **반복** 옵션을 선택하여 가용 공간을 채우기에 너무 작은 이미지를 타일링합니다. **배율** 옵션으로 **늘이기**를 선택한 경우 이 옵션은 적용되지 않습니다.
  - **페이지에서 스크롤 안함**을 선택하여 사용자가 페이지를 스크롤하는 동안 이미지를 고정시킵니다.
8. 배경색을 선택하려면 **색상** 메뉴에서 색상을 선택하거나 **자세히**를 눌러 전체 색 범위에서 색상을 선택합니다.
  - 오른쪽의 스펙트럼 막대를 눌러 색상을 선택하고 색 범위 화면에 색상 변형을 표시합니다. 화면 오른쪽 상단에 선택한 색상을 표시합니다. 선택한 색상의 음영은 왼쪽으로 갈수록 흰색 양이 늘어나고 아래쪽으로 갈수록 검정색 양이 늘어납니다.
  - 색상 변형을 선택하려면 색 범위 화면 안을 누르거나 선택 지점을 눌러 새 색상 위치로 끌어옵니다. 현재 색상 상자와 6자리 색상 코드에 선택한 색상을 표시합니다.
  - 투명도(알파 채널)를 조정하려면 투명도 슬라이더를 왼쪽으로 눌러 끌어서 투명도를 높입니다. 알파 값 0%는 색상을 투명하게 만들어서 하층의 색상과 콘텐츠가 완전히 비쳐보입니다. 알파 값 100%는 색상을 완전히 불투명하게 만들어서 하층의 색상과 콘텐츠를 모두 가립니다. 그 밖의 값은 다른 정도로 하층의 이미지와 색상이 비쳐보이고 하층 색상이 상층 색상과 섞일 수 있습니다.

- 현재 색상 선택을 적용하려면 **선택**을 누릅니다.



## 테마 변경

**저장**을 누르면 테마 변경이 업데이트의 일부로 기록됩니다. 업데이트를 게시해야 테마 변경사항이 사이트에 적용됩니다.

### 주:

모든 테마가 서로 호환되지는 않습니다. 각 테마는 서로 다른 탐색, 스타일 이름 또는 페이지 레이아웃 정보를 포함할 수 있습니다. 예를 들어, 다른 레이아웃 이름을 가진 테마를 선택하면 기존 페이지는 테마에 사용할 수 없는 레이아웃을 사용하게 됩니다. 해당 페이지는 공백으로 나타납니다. 페이지 콘텐츠는 삭제되지 않으며, 단지 선택한 테마에 동일한 이름의 레이아웃과 슬롯이 없는 한 콘텐츠를 표시할 수 없습니다. 호환되지 않는 테마를 선택한 경우 원래 테마로 다시 변경하여 페이지 콘텐츠를 복원할 수 있습니다. 반드시 호환되는 테마로 전환해야 합니다. 확실하지 않으면 테마 개발자에게 문의하십시오.

사이트에 사용된 테마를 변경하려면 다음과 같이 하십시오.

1. 편집을 위해 사이트를 엽니다.
2. 사이드바에서  을 누르고  을 누릅니다.
3. 사용 가능한 테마 목록에서 테마를 선택하고 **확인**을 누릅니다. 현재 업데이트에 테마가 적용됩니다. 현재 업데이트에 보유 중인 변경사항을 모두 저장하려면 **저장**을 누릅니다.



# 9

## 페이지 콘텐츠 배열

구성요소 및 콘텐츠 항목은 사용자가 원하는 기능 및 콘텐츠를 제공하며 섹션 레이아웃은 사용자가 이를 자동으로 배열할 수 있도록 해 줍니다.

- [구성요소 및 섹션 레이아웃 추가](#)
- [자산 및 콘텐츠 항목 작업](#)
- [트리거 및 작업 사용](#)
- [가로 섹션 레이아웃 사용](#)
- [세로 섹션 레이아웃 사용](#)
- [2열 및 3열 레이아웃 사용](#)
- [탭 섹션 레이아웃 사용](#)
- [슬라이더 섹션 레이아웃 사용](#)

개별 구성요소 사용에 대한 자세한 내용은 [내장 구성요소 사용](#)을 참조하십시오.

구성요소 및 레이아웃을 관리하는 방법을 알아보려면 [사용자정의 구성요소 및 레이아웃 관리](#)를 참조하십시오.





고유 구성요소 생성 방법에 대한 자세한 내용은 [구성요소 개발](#)을 참조하십시오.

## 구성요소 및 섹션 레이아웃 추가


구성요소는 웹 페이지의 개별 부품입니다. *섹션 레이아웃*은 여기에 추가되는 콘텐츠를 자동으로 구성하여 제공자가 페이지의 형식을 지정하는 데 시간을 들일 필요 없이 콘텐츠를 손쉽게 추가할 수 있게 해 줍니다.

### 구성요소 추가


페이지에 구성요소를 추가하려면 다음과 같이 하십시오.

1. 편집할 페이지로 이동하고 가 **편집**으로 설정되어 있는지 확인합니다.
2. 을 누르고 다음 구성요소 유형 중 하나를 누릅니다.
  - **테마**를 눌러서 사이트와 연관된 테마에 사용하도록 선택된 구성요소 목록을 표시합니다.
  - **사용자정의**를 눌러서 사용자에게 공유된 사용자정의 구성요소 목록을 표시합니다.
  - **시드**를 눌러서 서비스에 사용할 수 있는 기본 구성요소 목록을 표시합니다.
  - **모두**를 눌러서 사용자에게 공유된 모든 구성요소를 표시합니다.
3. 특정 아이콘을 누른 채 페이지의 원하는 위치로 끌어옵니다. 페이지로 항목을 끌어올 때 사용 가능한 슬롯, 섹션 레이아웃 및 기존 항목의 테두리가 표시됩니다. 배치 아이콘  또는 세로 배치 막대  (왼쪽 또는 오른쪽)는 새 콘텐츠의 이동 위치를 나타냅니다. 섹션 레이아웃 또는 콘텐츠 항목의 단색 테두리는 이 항목을 끌어올 수 있으며 자동으로 배치된다는 점을

나타냅니다. 슬롯에 여러 항목을 가질 수 있고 페이지의 새 위치로 항목을 끌어서 이동할 수 있습니다.

- 올바른 위치에 있을 때 항목을 페이지 위에 놓습니다.
- 항목의 속성을 조정하려면 항목의 메뉴 아이콘  을 누르고 **설정**을 선택합니다. 항목에 따라 텍스트를 추가하거나, 이미지에 링크를 지정하거나, 다른 사이트나 맵의 URL을 지정하거나, 항목 크기를 조정하거나, 기타 작업을 수행해야 합니다.



각 구성요소에는 크기, 맞춤, 간격, 색상, 테두리와 같은 설정이 있습니다. 이는 구성요소의 모양과 작동 방식을 정의합니다. 예를 들어, 단락 설정에는 글꼴 유형, 글꼴 크기와 텍스트가 표시되는 방법을 결정하는 다른 기능이 있습니다.

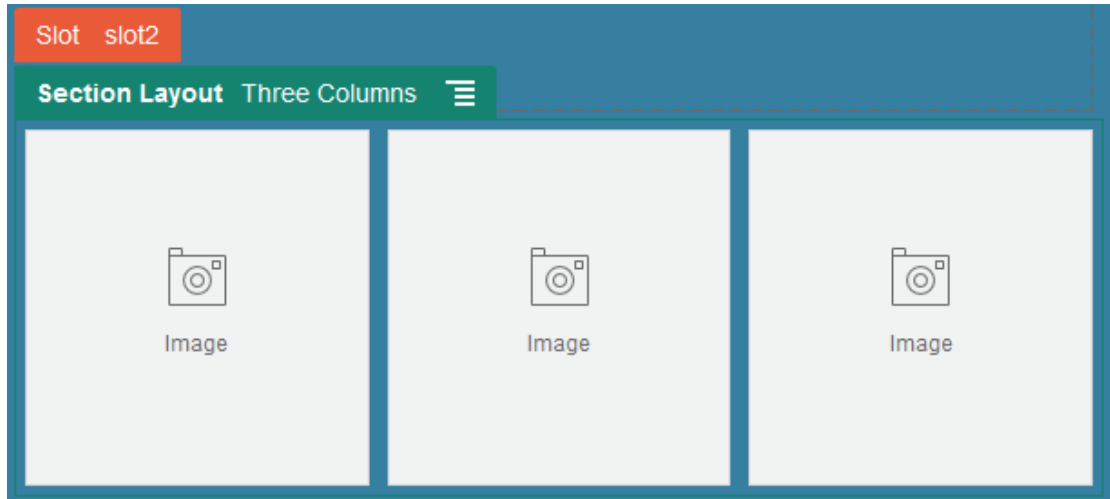
구성요소의 속성을 조정하려면 구성요소를 선택한 다음  메뉴를 누르고 **설정**을 선택합니다. 여러 구성요소는 다른 종류의 설정을 가집니다.


설정	설명
일반 사항	일반 설정은 구성요소 사이에 공통되는 간격, 맞춤 및 설정을 포함합니다.
스타일	스타일은 모양을 제어하는 명명된 기본값 집합입니다. 스타일은 테마에 정의됩니다. 여러 테마는 다른 스타일을 가질 수 있고, 한 테마 안에서 여러 구성요소는 다른 스타일을 가질 수 있습니다. 특정 구성요소 인스턴스에 대한 스타일 설정을 수동으로 지정할 수도 있습니다.
링크	링크 설정은 구성요소에서 사용된 이미지, 문서 또는 기타 리소스의 위치를 포함합니다. 작업을 수행할 단추 및 기타 구성요소의 경우, 링크 설정은 구성요소에서 지원되는 트리거 및 작업도 포함합니다.
구성요소	기사 구성요소와 같은 다른 표준 구성요소를 포함하는 구성요소는 개별 구성요소 목록을 제공하며 각 구성요소에 대한 설정에 액세스할 수 있습니다.
사용자정의	사용자정의 설정은 구성요소에 고유하며 표준 설정 그룹과 별도로 표시됩니다. 예를 들어, 원격 구성요소는 고유 URL에 환경설정을 저장하고 사용자정의 설정으로 표시할 수 있습니다.

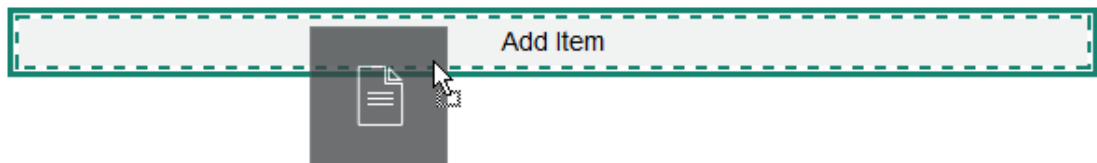
### 섹션 레이아웃 사용

섹션 레이아웃은 콘텐츠를 여러 열, 세로 목록 또는 일련의 탭 영역으로 자동으로 구성할 수 있습니다. 사이트 제공자는 슬롯에 섹션 레이아웃을 하나 이상 추가하여 콘텐츠를 구성할 수 있습니다. 페이지에 레이아웃을 추가하려면 다음을 수행합니다.

- 편집할 페이지로 이동하고  가 **편집**으로 설정되어 있는지 확인합니다.
-  을 눌러서 페이지에 섹션 레이아웃을 삽입합니다. 사용하려는 레이아웃 유형을 선택하고 구성요소를 배치할 때와 같은 지침에 따라 항목을 배치합니다.



배치 아이콘  은 섹션 레이아웃을 이동하는 위치(위, 아래, 왼쪽, 오른쪽)를 나타냅니다. 섹션 레이아웃 또는 콘텐츠 항목의 단색 테두리는 이 항목을 끌어올 수 있으며 자동으로 배치된다는 점을 나타냅니다:






구조화된 콘텐츠 항목으로 작업하는 엔터프라이즈 사용자는 페이지에 콘텐츠 항목 목록을 추가할 때 섹션 레이아웃을 지정할 수 있으며 이 섹션 레이아웃은 페이지의 항목을 자동으로 형식 지정합니다.

개발자는 제공된 레이아웃 외에도 추가 섹션 레이아웃을 생성하여 특정 레이아웃 문제를 해결하거나 제공자를 위해 작성을 간소화할 수 있습니다. 테마 디자이너는 페이지 레이아웃의 슬롯에 섹션 레이아웃을 구성할 수도 있습니다. [레이아웃 개발](#)을 참조하십시오.

## 자산 및 콘텐츠 항목 작업

엔터프라이즈 사용자인 경우 사이트에 *사이트 모음* 또는 연관된 저장소에 저장된 디지털 자산 및 콘텐츠 항목을 포함시킬 수 있습니다. 이 모음은 사이트에서 사용할 수 있는 저장소에 있는 자산 부분 집합입니다.

페이지에 디지털 자산 또는 콘텐츠 항목을 추가하려면  가 **편집**으로 설정되었는지 확인하고  을 누릅니다. 사이트에 여러 저장소가 사용될 경우 사용할 저장소를 선택합니다. 사이트에서 여러 저장소 사용에 대한 자세한 내용은 *Managing Assets with Oracle Content Management*의 사이트에 여러 저장소에 대한 액세스 권한 부여를 참조하십시오.

자산을 필터링하여 필요한 항목을 정확하게 찾을 수 있습니다.  을 누르고 선택 범위를 좁히기 위한 방법을 선택합니다. 자산이 표시되지 않을 경우 자산이 사이트 모음에 속하지 않는 것일 수 있습니다. 사이트 모음 대신 저장소의 모든 모음이 표시되도록 필터를 변경하여 이 방법이 유용한지 여부를 확인합니다. 자세한 내용은 자산 검색, 필터 및 정렬을 참조하십시오.

패널에서 디지털 자산 또는 콘텐츠 항목을 끌어와 페이지의 슬롯에 놓습니다. 대체 텍스트 입력 옵션을 사용하여 커서 위치의 단락 구성요소에 이미지를 직접 포함하고 이미지 높이 및 너비를 설정한 다음 맞춤을 설정할 수 있습니다.

사이트 모음의 항목을 페이지에 추가하는 경우 적합한 유형의 구성요소에 자동으로 배치됩니다. 예를 들어 이미지 디지털 자산을 추가하는 경우 이미지 구성요소에 자동으로 배치됩니다. 콘텐츠 항목을 추가하는 경우 콘텐츠 항목 구성요소에 자동으로 배치됩니다. 이미지 또는 콘텐츠 항목 구성요소를 먼저 추가하고 나중에 [콘텐츠] 패널에서 콘텐츠 항목을 구성요소로 끌어올 수도 있습니다. 또는 구성요소에 대해 **설정**을 선택한 후 **선택**을 눌러서 자산 또는 문서 목록에서 이미지를 선택할 수 있습니다.

## 트리거 및 작업 사용

단추 구성요소는 페이지 구성요소 표시/숨기기 및 메시지 표시와 같은 작업을 하나 이상 시작할 수 있습니다. 폴더 및 파일 목록과 같은 특정 구성요소는 사용자의 선택사항에 따라 동반 구성요소에서 작업을 시작할 수 있습니다.

예를 들어, 단추를 누르면(트리거) 사용자가 다른 페이지나 외부 URL로 이동하고 변경사항을 알리는 경보가 표시되도록(작업) 단추를 구성할 수 있습니다. 표시는 사용되는 구성요소 종류에 따라 변경됩니다.

**Configure Trigger Actions**

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

**Page Actions**

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

**Show Alert**

Message: You are going to the About page.

**And**

**Navigate to Site Page**

Page: About

Target: Open in Same Window

OK Cancel

여러 개의 작업을 사용하는 경우 작업 순서를 고려하여 수행할 순서대로 작업을 배치하십시오. 위의 예제에서는 먼저 경보 작업을 나열합니다. 사용자가 페이지로 재지정되기 전에 메시지를 읽고 해제할 시간이 주어집니다. 먼저 재지정 작업을 나열하면 사용자가 메시지를 읽기도 전에 새 페이지로 바뀔 수 있습니다.

### 페이지 작업

트리거 및 작업을 지원하는 모든 구성요소는 페이지 작업을 지원합니다.

- **사이트 페이지로 이동:** 현재 사이트의 페이지를 선택합니다.
- **외부 URL로 이동:** 외부 페이지나 사이트의 전체 URL을 지정합니다.
- **경보 표시:** 지정된 메시지를 창에 표시합니다.
- **구성요소 표시 또는 숨기기:** 현재 페이지의 구성요소 목록에서 표시하거나 숨기거나 토글할 구성요소를 선택합니다.
- **슬롯 표시 또는 숨기기:** 현재 페이지의 슬롯 목록에서 표시하거나 숨기거나 토글할 슬롯을 선택합니다.


### 구성요소 특정 작업

페이지 작업 외에도 구성요소 자체 작업을 정의할 수 있습니다. 이 작업을 통해 구성요소는 다른 구성요소와 통신하고 동반 구성요소에서 작업을 시작할 수 있습니다. 예를 들어, 사용자가 폴더 목록 구성요소에서 폴더를 선택하면 파일 목록 구성요소는 선택한 폴더의 파일을 표시할 수 있습니다. 이 경우 폴더 목록 구성요소는 **폴더가 선택됨** 트리거를 지원하고 파일 목록 구성요소는 **파일 표시** 작업을 지원합니다.

페이지에 작업을 지원하는 구성요소를 추가하면 구성요소와 이들이 지원하는 작업이 사용 가능한 작업 목록에 추가됩니다. 일부 구성요소는 특정 작업만 지원합니다. 일부 구성요소는 트리거를 제공하지만 자체적으로 어떤 작업을 지원하지 않습니다.

고유의 구성요소를 생성하는 방법을 알아보려면 [구성요소 개발\(를\)](#) 참조하십시오.

예를 들어, 단추 구성요소에 하나 이상의 작업을 지정하려면 다음과 같이 하십시오.


1. 단추의 메뉴 아이콘  을 누르고 **설정**을 선택합니다.
2. 설정 패널에서 **링크**를 누릅니다.
3. **링크 유형 선택**을 누르고 **트리거 작업**을 선택합니다.
4. [사용 가능한 트리거]에서 **단추 누르기**를 누릅니다.
5. [트리거 작업 구성] 창에서 왼쪽 열에서 작업을 눌러 끌어서 **첫번째 작업** 레이블이 붙은 슬롯에 놓습니다.

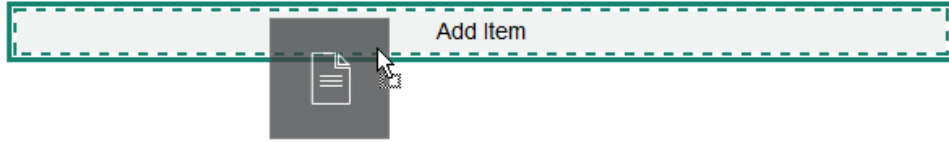
## 가로 섹션 레이아웃 사용

섹션 레이아웃을 사용하면 레이아웃에 추가하는 구성요소의 간격 및 배열이 자동으로 지정되도록 할 수 있습니다.

가로 레이아웃은 추가된 항목을 가로줄에 차례로 배열합니다. 페이지 너비가 늘어나거나 줄어들면 이에 비례하여 레이아웃이 변경됩니다. 기본적으로 항목은 한 줄에 맞춰지며 같은 가로 간격이 할당됩니다.


페이지에 레이아웃을 추가하려면 다음을 수행합니다.

1. 편집할 페이지로 이동하고  가 **편집**으로 설정되어 있는지 확인합니다.
2. 페이지에 레이아웃을 추가합니다.
3. 섹션 레이아웃에 다른 콘텐츠를 배치하려면 콘텐츠를 레이아웃으로 끌어 놓습니다.  
레이아웃은 단색 테두리와 [항목 추가]를 표시하는 배너로 강조 표시됩니다.



레이아웃에 항목을 계속 추가할 수 있으며 이 레이아웃은 항목 추가에 맞춰 형식이 지정됩니다. 다른 섹션 레이아웃을 추가하여 정교한 레이아웃을 생성할 수도 있습니다. 다음은 레이아웃의 항목 순서를 나타내기 위해 번호가 매겨진 텍스트 구성요소가 있는 가로 레이아웃입니다.




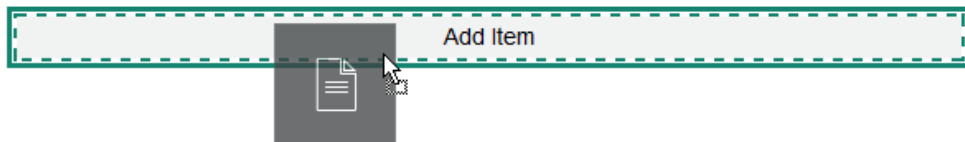
4. 구성요소 및 구성요소 모양을 편집하려면 메뉴 아이콘  을 누르고 **설정**을 선택합니다. 기본 비례 크기 조정을 사용하지 않으려는 경우 개별 영역에 대한 너비를 설정할 수 있습니다. 맞춤(왼쪽, 가운데 또는 오른쪽)도 설정할 수 있습니다.
5. [일반 사항] 탭을 사용하여 레이아웃의 개별 구성요소에 대한 설정을 수정합니다. 구성요소 이름을 눌러 해당 구성요소에 대한 설정을 표시합니다.
6. [배경] 탭을 사용하여 레이아웃에 대한 배경 설정을 수정합니다. [배경 또는 테마 변경](#)을 참조하십시오.

## 2열 및 3열 레이아웃 사용

다중 열 섹션 레이아웃을 사용하면 레이아웃에 추가하는 구성요소의 간격 및 배열이 자동으로 지정되도록 할 수 있습니다.

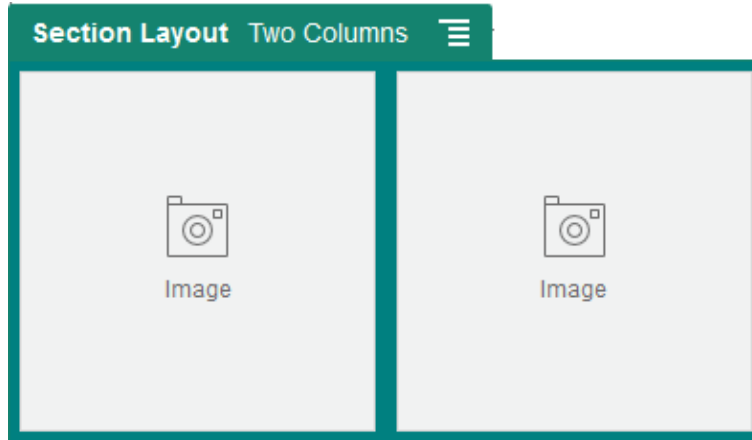
페이지에 레이아웃을 추가하려면 다음을 수행합니다.

1. 편집할 페이지로 이동하고  가 **편집**으로 설정되어 있는지 확인합니다.
2. 페이지에 레이아웃을 추가합니다.
3. 섹션 레이아웃에 다른 콘텐츠를 배치하려면 콘텐츠를 레이아웃으로 끌어 놓습니다. 레이아웃은 단색 테두리와 [항목 추가]를 표시하는 배너로 강조 표시됩니다.




레이아웃에 항목을 계속 추가할 수 있으며 이 레이아웃은 항목 추가에 맞춰 형식이 지정됩니다. 다른 섹션 레이아웃을 추가하여 정교한 레이아웃을 생성하거나 구성요소 그룹을 추가할 수도 있습니다.

다음은 위치 표시자인 이미지 구성요소가 사용된 2열 레이아웃입니다.



항목을 더 추가하면 추가 행에 표시되며 각 행에는 항목이 최대 두 개까지 포함됩니다.


4. 구성요소 및 구성요소 모양을 편집하려면 메뉴 아이콘  을 누르고 **설정**을 선택합니다.
5. [일반 사항] 탭을 사용하여 레이아웃의 개별 구성요소에 대한 설정을 수정합니다.  
구성요소 이름을 눌러 해당 구성요소에 대한 설정을 표시합니다.
6. [배경] 탭을 사용하여 레이아웃에 대한 배경 설정을 수정합니다.
7. **사용자정의 설정**을 선택하여 표시되는 콘텐츠에 대한 추가 기본값을 설정합니다.
  - **첫번째 열 너비(%)**: 열 너비를 레이아웃에 사용할 수 있는 공간의 백분율로 지정합니다.
  - **두번째 열 너비(%)**: 열 너비를 레이아웃에 사용할 수 있는 공간의 백분율로 지정합니다.
  - **세번째 열 너비(%)**: 열 너비를 레이아웃에 사용할 수 있는 공간의 백분율로 지정합니다.
  - **반응형 중단점(픽셀)**: 사용 가능한 표시 크기가 다른 경우 콘텐츠를 자동으로 재형식화하는 반응형 페이지 디자인의 경우 너비를 픽셀로 지정합니다. 이 경우 섹션 레이아웃은 표준 2열 레이아웃과 아래에서 지정된 **반응형 동작** 옵션 간을 전환합니다.
  - **반응형 동작**: 사용 가능한 표시 크기가 **반응형 중단점** 값보다 작은 경우 레이아웃을 변경하는 방법을 선택합니다.
    - **작업 없음**: 레이아웃 동작을 조정하지 않습니다.
    - **열 누적**: 열 하나에 위쪽에서 아래쪽으로 1열의 모든 항목이 나오고 그 뒤에 2열의 모든 항목이 오도록 배열합니다.
    - **첫번째 열 숨기기**: 첫번째 열의 콘텐츠를 숨겨 나머지 열에 더 많은 공간을 제공합니다.
    - **두번째 열 숨기기**: 두번째 열의 콘텐츠를 숨겨 나머지 열에 더 많은 공간을 제공합니다.
    - **세번째 열 숨기기**: 세번째 열의 콘텐츠를 숨겨 나머지 열에 더 많은 공간을 제공합니다.
    - **양쪽 열 숨기기**: 레이아웃의 모든 콘텐츠를 숨깁니다.
    - **첫번째 열 아래로 두번째 열 이동**: 열 하나에 항목을 배열하며 1열의 모든 항목 뒤로 2열의 모든 항목이 표시됩니다.
    - **두번째 열 아래로 첫번째 열 이동**: 열 하나에 항목을 배열하며 2열의 모든 항목 뒤로 1열의 모든 항목이 표시됩니다.

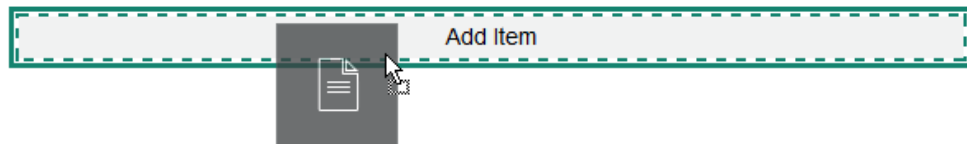
## 세로 섹션 레이아웃 사용

섹션 레이아웃을 사용하면 레이아웃에 추가하는 구성요소의 간격 및 배열이 자동으로 지정되도록 할 수 있습니다.

세로 레이아웃은 추가된 항목을 세로줄에 차례로 배열합니다.

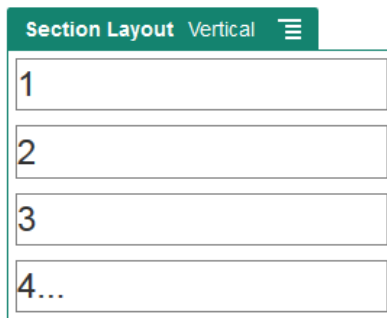
페이지에 레이아웃을 추가하려면 다음을 수행합니다.


1. 편집할 페이지로 이동하고 가 **편집**으로 설정되어 있는지 확인합니다.
2. 페이지에 레이아웃을 추가합니다.
3. 섹션 레이아웃에 다른 콘텐츠를 배치하려면 콘텐츠를 레이아웃으로 끌어 놓습니다.  
레이아웃은 단색 테두리와 [항목 추가]를 표시하는 배너로 강조 표시됩니다.



레이아웃에 항목을 계속 추가할 수 있으며 이 레이아웃은 항목 추가에 맞춰 형식이 지정됩니다. 다른 섹션 레이아웃을 추가하여 정교한 레이아웃을 생성할 수도 있습니다.

다음은 레이아웃의 항목 순서를 나타내기 위해 번호가 매겨진 텍스트 구성요소가 있는 세로 레이아웃입니다.





4. 구성요소 및 구성요소 모양을 편집하려면 메뉴 아이콘 을 누르고 **설정**을 선택합니다.
5. [일반 사항] 탭을 사용하여 레이아웃의 개별 구성요소에 대한 설정을 수정합니다.  
구성요소 이름을 눌러 해당 구성요소에 대한 설정을 표시합니다.
6. [배경] 탭을 사용하여 레이아웃에 대한 배경 설정을 수정합니다.



## 탭 섹션 레이아웃 사용



탭 섹션 레이아웃을 사용하여 레이아웃에 추가하는 구성요소의 간격 및 배열을 생성할 수 있습니다. 페이지에 레이아웃을 추가하려면 다음을 수행합니다.

1. 편집할 페이지로 이동하고 가 **편집**으로 설정되어 있는지 확인합니다.
2. 페이지에 레이아웃을 추가합니다. 기본적으로 단일 탭이 레이아웃에 추가됩니다. **새 탭**을 눌러서 탭을 추가합니다.
3. 탭에 다른 콘텐츠를 배치하려면 콘텐츠를 탭으로 끌어다 놓습니다.  
레이아웃에 항목을 계속 추가할 수 있으며 이 레이아웃은 항목 추가에 맞춰 형식이 지정됩니다. 다른 섹션 레이아웃을 추가하여 정교한 레이아웃을 생성하거나 구성요소 그룹을 추가할 수도 있습니다.
4. 탭 및 탭 모양을 편집하려면 메뉴 아이콘 을 누르고 **설정**을 선택합니다.
5. 배경 섹션을 사용해서 탭에 대한 배경 설정을 수정합니다.  
탭 이미지 사용, 위치 변경 등을 수행할 수 있습니다. **배경 또는 테마 변경**을 참조하십시오.
6. 스타일 섹션을 사용해서 탭의 다른 모양 설정을 수정할 수 있습니다. 실선, 프레임 등의 스타일을 선택하거나 탭의 테두리 및 모서리를 사용자가 정의할 수 있습니다.

## 슬라이더 섹션 레이아웃 사용

슬라이더 섹션 레이아웃을 사용하여 새 콘텐츠를 위한 공간이 확보되도록 한 쪽으로 슬라이드될 때까지 제자리에 유지되는 콘텐츠를 생성할 수 있습니다. 사용자는 슬라이드 섹션 아래의 탐색 점을 눌러 슬라이드를 탐색합니다.

페이지에 슬라이더 레이아웃을 추가하려면 다음과 같이 하십시오.

1. 편집할 페이지로 이동하고 가 **편집**으로 설정되어 있는지 확인합니다.
2. 페이지에 레이아웃을 추가합니다. 기본적으로 단일 슬라이드가 레이아웃에 추가됩니다. **새 슬라이드**를 눌러 다른 슬라이드를 추가합니다.
3. 특정 슬라이드에 콘텐츠를 배치하려면 슬라이더에 슬라이드를 표시한 다음 구성요소 또는 콘텐츠 항목을 슬라이드로 끌어 놓습니다.  
계속해서 슬라이드에 항목을 추가하고 필요에 따라 항목의 위치를 지정할 수 있습니다. 다른 섹션 레이아웃을 추가하여 정교한 레이아웃을 생성하거나 구성요소 그룹을 추가하는 방식으로 원하는 위치에 콘텐츠를 배치할 수도 있습니다.
4. 슬라이드 레이아웃 및 모양을 편집하려면 메뉴 아이콘 을 누른 다음 **설정**을 선택합니다.
5. [배경] 섹션에서 전체 슬라이드 영역에 대한 배경 설정을 수정합니다.  
**배경 또는 테마 변경**을 참조하십시오. 이미지 사용, 위치 변경 등을 수행할 수 있습니다.
6. [스타일] 섹션에서 기타 모양 설정을 수정합니다. 실선, 프레임 등의 스타일을 선택하거나 슬라이드의 테두리 및 모서리를 사용자정의합니다.
7. 각 슬라이드에 대한 설정을 개별적으로 수정하려면 슬라이드 이름을 눌러 해당 설정을 엽니다. 필요에 따라 각 슬라이드의 배경, 스타일 등을 변경합니다.

# 10

## 내장 구성요소 사용

구성요소, 섹션 레이아웃, 디지털 자산 및 구조화된 콘텐츠에는 사용자가 원하는 콘텐츠 및 기능을 제공할 수 있는 유연성이 있습니다.

이 절에서는 개별 구성요소 사용에 대한 정보를 제공합니다. 구성요소에 대한 다른 정보는 다음을 참조하십시오.

- 사이트에서 구성요소를 사용하는 방법을 알아보려면 [페이지 콘텐츠 배열](#)을 참조하십시오.
- 구성요소를 관리하는 방법을 알아보려면 [사용자정의 구성요소 및 레이아웃 관리](#)(를) 참조하십시오.
- 고유의 구성요소를 생성하는 방법을 알아보려면 [구성요소 개발](#)을 참조하십시오.

### 기본 구성요소

- [제목](#)
- [단락](#)
- [일반 텍스트](#)
- [단추](#)

### 구조 구성요소

- [구분선](#)
- [스페이스](#)

### 미디어 구성요소

- [이미지](#)
- [갤러리](#)
- [갤러리 그리드](#)
- [YouTube 비디오](#)
- [비디오](#)

### 문서 구성요소

- [문서](#)
- [폴더 목록](#)
- [파일 목록](#)
- [문서 관리자](#)
- [프로젝트 라이브러리](#)

### 소셜 구성요소

- [소셜 표시줄](#)

- Facebook 좋아요 및 추천
- Twitter 공유 및 팔로우
- 대화 구성요소
- 대화 목록

#### 프로세스 구성요소

- 프로세스 시작 폼
- 프로세스 작업 목록
- 작업 세부정보 폼

#### 콘텐츠 항목

- 콘텐츠 항목 구성요소
- 콘텐츠 위치 표시자
- 콘텐츠 목록
- 콘텐츠 검색
- 권장사항

#### 기타 구성요소

- 맵
- 헤드라인
- 기사
- 텍스트가 있는 이미지
- 구성요소 그룹
- 페이지에서 Cobrowse 사용
- Oracle Intelligent Advisor
- Oracle Visual Builder


## 기본 구성요소

기본 구성요소는 텍스트 콘텐츠의 빌딩 블록을 제공합니다.


- 제목
- 단락
- 일반 텍스트
- 단추

## 제목

제목 구성요소를 추가하고 서식을 지정하려면 다음과 같이 하십시오.

1. 편집할 페이지로 이동하고 가 편집으로 설정되어 있는지 확인합니다.
2. 페이지에 구성요소를 추가합니다.

3. 제목 구성요소를 눌러 제목 텍스트를 입력합니다. 구성요소의 기본 스타일 서식으로 텍스트가 표시됩니다.
4. 제목 텍스트 내에 링크를 추가하려면 다음을 수행합니다.

a. 링크 텍스트로 사용할 텍스트를 입력하고 선택한 다음  을 누릅니다.


b. 링크 유형 선택을 누르고 다음 옵션 중 하나를 선택합니다.

- **웹 페이지:** 외부 페이지나 사이트의 전체 URL을 지정하고 링크를 열 위치를 선택합니다.
- **사이트 페이지:** 페이지 선택기를 사용하여 현재 사이트에서 페이지를 선택하고 링크를 열 위치를 선택합니다. `key1=value1&key2=value2` 형식으로 추가 URL 매개변수를 지정할 수 있습니다. 비어 있는 값이 지원됩니다(예: `key1=&key2=value2`). 또한 URL 앵커를 지정할 수 있지만 앵커 링크를 확인할 대상 사이트 페이지의 위치에 특수한 앵커 섹션 레이아웃을 추가하고, 트리거 작업을 정의할 때 사용한 레이아웃 설정에서 동일한 앵커 이름을 지정해야 합니다.

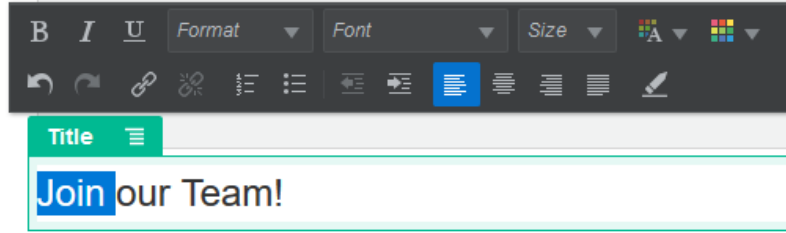
 **주:**

URL 앵커를 사용하는 데 필요한 앵커 섹션 레이아웃은 OCE Toolkit에서 배포됩니다. 툴킷을 얻는 방법에 대한 자세한 내용은 OCE Toolkit으로 개발을 참조하십시오.


- **파일 다운로드:** 저장소에서 선택한 파일을 다운로드합니다. 파일을 선택합니다. 디지털 자산을 선택한 경우, 특정 표현을 선택할 수 있습니다. 표현을 선택하지 않으면 원래 크기가 사용됩니다. 사이트가 게시될 때 자산의 최신 버전이 게시되도록 하려면 **자산의 최신 버전 사용**을 선택합니다. **자산의 최신 버전 사용**을 선택하지 않으면 가장 최근 초안 버전이 아닌 가장 최근 **게시된** 버전이 사용됩니다.
- **파일 미리보기:** 희미한 비활성 페이지 버전 위에 선택한 파일을 겹쳐 미리보고 필요에 따라 다운로드합니다.
- **콘텐츠 항목:** 연관된 자산 저장소에서 콘텐츠 항목을 선택하고, 표시할 세부정보 페이지를 선택하고, 페이지가 동일한 창 또는 새 창에서 열리도록 지정합니다.
- **전자메일:** 적합한 전자메일 주소와 제목(선택사항)을 지정합니다. 결과 메시지가 열리고 기본 전자 메일 클라이언트를 통해 전송됩니다.
- **맵:** 적합한 주소 또는 좌표를 입력하고 데스크톱 및 모바일 브라우저에서 맵을 열 위치를 선택합니다.
- **전화 통화:** 적합한 전화 번호를 입력합니다.

링크를 제거하려면 링크 텍스트 아무 곳이나 누른 후  을 누릅니다.

5. 기본 서식을 변경하려면 서식을 지정할 텍스트를 선택하고 서식 도구모음에서 글꼴, 색상, 맞춤과 같은 옵션을 선택합니다. 테마에 따라 일부 옵션이 표시되지 않을 수 있습니다.








변경사항이 즉시 적용됩니다. 이 서식 변경사항은 기본 스타일보다 우선 적용됩니다. 기본 스타일을 변경해도 우선 적용은 그대로 남습니다.

- 이 옵션으로 적용된 서식을 제거하려면 텍스트를 선택하고  을 누릅니다.

## 단락

단락 구성요소를 추가하고 서식을 지정하려면 다음과 같이 하십시오.

- 편집할 페이지로 이동하고  가 **편집**으로 설정되어 있는지 확인합니다.
- 페이지에 구성요소를 추가합니다.
- 단락 구성요소를 눌러 텍스트를 입력합니다. 구성요소의 기본 스타일 서식으로 텍스트가 표시됩니다.
- 비디오() , 이미지() 또는 콘텐츠 항목()을 단락에 추가하려면 적절한 아이콘을 누릅니다. 추가할 자산을 선택하고 필요에 따라 설정을 조정합니다. 예를 들어, 콘텐츠 항목을 삽입하고 다른 뷰를 선택하여 단락 내의 복잡한 레이아웃을 제어합니다.
- 단락 구성요소 내에 단락을 추가하려면 Enter만 누릅니다.
- 그래프 내에 링크를 추가하려면 다음을 수행합니다.
  - Enter를 누르고 링크 텍스트로 사용하려는 텍스트를 선택한 후  을 누릅니다.
  - 링크 유형 선택을 누르고 다음 옵션 중 하나를 선택합니다.
    - 웹 페이지:** 외부 페이지나 사이트의 전체 URL을 지정하고 링크를 열 위치를 선택합니다.
    - 사이트 페이지:** 페이지 선택기를 사용하여 현재 사이트에서 페이지를 선택하고 링크를 열 위치를 선택합니다. `key1=value1&key2=value2` 형식으로 추가 URL 매개변수를 지정할 수 있습니다. 비어 있는 값이 지원됩니다(예: `key1=&key2=value2`). 또한 URL 앵커를 지정할 수 있지만 앵커 링크를 확인할 대상 사이트 페이지의 위치에 특수한 앵커 섹션 레이아웃을 추가하고, 트리거 작업을 정의할 때 사용한 레이아웃 설정에서 동일한 앵커 이름을 지정해야 합니다.


### 주:

URL 앵커를 사용하는 데 필요한 앵커 섹션 레이아웃은 OCE Toolkit에서 배포됩니다. 툴킷을 얻는 방법에 대한 자세한 내용은 OCE Toolkit으로 개발을 참조하십시오.

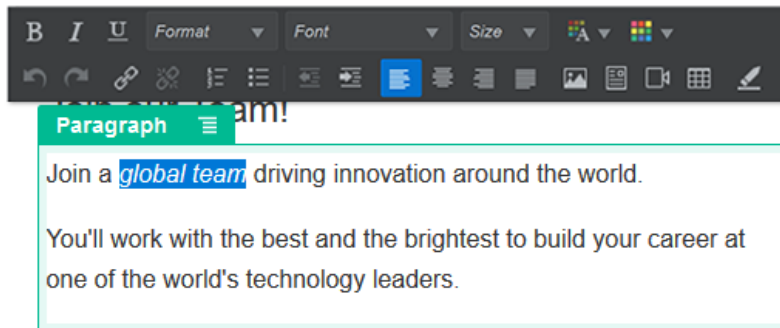
- 파일 다운로드:** 저장소에서 선택한 파일을 다운로드합니다. 파일을 선택합니다.

디지털 자산을 선택한 경우, 특정 표현을 선택할 수 있습니다. 표현을 선택하지 않으면 원래 크기가 사용됩니다. 사이트가 게시될 때 자산의 최신 버전이 게시되도록하려면 **자산의 최신 버전 사용**을 선택합니다. **자산의 최신 버전 사용**을 선택하지 않으면 가장 최근 초안 버전이 아닌 가장 최근 *게시된* 버전이 사용됩니다.


- **파일 미리보기:** 희미한 비활성 페이지 버전 위에 선택한 파일을 겹쳐 미리보고 필요에 따라 다운로드합니다.
- **콘텐츠 항목:** 연관된 자산 저장소에서 콘텐츠 항목을 선택하고, 표시할 세부정보 페이지를 선택하고, 페이지가 동일한 창 또는 새 창에서 열리도록 지정합니다.
- **전자메일:** 적합한 전자메일 주소와 제목(선택사항)을 지정합니다. 결과 메시지가 열리고 기본 전자 메일 클라이언트를 통해 전송됩니다.
- **맵:** 적합한 주소 또는 좌표를 입력하고 데스크톱 및 모바일 브라우저에서 맵을 열 위치를 선택합니다.
- **전화 통화:** 적합한 전화 번호를 입력합니다.

링크를 제거하려면 링크 텍스트 아무 곳이나 누른 후 을 누릅니다.

- 어떤 텍스트 부분의 기본 서식을 변경하려면 서식을 지정할 텍스트를 선택하고 서식 도구모음에서 글꼴, 색상, 맞춤과 같은 옵션을 선택합니다. 테마에 따라 일부 옵션이 표시되지 않을 수 있습니다.



변경사항이 즉시 적용됩니다. 이 서식 변경사항은 기본 스타일보다 우선 적용됩니다. 기본 스타일을 변경해도 우선 적용은 그대로 남습니다.


- 이러한 옵션이 적용된 서식을 제거하려면 텍스트를 선택하고 을 누릅니다.

## 일반 텍스트

텍스트 구성요소를 통해 페이지에 텍스트를 추가하고 테마에서 정의된 스타일을 사용하여 단독으로 서식을 지정할 수 있습니다. 콘텐츠를 편집할 때 서식 도구모음은 사용할 수 없으므로 선택된 스타일을 사용자정의 서식으로 무효화할 수 없습니다.

이는 머리글 등 요소에 대해 표준화된 서식을 사용하는 경우 유용할 수 있습니다. 테마에서 정의된 스타일을 업데이트하고 사이트 전체에서 업데이트를 방해하는 서식 무효화 없이 연관된 텍스트를 자동으로 업데이트할 수 있습니다.

텍스트 구성요소를 추가하려면 다음과 같이 하십시오.

1. 편집할 페이지로 이동하고 가 **편집**으로 설정되어 있는지 확인합니다.
2. 페이지에 구성요소를 추가합니다.

- 단락 구성요소를 눌러 텍스트를 입력합니다. 이 텍스트는 구성요소(<p> 또는 단락 요소)에 대한 기본 스타일의 서식에 사용됩니다.

다른 단락을 추가하려는 경우 Enter를 누르면 됩니다.

 **주:**

다른 소스의 콘텐츠를 복사하여 텍스트 구성요소로 붙여 넣을 수 있지만 이미지와 기본 HTML 태그가 제거됩니다.

- 텍스트 구성요소에 대한 기본 스타일을 변경하려면 다음 작업을 수행합니다.

- 구성요소의 메뉴 아이콘  을 누르고 **설정**을 선택합니다.

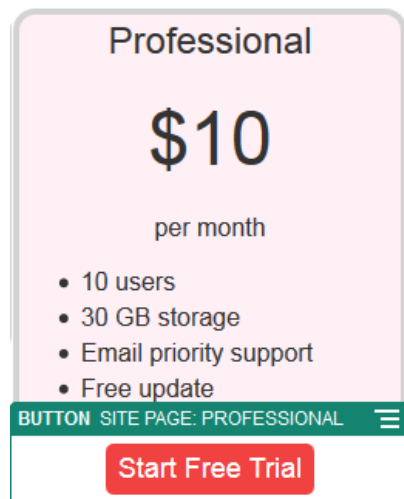
- 스타일 선택**을 누르고 메뉴에서 스타일을 선택합니다.

목록에는 단락 및 머리글에 대한 표준 HTML 태그가 포함됩니다. 선택하는 스타일은 콘텐츠에 연관된 태그를 지정합니다.



선택한 기본 스타일이 구성요소에 반영됩니다.

## 단추

단추를 사용하여 페이지에 링크나 다른 기능을 더 뚜렷이 만들 수 있습니다.



페이지에 단추를 추가하려면 다음과 같이 하십시오.

- 편집할 페이지로 이동하고  가 **편집**으로 설정되어 있는지 확인합니다.
- 페이지에 구성요소를 추가합니다.
- 단추의 속성을 조정하려면 메뉴 아이콘  을 누르고 **설정**을 선택합니다.  
단추 텍스트(레이블), 크기, 맞춤 및 기타 단추 표시 옵션을 지정할 수 있습니다.

 주:

너비 및 높이 필드를 0으로 설정할 경우 레이블에 지정된 텍스트에 맞게 단추 크기가 자동으로 조정됩니다.

배경색, 글꼴, 테두리 및 기타 설정을 지정하려면 **스타일** 탭을 누릅니다. 현재 테마에 미리 정의된 스타일 중에서 선택할 수 있습니다. 테마에 따라 **사용자정의**를 눌러 고유 값을 지정할 수도 있습니다.

4. 단추에 링크나 다른 동작을 연관시킵니다.
  - a. 설정 패널에서 **링크**를 누릅니다.
  - b. **링크 유형** 선택을 누르고 다음 옵션 중 하나를 선택합니다.
    - **링크 없음**: 사용자가 단추를 누를 때 아무 동작도 수행하지 않습니다.
    - **웹 페이지**: 외부 페이지나 사이트의 전체 URL을 지정하고 링크를 열 위치를 선택합니다.
    - **사이트 페이지**: 페이지 선택기를 사용하여 현재 사이트에서 페이지를 선택하고 링크를 열 위치를 선택합니다. `key1=value1&key2=value2` 형식으로 추가 URL 매개변수를 지정할 수 있습니다. 비어 있는 값이 지원됩니다(예: `key1=&key2=value2`). 또한 URL 앵커를 지정할 수 있지만 앵커 링크를 확인할 대상 사이트 페이지의 위치에 특수한 앵커 섹션 레이아웃을 추가하고, 트리거 작업을 정의할 때 사용한 레이아웃 설정에서 동일한 앵커 이름을 지정해야 합니다.

 주:

URL 앵커를 사용하는 데 필요한 앵커 섹션 레이아웃은 OCE Toolkit에서 배포됩니다. 툴킷을 얻는 방법에 대한 자세한 내용은 OCE Toolkit으로 개발을 참조하십시오.

- **파일 다운로드**: 저장소에서 선택한 파일을 다운로드합니다. 파일을 선택합니다. 디지털 자산을 선택한 경우, 특정 표현을 선택할 수 있습니다. 표현을 선택하지 않으면 원래 크기가 사용됩니다. 사이트가 게시될 때 자산의 최신 버전이 게시되도록 하려면 **자산의 최신 버전 사용**을 선택합니다. **자산의 최신 버전 사용**을 선택하지 않으면 가장 최근 초안 버전이 아닌 가장 최근 *게시된* 버전이 사용됩니다.
- **파일 미리보기**: 희미한 비활성 페이지 버전 위에 선택한 파일을 겹쳐 미리보고 필요에 따라 다운로드합니다.
- **콘텐츠 항목**: 연관된 자산 저장소에서 콘텐츠 항목을 선택하고, 표시할 세부정보 페이지를 선택하고, 페이지가 동일한 창 또는 새 창에서 열리도록 지정합니다.
- **전자메일**: 적합한 전자메일 주소와 제목(선택사항)을 지정합니다. 결과 메시지가 열리고 기본 전자 메일 클라이언트를 통해 전송됩니다.
- **트리거 작업**: 단추를 누를 때 수행할 페이지 작업을 하나 이상 선택합니다. 페이지 작업은 다음과 같습니다.
  - **사이트 페이지로 이동**: 현재 사이트의 페이지를 선택합니다.
  - **외부 URL로 이동**: 외부 페이지나 사이트의 전체 URL을 지정합니다.
  - **경보 표시**: 지정된 메시지를 창에 표시합니다.



- **구성요소 표시 또는 숨기기:** 현재 페이지에서 표시하거나 숨기거나 토글할 구성요소를 선택합니다.
  - **슬롯 표시 또는 숨기기:** 현재 페이지에서 표시하거나 숨기거나 토글할 슬롯을 선택합니다.
  - **맵:** 적합한 주소 또는 좌표를 입력하고 데스크톱 및 모바일 브라우저에서 맵을 열 위치를 선택합니다.
  - **전화 통화:** 적합한 전화 번호를 입력합니다.
- 트리거 및 작업 사용을 참조하십시오.

## 구조 구성요소


구조 구성요소는 페이지의 콘텐츠를 구분하도록 돕습니다.

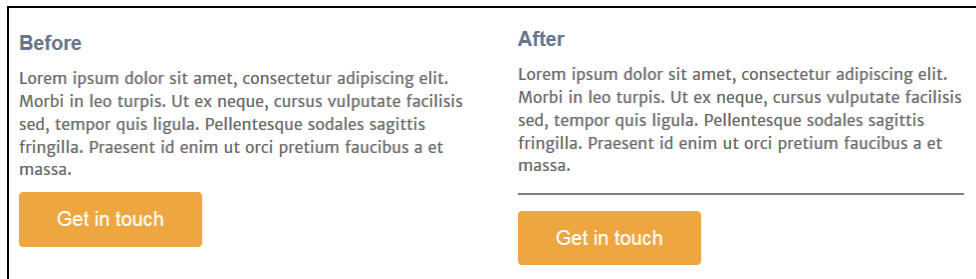
- 구분선
- 스페이서


## 구분선

구분선(수평선)을 사용하여 열 또는 페이지를 가로질러 시각적 구분을 만들 수 있습니다.

페이지에 구분선을 추가하려면 다음과 같이 하십시오.

1. 편집할 페이지로 이동하고 가 편집으로 설정되어 있는지 확인합니다.
2. 페이지에 구성요소를 추가합니다.




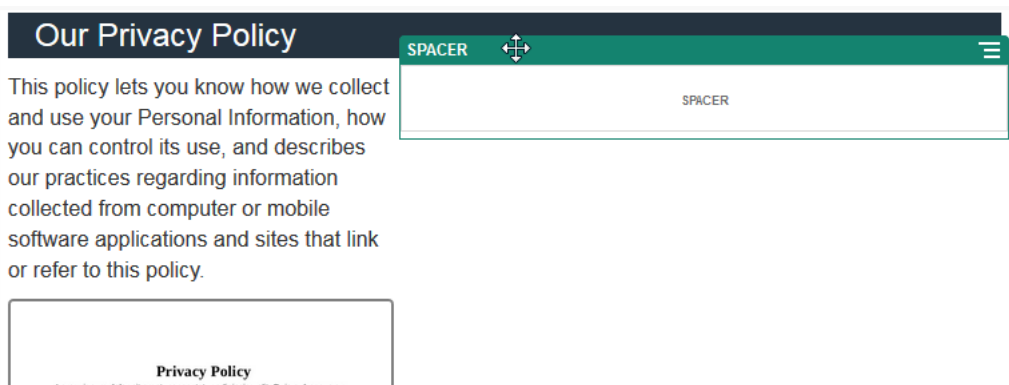
3. 구분선의 속성을 조정하려면 메뉴 아이콘 을 누르고 설정을 선택합니다.  
왼쪽 및 오른쪽 간격 옵션을 조정하여 구분선의 가로 배치와 길이를 지정할 수 있습니다. 선 색상, 두께 및 기타 설정은 구성요소 스타일로 정의됩니다. 현재 테마에 미리 정의된 스타일 중에서 선택하거나 사용자정의를 눌러 고유 값을 지정할 수 있습니다.


## 스페이서

페이지의 “공백”을 관리하면 페이지를 시각적으로 정돈하고 가독성을 높일 수 있습니다. 스페이서 구성요소를 사용하여 스타일이나 다른 구성요소에 정의된 간격을 무효화하지 않고도 공백을 추가할 수 있습니다.

페이지의 구성요소 사이에 수직 빈 공간을 추가하려면 다음과 같이 하십시오.

1. 편집할 페이지로 이동하고 가 편집으로 설정되어 있는지 확인합니다.
2. 페이지에 구성요소를 추가합니다.



- 스페이스의 높이를 조정하려면 메뉴 아이콘  을 누르고 **설정**을 선택합니다.

## 미디어 구성요소


미디어 구성요소는 이미지 및 비디오를 페이지에 가져옵니다.

- 이미지
- 갤러리
- 갤러리 그리드
- YouTube 비디오
- 비디오


## 이미지

이미지는 열람자의 관심을 유도해서 열람자가 페이지의 다른 콘텐츠 영역을 탐색하도록 이끌 수 있습니다.

페이지에 이미지를 추가하려면 다음과 같이 하십시오.

- 편집할 페이지로 이동하고  가 **편집**으로 설정되어 있는지 확인합니다.
- 페이지에 구성요소를 추가합니다. 사용할 이미지를 선택할 때까지 이미지 구성요소는 위치 표시자 이미지를 보여줍니다.

















- 이미지를 선택하려면 메뉴 아이콘  을 누르고, **설정**을 선택하고, 이미지 필드 옆에 있는 **선택**을 누릅니다.

4. 사이트 저장소의 이미지, 사용자에게 공유된 문서 폴더의 이미지를 선택하거나, 문서 폴더에 이미지를 업로드합니다.

 **주:**

모든 사용 가능한 파일이 창에 표시됩니다. 컨텍스트에 적합한 파일 유형을 선택해야 합니다. 예를 들어, 이미지 파일을 선택하는 경우 적합한 이미지 형식의 파일(GIF, JPG, JPEG, PNG, SVG)을 선택해야 합니다.

- a. 사용할 이미지를 찾아 선택합니다.
- 디지털 자산이 표시되지 않으면  을 누르고 모음 필터를 모두로 변경합니다.
- b. 문서 폴더에서 이미지를 선택한 경우, 이를 사이트에 복사하는 대신 파일에 링크할 수 있습니다. 파일에 링크하려면 **사이트에 파일을 복사하는 대신 원본 파일의 참조 사용**을 선택합니다. 이 옵션을 선택하지 않으면 파일의 복사본이 사이트에 저장되고, 사이트에서 참조됩니다. 원본 파일에 링크하면 콘텐츠가 복제되지 않습니다. 링크를 통해 사이트 방문자는 파일에 대한 권한이 변경되거나 보기 권한이 제한되는 경우에도 콘텐츠를 볼 수 있습니다.
- c. **확인**을 누릅니다.
- d. 디지털 자산을 선택한 경우, 특정 표현을 선택할 수 있습니다. 표현을 선택하지 않으면 원래 크기가 사용됩니다. 사이트가 게시될 때 자산의 최신 버전이 게시되도록 하려면 **자산의 최신 버전 사용**을 선택합니다.
5. 이미지를 편집하려면  을 누르고 다음 작업을 통해 이미지를 편집합니다.
- 이미지를 자르려면  **자르기**를 누릅니다. 자르기 도구 모음에서 미리 정의된 이미지 비율 중 하나를 선택하거나 이미지에서 자르기 핸들을 원하는 대로 끌어갑니다. 비율이 만족스러우면 자르기 도구 모음에서 **자르기**를 누릅니다.
  - 이미지를 회전하거나 뒤집으려면  **회전**을 누릅니다. 회전 도구모음에서 사용자정의 회전 각도를 입력하거나 이미지를 왼쪽 또는 오른쪽으로 회전하는 단추를 사용하거나 이미지를 좌우 또는 상하로 뒤집을지 여부를 선택합니다.
  - 이미지에 워터마크를 추가하려면  **워터마크**를 누릅니다. 워터마크 틀을 사용하여 필요에 따라 텍스트 크기, 스타일, 색상 및 불투명도를 변경하여 이미지에 텍스트를 추가합니다.
  - 이미지 형식을 변경하려면  **옵션**을 누른 후 **형식** 드롭다운 목록에서 새 형식을 선택합니다.
  - 배경 색상을 변경하려면  **옵션**을 누른 후 **배경 색상** 드롭다운 메뉴에서 옵션을 선택합니다.
  - .jpg 또는 .webp(Google Chrome 브라우저에서 사용 가능)를 편집 중인 경우 보다 작은 파일 크기가 생성되도록 이미지 품질을 변경할 수 있습니다.  **옵션**을 누른 후 **품질** 상자에 새 백분율을 입력합니다.
  - 변경사항을 실행 취소하거나 재실행하려면  또는  을 누릅니다. 모든 변경사항을 제거하려면 **재설정**을 누릅니다.

- 이미지 확대를 변경하려면 확대/축소 컨트롤을 사용합니다 (    ).
6. 설정 메뉴 의 일반 사항 패널을 사용하여 표제를 추가하거나 간격, 맞춤, 스타일 및 기타 옵션을 수정합니다.
  7. 이미지에 링크나 다른 동작을 연관시킬 수 있습니다.
    - a. 설정 패널에서 링크를 누릅니다.
    - b. 다음 옵션 중 하나를 선택합니다.
      - **링크 없음:** 사용자가 이미지를 누를 때 아무 동작도 수행하지 않습니다.
      - **웹 페이지:** 외부 페이지나 사이트의 전체 URL을 지정하고 링크를 열 위치를 선택합니다.
      - **사이트 페이지:** 페이지 선택기를 사용하여 현재 사이트에서 페이지를 선택하고 링크를 열 위치를 선택합니다. `key1=value1&key2=value2` 형식으로 추가 URL 매개변수를 지정할 수 있습니다. 비어 있는 값이 지원됩니다(예: `key1=&key2=value2`). 또한 URL 앵커를 지정할 수 있지만 앵커 링크를 확인할 대상 사이트 페이지의 위치에 특수한 앵커 섹션 레이아웃을 추가하고, 트리거 작업을 정의할 때 사용한 레이아웃 설정에서 동일한 앵커 이름을 지정해야 합니다.

 주:

URL 앵커를 사용하는 데 필요한 앵커 섹션 레이아웃은 OCE Toolkit에서 배포됩니다. 툴킷을 얻는 방법에 대한 자세한 내용은 OCE Toolkit으로 개발을 참조하십시오.


- **파일 다운로드:** 저장소에서 선택한 파일을 다운로드합니다. 파일을 선택합니다.  
디지털 자산을 선택한 경우, 특정 표현을 선택할 수 있습니다. 표현을 선택하지 않으면 원래 크기가 사용됩니다. 사이트가 게시될 때 자산의 최신 버전이 게시되도록 하려면 **자산의 최신 버전 사용**을 선택합니다. **자산의 최신 버전 사용**을 선택하지 않으면 가장 최근 초안 버전이 아닌 가장 최근 *게시된* 버전이 사용됩니다.
- **파일 미리보기:** 선택된 파일이 페이지에 중첩되어 나타납니다.
- **콘텐츠 항목:** 연관된 자산 저장소에서 콘텐츠 항목을 선택하고, 표시할 세부정보 페이지를 선택하고, 페이지가 동일한 창 또는 새 창에서 열리도록 지정합니다.
- **전자메일:** 적합한 전자메일 주소와 제목(선택사항)을 지정합니다. 결과 메시지가 열리고 기본 전자 메일 클라이언트를 통해 전송됩니다.
- **이미지 미리보기:** 선택된 이미지가 페이지에 중첩되어 나타납니다.
- **맵:** 적합한 주소 또는 좌표를 입력하고 데스크톱 및 모바일 브라우저에서 맵을 열 위치를 선택합니다.
- **전화 통화:** 적합한 전화 번호를 입력합니다.

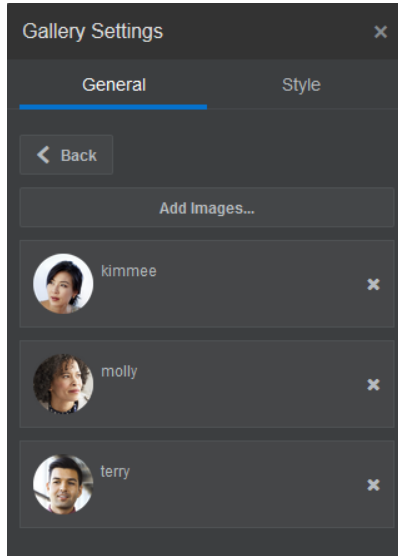
## 갤러리

이미지 갤러리를 사용하여 일련의 이미지를 표시할 수 있습니다. 이미지가 자동으로 순환하도록 하거나, 사용자가 수동으로 이미지를 진행시킬 수 있습니다.

페이지에 이미지 갤러리를 추가하려면 다음과 같이 하십시오.

1. 편집할 페이지로 이동하고 가 편집으로 설정되어 있는지 확인합니다.

2. 페이지에 구성요소를 추가합니다. 사용할 이미지를 선택할 때까지 갤러리 구성요소는 위치 표시자 이미지를 보여줍니다.
3. 하나 이상의 이미지를 갤러리에 추가하려면 메뉴 아이콘  을 누르고, **설정**을 선택하고, **일반 사항** 탭에서 **이미지**를 누릅니다.
4. **이미지 추가**를 누릅니다.




5. 하나 이상의 이미지를 선택합니다.

 **주:**

모든 사용 가능한 파일이 창에 표시됩니다. 컨텍스트에 적합한 파일 유형을 선택해야 합니다. 예를 들어, 이미지 파일을 선택하는 경우 적합한 이미지 형식의 파일(GIF, JPG, JPEG, PNG, SVG)을 선택해야 합니다.

- a. 사용할 이미지를 찾아 선택합니다.

디지털 자산이 표시되지 않으면  을 누르고 모음 필터를 **모두**로 변경합니다.

- b. 문서 폴더에서 이미지를 선택한 경우, 이를 사이트에 복사하는 대신 파일에 링크할 수 있습니다. 파일에 링크하려면 **사이트에 파일을 복사하는 대신 원본 파일의 참조 사용**을 선택합니다. 이 옵션을 선택하지 않으면 파일의 복사본이 사이트에 저장되고, 사이트에서 참조됩니다. 원본 파일에 링크하면 콘텐츠가 복제되지 않습니다. 링크를 통해 사이트 방문자는 파일에 대한 권한이 변경되거나 보기 권한이 제한되는 경우에도 콘텐츠를 볼 수 있습니다.

- c. **확인**을 누릅니다.

- d. 디지털 자산을 선택한 경우, 특정 표현을 선택할 수 있습니다. 표현을 선택하지 않으면 원래 크기가 사용됩니다. 사이트가 게시될 때 자산의 최신 버전이 게시되도록 하려면 **자산의 최신 버전 사용**을 선택합니다.

선택한 이미지가 이미지 목록에 추가됩니다. 이미지 끌어 놓기로 목록에서 재정렬합니다. 각 이미지의 기본 제목은 확장자 없는 파일 이름입니다.

6. 특정 이미지의 제목, 설명 또는 기타 옵션을 변경하려면 목록에서 이미지를 누르고 적절히 변경합니다.


또한 갤러리의 이미지에 링크나 다른 동작을 연관시킬 수 있습니다.

- a. 특정 이미지의 설정 패널에서 링크 필드를 누릅니다.
- b. 다음 옵션 중 하나를 선택합니다.
  - **링크 없음:** 사용자가 이미지를 누를 때 아무 동작도 수행하지 않습니다.
  - **웹 페이지:** 외부 페이지나 사이트의 전체 URL을 지정하고 링크를 열 위치를 선택합니다.
  - **사이트 페이지:** 페이지 선택기를 사용하여 현재 사이트에서 페이지를 선택하고 링크를 열 위치를 선택합니다. `key1=value1&key2=value2` 형식으로 추가 URL 매개변수를 지정할 수 있습니다. 비어 있는 값이 지원됩니다(예: `key1=&key2=value2`). 또한 URL 앵커를 지정할 수 있지만 앵커 링크를 확인할 대상 사이트 페이지의 위치에 특수한 앵커 섹션 레이아웃을 추가하고, 트리거 작업을 정의할 때 사용한 레이아웃 설정에서 동일한 앵커 이름을 지정해야 합니다.

 **주:**

URL 앵커를 사용하는 데 필요한 앵커 섹션 레이아웃은 OCE Toolkit에서 배포됩니다. 툴킷을 얻는 방법에 대한 자세한 내용은 OCE Toolkit으로 개발을 참조하십시오.



- **파일 다운로드:** 저장소에서 선택한 파일을 다운로드합니다. 파일을 선택합니다. 디지털 자산을 선택한 경우, 특정 표현을 선택할 수 있습니다. 표현을 선택하지 않으면 원래 크기가 사용됩니다. 사이트가 게시될 때 자산의 최신 버전이 게시되도록 하려면 **자산의 최신 버전 사용**을 선택합니다. **자산의 최신 버전 사용**을 선택하지 않으면 가장 최근 초안 버전이 아닌 가장 최근 *게시된* 버전이 사용됩니다.
  - **콘텐츠 항목:** 연관된 자산 저장소에서 콘텐츠 항목을 선택하고, 표시할 세부정보 페이지를 선택하고, 페이지가 동일한 창 또는 새 창에서 열리도록 지정합니다.
  - **전자메일:** 적합한 전자메일 주소와 제목(선택사항)을 지정합니다. 결과 메시지가 열리고 기본 전자 메일 클라이언트를 통해 전송됩니다.
  - **맵:** 적합한 주소 또는 좌표를 입력하고 데스크톱 및 모바일 브라우저에서 맵을 열 위치를 선택합니다.
  - **전화 통화:** 적합한 전화 번호를 입력합니다.
- c. **뒤로**를 눌러 이미지 설정 패널로 돌아갑니다. **뒤로**를 다시 눌러 이미지 목록으로 돌아가서 업데이트할 다른 이미지를 선택합니다.
  - d. 개별 이미지 업데이트가 완료되면 **뒤로**를 눌러 갤러리 옵션을 지정합니다.
7. 갤러리의 이미지 크기와 배율을 조정합니다.
- a. **너비**를 사용하여 슬롯 내에 갤러리의 너비(픽셀)를 지정합니다. **채우기** 이외의 맞춤 옵션을 눌러 너비를 지정합니다. 너비를 설정한 후 **채우기**를 사용하여 지정된 너비로 이미지를 늘릴 수 있습니다.
  - b. **확장/축소** 옵션을 선택하여 갤러리의 이미지 표시를 조정합니다.
    - **자르기:** 두 치수(너비 또는 높이)가 더 작으면 가용 공간에 맞게 조정하고, 더 큰 치수는 잘라서 이미지 늘이기를 막습니다.
    - **맞춤:** 전체 이미지가 이미지 왜곡 없이 가용 공간에 맞도록 각 이미지가 조정됩니다.
    - **늘이기:** 두 치수(너비 또는 높이)가 더 크면 가용 공간에 맞게 조정하고, 더 작은 치수는 늘여서 가용 공간을 채웁니다.

- **없음:** 이미지를 전체 픽셀 해상도로 사용하고 가용 공간에 맞지 않으면 균일하게 자릅니다.
8. 사용자의 갤러리 내 이동을 도와줍니다.
    - a. **탐색** 방식을 선택합니다.
      - **작은 그림:** 갤러리 아래에 차례로 갤러리의 이미지 목록을 표시합니다. 사용자가 작은 그림 이미지를 누르면 갤러리의 관련 이미지로 이동합니다.
      - **인덱서:** 갤러리의 각 이미지를 나타내는 일련의 단추 를 갤러리 아래에 표시합니다. 사용자가 단추를 누르면 갤러리의 관련 이미지로 이동합니다.
      - **없음:** 시각적 탐색을 제공하지 않습니다. 사용자가 오른쪽/왼쪽으로 쓸어넘기면 인접 이미지를 표시할 수 있습니다. 이 수동 방식은 항상 사용할 수 있습니다.
    - b. **이전/다음 표시**를 눌러 갤러리의 이전 또는 다음 이미지로 진행하는 화살표 아이콘을 각 이미지에 포함합니다.
  9. **캡션 표시** 상자를 선택하여 이미지의 캡션을 표시하거나 숨기도록 선택할 수 있습니다.
  10. 갤러리의 이미지가 자동으로 순환하도록 하려면 **자동 재생**을 선택하고 **변환 시간 및 표시 시간**을 지정합니다.
 

사용자는 설정된 모든 수동 탐색 기능을 계속 사용할 수 있습니다.

## 갤러리 그리드


갤러리 그리드를 사용하여 한 번에 여러 이미지를 행과 열로 표시할 수 있습니다. 페이지에 이미지 갤러리를 추가하려면 다음과 같이 하십시오.

1. 편집할 페이지로 이동하고 가 **편집**으로 설정되어 있는지 확인합니다.
2. 페이지에 구성요소를 추가합니다. 사용할 이미지를 선택할 때까지 갤러리 그리드 구성요소는 위치 표시자 이미지를 보여줍니다.
3. 하나 이상의 이미지를 갤러리에 추가하려면 메뉴 아이콘 을 누르고, **설정**을 선택하고, **일반 사항** 탭에서 **이미지**를 누릅니다.
4. **이미지 추가**를 누릅니다.
5. 하나 이상의 이미지를 선택합니다.

### 주:

모든 사용 가능한 파일이 창에 표시됩니다. 컨텍스트에 적합한 파일 유형을 선택해야 합니다. 예를 들어, 이미지 파일을 선택하는 경우 적합한 이미지 형식의 파일(GIF, JPG, JPEG, PNG, SVG)을 선택해야 합니다.

- a. 사용할 이미지를 찾아 선택합니다.
 

디지털 자산이 보이지 않으면 을 누르고 모음 필터를 **모두**로 변경합니다.
- b. 문서 폴더에서 이미지를 선택한 경우, 이를 사이트에 복사하는 대신 파일에 링크할 수 있습니다. 파일에 링크하려면 **사이트에 파일을 복사하는 대신 원본 파일의 참조 사용**을 선택합니다. 이 옵션을 선택하지 않으면 파일의 복사본이 사이트에

저장되고, 사이트에서 참조됩니다. 원본 파일에 링크하면 콘텐츠가 복제되지 않습니다. 링크를 통해 사이트 방문자는 파일에 대한 권한이 변경되거나 보기 권한이 제한되는 경우에도 콘텐츠를 볼 수 있습니다.

- c. **확인**을 누릅니다.
- d. 디지털 자산을 선택한 경우, 특정 표현을 선택할 수 있습니다. 표현을 선택하지 않으면 원래 크기가 사용됩니다. 사이트가 게시될 때 자산의 최신 버전이 게시되도록 하려면 **자산의 최신 버전 사용**을 선택합니다.

선택한 이미지가 이미지 목록에 추가됩니다. 이미지 끌어 놓기로 목록에서(그리드에서) 재정렬합니다. 각 이미지의 기본 제목은 확장자 없는 파일 이름입니다.

- 6. 특정 이미지의 제목, 설명 또는 기타 옵션을 변경하려면 목록에서 이미지를 누르고 적절히 변경합니다.

또한 갤러리의 이미지에 링크나 다른 동작을 연관시킬 수 있습니다.

- a. 특정 이미지의 설정 패널에서 링크 필드를 누릅니다.
- b. 다음 옵션 중 하나를 선택합니다.
  - **링크 없음**: 사용자가 이미지를 누를 때 아무 동작도 수행하지 않습니다.
  - **웹 페이지**: 외부 페이지나 사이트의 전체 URL을 지정하고 링크를 열 위치를 선택합니다.
  - **사이트 페이지**: 페이지 선택기를 사용하여 현재 사이트에서 페이지를 선택하고 링크를 열 위치를 선택합니다. `key1=value1&key2=value2` 형식으로 추가 URL 매개변수를 지정할 수 있습니다. 비어 있는 값이 지원됩니다(예: `key1=&key2=value2`). 또한 URL 앵커를 지정할 수 있지만 앵커 링크를 확인할 대상 사이트 페이지의 위치에 특수한 앵커 섹션 레이아웃을 추가하고, 트리거 작업을 정의할 때 사용한 레이아웃 설정에서 동일한 앵커 이름을 지정해야 합니다.

 **주:**

URL 앵커를 사용하는 데 필요한 앵커 섹션 레이아웃은 OCE Toolkit에서 배포됩니다. 툴킷을 얻는 방법에 대한 자세한 내용은 OCE Toolkit으로 개발을 참조하십시오.

- **파일 다운로드**: 저장소에서 선택한 파일을 다운로드합니다. 파일을 선택합니다. 디지털 자산을 선택한 경우, 특정 표현을 선택할 수 있습니다. 표현을 선택하지 않으면 원래 크기가 사용됩니다. 사이트가 게시될 때 자산의 최신 버전이 게시되도록 하려면 **자산의 최신 버전 사용**을 선택합니다. **자산의 최신 버전 사용**을 선택하지 않으면 가장 최근 초안 버전이 아닌 가장 최근 **게시된** 버전이 사용됩니다.
- **콘텐츠 항목**: 연관된 자산 저장소에서 콘텐츠 항목을 선택하고, 표시할 세부정보 페이지를 선택하고, 페이지가 동일한 창 또는 새 창에서 열리도록 지정합니다.
- **전자메일**: 적합한 전자메일 주소와 제목(선택사항)을 지정합니다. 결과 메시지가 열리고 기본 전자 메일 클라이언트를 통해 전송됩니다.
- **이미지 미리보기**: 선택된 이미지가 페이지에 중첩되어 나타납니다.
- **맵**: 적합한 주소 또는 좌표를 입력하고 데스크톱 및 모바일 브라우저에서 맵을 열 위치를 선택합니다.
- **전화 통화**: 적합한 전화 번호를 입력합니다.
- c. **뒤로**를 눌러 이미지 설정 패널로 돌아갑니다. **뒤로**를 다시 눌러 이미지 목록으로 돌아가서 업데이트할 다른 이미지를 선택합니다.



- d. 개별 이미지 업데이트가 완료되면 **뒤로**를 눌러 갤러리 옵션을 지정합니다.
- 7. 레이아웃 옵션을 사용하여 그리드에 이미지를 배열합니다.
  - 석조
  - 열
  - 사용자정의
 각 옵션은 다음 단계에서 설명합니다.
- 8. 석조 레이아웃을 선택하여 가용 공간 내에 행으로 이미지를 자동으로 배열합니다. 결과 행은 높이가 균일하지만 정의된 열은 없습니다.



- a. 높이를 지정하여 모든 이미지를 지정된 높이(픽셀)에 비례하여 조정합니다.
- b. 이미지 간격을 지정하여 행의 이미지 사이 간격을 늘리거나 줄입니다.
- 9. 열 레이아웃을 선택하여 이미지를 행과 열로 배열합니다.
  - a. 확장/축소 옵션을 선택하여 그리드의 이미지 표시를 조정합니다.
    - 자르기: 두 치수(너비 또는 높이)가 더 작으면 가용 공간에 맞게 조정하고, 더 큰 치수는 잘라서 이미지 늘이기를 막습니다.
    - 맞춤: 전체 이미지가 이미지 왜곡 없이 가용 공간에 맞도록 각 이미지가 조정됩니다.

예를 들어, 다음 그리드는 4개 열을 사용하고 이에 맞게 6개 이미지를 조정합니다.





은 이미지가 잘린 동일한 그리드입니다.


- b. 가로 세로 비율을 지정하여 그리드의 셀 모양을 결정합니다.
    - 사각형: 가로 세로 비율 1:1
    - 가로: 가로 세로 비율 16:9
    - 세로: 가로 세로 비율 9:16
    - 사용자정의: 가로 세로 비율에 고유의 숫자 값을 지정합니다.
  - c. 열 개수를 지정합니다.  
동일 너비의 열을 생성하도록 그리드가 자동으로 조정됩니다.
  - d. 이미지 간격을 지정하여 행과 열의 이미지 사이 간격을 늘리거나 줄입니다.
10. 사용자정의 레이아웃을 선택하여 사용자가 지정한 이미지 크기와 너비에 준하여 이미지를 행과 열로 배열합니다.
- a. 확장/축소 옵션을 선택하여 그리드의 이미지 표시를 조정합니다.
    - 자르기: 두 치수(너비 또는 높이)가 더 작으면 가용 공간에 맞게 조정하고, 더 큰 치수는 잘라서 이미지 늘이기를 막습니다.
    - 맞춤: 전체 이미지가 이미지 왜곡 없이 가용 공간에 맞도록 각 이미지가 조정됩니다.
  - b. 이미지 높이 및 이미지 너비를 지정하여 그리드의 셀 모양을 결정합니다.  
사용자가 지정한 치수의 셀을 생성하도록 그리드가 자동으로 조정됩니다.
  - c. 이미지 간격을 지정하여 행과 열의 이미지 사이 간격을 늘리거나 줄입니다.
11. 맞춤, 너비, 간격 옵션을 지정하여 슬롯 내에 그리드를 배치합니다.  
너비를 사용하여 슬롯 내에 갤러리의 너비(픽셀)를 지정합니다. 채우기 이외의 맞춤 옵션을 눌러 너비를 지정합니다. 너비를 설정한 후 채우기를 사용하여 지정된 너비로 이미지를 늘릴 수 있습니다.


## YouTube 비디오

페이지에 동작과 시각 효과가 추가되도록 스트리밍 YouTube 비디오를 포함합니다.

YouTube가 호스트하는 비디오가 아닌 다른 비디오를 사용하려면 **비디오**를 참조하십시오.

페이지에 YouTube 비디오를 추가하려면 다음과 같이 하십시오.

1. 편집할 페이지로 이동하고 가 **편집**으로 설정되어 있는지 확인합니다.

2. 페이지에 구성요소를 추가합니다. 사용할 비디오를 선택할 때까지 구성요소에 위치 표시자 이미지가 표시됩니다.
3. 사용할 YouTube 비디오를 지정하고 표시 속성을 조정하려면 메뉴 아이콘  을 누르고 **설정**을 선택합니다.
4. 사용할 비디오를 찾은 다음 URL을 복사하여 **YouTube URL** 필드에 붙여 넣습니다.
5. 표시 옵션을 지정합니다.
  - **컨트롤 표시**: 사용자가 비디오 재생을 수동으로 제어할 수 있도록 화면 및 장치 옵션을 사용으로 설정합니다.
  - **정보 표시**: 비디오 왼쪽 위에 비디오 설명을 일시적으로 포함합니다.
  - **자동 재생**: 비디오 재생을 자동으로 시작합니다.
  - **루프**: 비디오가 완료된 후 자동으로 반복합니다.
6. 비디오에 대해 정의된 **가로 세로 비율**을 사용하거나(**자동**) 다른 가로 세로 비율을 선택하여 비디오 표시 모양을 결정합니다.
7. 맞춤 또는 간격 옵션을 지정하여 비디오를 배치합니다.


## 비디오

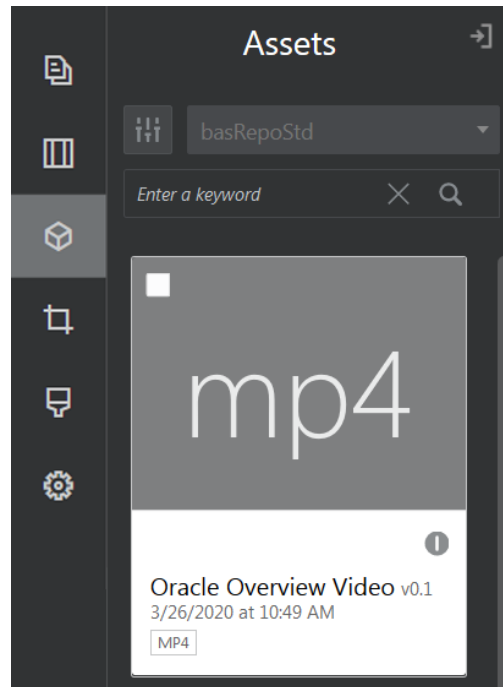
페이지에 동작과 시각 효과가 추가되도록 Oracle Content Management에서 비디오를 포함합니다.

Oracle Content Management는 비디오를 페이지에 추가하는 여러 옵션을 제공합니다. 사이트에 호스트되는 **YouTube 비디오**를 포함할 수도 있고, Oracle Content Management에서 저장, 관리 및 전달되는 비디오를 추가하여 자동 트랜스코딩 및 최적화된 스트리밍 Video Plus 혜택을 활용할 수도 있습니다. 다양한 기능의 여러 장치에 사이트가 제공될 때 유용합니다. 시스템 관리자가 Video Plus를 사용으로 설정하지 않은 경우 표준 비디오가 옵션입니다.

YouTube가 호스트하는 비디오를 사용하려면 **YouTube 비디오**를 참조하십시오.

Oracle Content Management 문서 또는 자산 저장소에 저장된 비디오를 사용하려면 다음과 같이 하십시오.

1. 편집할 페이지로 이동하고 편집 스위치()가 **편집**으로 설정되어 있는지 확인합니다.
  - a. 디지털 자산으로 저장된 비디오를 사용하려면 **자산** 패널을 열고 여기서 사용할 비디오를 검색한 다음, 선택 후 페이지에 끌어옵니다.



- b. 또는 구성요소 목록에서 비디오 구성요소를 페이지에 추가합니다. 사용할 비디오를 선택할 때까지 구성요소에 위치 표시자 이미지가 표시됩니다.
2. 자산 패널에서 자산을 끌어오거나 구성요소 목록에서 비디오 구성요소를 끌어왔으면 메뉴 아이콘(☰)을 사용하여 비디오 표시 속성을 조정하거나 비디오 구성요소의 비디오로 이동합니다. 메뉴를 누르고 **설정**을 선택합니다.
3. 비디오 구성요소의 비디오를 선택하려면 **선택**을 눌러 **문서 및 디지털 자산** 탭을 열고 Oracle Content Management에서 추가할 비디오로 이동합니다. 항목을 선택하고 **확인**을 누릅니다.
  - a. Video Plus가 사용으로 설정된 경우 비디오로 이동할 때 다른 저장소, 검색 채널, 모음 및 키워드를 선택하고 자산을 필터링하면 검색 범위를 좁힐 수 있습니다.
  - b. 표준 비디오를 사용하는 경우 MP4 유형의 비디오 파일을 선택합니다. MP4 비디오 형식은 지원되는 모든 브라우저에 공통됩니다. 또한 일부 표준 비디오에는 미리 정의된 미리보기 이미지가 있습니다. 그러면 미리보기 이미지로 빈 화면을 표시하지 않습니다. 빈 화면 대신 그래픽을 사용하려면 비디오 설정 대화상자에서 **선택**을 누르고 Oracle Content Management에서 미리보기 이미지를 선택합니다.
4. 자산 패널에서 비디오를 끌어오거나 설정 대화상자에서 비디오를 선택하여 항목을 선택했으면 설정 대화상자를 사용하여 표시 옵션을 지정합니다.
  - **컨트롤 표시**: 사용자가 비디오 재생을 수동으로 제어할 수 있도록 화면 및 장치 옵션을 사용으로 설정합니다.
  - **자동 재생**: 비디오 재생을 자동으로 시작합니다.
  - **루프**: 비디오가 완료된 후 자동으로 반복합니다.
  - **음소거됨**: 페이지를 로드할 때 자동으로 비디오를 음소거합니다.
5. 너비, 맞춤 또는 간격 옵션을 지정하여 비디오를 배치합니다.
6. 스타일을 눌러 비디오 테두리에 대한 사용자정의 스타일을 선택하거나 정의합니다.

## 문서 구성요소

파일 및 폴더를 액세스하고 표시할 수 있는 구성요소를 살펴보겠습니다.



- 문서
- 폴더 목록
- 파일 목록
- 문서 관리자
- 프로젝트 라이브러리

## 문서

페이지에서 직접 다중 페이지 문서와 슬라이드 프레젠테이션을 볼 수 있습니다.



페이지에 문서를 추가하려면 다음과 같이 하십시오.


1. 편집할 페이지로 이동하고 가 **편집**으로 설정되어 있는지 확인합니다.
2. 페이지에 구성요소를 추가합니다. 표시할 문서를 지정할 때까지 문서 구성요소는 위치 표시자 이미지를 보여줍니다.
3. 문서를 선택하려면 메뉴 아이콘 을 누르고, **설정**을 선택하고, 문서 필드 옆에 있는 **선택**을 누릅니다.

사이트 저장소나 액세스 권한이 있는 다른 저장소에 문서를 저장해야 합니다. 또한 나와 공유된 문서를 사용하거나 로컬 또는 네트워크 파일 위치에서 업로드한 문서를 사용할 수 있습니다.

4. 문서를 선택하고 **확인**을 누릅니다.

 주:

모든 사용 가능한 파일이 창에 표시됩니다. 컨텍스트에 적합한 파일 유형을 선택해야 합니다. 예를 들어, 문서 파일을 선택하는 경우 적합한 문서 형식(TXT 또는 DOC)의 파일을 선택해야 합니다.



5. [설정] 패널에서 표제를 추가하거나 간격, 맞춤, 스타일 및 기타 표시 옵션을 수정할 수 있습니다.
6. 사용자의 문서 내 이동을 도와줍니다.
  - a. **페이지 번호 표시**를 눌러 각 페이지 아래에 페이지 번호를 표시합니다.
  - b. **탐색 방식**을 선택합니다.
    - **작은 그림**: 문서 아래에 차례로 문서의 페이지 목록을 표시합니다. 사용자가 작은 그림 이미지를 누르면 연관된 페이지로 이동합니다.
    - **인덱서**: 문서의 각 페이지를 나타내는 일련의 단추  을 문서 아래에 표시합니다. 사용자가 단추를 누르면 연관된 페이지로 이동합니다.
    - **없음**: 시각적 탐색을 제공하지 않습니다. 사용자가 오른쪽/왼쪽으로 쓸어넘기면 인접 페이지를 표시할 수 있습니다. 이 수동 방식은 항상 사용할 수 있습니다.
  - c. **이전/다음 표시**를 눌러 문서의 이전 또는 다음 페이지로 진행하는 화살표 아이콘을 각 페이지에 포함합니다.

## 폴더 목록

폴더 목록을 사용하여 Oracle Content Management 계정에서 지정된 폴더 내에 폴더를 나열할 수 있습니다.

이 구성요소를 페이지에서 하나 이상의 파일 목록 또는 문서 관리자 구성요소와 함께 사용할 경우 해당 구성요소는 폴더 목록에서 선택된 폴더의 내용을 자동으로 표시할 수 있습니다.

페이지에 폴더 목록 구성요소를 추가하려면 다음과 같이 하십시오.

1. 편집할 페이지로 이동하고  가 **편집**으로 설정되어 있는지 확인합니다.
2. 페이지에 구성요소를 추가합니다.
3. 구성요소 및 구성요소 모양을 편집하려면 메뉴 아이콘  을 누르고 **설정**을 선택합니다.
4. **사용자정의 설정**을 선택하여 표시되는 콘텐츠에 대한 기본 세부정보를 설정합니다.
5. **폴더 선택** 옆에 있는 **선택**을 눌러 표시에 사용할 폴더를 변경합니다. 문서 저장소 홈 페이지 이외의 폴더를 선택해야 합니다. 완료되면 **뒤로**를 누릅니다.

 주:

폴더 목록은 모든 사용자에게 다운로드 액세스 권한을 부여합니다. 사용자는 롤에 관계없이 파일을 보고 다운로드할 수 있습니다. 사이트 방문자의 권한이 구성요소에 대해 지정된 권한보다 높은 경우 개별 권한이 구성요소에 대해 설정된 권한을 무효화합니다.

6. 다음 중에서 선택하여 표시되는 콘텐츠에 대한 추가 기본값을 설정합니다.



- **기본 선택 항목 선택:** Oracle Documents 폴더를 선택한 경우 폴더(있는 경우)를 선택하여 목록에 선택한 대로 표시합니다.
  - **폴더 이름 머리글 표시:** 포함된 구성요소의 머리글에 폴더 이름을 표시하려면 선택합니다.
  - **폴더 정렬:** 처음에 항목이 표시되는 방식(이름에 따라 사전순으로 또는 최종 항목 업데이트 날짜순으로)을 선택합니다.
  - **하위 폴더 표시:** 슬라이드 막대를 사용하여 표시되는 하위 폴더 수를 제한합니다.
7. [일반 사항] 탭에서 간격, 맞춤 및 기타 표시 옵션을 수정할 수 있습니다.
  8. [스타일] 탭에서 미리 정의된 스타일이나 사용자정의 옵션을 사용하여 구성요소가 포함된 프레임의 서식을 지정합니다.

## 파일 목록

파일 목록을 사용하여 Oracle Content Management 계정에서 지정된 폴더의 파일 뷰를 제공할 수 있습니다.

이 구성요소를 페이지에서 하나 이상의 폴더 목록 구성요소와 함께 사용할 경우 파일 목록 구성요소는 폴더 목록에서 선택된 폴더의 내용을 자동으로 표시할 수 있습니다. 구성요소를 구성하여 사용자가 목록의 파일을 누르면 작업을 하나 이상 수행할 수도 있습니다. 예를 들어 별도의 브라우저 창이나 "Lightbox" 중첩으로 또는 페이지의 문서 관리자 구성요소에서 선택한 파일을 미리볼 수 있습니다.

페이지에 파일 목록 구성요소를 추가하려면 다음과 같이 하십시오.

1. 편집할 페이지로 이동하고 가 **편집**으로 설정되어 있는지 확인합니다.
2. 페이지에 구성요소 그룹을 추가합니다.
3. 구성요소 및 구성요소 모양을 편집하려면 메뉴 아이콘 을 누르고 **설정**을 선택합니다.
4. **사용자정의 설정**을 선택하여 표시되는 콘텐츠에 대한 기본 세부정보를 설정합니다.
5. **폴더 선택** 옆에 있는 **선택**을 눌러 표시에 사용할 폴더를 변경합니다. 문서 홈 페이지 이외의 폴더를 선택해야 합니다. 완료되면 **뒤로**를 누릅니다.

### 주:

파일 목록은 모든 사용자에게 다운로드 액세스 권한을 부여합니다. 사용자는 룰에 관계없이 파일을 보고 다운로드할 수 있습니다. 사이트 방문자의 권한이 구성요소에 대해 지정된 권한보다 높은 경우 개별 권한이 구성요소에 대해 설정된 권한을 무효화합니다.

6. 다음 중에서 선택하여 표시되는 콘텐츠에 대한 추가 기본값을 설정합니다.
  - **표시 옵션:** 나열된 파일과 함께 표시될 세부정보를 선택합니다.
    - **폴더 이름 머리글:** 머리글에 폴더 이름을 표시합니다.
    - **파일 설명:** 제공된 경우 파일 설명을 표시합니다.
    - **파일 구분 기호:** 행으로 각 파일을 구분합니다.

- **다운로드 아이콘:** 사용자가 원하는 경우 파일을 다운로드할 수 있도록 다운로드 아이콘을 포함합니다.
  - **최종 업데이트:** 파일이 최종 업데이트된 날짜를 표시합니다.
  - **파일 크기:** 파일 크기를 표시합니다.
  - **이미지:** 파일 콘텐츠의 작은 그림 이미지를 표시합니다.
  - **트리거 및 작업:** 폴더 목록 구성요소에서 선택된 폴더의 내용으로 파일 목록을 자동으로 새로고침하려는 경우 선택합니다. 파일을 선택할 때 [파일이 선택됨] 트리거를 사용하여 설정하도록 선택할 수도 있습니다. [링크] 탭에서 **파일이 선택됨** 트리거와 작업을 연관시킵니다.
  - **파일 정렬:** 처음에 항목이 표시되는 방식(이름에 따라 사전순으로 또는 최종 항목 업데이트 날짜순으로)을 선택합니다.
  - **파일 표시:** 모든 파일을 표시하거나 슬라이드 막대를 사용하여 파일 수를 제한하려는 경우 선택합니다.
7. [일반 사항] 탭에서 간격, 맞춤 및 기타 표시 옵션을 수정할 수 있습니다.
  8. [스타일] 탭에서 미리 정의된 스타일이나 사용자정의 옵션을 사용하여 구성요소가 포함된 프레임의 서식을 지정합니다.
  9. [링크] 탭에서 **파일이 선택됨** 트리거와 작업을 연관시킵니다. 예를 들어 파일 목록 구성요소를 구성하여 선택한 파일을 Lightbox 중첩으로 미리봅니다.
    - a. [파일 목록 설정] 창에서 **파일이 선택될 때 트리거 활성화**를 선택했는지 확인합니다.
    - b. 파일 목록 설정의 [링크] 탭에서 **파일이 선택됨** 트리거를 누릅니다.
    - c. **Lightbox 미리보기** 페이지 작업을 누르고 작업 목록으로 끕니다.
    - d. [파일 ID] 또는 [URL] 필드에서 **파일 링크**를 선택합니다.
 



**파일 링크**는 다운로드 권한이 있는 참조 링크를 사용하기 때문에 모든 사이트 방문자가 파일을 미리보고 필요에 따라 다운로드할 수 있습니다. **파일 ID**를 사용하는 경우 멤버 링크가 사용됩니다. 멤버 링크를 사용하면 등록된 사용자만 선택한 파일을 미리보고 필요에 따라 다운로드할 수 있습니다.

사용자가 파일 목록의 파일을 누르면 희미한 비활성화 페이지 버전 위에 파일 미리보기가 열립니다.

## 문서 관리자

문서 관리자를 사용하여 Oracle Content Management에서 홈 페이지 또는 파일의 뷰를 제공할 수 있습니다.

페이지에 문서 관리자 구성요소를 추가하려면 다음과 같이 하십시오.

1. 편집할 페이지로 이동하고 가 **편집**으로 설정되어 있는지 확인합니다.
2. 페이지에 구성요소를 추가합니다.
3. 구성요소 및 구성요소 모양을 편집하려면 메뉴 아이콘 을 누르고 **설정**을 선택합니다.
4. **사용자정의 설정**을 선택하여 표시되는 콘텐츠에 대한 기본 세부정보를 설정합니다.
5. **폴더 선택** 옆에 있는 **선택**을 눌러 표시에 사용할 폴더를 변경합니다.

폴더를 선택하려면 폴더 이름 옆에 있는 상자를 선택합니다. 폴더를 열려면 폴더 이름을 누릅니다. 해당 폴더로 돌아가려면 경로에 있는 폴더 이름을 누르고, 홈 폴더로 돌아가려면 **홈으로 이동**을 누릅니다. 완료되면 **뒤로**를 누릅니다.



6. **폴더 액세스** 선택을 누르고 방문자에게 부여할 액세스 롤을 선택합니다.

방문자는 주어진 롤과 **찾아보기 옵션**(다음 단계에서 사용으로 설정)에 따라 폴더 콘텐츠를 확인 및 사용할 수 있습니다.

- **멤버 액세스:** 방문자는 폴더 멤버에 제공되는 기능(예: 대화, 주석 또는 항목의 사용자정의 속성 보기)을 사용할 수 있습니다.
- **열람자:** 열람자는 파일 및 폴더를 볼 수 있지만 변경할 수 없습니다.
- **다운로더:** 다운로더는 파일을 다운로드하여 자신의 컴퓨터에 저장할 수도 있습니다.
- **제공자:** 제공자는 파일 수정, 업데이트, 새 파일 업로드 및 파일 삭제를 수행할 수도 있습니다.

폴더 액세스를 설정할 때는 다음 사항을 고려합니다.

- 사이트 작성자는 보유한 액세스 권한에서 벗어나는 폴더 액세스 권한을 부여할 수 없습니다. 예를 들어, 작성자의 폴더 액세스 권한이 다운로더인 경우 제공자에게 사이트 방문자 권한을 부여할 수 없습니다.
- 구성요소에서 폴더에 대해 설정된 권한은 방문자의 권한을 보강할 수 있습니다. 예를 들어, 방문자의 폴더 권한이 열람자(또는 권한 없음)인 경우 구성요소는 선택된 롤에 따라 이보다 높은 권한을 부여할 수 있습니다. 이와 같이 향상된 권한은 구성요소 자체에서만 적합합니다.
- 사이트 방문자의 권한이 구성요소에 대해 지정된 권한보다 높은 경우 개별 권한이 구성요소에 대해 설정된 권한을 무효화합니다.
- 폴더에 대해 부여된 권한은 해당 폴더에 중첩된 폴더 및 파일에 적용됩니다.


7. 다음 중에서 선택하여 표시되는 콘텐츠에 대한 추가 기본값을 설정합니다.

- **레이아웃:** 폴더 및 파일에 대해 초기 그리드, 목록 또는 축소 목록 레이아웃을 선택합니다. 사용자는 완성 후 포함된 구성요소를 볼 때 레이아웃을 변경할 수 있습니다.
- **색상표:** 포함된 폴더 목록에 대해 사용 가능한 색상표 중 하나를 선택합니다.
- **정렬 순서:** 처음에 항목이 표시되는 방식(이름에 따라 사전순으로 또는 최종 항목 업데이트 날짜순으로)을 선택합니다.
- **찾아보기 옵션:** 항목을 선택할 때 사용자에게 제공되는 옵션을 선택합니다. 예를 들어, 사용자의 파일 확인, 파일 다운로드, 파일 및 폴더 공유 또는 복사, 파일 삭제를 허용할지 선택할 수 있습니다. 사용자가 파일 및 폴더로 수행할 수 있는 작업을 제한하려면 이 목록에서 옵션의 선택을 해제합니다. 이전 단계에서 [멤버 액세스]를 선택한 경우 대화, 주석 또는 사용자정의 속성이 표시될 측면 창을 표시하도록 선택할 수 있습니다.
- **뷰어 옵션:** 사용자가 파일을 볼 방식을 선택합니다. 포함된 프레임 내에서 또는 다른 탭(또는 브라우저 설정에 따라 창)에서 파일을 볼 수 있습니다. 또한 작은 그림을 숨기거나 표시하고 비디오 재생 방식을 사용자가 정의할 수 있습니다.
- **확대/축소 컨트롤 표시:** 폴더의 포함된 뷰에 슬라이드 막대 또는 확대/축소 컨트롤을 표시할지 여부를 선택합니다.
- **뷰어 맞춤 모드:** 파일 표시 방식으로 페이지 맞춤, 페이지 너비 맞춤 또는 원래 크기를 선택할 수 있습니다.
- **트리거 및 작업:** 폴더 목록 구성요소에서도 사용되는 경우 파일 목록을 새로고침하려면 선택합니다.

8. [일반 사항] 탭에서 간격, 맞춤 및 기타 표시 옵션을 수정할 수 있습니다.

9. [스타일] 탭에서 미리 정의된 스타일이나 사용자정의 옵션을 사용하여 구성요소가 포함된 프레임의 서식을 지정합니다.

문서 관리자가 추가되면 사이트의 프레임에 포함된 선택한 폴더의 뷰가 열람자에게 표시됩니다. 사용자는 제공된 표시 옵션을 사용하여 폴더 및 파일의 나열 방법을 변경할 수 있습니다. 사용자가 파일 또는 폴더를 선택하는 경우 메뉴 모음 또는 마우스 오른쪽 단추 누르기 메뉴에서 제공된 옵션을 선택하고, 주어진 롤과 지정된 **찾아보기 옵션**에 따라 허용되는 작업을 수행할 수 있습니다.

대화가 항목과 연관된 경우 항목과 함께  이 표시됩니다. 이 아이콘을 누르면 주석 및 설명을 볼 수 있는 대화 창이 열립니다. 폴더와 연관되지 않은 독립 대화는 대화 구성요소를 사용하여 추가해야 합니다.



항목에 사용자정의 속성이 있을 경우 창에서 해당 속성을 볼 수 있습니다. **자세히**를 누른 다음 **사용자정의 속성**을 선택하여 속성 창을 엽니다.

## 프로젝트 라이브러리

프로젝트 라이브러리를 사용하면 원본 폴더 또는 해당 위치를 변경하지 않고 저장소의 다른 위치에서 폴더를 어셈블할 수 있습니다. 프로젝트 라이브러리 구성요소를 사용하면 각 폴더에 다른 권한 레벨을 지정하여 여러 프로젝트 팀 롤을 조정할 수도 있습니다.

예를 들어, 프로젝트 라이브러리의 폴더 하나에 방문자 롤(열람자, 다운로더 또는 제공자) 중 하나를 부여하면 방문자는 이 롤과 연관된 권한이 있는 폴더 콘텐츠를 확인하고 상호 작용할 수 있습니다. 멤버 액세스를 지정하는 경우 멤버만 프로젝트 라이브러리에 나와 있는 항목을 볼 수 있습니다. 멤버는 원본 폴더에서 지정된 권한으로 폴더 콘텐츠와 상호 작용합니다. 파일 목록 하나 이상 또는 페이지의 문서 관리자 구성요소와 함께 이 구성요소를 사용하는 경우 이러한 구성요소는 프로젝트 라이브러리에서 선택된 폴더의 콘텐츠를 자동으로 표시할 수 있습니다.

페이지에 프로젝트 라이브러리 구성요소를 추가하려면 다음과 같이 하십시오.

1. 편집할 페이지로 이동하고  가 **편집**으로 설정되어 있는지 확인합니다.
2. 페이지에 구성요소를 추가합니다.
3. 구성요소 및 구성요소 모양을 편집하려면 메뉴 아이콘  을 누르고 **설정**을 선택합니다.
4. **사용자정의 설정**을 선택하여 대화를 선택하고 표시 옵션을 선택합니다.
5. 제목을 지정합니다. 기본 제목은 프로젝트 라이브러리입니다. 아래의 **제목** 표시 옵션을 선택하거나 선택을 해제하여 제목을 표시하거나 숨기도록 선택할 수 있습니다.
6. **색상표**를 선택합니다.

### 주:

프로젝트 라이브러리를 문서 관리자 구성요소와 연관시키는 데 트리거 및 작업을 사용하는 경우 프로젝트 라이브러리에 대해 선택하는 색상표가 선택된 폴더에 적용되어 문서 관리자 구성요소에 표시됩니다. 프로젝트 라이브러리를 폴더 목록 또는 파일 목록 구성요소와 연관시키는 경우 해당 구성요소가 색상표를 지원하지 않으므로 적용되지 않습니다.

7. 폴더 목록 옆에 있는 **추가**를 눌러 기존 폴더를 추가하거나 새 폴더를 생성합니다.
  - a. 검색하여 폴더를 하나 이상 선택하거나 **생성**을 눌러 폴더를 생성합니다. 이 창에서 선택하는 각 폴더는 프로젝트 라이브러리에 추가됩니다.

보기 및 정렬 옵션을 비롯하여 도구 모음에 나와 있는 모든 옵션을 사용할 수 있습니다.

- b. 완료되면 뒤로를 누릅니다.
8. 목록을 구성하려면 폴더를 선택하고 **위로 이동** 또는 **아래로 이동**을 눌러 목록 순서에서 위치를 변경하거나 **제거**를 눌러 목록에서 폴더를 제거합니다.
  9. 사용자 룰을 기반으로 폴더에 대한 액세스 권한을 지정하려면 폴더 목록에서 폴더를 선택하고 **폴더 권한 선택**을 누른 다음 액세스 룰을 선택합니다.
    - **멤버 액세스:** 폴더에 대한 권한을 가진 등록된 사용자만 프로젝트 라이브러리에 나와 있는 폴더를 볼 수 있습니다. 멤버는 원본 폴더에서 지정된 권한으로 폴더 콘텐츠와 상호 작용합니다.
    - **열람자:** 열람자는 파일 및 폴더를 볼 수 있지만 변경할 수 없습니다.
    - **다운로더:** 다운로더는 파일을 다운로드하여 자신의 컴퓨터에 저장할 수도 있습니다.
    - **제공자:** 제공자는 파일 수정, 업데이트, 새 파일 업로드 및 파일 삭제를 수행할 수도 있습니다.

폴더 액세스를 설정할 때는 다음 사항을 고려합니다.

- 사이트 작성자는 보유한 액세스 권한에서 벗어나는 폴더 액세스 권한을 부여할 수 없습니다. 예를 들어, 작성자의 폴더 액세스 권한이 다운로더인 경우 제공자에게 사이트 방문자 권한을 부여할 수 없습니다.
  - 구성요소에서 폴더에 대해 설정된 권한은 방문자의 권한을 보강할 수 있습니다. 예를 들어, 방문자의 폴더 권한이 열람자(또는 권한 없음)인 경우 구성요소는 선택된 룰에 따라 이보다 높은 권한을 부여할 수 있습니다. 이와 같이 향상된 권한은 구성요소 자체에서만 적합합니다.
  - 사이트 방문자의 권한이 구성요소에 대해 지정된 권한보다 높은 경우 개별 권한이 구성요소에 대해 설정된 권한을 무효화합니다.
  - 폴더에 대해 부여된 권한은 해당 폴더에 중첩된 폴더 및 파일에 적용됩니다.
10. **멤버 액세스** 권한이 있는 폴더의 경우 **문서 관리자에서 대화 창 표시**를 눌러 폴더 또는 콘텐츠와 연관된 대화를 표시할 수 있습니다.

이 옵션을 선택하고 페이지에서 문서 관리자 구성요소를 구성하여 선택된 폴더를 표시하는 경우 사용자는 대화 아이콘을 눌러 이 폴더와 연관된 모든 대화를 표시할 수 있습니다. 이 옵션을 선택하지 않으면 대화 아이콘이 표시되지 않습니다.

11. 완료되면 창을 닫습니다.
12. [일반 사항] 탭에서 간격, 맞춤 및 기타 표시 옵션을 수정할 수 있습니다.
13. [스타일] 탭에서 미리 정의된 스타일이나 사용자정의 옵션을 사용하여 구성요소가 포함된 프레임의 서식을 지정합니다.
14. [링크] 탭을 사용하여 **폴더가 선택됨** 트리거에 작업을 연관합니다. 예를 들어, 페이지에 문서 관리자 구성요소도 추가하는 경우 문서 관리자 구성요소를 사용하여 프로젝트 폴더에서 선택된 폴더의 콘텐츠를 표시할 수 있습니다.
  - a. 프로젝트 라이브러리 설정의 [링크] 탭에서 **폴더가 선택됨** 트리거를 누릅니다.
  - b. [트리거 작업 구성] 창에서 [문서 관리자]를 누릅니다.
  - c. **문서 표시** 작업을 눌러 작업 목록으로 끕니다.
  - d. [폴더 ID] 또는 [URL] 필드에서 **선택된 폴더**를 선택합니다.

사용자가 프로젝트 라이브러리에서 특정 폴더를 누르면 이 폴더 콘텐츠가 페이지의 문서 관리자 구성요소에 표시됩니다.

## 소셜 구성요소

소셜 구성요소는 사용자가 연결 상태를 유지하고 소통할 수 있도록 돕습니다.



- [소셜 표시줄](#)
- [Facebook 좋아요 및 추천](#)
- [Twitter 공유 및 팔로우](#)
- [대화 구성요소](#)
- [대화 목록](#)



## 소셜 표시줄

Facebook, Twitter와 같은 인기있는 소셜 미디어의 아이콘과 링크를 쉽게 추가할 수 있습니다. 소셜 표시줄에는 기본적으로 일부 대상이 포함되지만, 소셜 표시줄의 항목을 추가/제거할 수 있습니다.



소셜 표시줄을 추가하고 수정하려면 다음과 같이 하십시오.

1. 편집할 페이지로 이동하고 가 **편집**으로 설정되어 있는지 확인합니다.
2. 페이지에 구성요소를 추가합니다. 소셜 표시줄에 기본적으로 포함된 아이콘이 표시됩니다.
3. 소셜 표시줄에 아이콘을 추가하거나 변경하려면 메뉴 아이콘 을 누르고, **설정**을 선택하고, **일반 사항** 탭의 상단에서 **아이콘**을 누릅니다.
 

소셜 아이콘을 제거하려면 이름 옆에 있는 x를 누릅니다. 아이콘을 추가하려면 사이트 저장소나 액세스 권한이 있는 다른 저장소에 아이콘을 저장해야 합니다. 또한 나와 공유된 이미지를 사용하거나 로컬 또는 네트워크 파일 위치에서 업로드한 이미지를 사용할 수 있습니다.
4. 로컬 또는 네트워크 위치에서 하나 이상의 아이콘을 업로드합니다.
  - a. 패널 상단에서 **아이콘 추가**를 누릅니다.
  - b. 이미지를 저장할 저장소의 위치로 이동하거나 을 눌러 현재 위치에 새 폴더를 추가합니다.
  - c. 을 누릅니다.
  - d. 이미지 파일을 찾아 선택하고 **열기**를 누릅니다.
 

저장소의 현재 위치로 이미지 파일이 업로드됩니다.
5. 저장소에서 하나 이상의 이미지를 선택하고 **확인**을 누릅니다.
  - a. 사용할 이미지를 찾아 누릅니다.
  - b. 저장소의 파일에 링크하려면 **사이트에 파일을 복사하는 대신 원본 파일의 참조 사용**을 선택합니다. 이 옵션을 선택하지 않을 경우 파일의 복사본이 사이트와 함께 저장되며 사이트에서 참조됩니다. 원본 파일에 링크하면 콘텐츠가 복제되지 않습니다. 링크를 통해 사이트 방문자는 파일에 대한 권한이 변경되거나 보기 권한이 제한되는 경우에도 콘텐츠를 볼 수 있습니다.

## c. 확인을 누릅니다.

선택한 이미지가 이미지 목록에 추가됩니다. 이미지 끌어 놓기로 목록에서 재정렬합니다. 각 이미지의 기본 제목은 확장자 없는 파일 이름입니다.

6. 특정 이미지의 대상 URL, 제목, 설명 또는 기타 옵션을 변경하려면 목록에서 이미지를 누르고 적절히 변경합니다.

개별 아이콘 업데이트가 완료되면 뒤로를 눌러 일반 옵션을 지정합니다.

7. 소셜 표시줄의 모든 아이콘에 대해 크기, 간격, 방향, 맞춤을 지정합니다.



## Facebook 좋아요 및 추천

열람자가 Facebook에서 간편하게 내 사이트에 좋아요를 표시할 수 있도록 [Facebook 좋아요] 단추를 추가할 수 있습니다.

1. 편집할 페이지로 이동하고 가 편집으로 설정되어 있는지 확인합니다.
2. 페이지에 구성요소를 추가합니다.
3. 앱 및 앱 모양을 편집하려면 메뉴 아이콘 을 누르고 설정을 선택합니다.
4. Facebook 좋아요 설정 또는 Facebook 추천 설정을 선택하여 URL을 설정하고 필요에 따라 공유 단추를 추가하면 Facebook 페이지에서 내 사이트에 대한 링크를 간편하게 게시할 수 있습니다.
5. [일반 사항] 탭에서 간격, 맞춤 및 기타 표시 옵션을 수정할 수 있습니다.
6. [스타일] 탭에서 미리 정의된 스타일이나 사용자정의 옵션을 사용하여 앱이 포함된 프레임의 서식을 지정합니다.

## Twitter 공유 및 팔로우

열람자가 Twitter 계정에서 빠르게 링크를 공유할 수 있도록 페이지에 [Twitter 공유] 단추를 추가할 수 있습니다.

1. 편집할 페이지로 이동하고 가 편집으로 설정되어 있는지 확인합니다.
2. 페이지에 구성요소를 추가합니다.
3. 앱 및 앱 모양을 편집하려면 메뉴 아이콘 을 누르고 설정을 선택합니다.
4. Twitter 팔로우 설정을 선택하여 Twitter 사용자 이름을 설정하고 사용자 이름을 표시할지 여부를 선택한 다음 팔로우 단추의 크기를 선택합니다. Twitter 공유 설정을 선택하여 다음 기본값을 설정합니다. 사용자는 단추를 사용할 때 값을 변경할 수 있습니다.
  - URL 공유: 사용자가 공유할 수 있는 사이트의 URL입니다.
  - 트윗 텍스트: 페이지에 대한 트윗 텍스트입니다.
  - 수단 @: 트윗에 사용되는 Twitter 사용자 계정입니다.
  - 추천 @: Twitter 추천에 사용되는 Twitter 사용자 계정입니다.
  - 해시 태그 #: 게시물에 사용할 해시 태그입니다.
  - 개수: 세로 또는 가로로 공유 개수를 표시합니다.
  - 큰 단추: 앱에 대해 큰 단추 또는 보다 작은 단추를 선택합니다.

5. [일반 사항] 탭에서 간격, 맞춤 및 기타 표시 옵션을 수정할 수 있습니다.
6. [스타일] 탭에서 미리 정의된 스타일이나 사용자정의 옵션을 사용하여 앱이 포함된 프레임의 서식을 지정합니다.



## 대화 구성요소

대화를 통해 사이트에서 직접 주제에 대한 토론을 진행할 수 있습니다.

### 주:

사이트에서 대화를 사용하려면 해당 사이트가 지정된 사용자로 제한되거나 Oracle Content Management 사용자 룰을 보유한 사용자로 제한되는 보안 사이트여야 합니다. [사이트 보안 변경](#)을 참조하십시오.

페이지에 대화 구성요소를 추가하려면 다음과 같이 하십시오.

1. 편집할 페이지로 이동하고 가 **편집**으로 설정되어 있는지 확인합니다.
2. 페이지에 구성요소를 추가합니다.
3. 구성요소 및 구성요소 모양을 편집하려면 메뉴 아이콘 을 누르고 **설정**을 선택합니다.
4. **사용자정의 설정**을 선택하여 대화를 선택하고 표시 색상표를 선택합니다.
5. **대화 선택** 옆에 있는 **선택**을 눌러 기존 대화를 선택하거나 새 대화를 생성합니다.
  - a. 사용 가능한 대화 목록에서 대화를 선택하거나 **생성**을 눌러 새 대화를 생성하여 이름을 지정합니다.
 

목록에는 액세스 권한이 있는 모든 대화가 포함됩니다.  
목록을 필터링할 수 있습니다.

    - **모두**: 음소거됨으로 표시된 대화를 제외한 모든 대화를 표시합니다.
    - **즐거찾기**: 즐겨찾기로 표시된 대화를 표시합니다.
    - **음소거됨**: 음소거되어 대화 목록에서 제외된 대화를 표시합니다.
    - **단함**: 단함으로 표시된 모든 대화를 표시합니다.

목록을 정렬할 수도 있습니다.

    - **최종 업데이트**: 가장 최근에 업데이트된 대화부터 오래 전에 업데이트된 대화 순으로 대화를 나열합니다.
    - **이름**: 대화를 영숫자 순서대로 오름차순으로 나열합니다.
    - **읽지 않음**: 읽지 않은 의견이 가장 많은 대화부터 읽지 않은 의견이 가장 적은 대화 순으로 읽지 않은 대화를 먼저 나열합니다.
  - b. 완료되면 **뒤로**를 누릅니다.
  - c. 파일 또는 폴더를 기반으로 하지 않는 독립 대화를 선택한 경우 대화를 보는 사람에 대한 액세스 권한을 설정할 수 있습니다.
  - d. **색상표**를 선택합니다.
  - e. 페이지의 대화 목록 구성요소와 함께 대화 구성요소를 사용하려면 **대화 목록 구성요소의 선택항목을 기반으로 대화를 자동으로 새로고침**을 선택합니다.

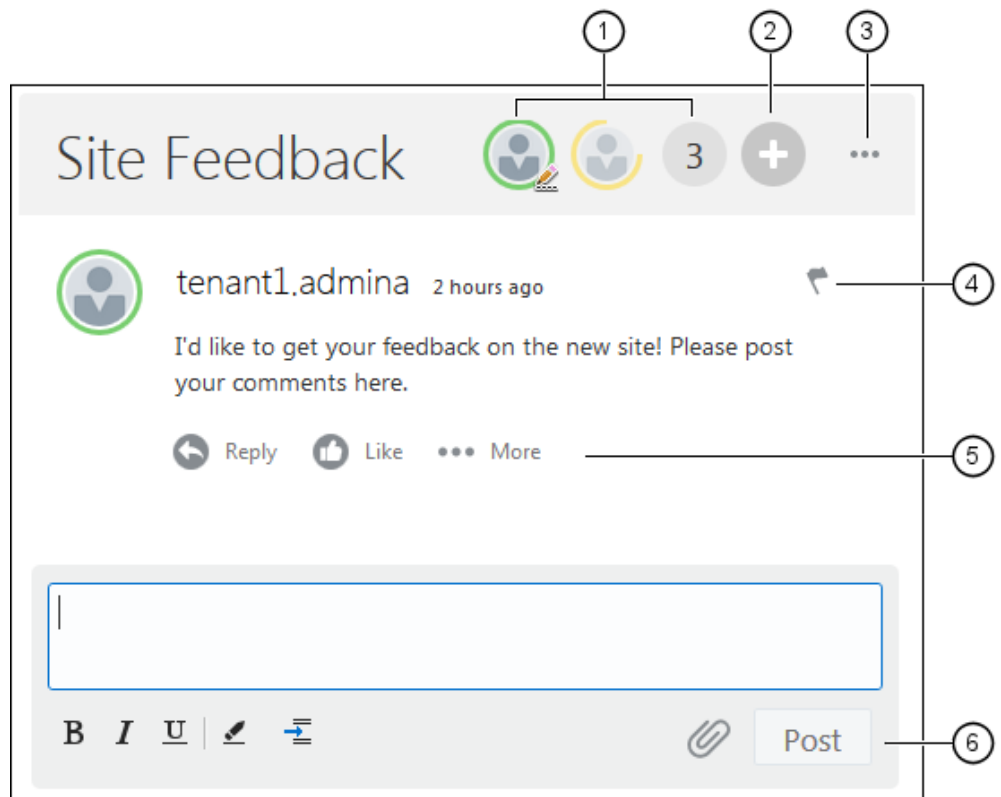
사용자가 목록에서 대화를 선택하면 대화 구성요소에 표시됩니다.

- f. 완료되면 창을 닫습니다.
- 6. [일반 사항] 탭에서 간격, 맞춤 및 기타 표시 옵션을 수정할 수 있습니다.
- 7. [스타일] 탭에서 미리 정의된 스타일이나 사용자정의 옵션을 사용하여 구성요소가 포함된 프레임의 서식을 지정합니다.

게시된 사이트에서 방문자에게는 사이트의 프레임 내에서 대화가 표시됩니다. 사용자는 대화 내에서 이동을 통해 의견을 읽고 이에 응답할 수 있습니다.

**주:**

방문자가 명시적으로 대화 멤버로 추가되지 않은 경우 의견을 읽고 댓글을 달 수 있지만 방문자에게는 아래 이미지의 1-4 항목이 표시되지 않습니다.



- **메뉴 모음 (1)**에는 개별 대화 참가자에 대한 정보가 들어 있습니다. 사용자 아이콘을 누르면 상태 정보가 제공되고 사용자 관련 작업에 대한 옵션이 표시됩니다. 숫자 아이콘은 총 참가자 수를 보여줍니다. 아이콘을 누르면 전체 목록이 표시됩니다.
- 대화에 사용자를 추가하려면 **사용자 추가 (2)**를 누릅니다.
- 대화 관련 작업에 대한 옵션 목록을 확인하려면 **기타 옵션 (3)**을 누릅니다. 옵션 목록은 사용자 룰에 따라 다릅니다. 폴더 및 파일과 연관된 대화의 경우 옵션 목록은 폴더 또는 파일에 대해 설정된 권한에 따라 달라집니다.

- 통지 플래그를 지정하여 특정 사용자에게 경보를 알려려면 **플래그 (4)**를 누릅니다. 사용자에게는 환경설정에 지정된 전자메일을 통해 경보가 전송됩니다.
- **의견 옵션 (5)**을 사용하여 대화의 특정 의견에 대한 작업(예: 의견에 댓글 달기 또는 의견에 좋아요 표시, 의견 편집 또는 삭제)을 수행할 수 있습니다.
- 의견을 추가 또는 편집하는 경우 **편집 옵션 (6)**을 사용하여 기본 서식(예: 굵은체 또는 밑줄)을 추가 또는 제거하고 첨부 파일을 추가하며 대화에 의견을 게시할 수 있습니다. 사이트 방문자는 대화 멤버가 아닌 경우 첨부 파일을 추가할 수 없습니다.

## 대화 목록



대화를 통해 사이트에서 직접 주제에 대한 토론을 진행할 수 있습니다.

이 구성요소를 페이지에서 하나 이상의 대화 목록 구성요소와 함께 사용할 경우 대화 목록 구성요소는 대화 목록에서 선택된 대화의 내용을 자동으로 표시할 수 있습니다.

### 주:

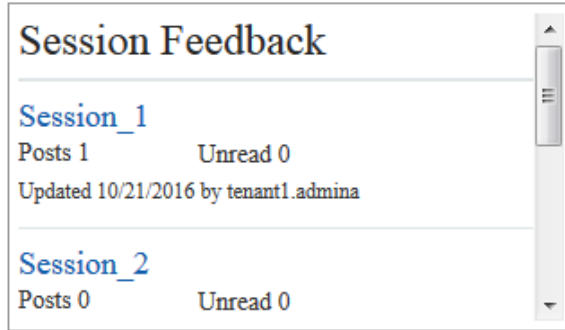
사이트에서 대화 또는 대화 목록을 사용하려면 해당 사이트가 지정된 사용자로 제한되거나 Oracle Content Management 사용자 롤을 보유한 사용자로 제한되는 보안 사이트여야 합니다. [사이트 보안 변경](#)을 참조하십시오.

페이지에 대화 목록 구성요소를 추가하려면 다음과 같이 하십시오.

1. 편집할 페이지로 이동하고 가 **편집**으로 설정되어 있는지 확인합니다.
2. 페이지에 구성요소를 추가합니다.
3. 구성요소 및 구성요소 모양을 편집하려면 메뉴 아이콘 을 누르고 **설정**을 선택합니다.
4. **사용자정의 설정**을 선택하여 대화를 선택하고 표시 옵션을 선택합니다.
5. 제목을 지정합니다. 기본 제목은 대화 목록입니다. 아래의 **제목** 표시 옵션을 선택하거나 선택을 해제하여 제목을 표시하거나 숨기도록 선택할 수 있습니다.
6. 대화 목록 옆에 있는 **추가**를 눌러 기존 대화를 추가하거나 새로 생성합니다.
  - a. 사용 가능한 대화 목록에서 대화를 하나 이상 선택하거나 **생성**을 눌러 새 대화를 생성하여 이름을 지정합니다.
  - b. 완료되면 **뒤로**를 누릅니다.
  - c. 대화 목록을 보는 사람에 대해 허용되는 권한을 선택합니다.
7. 대화 목록을 구성하려면 대화를 선택하고 **위로 이동** 또는 **아래로 이동**을 눌러 목록 순서에서 위치를 변경하거나 **제거**를 눌러 목록에서 대화를 제거합니다.
8. **색상표**를 선택하고 추가 **표시 옵션**을 선택합니다. 선택한 표시 옵션은 목록의 대화 이름 아래에 표시됩니다.
9. 완료되면 창을 닫습니다.
10. [일반 사항] 탭에서 간격, 맞춤 및 기타 표시 옵션을 수정할 수 있습니다.
11. [스타일] 탭에서 미리 정의된 스타일이나 사용자정의 옵션을 사용하여 구성요소가 포함된 프레임의 서식을 지정합니다.

다음 이미지는 모든 표시 옵션이 선택된 상태로 *Session Feedback*이라는 대화 목록을 보여줍니다.





## 프로세스 구성요소

프로세스 구성요소는 사이트 사용자가 미리 정의된 프로세스에 대한 작업을 시작하고 관리할 수 있게 해줍니다.

### 주:

프로세스 구성요소를 사용하려면 Oracle Process Cloud Service 릴리스 17.1.3 이상을 사용해야 합니다.

- [프로세스 시작 폼](#)
- [프로세스 작업 목록](#)
- [작업 세부정보 폼](#)


## 프로세스 시작 폼


프로세스 시작 폼을 사용하여 Oracle Process Cloud Service에서 정의된 프로세스를 시작할 수 있습니다.

사이트에서 프로세스 시작 폼이 작동하려면 다음이 충족되어야 합니다.

- 연관된 프로세스 및 프로세스 시작 폼이 Oracle Process Cloud Service에서 정의되어 있어야 이 구성요소로 연관된 프로세스 및 프로세스 시작 폼을 표시할 수 있습니다. *Using Processes in Oracle Integration*의 *Developing Structured Processes*를 참조하십시오.
- 프로세스 시작 폼을 사용하려면 사용자에게 시작 폼이 포함되어 있는 프로세스 스윙레인과 연관된 롤이 지정되어 있어야 합니다.
- 관리자가 Oracle Process Cloud Service와 Oracle Content Management 간의 통합을 설정해야 합니다. *Administering Oracle Content Management*의 Oracle Process Cloud Service와 통합을 참조하십시오. 두 서비스 간의 통합을 위해서는 SSO 사인온이 필요하므로 두 서비스가 동일한 ID 도메인에 있어야 합니다.

페이지에 프로세스 시작 폼 구성요소를 추가하려면 다음과 같이 하십시오.

1. 편집할 페이지로 이동하고 가 **편집**으로 설정되어 있는지 확인합니다.
2. 페이지에 구성요소를 추가합니다.

3. 구성요소 및 구성요소 모양을 편집하려면 메뉴 아이콘  을 누르고 **설정**을 선택합니다.
4. **사용자정의 설정**을 선택하여 프로세스 시작 품을 선택하고 품 기본값을 설정합니다.
  - a. 사이트가 공용 사이트인 경우 **프록시 서비스**를 선택합니다.
  - b. 시작 품에 대한 분할 영역을 선택합니다. [테스트] 분할 영역을 사용하여 프로세스가 계획대로 작동하는지를 확인하거나 [운영] 분할 영역을 사용하여 일반적인 용도로 프로세스를 배치할 수 있습니다.
  - c. 항상 기본값으로 선택된 프로세스 버전을 사용하려면 **기본 프로세스 버전 사용**을 선택합니다. 이 옵션을 선택하지 않으면 특정 버전을 선택하고, 프로세스가 업데이트되면 이 설정에서 선택사항을 업데이트해야 합니다.
  - d. 프로세스를 선택합니다.
  - e. 시작 품을 선택합니다. 품에는 `process type:version:processname:start`와 같은 구문이 있습니다. 예를 들면 `Basic Approval:1.0:Process:Start Basic Approval`과 같습니다.

창 맨 위에 나오는 메시지에 "Process Cloud Service 접속 없음"이 표시되면 Oracle Process Cloud Service와 Oracle Content Management 간 통합이 구성되지 않은 것입니다. 관리자에게 문의하십시오.

프로세스 작성자가 사용자를 프로세스 게시자로 추가해야 목록에서 프로세스를 확인할 수 있습니다. 프로세스 작성자는 모든 사이트 방문자를 프로세스의 게시자로 추가해야 합니다. 그렇지 않으면 방문자가 품을 작성할 수 있지만 프로세스를 게시할 수는 없습니다.

- f. 다음 중에서 선택하여 품에 대한 추가 기본값을 설정합니다.
  - **품 제목:** 필요에 따라 기본 품 제목을 원하는 제목으로 바꿉니다.
  - **제출 단추 이름:** 필요에 따라 **제출** 단추 이름을 지정하는 값으로 바꿉니다.
  - **제출 단추 표시:** 필요에 따라 구성요소에 **제출** 단추를 표시하거나 숨깁니다 (기본적으로 표시됨). 유사한 단추가 품 자체에 제공될 경우 구성요소 **제출** 단추를 숨길 수 있습니다.
  - **제출 확인:** 필요에 따라 기본 확인 메시지를 자신만의 메시지로 바꿉니다.
  - **제출 확인 표시:** 필요에 따라 **제출** 단추를 누를 때 확인 메시지를 표시하거나 숨깁니다. 기본적으로 프로세스 시작 품 구성요소 내에 표시됩니다.
  - **저장 단추 표시:** 필요에 따라 구성요소에 **저장** 단추를 표시하거나 숨깁니다(기본값은 숨김). 관련 프로세스에서 작업을 저장하고 나중에 되돌릴 수 있는 경우 구성요소 **저장** 단추를 표시할 수 있습니다.
  - **무시 단추 표시:** 필요에 따라 구성요소에 **무시** 단추를 표시하거나 숨깁니다(기본값은 숨김). 사용자가 품 콘텐츠를 무시하고 다시 시작하도록 허용하려면 **무시** 단추를 표시할 수 있습니다.
  - **첨부 파일 표시:** 필요에 따라 파일을 업로드할 수 있는 기능이 제공되는 품에 **첨부 파일** 영역을 표시하거나 숨깁니다. 업로드된 파일은 Oracle Process Cloud Service에서 프로세스의 일부로 저장됩니다.
  - **기본값 사용자정의:** 시작 품을 미리 채울 이름 및 값 쌍을 설정할 수 있습니다. **필드** 추가를 누르고 필드 이름 및 값을 입력합니다. 이름은 품에서 사용되는 레이블이 아니라 품 필드 중 하나이며 값은 해당 필드에 허용되는 값입니다. 현재 로그인한 사용자 데이터를 삽입하려면 특수 값 `%%username%%` 및 `%%userid%%`를 사용합니다.
5. [일반 사항] 탭에서 간격, 맞춤 및 기타 표시 옵션을 수정할 수 있습니다.

6. [스타일] 탭에서 미리 정의된 스타일이나 사용자정의 옵션을 사용하여 구성요소가 포함된 프레임의 서식을 지정합니다.
7. [링크] 탭에서 구성요소가 제공한 트리거에 작업을 지정합니다.
  - 시작 폼이 제출됨: 이 트리거는 사용자가 제출 단추를 누를 때 발생합니다.
  - 시작 폼이 저장됨: 이 트리거는 사용자가 저장 단추를 누를 때 발생합니다.
  - 시작 폼이 무시됨: 이 트리거는 사용자가 무시 단추를 누를 때 발생합니다.
 트리거를 눌러 작업을 지정합니다. 트리거 및 작업 지정에 대한 자세한 내용은 [트리거 및 작업 사용](#)(을) 참조하십시오.



## 프로세스 작업 목록

프로세스 작업 목록을 사용하여 Oracle Process Cloud Service에서 정의된 프로세스를 선택적으로 나열할 수 있습니다. 작업 목록 구성요소의 작업에 대한 자세한 정보를 표시하거나 작업 세부정보 구성요소와 함께 프로세스 작업 목록 구성요소를 사용하여 목록을 단순화하고 선택한 작업에 대한 자세한 정보만 표시할 수 있습니다.

사이트에서 작업할 프로세스 작업 목록의 경우 다음이 충족되어야 합니다.

- 이 구성요소로 연관된 프로세스를 표시하려면 연관된 프로세스가 Oracle Process Cloud Service에서 정의되어 있어야 합니다. *Using Processes in Oracle Integration*의 Developing Structured Processes를 참조하십시오.
- 사이트는 지정된 사용자로 제한되거나 Oracle Content Management 사용자 롤을 보유한 사용자로 제한되는 보안 사이트여야 합니다. [사이트 보안 변경](#)을 참조하십시오.
- 관리자가 Oracle Process Cloud Service와 Oracle Content Management 간의 통합을 설정해야 합니다. *Administering Oracle Content Management*의 Oracle Process Cloud Service와 통합을 참조하십시오. 두 서비스 간의 통합을 위해서는 SSO 사인온이 필요하므로 두 서비스가 동일한 ID 도메인에 있어야 합니다.

페이지에 프로세스 작업 목록 구성요소를 추가하려면 다음을 수행합니다.

1. 편집할 페이지로 이동하고 가 편집으로 설정되어 있는지 확인합니다.
2. 페이지에 구성요소를 추가합니다.
3. 구성요소 및 구성요소 모양을 편집하려면 메뉴 아이콘 을 누르고 설정을 선택합니다.
4. 사용자정의 설정을 선택하여 표시 옵션을 설정하고 사용 가능한 작업을 필터링합니다.
5. 작업 목록에 대한 표시 옵션을 설정하려면 다음 중에서 선택합니다.

### 주:

구성요소에 대한 표시 옵션을 설정할 수 있지만 작업 자체에 대한 표현은 Oracle Process Cloud Service의 디자인에서 결정됩니다.

- 세부정보 표시: 작업 목록 자체에 작업 세부정보를 포함하려면 선택합니다. 작업 세부정보 구성요소와 함께 프로세스 작업 목록 구성요소를 사용하여 목록을 간소화하고 선택한 작업에 대한 세부정보만 표시합니다.

- **검색 표시:** 작업 목록 맨 위에 검색 도구 모음을 포함시키려면 선택합니다. **검색 표시**를 선택하는 경우 **필터 표시**를 추가로 선택하여 검색 도구 모음에 필터 옵션을 포함시킬 수 있습니다. 아래 필터 옵션을 사용하여 기본 필터 값을 설정합니다.
  - **모두 선택 표시:** 사용자가 처리를 위해 표시된 모든 작업을 선택할 수 있도록 하려면 선택합니다.
  - **페이지 크기:** 표시할 최대 작업 수를 지정하려면 조정합니다. 지정된 최대값보다 작업이 적은 경우 구성요소는 더 작은 숫자로 자동으로 조정합니다. 지정된 최대값보다 작업이 많은 경우 이 구성요소는 최대 수를 표시하고 추가 페이지에 대한 링크를 추가합니다.
6. 목록에서 사용 가능한 작업을 필터링하려면 다음 중에서 선택합니다.
- 적용하는 필터는 작업의 초기 목록을 결정합니다. 위의 **검색 표시** 및 **필터 표시** 옵션을 선택하는 경우 사용자는 설정을 조정하여 **보낸 사용자**를 제외한 모든 설정의 필터를 수정할 수 있습니다.
- **키워드 검색:** 필요에 따라 검색어를 하나 이상 지정합니다. 검색 결과에는 지정된 모든 키워드가 포함되는 제목이 있는 작업만 포함됩니다. **검색 표시**를 선택한 경우 이러한 키워드는 검색 도구 모음에 표시되며 사용자가 제거하거나 수정할 수 있습니다.
  - **상태:** 사용 가능한 [상태] 값 중 하나를 선택합니다. 기본값은 **지정됨**입니다.
  - **피지정자:** 사용 가능한 [피지정자] 값 중 하나를 선택합니다. 기본값은 **나와 내 그룹 모두**이며 여기에는 사용자 및 그룹에 사용할 수 있는 작업(획득했지만 작업할 수 없는 경우 포함)이 포함됩니다.
  - **보낸 사용자:** 필요에 따라 작업을 시작한 사용자를 한 명 이상 선택합니다. 사용자 이름을 입력하여 연관된 Oracle Process Cloud Service에서 사용 가능한 사용자 검색을 시작합니다. 지정하는 값은 일반 사용자에게 표시되지 않으며 제거할 수 없습니다. 사용자는 다른 사용자 이름을 추가하여 사용자 목록을 추가로 확장할 수 있습니다.
  - **만기 날짜:** 필요에 따라 달력에서 선택한 날짜, 선택한 날짜 이전 또는 이후, 선택한 날짜 사이인 만기 날짜를 선택합니다.
  - **애플리케이션:** 필요에 따라 특정 애플리케이션 이름을 선택합니다. 사용자가 사용할 수 있는 애플리케이션 목록에서 애플리케이션을 하나 이상 선택할 수 있습니다.
7. [일반 사항] 탭에서 간격, 맞춤 및 기타 표시 옵션을 수정할 수 있습니다.
8. [스타일] 탭에서 미리 정의된 스타일이나 사용자정의 옵션을 사용하여 구성요소가 포함된 프레임의 서식을 지정합니다.

 **주:**

페이지에 프로세스 작업 목록 및 작업 세부정보 폼이 있는 경우 작업 세부정보 구성요소는 목록에서 선택한 작업에 대한 세부정보를 자동으로 표시합니다. 트리거 및 작업에 대한 자세한 내용은 [트리거 및 작업 사용](#)을 참조하십시오.



## 작업 세부정보 폼

프로세스 작업 목록과 함께 작업 세부정보 폼을 사용하여 선택한 작업에 대한 세부정보를 표시할 수 있습니다. 페이지에 프로세스 작업 목록 및 프로세스 세부정보 폼이 있는 경우 프로세스 작업 세부정보 구성요소는 목록에서 선택한 작업에 대한 세부정보를 자동으로 표시합니다.

사이트에서 작업할 작업 세부정보 폼의 경우 다음이 충족되어야 합니다.

- 이 구성요소로 연관된 프로세스를 표시하려면 연관된 프로세스가 Oracle Process Cloud Service에서 정의되어 있어야 합니다. *Using Processes in Oracle Integration*의 Developing Structured Processes를 참조하십시오.
- 사이트는 지정된 사용자로 제한되거나 Oracle Content Management 사용자 롤을 보유한 사용자로 제한되는 보안 사이트여야 합니다. [사이트 보안 변경](#)을 참조하십시오.
- 관리자가 Oracle Process Cloud Service와 Oracle Content Management 간의 통합을 설정해야 합니다. *Administering Oracle Content Management*의 Oracle Process Cloud Service와 통합을 참조하십시오. 두 서비스 간의 통합을 위해서는 SSO 사인온이 필요하므로 두 서비스가 동일한 ID 도메인에 있어야 합니다.

페이지에 작업 세부정보 구성요소를 추가하려면 다음을 수행합니다.

1. 편집할 페이지로 이동하고 가 **편집**으로 설정되어 있는지 확인합니다.
2. 페이지에 구성요소를 추가합니다.
3. 구성요소 및 구성요소 모양을 편집하려면 메뉴 아이콘 을 누르고 **설정**을 선택합니다.
4. **사용자정의 설정**을 선택하여 표시 옵션을 설정합니다.
5. 작업 세부정보 폼에 대한 표시 옵션을 설정하려면 다음 중에서 선택합니다.

#### 주:

구성요소에 대한 표시 옵션을 설정할 수 있지만 작업 세부정보에 대한 표현은 Oracle Process Cloud Service의 디자인에서 결정됩니다.

- **작업 표시:** 승인, 거부 등과 같이 사용자가 사용할 수 있는 작업을 표시하려면 선택합니다.
  - **저장 표시:** 저장 단추를 표시하려면 선택합니다.
  - **닫기 표시:** 닫기 단추를 표시하려면 선택합니다.
  - **첨부 파일 표시:** 세부정보 폼에 **첨부 파일** 섹션을 표시하려면 선택합니다.
  - **의견 표시:** 세부정보 폼에 **의견** 섹션을 표시하려면 선택합니다.
  - **내역 표시:** 세부정보 폼에 **내역** 섹션을 표시하려면 선택합니다.
  - **추가 정보 표시:** 세부정보 폼에 **추가 정보** 섹션을 표시하려면 선택합니다.
  - **링크 표시:** 세부정보 폼에 **링크** 섹션을 표시하려면 선택합니다.
6. [일반 사항] 탭에서 간격, 맞춤 및 기타 표시 옵션을 수정할 수 있습니다.  
기본적으로 작업 세부정보 구성요소는 지정된 모든 세부정보를 표시하도록 확장됩니다. **높이 설정**을 누르고 높이 값을 조정하여 특정 높이를 설정합니다.
  7. [스타일] 탭에서 미리 정의된 스타일이나 사용자정의 옵션을 사용하여 구성요소가 포함된 프레임의 서식을 지정합니다.
  8. [링크] 탭에서 구성요소가 제공한 트리거에 작업을 지정합니다.
    - **제출한 작업 세부정보:** 사용자가 **제출** 단추를 누르면 이 트리거가 발생합니다.
    - **승인된 작업:** 사용자가 **승인** 단추를 누르면 이 트리거가 발생합니다.
    - **거부된 작업:** 사용자가 **거부** 단추를 누르면 이 트리거가 발생합니다.

- **닫힌 작업:** 사용자가 닫기 단추를 누르면 이 트리거가 발생합니다.
- **저장된 작업:** 사용자가 저장 단추를 누르면 이 트리거가 발생합니다.
- **의견 추가된 작업:** 사용자가 의견 텍스트를 추가하고 의견 게시 단추를 누르면 이 트리거가 발생합니다.

트리거를 눌러 작업을 지정합니다.

#### 주:

프로세스 작업 목록 구성요소는 수동 트리거 또는 작업을 지원하지 않습니다. 하지만 페이지에 작업 세부정보 구성요소를 추가하는 경우 목록에서 선택한 작업에 대한 세부정보가 자동으로 표시됩니다. 트리거 및 작업에 대한 자세한 내용은 [트리거 및 작업 사용](#)을 참조하십시오.

Oracle Process Cloud Service 사용에 대한 자세한 내용은 [Getting Started with Process](#)를 참조하십시오.

## 콘텐츠 항목

엔터프라이즈 사용자의 경우 모든 사이트에는 사이트와 연관된 디지털 자산 및 콘텐츠 항목이 포함되어 있는 모음이 있습니다. 콘텐츠 항목 구성요소를 사용하면 사이트에 항목을 손쉽게 추가할 수 있습니다.

편집기의 [콘텐츠] 패널에서 디지털 자산 및 콘텐츠 항목을 직접 끌어 놓을 수 있으며, 디지털 자산 이미지가든 구조화된 콘텐츠 항목이든 상관없이 해당 구성요소가 자동으로 사용됩니다.

[구성요소] 패널에서 선택한 추가 구성요소를 사용하여 콘텐츠 유형에 따라 콘텐츠 항목을 동적으로 표시할 수 있습니다.


- [콘텐츠 항목 구성요소](#)
- [콘텐츠 위치 표시자](#)
- [콘텐츠 목록](#)
- [콘텐츠 검색](#)
- [권장사항](#)



## 콘텐츠 항목 구성요소

엔터프라이즈 사용자는 콘텐츠 항목 구성요소를 사용하여 페이지의 레이아웃을 지정하고 콘텐츠 항목을 추가할 준비가 될 때까지 페이지 상호 작용을 설정합니다.

[콘텐츠] 패널에서 페이지로 콘텐츠 항목을 끌어 놓을 경우 항목을 유지하기 위해 콘텐츠 항목 구성요소가 자동으로 삽입됩니다(콘텐츠 항목이 사용자정의 디지털 자산 유형이 아닌 경우). 사용자정의 디지털 자산을 페이지로 끌어오면 자산이 이미지 또는 비디오 구성요소로 삽입됩니다. 먼저 제거하고 콘텐츠 항목 구성요소를 추가한 다음, [콘텐츠] 패널의 사용자정의 디지털 자산 또는 표준 콘텐츠 항목을 구성요소로 끌어오면 올바르게 표시됩니다.



콘텐츠 항목 구성요소를 페이지에 추가하려면 다음을 수행합니다.

1. 편집할 페이지로 이동하고 가 **편집**으로 설정되어 있는지 확인합니다.
2. 페이지에 구성요소를 추가합니다.

3. 콘텐츠 항목 구성요소 콘텐츠를 편집하려면 메뉴 아이콘  을 누르고 **편집**를 선택하거나 **설정**을 선택하여 설정 패널을 열고 편집 아이콘  을 누릅니다.

 주:

콘텐츠 항목 구성요소에 사용된 콘텐츠 유형이 표준 콘텐츠 유형이 아닌 사용자정의 디지털 자산인 경우 사용자정의 디지털 자산 유형을 편집할 수 없습니다. 구성요소 메뉴에 **편집** 옵션이 없고 설정 패널에 편집 아이콘이 없습니다.

4. 콘텐츠 항목을 변경하고 **저장**을 누릅니다. 콘텐츠 항목이 콘텐츠 저장소에 새 개정으로 저장됩니다.
5. 구성요소 모양을 편집하려면 메뉴 아이콘  을 누르고 **설정**을 선택합니다.
6. [일반 사항] 탭에서 간격, 맞춤 및 기타 표시 옵션을 수정할 수 있습니다.  
콘텐츠 항목이 구성요소에 지정된 경우 콘텐츠 항목에 대한 작은 그림 보기가 표시됩니다. 콘텐츠 항목이 아직 지정되지 않았으면 위치 표시자 이미지 및 메시지가 표시됩니다.
7. 다음 중에서 선택하여 표시되는 콘텐츠에 대한 추가 기본값을 설정합니다.
  - **사용할 버전: 자산의 최신 버전 사용**을 선택하고 콘텐츠 항목의 게시되지 않은 최신 버전이 있는 경우 현재 사이트 업데이트가 게시되면 최신 버전이 자동으로 게시됩니다. 특별히 요청하지 않는 한 모든 항목은 초안 또는 최신 버전이 됩니다.
  - **항목 뷰**: 콘텐츠 항목을 표시하는 데 사용된 레이아웃을 선택합니다. 제공된 기본 레이아웃은 콘텐츠 항목의 모든 필드를 표시합니다. 콘텐츠 항목에 이에 맞게 제작된 다른 사용자정의 레이아웃이 있는 경우 사용 가능한 레이아웃 중 어느 것이든 선택할 수 있습니다. **사이트에서 사용하는 경우 사용자정의 설정에 대한 지원 추가**가 사용으로 설정된 사용자정의 레이아웃을 선택한 경우 사용자정의 데이터를 레이아웃에 추가할 수 있습니다.  을 누르고 **사용자정의 데이터** 필드에 데이터를 입력하고 **뒤로**를 눌러 일반 사항 탭으로 돌아갑니다.
  - **개별 항목을 표시하는 페이지**: 하나 이상의 페이지를 세부정보 페이지로 지정한 경우 페이지가 여기에 나열됩니다. 사용자가 자세한 정보를 보기 위해 콘텐츠 항목에서 링크를 누를 때 자세한 정보를 표시할 페이지를 선택합니다.

 주:

세부정보 페이지를 생성하지 않는 경우 세부정보를 표시할 링크가 기본 레이아웃의 콘텐츠 항목에 표시되지 않습니다.



8. [스타일] 탭에서 미리 정의된 스타일이나 사용자정의 옵션을 사용하여 구성요소가 포함된 프레임의 서식을 지정합니다.

## 콘텐츠 위치 표시자

엔터프라이즈 사용자는 콘텐츠 위치 표시자 구성요소를 사용하여 하나 이상의 유형에 대한 콘텐츠 항목을 동적으로 표시할 수 있습니다.

예를 들어 지정된 세부정보 페이지에서 콘텐츠 항목 위치 표시자를 사용할 수 있으며 사용자가 특정 콘텐츠 항목에 대한 자세한 정보를 얻기 위해 링크를 누르면 관련 콘텐츠 항목에 대한 세부정보 뷰가 자동으로 로드됩니다. 예를 들어 여러 문서가 제공되는 페이지가 있고 각 문서에 헤드라인과 이미지가 포함되어 있는 경우 사용자가 특정 문서를 누르면 세부정보 페이지에는 선택한 문서와 상관없이 전체 문서가 표시됩니다.

콘텐츠 위치 표시자 구성요소를 페이지에 추가하려면 다음을 수행합니다.

1. 편집할 페이지로 이동하고 가 **편집**으로 설정되어 있는지 확인합니다.
2. 페이지에 구성요소를 추가합니다.
3. 구성요소 및 구성요소 모양을 편집하려면 메뉴 아이콘 을 누르고 **설정**을 선택합니다.
4. [일반 사항] 탭에서 간격, 맞춤 및 기타 표시 옵션을 수정할 수 있습니다.
5. 다음 중에서 선택하여 표시되는 콘텐츠에 대한 추가 기본값을 설정합니다.
  - **콘텐츠 유형:** 사용 가능한 콘텐츠 유형을 하나 이상 선택합니다. 콘텐츠 유형은 사이트 모음의 콘텐츠 항목 유형으로, 사용자정의 디지털 자산 유형을 포함합니다.
  - **항목 뷰:** 콘텐츠 항목을 표시하는 데 사용된 레이아웃을 선택합니다. 제공된 기본 레이아웃은 콘텐츠 항목의 모든 필드를 표시합니다. 콘텐츠 항목에 이에 맞게 제작된 다른 사용자정의 레이아웃이 있는 경우 사용 가능한 레이아웃 중 어느 것이든 선택할 수 있습니다.
  - **개별 항목을 표시하는 페이지:** 하나 이상의 페이지를 세부정보 페이지로 지정한 경우 페이지가 여기에 나열됩니다. 사용자가 자세한 정보를 보기 위해 콘텐츠 항목에서 링크를 누를 때 자세한 정보를 표시할 페이지를 선택합니다.

### 주:

세부정보 페이지를 생성하지 않는 경우 세부정보를 표시할 링크가 기본 레이아웃의 콘텐츠 항목에 표시되지 않습니다.

6. [스타일] 탭에서 미리 정의된 스타일이나 사용자정의 옵션을 사용하여 구성요소가 포함된 프레임의 서식을 지정합니다.




## 콘텐츠 목록

엔터프라이즈 사용자는 콘텐츠 목록 구성요소를 사용하여 특정 유형의 콘텐츠 항목을 동적으로 표시할 수 있습니다.

예를 들어 지정된 세부정보 페이지에서 콘텐츠 항목 목록을 사용할 수 있으며 사용자가 특정 콘텐츠 항목에 대한 자세한 정보를 얻기 위해 링크를 누르면 관련 콘텐츠 항목에 대한 세부정보 뷰가 자동으로 로드됩니다. 예를 들어 여러 문서가 제공되는 페이지가 있고 각 문서에 헤드라인과 이미지가 포함되어 있는 경우 사용자가 특정 문서를 누르면 세부정보 페이지에는 선택한 문서와 상관없이 전체 문서가 표시됩니다.

페이지에 콘텐츠 목록 구성요소를 추가하려면 다음과 같이 하십시오.



1. 편집할 페이지로 이동하고 가 **편집**으로 설정되어 있는지 확인합니다.
2. 페이지에 구성요소를 추가합니다.
3. 구성요소 및 구성요소 모양을 편집하려면 메뉴 아이콘 을 누르고 **설정**을 선택합니다.
4. [일반 사항] 탭에서 간격, 맞춤 및 기타 표시 옵션을 수정할 수 있습니다.
5. 다음 중에서 선택하여 표시되는 콘텐츠에 대한 추가 기본값을 설정합니다.
  - **콘텐츠 유형:** 사용 가능한 콘텐츠 유형 중 하나를 선택합니다. 콘텐츠 유형은 사이트 모음의 콘텐츠 항목 유형으로, 사용자정의 디지털 자산 유형을 포함합니다.
  - **최대 항목 및 항목에서 시작:** 표시할 최대 항목 수 및 표시를 시작할 위치를 지정합니다. 추가 항목은 표시되지 않습니다.
  - **페이지 매김:** 페이지 매김 옵션을 목록에 포함할지 여부를 지정합니다. 선택하는 경우 단추 또는 페이지 번호와 다른 레이블로 페이지 매김이 표시되는 방식을 조정할 수 있습니다.
  - **날짜:** 선택한 날짜 전후, 사이 또는 이내의 항목 생성 날짜나 날짜 범위를 기준으로 표시할 콘텐츠 항목을 선택하려면 제공된 옵션을 사용합니다.
  - **범주:** 콘텐츠 항목 목록을 필터링할 범주를 선택하여 특정 범주의 항목만 표시합니다. **범주 선택**을 눌러 사이트 저장소에서 범주 선택에 사용할 수 있는 슬라이드아웃 패널을 엽니다. 동일한 분류법에서 둘 이상의 범주를 선택하면 표시 항목은 선택한 범주 중 하나에만 지정되면 됩니다. 상이한 분류법에서 둘 이상의 범주를 선택하면 표시 항목은 선택한 범주 모두에 지정되어야 합니다. 예를 들어, **목적지** 분류법에서 **도시** 및 **공원** 범주를 선택하면 도시 또는 공원에 속한 항목이 표시됩니다. **목적지** 분류법에서 **도시** 범주를 선택하고 **지역** 분류법에서 **유럽** 범주를 선택하면 유럽의 도시에 속한 항목이 표시됩니다.  
  
기본적으로 선택한 범주의 모든 하위 항목을 사용할 수 있습니다. 특정 범주 노드로 항목 목록을 제한하려면 노드를 선택하고 **하위 범주 포함**을 선택 해제합니다.
  - **언어:** 저장소와 연관된 항목 중에서 언어를 선택합니다.
  - **추가 질의 문자열(선택사항):** field.dept eq "Finance"와 유사한 구문을 사용하여 표시된 항목 목록을 더욱 세분화하려면 추가 질의 매개변수를 지정합니다. 사용 가능한 연산자 목록은 아래 표를 참조하십시오.
  - **정렬 기준:** 항목을 이름 또는 날짜 기준으로 오름차순이나 내림차순으로 정렬합니다. 콘텐츠 유형에 다른 날짜, 숫자 또는 십진수 필드가 포함되면 해당 필드 기준으로 정렬할 수도 있습니다.  
  
**사용자정의**를 선택하고 사용자정의 표현식을 입력할 수도 있습니다. 예를 들어, 부서 번호 기준으로 오름차순으로 정렬하려면 fields.deptno:asc를 입력할 수 있습니다. 정렬을 위한 URL 매개변수를 정의할 수도 있습니다. 예를 들어, {{URLParams.sortBy}}를 입력한 다음 사이트 URL 끝에 ?sortBy=fields.deptno:asc 매개변수를 추가할 수 있습니다.
  - **항목 뷰:** 콘텐츠 항목을 표시하는 데 사용된 레이아웃을 선택합니다. 제공된 기본 레이아웃은 콘텐츠 항목의 모든 필드를 표시합니다. 콘텐츠 항목에 이에 맞게 제작된 다른 사용자정의 레이아웃이 있는 경우 사용 가능한 레이아웃 중 어느 것이든 선택할 수 있습니다. **사이트에서 사용하는 경우 사용자정의 설정에 대한 지원 추가**가 사용으로 설정된 사용자정의 레이아웃을 선택한 경우 사용자정의 데이터를 레이아웃에 추가할 수 있습니다. 을 누르고 **사용자정의 데이터** 필드에 데이터를 입력하고 **뒤로**를 눌러 일반 사항 탭으로 돌아갑니다.

- **개별 항목을 표시하는 페이지:** 하나 이상의 페이지를 세부정보 페이지로 지정한 경우 페이지가 여기에 나열됩니다. 사용자가 자세한 정보를 보기 위해 콘텐츠 항목에서 링크를 누를 때 자세한 정보를 표시할 페이지를 선택합니다.

 **주:**

세부정보 페이지를 생성하지 않는 경우 세부정보를 표시할 링크가 기본 레이아웃의 콘텐츠 항목에 표시되지 않습니다.

- **목록 뷰:** 항목을 배열할 섹션 레이아웃을 선택합니다. 이 목록에는 사용자정의 레이아웃이 포함될 수 있습니다. 다음 레이아웃이 제공됩니다.
  - **가로:** 항목을 가로줄에 차례로 배열합니다.
  - **세로:** 항목을 세로줄에 차례로 배열합니다. 레이아웃을 선택하지 않는 경우 기본값입니다.
  - **2열:** 항목을 여러 행에 두 개씩 배열합니다.
  - **3열:** 항목을 여러 행에 세 개씩 배열합니다.

선택한 레이아웃 옆에 있는 오른쪽 화살표를 눌러 레이아웃 설정을 수정합니다.
- **빈 목록 뷰:** 빈 목록을 표시할 방법을 선택합니다.
- **옵션:** 콘텐츠 검색 구성요소에 지정된 검색 질의에 자동 새로고침이 있어야 하는 경우 선택합니다.
- **간격:** 항목을 표시할 방법을 선택합니다.

6. [스타일] 탭에서 미리 정의된 스타일이나 사용자정의 옵션을 사용하여 구성요소가 포함된 프레임의 서식을 지정합니다.

**표 10-1 질의 연산자**

연산자	예제	지원되는 데이터 유형	설명
eq	?q=name eq "John" ?q=type eq "DigitalAsset" ?q=type eq "Employee" and fields.DOB eq "1994/09/26T16:23:45.208" ?q=type eq "Employee" and fields.DOB eq "1994/09/26T16:23:45.208"	텍스트, 참조, 숫자, 십진수, 부울, 날짜 시간	같은 연산자(eq)는 질의에서 제공된 정확한 값과 일치합니다. 이 연산자는 다중 값 데이터 유형에 적용할 수 없습니다. 이 연산자와 함께 제공되는 값은 표준 필드를 제외하고 대소문자를 구분하지 않습니다. 이 연산자는 값의 특수 문자도 고려합니다.

표 10-1 (계속) 질의 연산자

연산자	예제	지원되는 데이터 유형	설명
co	<pre>?q=(type eq "Employee" AND name co "john alex") ?q=(type eq "Car" AND fields.features co "manual")</pre>	텍스트, 참조, 숫자, 십진수, 날짜 시간, 큰 텍스트	포함 연산자(co)는 기준에서 제공된 모든 단어와 일치합니다. 단어는 특수 문자로 값을 분할하여 구성됩니다. 한 단어 이상이 포함된 결과가 제공됩니다(이 예의 경우 john 및/또는 alex). 이 연산자는 검색 시 값의 특수 문자를 고려하지 않습니다. 이 연산자는 불용 단어에 대해 검색을 수행하지 않습니다. 불용 단어에 대해 자세히 알아보려면 Apache Lucene 설명서를 참조하십시오. 이 연산자는 단일 값 속성의 경우 <i>텍스트</i> , <i>큰 텍스트</i> 에 적용할 수 있으며, 다중 값 속성의 경우 <i>텍스트</i> , <i>참조</i> , <i>숫자</i> , <i>십진수</i> , <i>날짜 시간</i> , <i>큰 텍스트</i> 에 적용할 수 있습니다. 가능한 날짜 시간 형식을 파악하려면 아래의 <i>지원되는 날짜/날짜 시간 형식(24시간)</i> 표를 참조하십시오. 이 연산자와 함께 제공되는 값은 대소문자를 구분하지 않습니다.
sw	<pre>?q=type eq "Employee" AND name sw "Joh" ?q=type eq "Employee" AND fields.city sw "Los"</pre>	텍스트	다음으로 시작 연산자(sw)는 필드 조건에 제공된 이니셜 문자 값과만 일치합니다. 이 연산자는 다중 값 데이터 유형에 적용할 수 없습니다. 이 연산자와 함께 제공되는 값은 대소문자를 구분하지 않습니다.
ge	<pre>?q=(type eq "Employee" AND fields.age ge "40") ?q=type eq "DigitalAsset" AND updatedAt ge "20171026"</pre>	숫자, 십진수, 날짜 시간	보다 크거나 같은 연산자(ge)는 숫자 및 날짜 시간 값과만 일치합니다. 가능한 날짜 시간 형식을 파악하려면 아래의 <i>지원되는 날짜/날짜 시간 형식(24시간)</i> 표를 참조하십시오. 이 연산자는 다중 값 데이터 유형에 적용할 수 없습니다.
le	<pre>?q=(type eq "Employee" AND fields.weight le "60.6")</pre>	숫자, 십진수, 날짜 시간	보다 작거나 같은 연산자(le)는 숫자 및 날짜 시간 값과만 일치합니다. 가능한 날짜 시간 형식을 파악하려면 아래의 <i>지원되는 날짜/날짜 시간 형식(24시간)</i> 표를 참조하십시오. 이 연산자는 다중 값 데이터 유형에 적용할 수 없습니다.

표 10-1 (계속) 질의 연산자

연산자	예제	지원되는 데이터 유형	설명
gt	?q=(type eq "Employee" AND fields.age gt "20")	숫자, 십진수, 날짜 시간	보다 큼 연산자(gt)는 숫자 및 날짜 시간 값과만 일치합니다. 가능한 날짜 시간 형식을 파악하려면 아래의 <b>지원되는 날짜/날짜 시간 형식 (24시간)</b> 표를 참조하십시오. 가능한 날짜 시간 형식을 파악하려면 아래의 <b>지원되는 날짜/날짜 시간 형식 (24시간)</b> 표를 참조하십시오. 이 연산자는 다중 값 데이터 유형에 적용할 수 없습니다.
lt	?q=(type eq "Employee" AND fields.age lt "20") ?q=type eq "Employee" AND createdAt lt "1994/09/26T16:23:45.208"	숫자, 십진수, 날짜 시간	보다 작음 연산자(lt)는 숫자 및 날짜 시간 값과만 일치합니다. 가능한 날짜 시간 형식을 파악하려면 <b>지원되는 날짜/날짜 시간 형식 (24시간)</b> 섹션을 참조하십시오. 이 연산자는 다중 값 데이터 유형에 적용할 수 없습니다.
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*")	텍스트, 큰 텍스트	구문 질의 또는 근접 검색(일치) 연산자(mt)를 통해 다른 단어와의 간격이 특정 범위에 속하는 단어를 찾을 수 있습니다. 결과는 가장 일치하는 항목별로 정렬됩니다. 이는 "petrol 20kmpl" 기준에서 제공된 값이 "petrol fuel mileage runs 20KMPL in the speed way"를 포함할 수 있는 실제 콘텐츠를 검색해야 하는 경우 콘텐츠 항목 검색에 유용합니다. 일치 연산자는 제공된 값에서 와일드 카드 문자를 사용할 수도 있으며 단일 값 내에서 단일 문자와 다중 문자 와일드 카드 문자 검색을 지원합니다. 단일 문자 와일드 카드 문자에는 ?를 사용하고 다중 문자에는 *를 사용합니다. "John"과 "Joan"은 단일 문자의 경우 "Jo?n"으로 검색되고 다중 문자의 경우 "Jo*"로 검색될 수 있습니다. 이 연산자는 단일 값 및 다중 값 데이터 유형에 적용할 수 있습니다. 이 연산자는 불용 단어에 대해 검색을 수행하지 않습니다. 불용 단어에 대해 자세히 알아보려면 Apache Lucene 설명서를 참조하십시오. 이 연산자와 함께 제공되는 값은 대소문자를 구분하지 않습니다.

표 10-1 (계속) 질의 연산자



연산자	예제	지원되는 데이터 유형	설명
sm	?q=(type eq "Employee" And fields.city sm "Rome")	텍스트, 큰 텍스트	유사 질의 연산자입니다. 이 연산자로는 지정된 기준과 발음이 유사한 값을 검색할 수 있습니다. 퍼지 검색이라고도 하며, 기본적으로 결과 일치율을 위해 최대 두 가지 편집이 사용됩니다. "Rome"은 "Dome"과 유사합니다. 이 연산자는 단일 값 및 다중 값 데이터 유형에 적용할 수 있습니다. 이 연산자와 함께 제공되는 값은 대소문자를 구분하지 않습니다.
AND	?q=(type eq "Employee" AND name eq "John" AND fields.age ge "40")	해당 사항 없음	AND 연산자는 다중 질의 조건 사이에 AND 조건을 넣는 데 사용할 수 있습니다. 이 연산자가 OR보다 우선합니다.
OR	type eq "Employee" AND name eq "John" OR fields.age ge "40"	해당 사항 없음	OR 연산자는 다중 질의 조건 사이에 OR 조건을 넣는 데 사용할 수 있습니다.
()	?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)	해당 사항 없음	기준에서 조건을 그룹화할 괄호, 둘러싸기 연산자입니다. 이 연산자의 우선순위가 가장 높으며 그 다음은 AND, OR순입니다.

## 콘텐츠 검색

엔터프라이즈 사용자는 콘텐츠 검색 구성요소를 사용하고 검색으로 반환된 작업을 지정할 수 있습니다.

사용자정의된 검색 도구 모음을 삽입하여 페이지에 표시되는 콘텐츠를 변경/새로고침하거나, 검색 결과 페이지를 열거나 경보를 표시하는 등의 다른 작업을 선택할 수 있습니다.

콘텐츠 검색 구성요소를 페이지에 추가하려면 다음을 수행합니다.

1. 편집할 페이지로 이동하고 가 **편집**으로 설정되어 있는지 확인합니다.
2. 페이지에 콘텐츠 검색 구성요소를 추가합니다.
3. 구성요소 및 구성요소 모양을 편집하려면 메뉴 아이콘 을 누르고 **설정**을 선택합니다.
4. [일반 사항] 탭에서 위치 표시자 텍스트, 간격, 맞춤 및 기타 표시 옵션을 수정합니다.
5. [스타일] 탭에서 미리 정의된 스타일이나 사용자정의 옵션(글꼴, 테두리, 배경색 등)을 사용하여 구성요소가 포함된 프레임의 서식을 지정합니다.

6. 검색 구성요소를 사용하여 현재 페이지의 데이터를 새로고침하려면 페이지에 콘텐츠 목록 구성요소를 추가합니다. **설정**으로 이동하고 콘텐츠 유형을 비롯하여 표시할 항목 수, 결과 페이지 매김 여부 또는 스크롤 시 지연 로드(페이지가 스크롤될 때 콘텐츠 로드)와 같은 기타 질의 정보를 선택합니다. 그러면 보기 모드로 이동하여 검색을 수행할 수 있습니다.
7. 검색 구성요소를 사용하여 현재 페이지의 데이터를 새로고침하려는 경우 페이지에 콘텐츠 목록 구성요소가 두 개 이상 있으면 결과를 표시할 콘텐츠 목록을 제외한 모든 콘텐츠 목록에 대한 자동 질의를 해제해야 합니다. 콘텐츠 목록 설정으로 이동하고 **콘텐츠 검색 구성요소에서 검색 질의 시 자동 새로고침**의 선택을 취소합니다.
8. 두 개 이상의 콘텐츠 검색 구성요소(각 검색 구성요소가 결과에 대한 특정 콘텐츠 목록에 영향을 줌)가 페이지에 포함되도록 하려면 콘텐츠 목록 구성요소에서 모든 자동 새로고침 옵션의 선택을 취소하고 트리거/작업을 사용하여 콘텐츠 검색 구성요소를 개별 콘텐츠 목록에 연관합니다.
9. 각 검색 구성요소에 대해 [링크] 탭에서 작업을 구성요소에 연관합니다. [링크 유형 선택] 드롭다운 목록에서 옵션을 선택합니다. **트리거 작업**을 선택하고 **검색 질의 시**를 눌러 사용 가능한 트리거를 보거나 새 트리거를 생성합니다. 검색할 콘텐츠 목록을 찾아 확장하고 **콘텐츠 검색**을 작업 목록으로 끌어옵니다. **검색** 아래에서 **검색 문자열**을 선택합니다. 선택한 콘텐츠 목록이 **이 작업 수행 위치** 목록에서 선택됩니다.
10. 콘텐츠 검색 구성요소를 사용하여 검색 결과 페이지에 질의를 전달할 수도 있습니다. 검색 구성요소의 [링크] 탭에서 **검색 페이지**를 선택하여 검색 결과 페이지를 표시합니다. 기본 페이지를 사용하거나 직접 생성한 검색 결과 페이지의 링크를 사용할 수 있습니다. [페이지] 드롭다운 목록에서 페이지를 선택하고 페이지의 표시 작업을 선택합니다. 특정 콘텐츠 유형에 맞게 결과를 조정할 수도 있고, 새 창에서 결과를 열 수도 있습니다.

다음 사항은 효과적인 검색 결과 페이지를 설계하는 데 도움이 될 수 있습니다.



- 페이지를 생성하여 검색 페이지로 지정합니다. 기본적으로 해당 페이지는 숨김으로 표시되지만 페이지 속성에서 설정을 변경할 수 있습니다.
- 페이지를 편집하고 페이지에 콘텐츠 목록을 추가합니다. 콘텐츠 목록에 대한 설정을 편집하여 콘텐츠 유형을 지정할 수 있습니다. 또는 검색 구성요소 링크 설정에서 콘텐츠 유형을 지정할 수도 있습니다(위 참조). 링크 설정을 사용하는 경우 검색 시작에 사용되는 구성요소에 따라 다양한 콘텐츠 유형의 결과를 표시할 수 있는 검색 결과 페이지가 제공됩니다.
- 표시를 편집하려면 설정을 변경합니다. 예를 들어, 페이지가 검색 결과에 배타적으로 사용될 수도 있으므로 지연 로드 페이지 매김을 선택합니다.
- 검색 결과 페이지에도 콘텐츠 검색 구성요소를 넣을 수 있습니다. 그러면 사용자가 필요에 따라 검색 결과를 세분화할 수 있도록 페이지 실행에 사용된 검색 문자열이 표시됩니다.

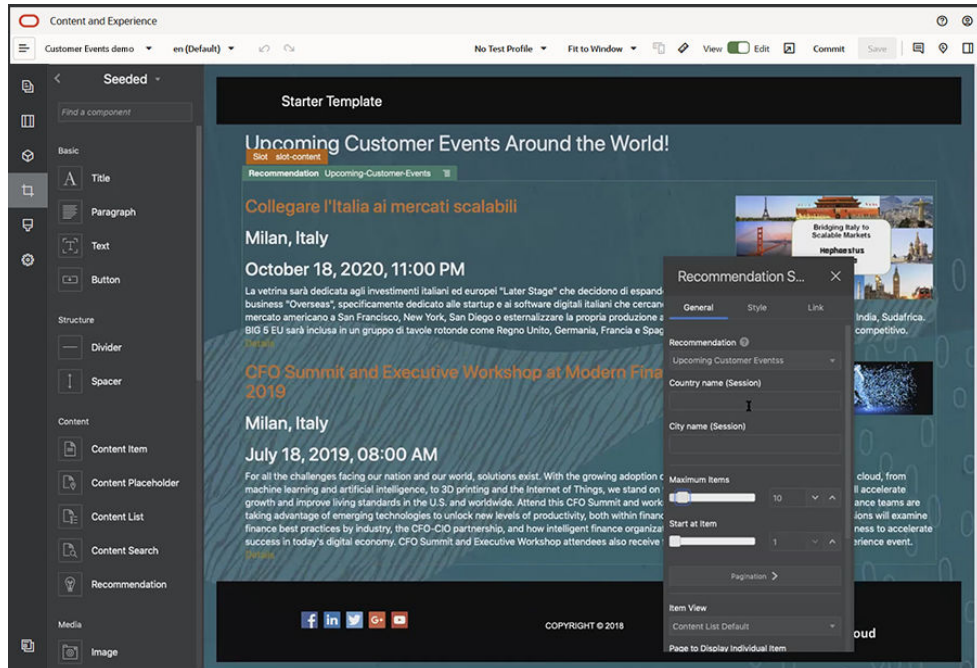
## 권장사항

엔터프라이즈 사용자는 **권장사항** 구성요소를 사용하여 위치 또는 관심 분야를 기반으로 자산을 표시하여 웹 사이트 방문자를 위한 개인설정 환경을 제공할 수 있습니다. 저장소 제공자가 권장사항을 생성할 때 대상자 속성(예: 사이트 방문자의 지리적 위치 정보)과 일치하는 자산을 찾는 일련의 규칙을 정의합니다. 예를 들어, 유럽 IP 주소를 사용하는 사이트 방문자에게는 사이트 홈페이지에서 유럽에 대한 이벤트 공지가 표시될 수 있으며, 북미의 사이트 방문자에게는 미국 및 캐나다의 이벤트가 표시됩니다.

디지털 자산과 마찬가지로 권장사항은 저장소와 연관되며, 저장소에 대한 콘텐츠 제공자인 사용자의 검토 및 게시를 위해 워크플로우를 통해 편집 및 이동될 수 있습니다. 단, 제공자 권한이 없는 경우에도 권장사항을 확인하고 테스트하여 사이트 또는 헤드리스 환경에서 사용하기 전에 작동 방식을 확인할 수 있습니다.

페이지에 권장사항을 추가하려면 다음과 같이 하십시오.

1. 측면 메뉴에서 **사이트**를 누르고 권장사항을 추가할 사이트를 선택한 다음, 마우스 오른쪽 단추 누르기 메뉴에서 **열기**를 선택하거나 작업 모음에서  을 누릅니다.
2. 사이트를 **편집**으로 토글하고 사용할 업데이트를 선택하거나 새로 생성합니다.
3. 측면 메뉴에서 **구성요소**를 선택합니다.
4. **시드**를 엽니다.
5. 콘텐츠 섹션 아래의 **권장사항**을 눌러 끌어서 페이지 위에 놓습니다.
6. 권장사항 메뉴()에서 **설정**을 선택합니다.



7. 권장사항 설정 **일반 사항** 탭에서 사용할 권장사항을 선택합니다. 페이지가 새로고침되고 권장 자산을 표시합니다.
8. 선택적으로, 선택한 권장사항 옆에 있는 화살표를 눌러 이 권장사항에 사용되는 대상자 속성의 기본값을 입력합니다. 값 추가가 완료되면 **뒤로**를 누릅니다.
9. 사용할 버전, 표시할 최대 항목, 기본 콘텐츠 레이아웃 등 원하는 추가 속성을 변경합니다. 완료되면 권장사항 설정을 닫고 **저장**을 누릅니다.

 **주:**

미리 정의된 대상자 속성값으로 테스트 프로파일을 생성한 경우, 메뉴 모음에 있는 테스트 프로파일 메뉴에서 프로파일을 선택하여 해당 입력에 권장사항이 어떻게 반응하는지 미리 봅니다. 테스트 프로파일 값은 구성요소 설정에서 지정한 기본값을 무효화합니다.

10. 권장사항이 규칙으로 현재 날짜(시스템)를 사용하는 경우 다른 날짜를 이용해서 권장사항을 테스트하려면 테스트 프로파일 메뉴에서 **시스템 날짜**를 선택하고 테스트할 날짜를 선택합니다.

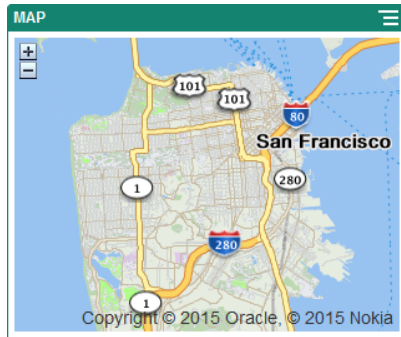
## 기타 구성요소

다른 유형의 콘텐츠를 결합한 구성요소를 살펴보겠습니다.



- 맵
- 헤드라인
- 기사
- 텍스트가 있는 이미지
- 구성요소 그룹
- 페이지에서 Cobrowse 사용
- Oracle Intelligent Advisor
- Oracle Visual Builder

### 맵

사이트에 맵을 추가하면 사용자가 위치 주변 지역을 대화식으로 탐색할 수 있습니다.



페이지에 맵을 추가하려면 다음과 같이 하십시오.





1. 편집할 페이지로 이동하고 가 **편집**으로 설정되어 있는지 확인합니다.
2. 페이지에 구성요소를 추가합니다. Oracle 맵 구성요소에 기본 위치가 표시됩니다.
3. 맵의 위치를 변경하고 표시 속성을 조정하려면 메뉴 아이콘 을 누르고 **설정**을 선택합니다.
4. 맵의 시작 **위치**를 입력합니다. 주소, 우편 번호 또는 콤마로 구분된 위도 및 경도(예: 40.5,-57.6)를 사용할 수 있습니다.  
주소를 사용하는 경우 단일 위치와 일치하는 충분한 정보를 제공해야 합니다. 주소가 여러 개의 위치와 일치할 경우 맵이 공백으로 남습니다.
5. 맵의 초기 **확대/축소** 레벨을 지정합니다. 선택적으로 사용자가 마우스, 트랙패드 또는 화면 컨트롤을 사용하여 확대/축소 레벨을 조정하도록 허용할 수 있습니다.
6. 스타일, 맞춤 및 표시 옵션을 지정합니다.
  - **확대/축소**: 사용자가 맵의 확대/축소 레벨을 조정할 수 있도록 화면 및 장치 옵션을 사용으로 설정합니다.






- **이동:** 사용자가 맵의 포커스 영역을 이동할 수 있도록 화면 및 장치 옵션을 사용으로 설정합니다.
- **표시자 표시:** 시작 위치를 핀 아이콘으로 표시합니다.
- **배율:** 현재 확대/축소 레벨의 눈금을 보여주는 표시기를 포함합니다.
- **개요:** 큰 맥락에서 현재 뷰를 보여주는 인셋 맵을 포함합니다.

## 헤드라인








사이트의 특정 위치가 강조되도록 제목 텍스트, 지원 이미지 및 단락과 함께 헤드라인을 추가할 수 있습니다.

1. 편집할 페이지로 이동하고 가 **편집**으로 설정되어 있는지 확인합니다.
2. 페이지에 구성요소를 추가합니다.
3. 구성요소의 다른 영역을 눌러 텍스트를 추가합니다. 기본 헤드라인 및 기본 헤드라인 아래의 단락에 콘텐츠를 추가합니다. 구성요소의 기본 스타일 서식으로 텍스트가 표시됩니다. Enter를 눌러 다른 텍스트 행을 추가합니다.
4. 어떤 텍스트 부분의 기본 서식을 변경하려면 서식을 지정할 텍스트를 선택하고 서식 도구모음에서 글꼴, 색상, 맞춤과 같은 옵션을 선택합니다.
5. 이 옵션으로 적용된 서식을 제거하려면 텍스트를 선택하고 을 누릅니다.
6. 구성요소 및 구성요소 모양을 편집하려면 메뉴 아이콘 을 누르고 **설정**을 선택합니다.
7. **구성요소**를 선택하여 표시되는 콘텐츠에 대한 세부정보를 설정합니다.
  - **이미지:**
    - 헤드라인에 이미지를 사용하려면 **선택**을 누릅니다. 사이트 저장소의 이미지, 사용자에게 공유된 문서 폴더의 이미지를 선택하거나, 문서 폴더에 이미지를 업로드합니다. 사용할 이미지를 찾아 선택합니다. 디지털 자산이 표시되지 않으면 을 누르고 모음 필터를 **모두**로 변경합니다. 문서 폴더에서 이미지를 선택한 경우, 이를 사이트에 복사하는 대신 파일에 링크할 수 있습니다. 파일에 링크하려면 **사이트에 파일을 복사하는 대신 원본 파일의 참조 사용**을 선택합니다. 이 옵션을 선택하지 않으면 파일의 복사본이 사이트에 저장되고, 사이트에서 참조됩니다. 원본 파일에 링크하면 콘텐츠가 복제되지 않습니다. 링크를 통해 사이트 방문자는 파일에 대한 권한이 변경되거나 보기 권한이 제한되는 경우에도 콘텐츠를 볼 수 있습니다. **확인**을 누릅니다. 디지털 자산을 선택한 경우, 특정 표현을 선택할 수 있습니다. 표현을 선택하지 않으면 원래 크기가 사용됩니다. 사이트가 게시될 때 자산의 최신 버전이 게시되도록 하려면 **자산의 최신 버전 사용**을 선택합니다.

이미지를 편집하려면 을 누르고 다음 작업을 통해 이미지를 편집합니다.



- \* 이미지를 자르려면  **자르기**를 누릅니다. 자르기 도구 모음에서 미리 정의된 이미지 비율 중 하나를 선택하거나 이미지에서 자르기 핸들을 원하는 대로 끌어갑니다. 비율이 만족스러우면 자르기 도구 모음에서 **자르기**를 누릅니다.
- \* 이미지를 회전하거나 뒤집으려면  **회전**을 누릅니다. 회전 도구모음에서 사용자정의 회전 각도를 입력하거나 이미지를 왼쪽 또는 오른쪽으로


회전하는 단추를 사용하거나 이미지를 좌우 또는 상하로 뒤집을지 여부를 선택합니다.

- \* 이미지에 워터마크를 추가하려면  **워터마크**를 누릅니다. 워터마크 툴을 사용하여 필요에 따라 텍스트 크기, 스타일, 색상 및 불투명도를 변경하여 이미지에 텍스트를 추가합니다.
  - \* 이미지 형식을 변경하려면  **옵션**을 누른 후 **형식** 드롭다운 목록에서 새 형식을 선택합니다.
  - \* 배경 색상을 변경하려면  **옵션**을 누른 후 **배경 색상** 드롭다운 메뉴에서 옵션을 선택합니다.
  - \* .jpg 또는 .webp(Google Chrome 브라우저에서 사용 가능)를 편집 중인 경우 보다 작은 파일 크기가 생성되도록 이미지 품질을 변경할 수 있습니다.  **옵션**을 누른 후 **품질** 상자에 새 백분율을 입력합니다.
  - \* 변경사항을 실행 취소하거나 재실행하려면  또는  을 누릅니다. 모든 변경사항을 제거하려면 **재설정**을 누릅니다.
  - \* 이미지 확대를 변경하려면 확대/축소 컨트롤을 사용합니다 (  ).
    - **제목:** 도구 설명에 표시할 텍스트를 입력합니다.
    - **대체 텍스트:** 접근성을 위해 표시될 대체 텍스트를 입력합니다.
    - **표제:** 이미지 아래에 표시되는 표제를 입력합니다.
    - **맞춤, 너비 및 간격:** 필요에 따라 이미지 레이아웃을 변경합니다.
  - **제목:** 헤드라인 제목의 간격을 변경합니다.
  - **단락:** 헤드라인 아래에 표시되는 텍스트의 간격을 변경합니다.
8. [일반 사항] 탭에서 간격, 맞춤 및 기타 표시 옵션을 수정할 수 있습니다.
9. [스타일] 탭에서는 텍스트 서식을 추가하고 배경색, 글꼴 및 테두리를 사용자가 정의할 수 있습니다.


## 기사


헤드라인, 단락 및 이미지 구성요소를 사용이 간편한 하나의 구성요소로 결합하는 문서를 사이트에 추가할 수 있습니다.










1. 편집할 페이지로 이동하고  가 **편집**으로 설정되어 있는지 확인합니다.
2. 페이지에 구성요소를 추가합니다.
3. 구성요소의 다른 영역을 눌러 텍스트를 추가합니다. 문서 헤드라인 및 헤드라인 아래의 부제목에 콘텐츠를 추가합니다. 그러면 부제목 아래에 문서 텍스트를 추가할 수 있습니다. 모든 텍스트에는 구성요소에 대한 기본 스타일 서식이 적용됩니다. Enter를 눌러 다른 텍스트 행을 추가합니다.
4. 어떤 텍스트 부분의 기본 서식을 변경하려면 서식을 지정할 텍스트를 선택하고 서식 도구모음에서 글꼴, 색상, 맞춤과 같은 옵션을 선택합니다.
5. 이 옵션으로 적용된 서식을 제거하려면 텍스트를 선택하고  을 누릅니다.

6. 구성요소 및 구성요소 모양을 편집하려면 메뉴 아이콘  을 누르고 **설정**을 선택합니다.
7. **구성요소**를 선택하여 표시되는 콘텐츠에 대한 세부정보를 설정합니다.

- **이미지:**

- 문서에 이미지를 사용하려면 **선택**을 누릅니다. 사이트 저장소의 이미지, 사용자에게 공유된 문서 폴더의 이미지를 선택하거나, 문서 폴더에 이미지를 업로드합니다. 사용할 이미지를 찾아 선택합니다. 디지털 자산이 표시되지 않으면  을 누르고 모음 필터를 **모두**로 변경합니다. 문서 폴더에서 이미지를 선택한 경우, 이를 사이트에 복사하는 대신 파일에 링크할 수 있습니다. 파일에 링크하려면 **사이트에 파일을 복사하는 대신 원본 파일의 참조 사용**을 선택합니다. 이 옵션을 선택하지 않으면 파일의 복사본이 사이트에 저장되고, 사이트에서 참조됩니다. 원본 파일에 링크하면 콘텐츠가 복제되지 않습니다. 링크를 통해 사이트 방문자는 파일에 대한 권한이 변경되거나 보기 권한이 제한되는 경우에도 콘텐츠를 볼 수 있습니다. **확인**을 누릅니다. 디지털 자산을 선택한 경우, 특정 표현을 선택할 수 있습니다. 표현을 선택하지 않으면 원래 크기가 사용됩니다. 사이트가 게시될 때 자산의 최신 버전이 게시되도록 하려면 **자산의 최신 버전 사용**을 선택합니다.




이미지를 편집하려면  을 누르고 다음 작업을 통해 이미지를 편집합니다.

- \* 이미지를 자르려면  **자르기**를 누릅니다. 자르기 도구 모음에서 미리 정의된 이미지 비율 중 하나를 선택하거나 이미지에서 자르기 핸들을 원하는 대로 끌어갑니다. 비율이 만족스러우면 자르기 도구 모음에서 **자르기**를 누릅니다.
- \* 이미지를 회전하거나 뒤집으려면  **회전**을 누릅니다. 회전 도구모음에서 사용자정의 회전 각도를 입력하거나 이미지를 왼쪽 또는 오른쪽으로 회전하는 단추를 사용하거나 이미지를 좌우 또는 상하로 뒤집을지 여부를 선택합니다.
- \* 이미지에 워터마크를 추가하려면  **워터마크**를 누릅니다. 워터마크 툴을 사용하여 필요에 따라 텍스트 크기, 스타일, 색상 및 불투명도를 변경하여 이미지에 텍스트를 추가합니다.
- \* 이미지 형식을 변경하려면  **옵션**을 누른 후 **형식** 드롭다운 목록에서 새 형식을 선택합니다.
- \* 배경 색상을 변경하려면  **옵션**을 누른 후 **배경 색상** 드롭다운 메뉴에서 옵션을 선택합니다.
- \* .jpg 또는 .webp(Google Chrome 브라우저에서 사용 가능)를 편집 중인 경우 보다 작은 파일 크기가 생성되도록 이미지 품질을 변경할 수 있습니다.  **옵션**을 누른 후 **품질** 상자에 새 백분율을 입력합니다.
- \* 변경사항을 실행 취소하거나 재실행하려면  또는  을 누릅니다. 모든 변경사항을 제거하려면 **재설정**을 누릅니다.
- \* 이미지 확대를 변경하려면 확대/축소 컨트롤을 사용합니다 (-  +).

- **제목:** 도구 설명에 표시할 텍스트를 입력합니다.
  - **대체 텍스트:** 접근성을 위해 표시될 대체 텍스트를 입력합니다.
  - **표제:** 이미지 아래에 표시되는 표제를 입력합니다.
  - **맞춤, 너비 및 간격:** 필요에 따라 이미지 레이아웃을 변경합니다.
  - **제목:** 문서 제목의 간격을 변경합니다.
  - **단락(부제목) 및 단락:** 문서 제목 아래에 표시되는 텍스트의 간격을 변경합니다.
8. [일반 사항] 탭에서 구성요소에 대한 간격, 맞춤 및 기타 표시 옵션을 수정할 수 있습니다.
  9. [스타일] 탭에서는 텍스트 서식을 추가하고 배경색, 글꼴 및 테두리를 사용자가 정의할 수 있습니다.


## 텍스트가 있는 이미지


단락 및 이미지 구성요소를 사용이 간편한 하나의 구성요소로 결합하는 이미지 및 텍스트 구성요소를 추가할 수 있습니다.


1. 편집할 페이지로 이동하고 가 **편집**으로 설정되어 있는지 확인합니다.
2. 페이지에 구성요소를 추가합니다.
3. 구성요소의 단락 영역을 눌러 텍스트를 추가합니다. 모든 텍스트에는 구성요소에 대한 기본 스타일 서식이 적용됩니다. Enter를 눌러 다른 텍스트 행을 추가합니다.
4. 어떤 텍스트 부분의 기본 서식을 변경하려면 서식을 지정할 텍스트를 선택하고 서식 도구모음에서 글꼴, 색상, 맞춤과 같은 옵션을 선택합니다.
5. 이 옵션으로 적용된 서식을 제거하려면 텍스트를 선택하고 을 누릅니다.
6. 구성요소 및 구성요소 모양을 편집하려면 메뉴 아이콘 을 누르고 **설정**을 선택합니다.
7. **구성요소**를 선택하여 표시되는 콘텐츠에 대한 세부정보를 설정합니다.









- **이미지:**

- 이미지를 사용하려면 **선택**을 누릅니다. 사이트 저장소의 이미지, 사용자에게 공유된 문서 폴더의 이미지를 선택하거나, 문서 폴더에 이미지를 업로드합니다. 사용할

이미지를 찾아 선택합니다. 디지털 자산이 표시되지 않으면 을 누르고 모음 필터를 모두로 변경합니다. 문서 폴더에서 이미지를 선택한 경우, 이를 사이트에 복사하는 대신 파일에 링크할 수 있습니다. 파일에 링크하려면 **사이트에 파일을 복사하는 대신 원본 파일의 참조 사용**을 선택합니다. 이 옵션을 선택하지 않으면 파일의 복사본이 사이트에 저장되고, 사이트에서 참조됩니다. 원본 파일에 링크하면 콘텐츠가 복제되지 않습니다. 링크를 통해 사이트 방문자는 파일에 대한 권한이 변경되거나 보기 권한이 제한되는 경우에도 콘텐츠를 볼 수 있습니다. **확인**을 누릅니다. 디지털 자산을 선택한 경우, 특정 표현을 선택할 수 있습니다. 표현을 선택하지 않으면 원래 크기가 사용됩니다. 사이트가 게시될 때 자산의 최신 버전이 게시되도록 하려면 **자산의 최신 버전 사용**을 선택합니다.

이미지를 편집하려면 을 누르고 다음 작업을 통해 이미지를 편집합니다.



- \* 이미지를 자르려면  **자르기**를 누릅니다. 자르기 도구 모음에서 미리 정의된 이미지 비율 중 하나를 선택하거나 이미지에서 자르기 핸들을 원하는 대로 끌어갑니다. 비율이 만족스러우면 자르기 도구 모음에서 **자르기**를 누릅니다.


- \* 이미지를 회전하거나 뒤집으려면  **회전**을 누릅니다. 회전 도구모음에서 사용자정의 회전 각도를 입력하거나 이미지를 왼쪽 또는 오른쪽으로 회전하는 단추를 사용하거나 이미지를 좌우 또는 상하로 뒤집을지 여부를 선택합니다.
  - \* 이미지에 워터마크를 추가하려면  **워터마크**를 누릅니다. 워터마크 툴을 사용하여 필요에 따라 텍스트 크기, 스타일, 색상 및 불투명도를 변경하여 이미지에 텍스트를 추가합니다.
  - \* 이미지 형식을 변경하려면  **옵션**을 누른 후 **형식** 드롭다운 목록에서 새 형식을 선택합니다.
  - \* 배경 색상을 변경하려면  **옵션**을 누른 후 **배경 색상** 드롭다운 메뉴에서 옵션을 선택합니다.
  - \* .jpg 또는 .webp(Google Chrome 브라우저에서 사용 가능)를 편집 중인 경우 보다 작은 파일 크기가 생성되도록 이미지 품질을 변경할 수 있습니다.  **옵션**을 누른 후 **품질** 상자에 새 백분율을 입력합니다.
  - \* 변경사항을 실행 취소하거나 재실행하려면  또는  을 누릅니다. 모든 변경사항을 제거하려면 **재설정**을 누릅니다.
  - \* 이미지 확대를 변경하려면 확대/축소 컨트롤을 사용합니다 ().
- **제목:** 도구 설명에 표시할 텍스트를 입력합니다.
  - **대체 텍스트:** 접근성을 위해 표시될 대체 텍스트를 입력합니다.
  - **표제:** 이미지 아래에 표시되는 표제를 입력합니다.
  - **맞춤, 너비 및 간격:** 필요에 따라 이미지 레이아웃을 변경합니다.
  - **단락:** 텍스트의 간격을 변경합니다.
8. [일반 사항] 탭에서 구성요소에 대한 간격, 맞춤 및 기타 표시 옵션을 수정할 수 있습니다.
  9. [스타일] 탭에서는 텍스트 서식을 추가하고 배경색, 글꼴 및 테두리를 사용자가 정의할 수 있습니다.

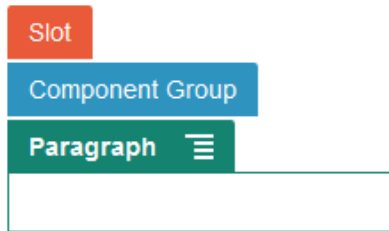
## 구성요소 그룹

하나 이상의 구성요소를 결합하여 구성요소 그룹을 생성한 후 이름을 지정하고 재사용할 수 있습니다.


구성요소 그룹을 저장할 때 구성요소 그룹은 사용자가 지정한 이름으로 사용자정의 구성요소로 저장되며 편집기의 사용자정의 구성요소 목록에 나타납니다.

1. 편집할 페이지로 이동하고  가 **편집**으로 설정되어 있는지 확인합니다.
2. 페이지에 구성요소 그룹을 추가합니다. 구성요소 그룹은  으로 식별됩니다.
3. 하나 이상의 구성요소를 사용자정의 구성요소 프레임으로 끌어 놓습니다.

- 슬롯 내 구성요소와 동일한 방법으로 구성요소 그룹 내의 구성요소 위치와 크기를 조정합니다.
- 구성요소 및 구성요소 모양을 편집하려면 메뉴 아이콘  을 누르고 **설정**을 선택합니다. 메뉴 아이콘 대신 구성요소 이름을 누르면 구성요소 그룹(또는 슬롯)의 메뉴 아이콘을 보고 선택할 수 있습니다.



구성요소 그룹의 [설정] 탭에서 구성요소 그룹의 위치, 배경 이미지 및 기타 전체 구성요소 그룹에 적용되는 설정을 지정할 수 있습니다.



- 구성요소 그룹에 변경사항을 저장할 준비가 되면 구성요소 그룹을 누르고 메뉴 아이콘  을 누르고 **저장**을 선택합니다.
  - 대화상자에 구성요소 그룹의 이름을 입력합니다. 문자, 숫자, 밑줄(\_) 및 하이픈(-)을 사용할 수 있습니다. 공백을 입력하면 자동으로 하이픈으로 바뀝니다.  
 새 구성요소 그룹인 경우 기존 사용자정의 구성요소 이름을 사용할 수 없습니다. 기존 구성요소 그룹을 페이지에 추가하고 구성요소 그룹을 수정한 후 변경사항을 저장하려는 경우, 이름을 제공하여 새 구성요소 그룹을 생성하거나 **기존 구성요소 그룹 겹쳐쓰기**를 선택하여 기존 구성요소 그룹을 변경사항으로 업데이트할 수 있습니다.
  - 저장**을 누릅니다.  
 구성요소 그룹이 지정된 이름의 사용자정의 구성요소로 저장됩니다. 내가 소유자이거나 누군가 나와 구성요소를 공유한 경우 구성요소 관리자의 고유 폴더와 편집기의 사용자정의 구성요소 목록에 나타납니다. 다른 사용자정의 구성요소와 마찬가지로 구성요소 그룹을 공유할 수 있습니다.

## 페이지에서 Cobrowse 사용

Oracle Cobrowse Cloud Service는 다른 사용자와 화면을 공유하거나 Cobrowsing 세션을 시작하는 데 사용할 수 있는 공동 작업 툴입니다. 예를 들어 고객이 주문을 진행하는 동안 담당자가 고객 화면을 볼 수 있도록 주문 양식에 이를 포함해야 할 수 있습니다.



이 기능을 사용하려면 먼저 사이트에 대해 사용으로 설정되어 있어야 합니다. 사용으로 설정할 수 있는 시작 프로그램 스크립트는 두 가지 유형이 있습니다. 하나는 사용자정의된 단추(시작 지점 2)를 사용하는 유형이고, 다른 하나는 기본 Cobrowse 단추(시작 지점 1)를 사용하는 유형입니다. 사이트에 대해 기능을 사용으로 설정할 때 사용할 시작 프로그램 유형을 결정하고 필요한 스크립트를 추가합니다. 자세한 내용은 [Cobrowse 통합 사용](#)을 참조하십시오.

사이트에 대해 Cobrowse가 사용으로 설정된 후에는 Cobrowsing을 허용하도록 모든 페이지를 구성할 수 있습니다.

- 편집할 페이지로 이동하고  가 **편집**으로 설정되어 있는지 확인합니다.
-  를 누릅니다. Cobrowse 섹션에서 체크박스를 선택합니다.
- 닫기**를 누릅니다.

시작 지점 1 스크립트가 사용으로 설정된 경우, 다음에 페이지를 표시할 때, 기본 Cobrowse 단추가 표시되거나 핫 키를 누른 다음(구성된 경우) 표시되는 것을 볼 수 있습니다.

시작 지점 2 스크립트가 사용으로 설정된 경우 사용자정의 단추를 페이지에 추가해야 합니다.

1. 가 **편집**으로 설정되었는지 확인합니다.
2. 페이지에 구성요소를 추가합니다. Cobrowse 시작 프로그램 구성요소는 구성요소의 통합 섹션에 나열됩니다.
3. 구성요소 및 구성요소 모양을 편집하려면 메뉴 아이콘 을 누르고 **설정**을 선택합니다. 단추 레이블, 모양, 크기 및 정렬을 조정할 수 있습니다. [스타일] 탭에서는 텍스트 서식을 추가하고 배경색, 글꼴 및 테두리를 사용자가 정의할 수 있습니다. 고급 스타일 지정을 위해서는 현재 사이트 템플릿의 테마 디자인 폴더에서 design.json 및 design.css 파일에 있는 스타일 클래스를 편집하거나 추가합니다. 스타일 클래스 접두어는 scs-cobrowse입니다.

사이트가 게시된 다음에는 사이트 방문자가 Cobrowse 단추를 사용해서 조직 담당자와 세션을 시작할 수 있습니다. 방문자가 Cobrowse 단추를 누르면 보안 세션 ID가 제공됩니다. 방문자는 전화 호출을 통해 ID를 Cobrowse 에이전트 콘솔에 액세스할 수 있는 조직 담당자에게 릴레이합니다. 에이전트는 콘솔을 사용해서 세션을 시작하고, 방문자 또는 에이전트가 이를 종료할 때까지 계속됩니다. 시작 지점 2 설정을 사용할 때 시작 프로그램 구성요소에는 cec-start-cobrowse라는 고정 ID가 사용됩니다. Cobrowse 콘솔에서 이 ID를 사용합니다.

### 사용법 참고

Cobrowse 인스턴트 모드(ICB)를 사용할 때 비디오 또는 포함된 iFrame은 iFrame 콘텐츠가 동일한 Cobrowse 사이트 ID를 사용해서 사용으로 설정되지 않은 한 페이지에 표시되지 않습니다. 따라서 일부 Oracle Content Management 구성요소가 ICB 모드의 Cobrowse 에이전트 콘솔에서 렌더링되지 않습니다. Cobrowse 고급 모드(ACB)를 사용해서 다음 구성요소를 렌더링합니다.

- 비디오
- Youtube
- 문서 관리자
- Facebook 좋아요
- Twitter 팔로우
- Twitter 공유
- Facebook 추천
- 대화

콘텐츠를 가져오기 위해 iFrame을 사용하는 사용자정의 구성요소는 또한 인스턴트 모드에서 렌더링되지 않습니다.

Oracle Cobrowse Cloud Service에 대한 자세한 내용은 *Cobrowse Deployment and Use Guide*의 [Cobrowse Overview](#)를 참조하십시오. 보안 사이트 또는 개발 중인 사이트에서 Cobrowse 사용에 대한 자세한 내용은 [Cobrowse 통합 사용](#)을 참조하십시오.



## Oracle Intelligent Advisor

Oracle Intelligent Advisor(이전의 Oracle Policy Automation)는 온라인 인터뷰 시나리오(예: 문제 해결 또는 서비스 적격성 평가를 위한 피드백) 구현에 사용됩니다. OPA는 자연어 Microsoft Word 및 Excel 문서에서 규칙을 캡처하고 해당 규칙에 대해 인터뷰라는 대화식 고객 서비스 경험을 구축하여 채널 전반에 걸쳐 권고사항을 제공합니다.

Intelligent Advisor 기능을 사용하려면 먼저 이를 구성하고 사용으로 설정해야 합니다. 서비스 관리자가 사용 중인 Intelligent Advisor 허브에 대한 호스트 이름, URL, 사용자 이름 및 비밀번호를 추가하는 등 서비스에 대한 기능을 사용으로 설정합니다. 두 서비스 간의 통합을 위해서는 SSO 사인온이 필요하므로 두 서비스가 동일한 ID 도메인에 있어야 합니다. *Oracle Content Management 통합 및 확장*의 Intelligent Advisor와 통합을 참조하십시오.

Intelligent Advisor 측에서 인터뷰를 생성하고 호스트 사이트에 저장해야 합니다. 또한 Intelligent Advisor 호스트에서 사용할 수 있도록 Oracle Content Management 서비스에 권한이 부여되어야 합니다.

Intelligent Advisor가 구성되고 사용으로 설정된 다음에는 사용자가 자신의 사이트 페이지에 Intelligent Advisor 구성요소를 추가할 수 있습니다.

1. 편집할 페이지로 이동하고 가 편집으로 설정되어 있는지 확인합니다.
2. 페이지에 구성요소를 추가합니다. [구성요소] 목록의 [통합] 섹션에 구성요소가 표시됩니다.
3. 구성요소 및 구성요소 모양을 편집하려면 메뉴 아이콘 을 누르고 설정을 선택합니다. 구성요소 레이블, 모양, 크기 및 맞춤을 조정할 수 있습니다. [스타일] 탭에서는 Intelligent Advisor 호스트에서 인터뷰와 연관된 기본 스타일을 사용합니다. 고급 스타일 지정을 위해서는 현재 사이트 템플릿의 테마 디자인 폴더에서 design.css 파일에 있는 스타일 클래스를 편집하거나 추가합니다. 스타일 클래스 접두어는 scs-opainterview-입니다.

사이트가 게시되면 Intelligent Advisor 구성요소에서 선택된 대화식 인터뷰가 사이트 방문자에게 표시됩니다. Intelligent Advisor에 대한 자세한 내용은 [Intelligent Advisor 설명서 라이브러리](#)를 참조하십시오.

## Oracle Visual Builder

Oracle Visual Builder는 호스트된 애플리케이션 개발 인프라 환경입니다. Oracle Cloud 내에서의 애플리케이션 개발, 공동 작업 및 배치를 위해 오픈 소스 표준 기반 솔루션이 제공됩니다.

### 초기 단계

Oracle Visual Builder를 사용하기 전에 사용으로 설정되고 구성되어야 합니다. 서비스 관리자가 앱 생성 및 저장 위치에 대한 호스트 이름을 추가하는 등 서비스에 대한 기능을 사용으로 설정합니다. *Integrating and Extending Oracle Content Management*의 Integrate with Oracle Visual Builder를 참조하십시오. 두 서비스 간의 통합을 위해서는 SSO가 필요하므로 두 서비스가 동일한 ID 도메인에 있어야 합니다.

Oracle Visual Builder에서 다음 작업이 완료되어야만 Oracle Content Management에서 이 기능을 사용할 수 있습니다.

- CORS(교차 출처 리소스 공유)가 Oracle Visual Cloud Service 사이트에서 사용으로 설정되어야 합니다.
- 앱이 생성된 후 포함될 수 있도록 제공되고 Oracle Content Management에서 사용되도록 구성되어야 합니다.



- 웹 애플리케이션이 생성된 후 iframe에 포함될 수 있도록 제공되어야 합니다. 사이트 SDK가 임포트된 후 웹 애플리케이션에서 참조되어야 합니다. "id"라는 페이지 URL 매개변수가 웹 애플리케이션에 추가되어야 합니다.

### Oracle Visual Builder 구성요소 생성

통합이 사용으로 설정되고 앱 및 웹 애플리케이션이 생성되어 사용 준비가 완료되면 사이트 페이지에 추가할 각 앱에 대한 새 구성요소를 생성해야 합니다.

1. Oracle Visual Builder에서 게시된 웹 애플리케이션의 URL을 가져옵니다. 웹 애플리케이션이 포함된 라이브 프로젝트를 누르고 웹 애플리케이션을 누릅니다. 주소 표시줄에서 URL을 복사합니다.
2. Oracle Content Management에서 **개발자**를 누르고 **모든 구성요소 보기**를 누릅니다. 등록된 원격 구성요소 및 레이아웃이 표시됩니다.
3. **생성**을 누르고 **Visual Builder 구성요소 생성**을 선택합니다.
4. 구성요소의 이름을 입력합니다. 다른 구성요소 또는 레이아웃에 사용되는 이름은 사용할 수 없습니다.



문자, 숫자, 밑줄(\_) 및 하이픈(-)을 사용할 수 있습니다. 공백을 입력하면 자동으로 밑줄로 바꿉니다.

템플릿, 테마, 구성요소, 사이트 또는 사이트 페이지에 다음 이름을 사용하지 마십시오: authsite, content, pages, scstemplate\_\*, \_comps, \_components, \_compsdelivery, \_idcservice, \_sitescloud, \_sitesclouddelivery, \_themes, \_themesdelivery. 사이트 페이지에는 다음 이름을 사용할 수 있지만 템플릿, 테마, 구성요소 또는 사이트에는 사용하지 마십시오: documents, sites.

5. 선택적으로 구성요소에 대한 설명을 입력합니다.
6. 라이브 웹 애플리케이션 또는 라이브 클래식 애플리케이션에 URL을 붙여 넣습니다.
7. **생성**을 누릅니다. 구성요소가 생성되면 구성요소 목록에 이름이 나타납니다. 목록에서 구성요소 이름을 눌러 구성요소 또는 레이아웃을 구성하는 폴더 및 파일을 탐색할 수 있습니다.
8. 구성요소에 지정된 기본 아이콘 이외의 아이콘을 선택하려면 다음을 수행합니다.
  - a. 목록에서 구성요소를 선택합니다.
  - b. **속성**을 누릅니다.
  - c. **구성요소 로고** 탭을 누릅니다.
  - d. 로고 갤러리에서 로고를 누른 다음 **완료**를 누릅니다.

### 사이트 페이지에 구성요소 추가

이제 사이트 페이지 중 하나에 앱의 구성요소를 추가할 수 있습니다. 구성요소는 공용 또는 보안 페이지에 추가할 수 있습니다.

1. 편집할 페이지로 이동하고 가 **편집**으로 설정되어 있는지 확인합니다.
2. 페이지에 구성요소를 추가합니다. [사용자정의] 섹션에 구성요소가 표시됩니다.
3. 구성요소 및 구성요소 모양을 편집하려면 메뉴 아이콘 을 누르고 **설정**을 선택합니다. 구성요소 레이블, 모양, 크기 및 맞춤을 조정할 수 있습니다. [스타일] 탭에서는 Oracle Visual Builder 호스트에서 구성요소와 연관된 기본 스타일을 사용합니다. 고급 스타일 지정은 위해서는 현재 사이트 템플릿의 테마 디자인

폴더에서 `design.css` 파일에 있는 스타일 클래스를 편집하거나 추가합니다. 스타일 클래스 접두어는 `scs-component`입니다.

사이트를 편집할 때 미리보기 모드로 구성요소를 볼 수 있습니다. 사이트가 게시되면 페이지의 `iFrame`에서 실행되는 해당 구성요소를 사용하도록 선택된 Oracle Visual Builder 앱이 사이트 방문자에게 표시됩니다.

# III부

## 사이트 게시 및 관리

이 부분에서는 사이트 작업, 보안, 향상 및 게시 방법을 자세히 설명합니다. 다음 장이 포함됩니다.

- [사이트 관리](#)
- [사이트 게시](#)
- [사이트 보안](#)
- [다국어 사이트 작업](#)
- [사이트 재지정 또는 URL 매핑 사용](#)
- [사이트 성능 향상](#)

# 11

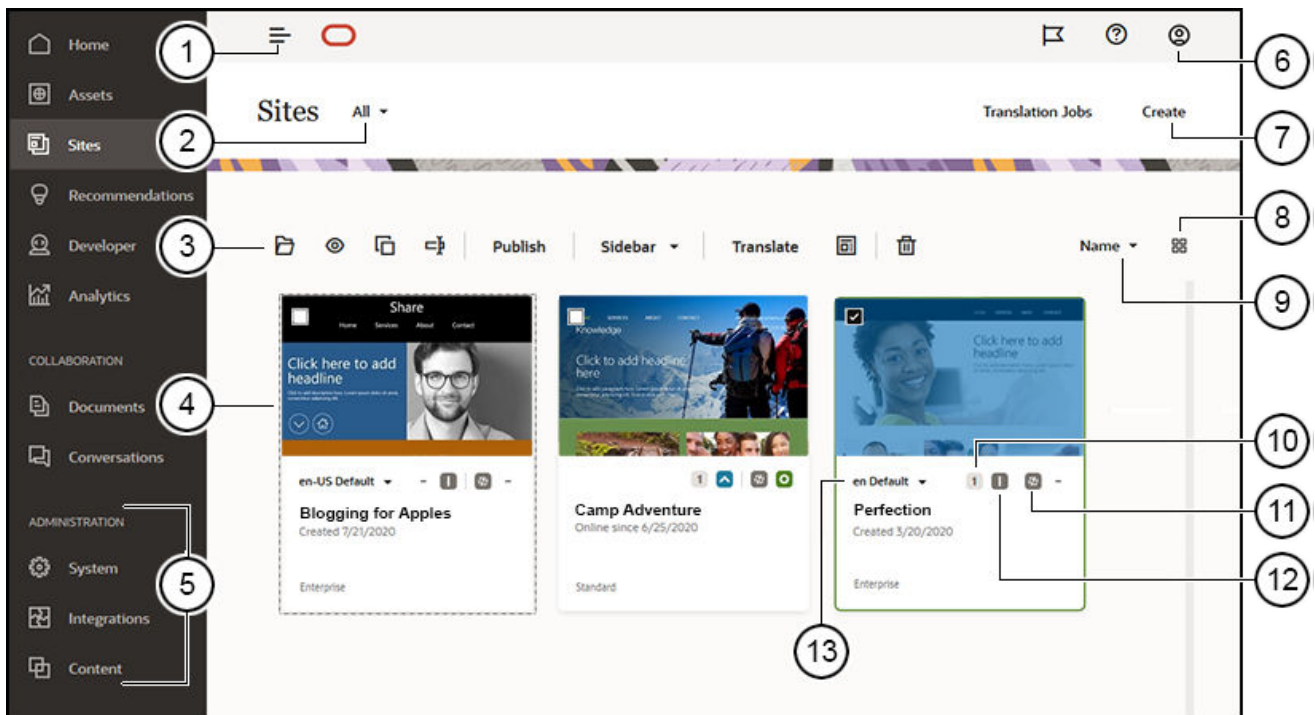
## 사이트 관리

사이트 레이아웃 또는 콘텐츠를 변경하려면 편집기에서 업데이트를 생성하고 엽니다. 사이트 자체와 사이트 속성을 생성하고 관리하려면 사이트 관리자의 옵션을 사용합니다.

- [사이트 페이지 알아두기](#)
- [사이트 및 사이트 설정 관리](#)
- [사이트를 온라인으로 전환 또는 오프라인으로 설정](#)
- [사이트 설명, 로고 또는 포함 속성 변경](#)
- [검색 엔진 속성 설정](#)
- [사이트 설정 사용자정의](#)
- [Cobrowse 통합 사용](#)
- [Analytics 추적 추가](#)

## 사이트 페이지 알아두기

[사이트] 페이지는 웹 사이트 작업의 관문입니다.













콜아웃	설명
1	<b>패널 토글</b> 은 탐색 메뉴를 표시하고 숨깁니다.
2	<b>필터 메뉴</b> 로는 사이트 목록을 필터링하여 액세스 권한이 있는 모든 사이트를 표시하거나 국한된 사이트를 표시할 수 있습니다. 사이트 제어가 사용으로 설정된 경우 보류 중인 사이트 요청을 볼 수 있습니다.
3	<p>목록에서 사이트를 선택한 후 <b>작업 모음</b>을 사용할 수 있습니다. 작업 모음의 옵션을 사용하여 사이트 열기, 사이트 보기, 사이트 상태 변경, 사이트 이름 바꾸기 및 기타 작업을 수행할 수 있습니다.</p> <p>표시되는 옵션은 선택한 사이트에 지정된 롤에 따라 다릅니다. 예를 들어, 사이트를 직접 생성한 경우 해당 사이트의 관리자 롤이 지정되므로 나열된 모든 작업을 수행할 수 있습니다. 그러나 누군가 나와 사이트를 공유했으며 내게 열람자 롤을 지정한 경우 사이트 속성을 볼 수 있지만 어떤 내용도 변경할 수 없습니다.</p> <p>작업 모음 옵션은 현재 사이트 상태에 따라서도 달라집니다. <b>이름 바꾸기</b> 및 <b>삭제</b> 옵션은 사이트가 오프라인일 때만 사용할 수 있습니다.</p> <p><b>팁:</b> 바로가기를 찾습니까? 사이트를 마우스 오른쪽 단추로 눌러 컨텍스트 메뉴를 열고 옵션을 선택합니다.</p>
4	<b>사이트 목록</b> 은 직접 소유했거나 나와 공유된 사이트를 모두 보여줍니다. 목록에는 사이트 이름, 업데이트 수, 보안 사이트 여부, 사이트 상태(온라인, 오프라인 또는 보류 중인 업데이트)가 포함됩니다.
5	<b>관리 메뉴</b> 는 콘텐츠 관리자로 로그인한 경우 표시됩니다. 여기의 옵션으로 서비스 설정을 변경하고 통합을 설정하며 저장소 및 기타 자산 관련 기능을 설정할 수 있습니다.
6	<b>사용자 메뉴</b> 에는 환경설정을 지정하고, 피드백을 제공하고, 도움말에 액세스하고, 사인아웃하는 옵션이 있습니다. 사용자 사진을 눌러 메뉴 옵션을 표시합니다.
7	<b>생성</b> 을 눌러 웹 사이트 생성 과정을 시작합니다. 사이트를 생성하려면 먼저 관리자가 사이트 생성을 사용으로 설정하고 템플릿을 하나 이상 제공해야 합니다. 사이트 페이지에 <b>생성</b> 옵션이 표시되지 않거나 템플릿 페이지에 템플릿이 표시되지 않으면 서비스 관리자에게 문의하십시오.
8	<b>뷰 아이콘</b> 을 눌러 사이트를 그리드, 목록 또는 테이블 뷰로 표시할지 선택합니다.
9	<b>정렬 옵션</b> 을 사용하여 사이트의 표시 순서를 변경합니다.
10	<b>업데이트 수</b> 를 눌러 사이트의 기존 업데이트를 편집하거나 사이트에 대한 새 업데이트를 생성합니다. 사이트 업데이트가 없을 경우 사이트를 열고 사이트 작성기를 편집 모드로 변경하여 새 업데이트를 생성할 수 있습니다.
11	<b>오프라인, 온라인 또는 재게시</b> 아이콘을 눌러 사이트 상태를 변경하거나 커밋되었지만 실시간 사이트에 게시되지 않은 변경사항을 게시합니다.
12	<b>보안</b> 아이콘은 사이트에 로그인이 필요한지 여부를 보여줍니다.
13	<b>언어 목록</b> 을 사용하여 사이트의 연관된 지역화 정책에서 지정된 언어를 확인합니다. 지역화된 버전을 미리 보려면 언어를 선택하고 사이트를 엽니다.

## 사이트 및 사이트 설정 관리

사이트 관리에는 사이트 생성 및 편집과 사이트에 사용된 콘텐츠 관리가 포함됩니다. *사이트 작성기*를 사용하여 사이트 페이지와 페이지 콘텐츠를 생성, 복사 및 삭제할 수 있습니다. *폴더 및 파일 관리자*를 사용하여 전체 사이트를 생성, 복사, 공유, 삭제할 수 있습니다.


[사이트] 페이지 및 사이트 작성기에서 메뉴 모음 또는 마우스 오른쪽 단추 누르기 메뉴의 옵션을 사용하여 다음 작업을 수행할 수 있습니다.

작업	설명
사이트 뷰 필터링	<p>사이트 목록을 필터링하려면 필터 메뉴에서 다음 옵션 중 하나를 선택합니다.</p> <ul style="list-style-type: none"> <li>모두 - 액세스 권한이 있는 모든 사이트를 보여줍니다. 사이트 관리자인 경우 사이트 제어가 사용으로 설정되어 있으면 환경의 모든 사이트에 액세스할 수 있습니다.</li> <li>내가 소유한 항목 - 내가 소유한 모든 사이트를 보여줍니다.</li> <li>나와 공유된 항목 - 나와 공유된 모든 사이트를 보여줍니다.</li> <li>온라인 - 온라인(라이브) 상태의 사이트를 보여줍니다.</li> <li>오프라인 - 오프라인 상태의 사이트를 보여줍니다.</li> <li>요청(사이트 제어가 사용으로 설정된 경우에만 사용 가능) - 보류 중인 사이트 요청을 보여줍니다. 사이트 관리자인 경우 시스템의 모든 사이트 요청이 표시됩니다. 그렇지 않은 경우 제출한 사이트 요청만 표시됩니다.</li> <li>휴지통 - 내가 소유한 삭제된 사이트를 보여줍니다. 사이트 관리자인 경우 제어가 사용으로 설정되어 있으면 모든 삭제된 사이트를 보여줍니다.</li> </ul>
사이트 생성	<p>사이트를 생성하려면 <b>생성</b>을 누릅니다. 사이트를 생성할 때 템플릿으로 시작합니다. 템플릿은 사이트 코드 프레임워크, 샘플 페이지와 콘텐츠로 구성된 기본 사이트, 스타일이 적용된 테마, 리소스(예: 이미지)는 물론 사용자정의 구성요소까지 사이트를 시작할 때 필요한 모든 것을 포함합니다. <b>사이트 생성</b>을 참조하십시오.</p> <p><b>사이트</b> 페이지의 필터 메뉴를 통해 사이트 제어가 사용으로 설정되어 있는지 여부를 확인할 수 있습니다. <b>요청</b> 옵션이 표시되면 사이트 제어가 사용으로 설정된 것입니다. <b>사이트 제어 이해</b>를 참조하십시오.</p>
사이트 요청 확인 및 관리	<p>보류 중인 사이트 요청(사이트 제어가 사용으로 설정된 경우에만 사용 가능)을 보려면 필터 메뉴에서 <b>요청</b>을 선택합니다. 사이트 관리자인 경우 시스템의 모든 사이트 요청이 표시됩니다. 그렇지 않은 경우 제출한 사이트 요청만 표시됩니다. <b>사이트 요청 관리</b>를 참조하십시오.</p>
사이트 편집	<p>사이트를 편집하려면 마우스 오른쪽 단추 누르기 메뉴에서 <b>열기</b>를 선택하거나 작업 모음에서  을 누릅니다. <b>사이트 편집</b>을 참조하십시오.</p>
사이트 토론 또는 주석 추가	<p>사이트 작성기에서 사이트를 토론하고 주석을 추가할 수 있습니다. 사이트를 선택하고 마우스 오른쪽 단추 누르기 메뉴에서 <b>열기</b>를 선택하거나 작업 모음에서  을 누릅니다. 사이트를 토론하려면  을 눌러서 대화 패널을 엽니다. 주석을 추가하려면  을 누릅니다. <b>사이트 작성기 페이지 알아두기</b>를 참조하십시오.</p>
사이트 보기	<p>라이브(온라인) 상태일 때 사이트의 모양을 보려면 사이트를 선택하고 마우스 오른쪽 단추 누르기 메뉴에서 <b>보기</b>를 선택하거나 작업 모음에서  을 누릅니다.</p>
사이트 게시	<p>사이트를 게시하려면 선택 후 마우스 오른쪽 단추 누르기 메뉴 또는 작업 모음에서 <b>게시</b>를 선택합니다. <b>사이트 변경사항 게시</b>를 참조하십시오.</p>
재게시	<p>사이트를 이미 게시했지만 연관된 항목 또는 정책이 변경된 경우(예: 연관된 지역화 정책이 보다 많은 언어를 포함하도록 업데이트된 경우) 사이트 <b>재게시</b>를 통해 해당 변경사항을 새로고침할 수 있습니다. 사이트를 선택하고 마우스 오른쪽 단추 누르기 메뉴 또는 작업 모음에서 <b>재게시</b>를 선택합니다.</p>
게시 취소	<p>사이트를 이미 게시한 경우 Oracle Cloud의 호스팅 위치에서 사이트 파일을 제거하려면 사이트를 선택하고 마우스 오른쪽 단추 누르기 메뉴 또는 작업 모음에서 <b>게시 취소</b>를 선택합니다. 게시를 취소하려면 사이트가 오프라인 상태여야 합니다.</p>

작업	설명
사이트를 온라인으로 전환 또는 오프라인으로 설정	<p>사이트를 온라인으로 전환하려면 먼저 사이트가 게시되어 있어야 합니다. 사이트를 온라인으로 전환하거나 오프라인으로 설정하려면 사이트를 선택하고 마우스 오른쪽 단추 누르기 메뉴에서 <b>온라인으로 전환</b> 또는 <b>오프라인으로 전환</b>을 선택합니다. 사이트 타일에서  을 눌러 사이트를 온라인으로 전환하거나, 사이트 타일에서  을 눌러 사이트를 오프라인으로 전환할 수도 있습니다. 사이트가 <i>온라인</i>일 때 사용자는 지정된 주소(URL)에서 웹 브라우저로 사이트를 볼 수 있습니다. 사이트가 <i>오프라인</i> 상태일 때는 공개적으로 사이트가 제공되지 않습니다. Oracle Content Management에서만 사이트를 볼 수 있습니다. <a href="#">사이트를 온라인으로 전환 또는 오프라인으로 설정</a>을 참조하십시오.</p>
사이트 이름 바꾸기	<p>사이트 이름을 바꾸려면 마우스 오른쪽 단추로 누르고 마우스 오른쪽 단추 누르기 메뉴에서 <b>이름 바꾸기</b>를 선택하거나 작업 모음에서  을 누릅니다. 동일 서버의 기존 사이트 이름과 다르게 사이트의 이름을 입력합니다.</p> <p>온라인 상태의 사이트는 이름을 바꿀 수 없습니다. 사이트를 오프라인으로 가져오려면 사이트 소유자이거나 사이트의 관리자 롤을 보유해야 합니다.</p>
사이트 복사	<p>사이트를 복사하여 사이트를 구축할 때 유리하게 출발할 수 있습니다. 테마, 모든 미해결 업데이트, 페이지, 페이지 콘텐츠 및 기타 모든 이미지 자산을 비롯한 원본 사이트의 모든 내용이 제공한 새 이름으로 새 사이트에 복사됩니다. 새 사이트는 오프라인이고 편집할 준비가 되어 있습니다.</p> <p>주: 제어가 사용으로 설정된 경우 사본이 생성되기 전 사이트가 승인되어야 합니다. 여러 저장소의 콘텐츠를 사용하는 사이트를 복사하는 경우 OCE 툴킷을 사용하여 작업을 수행해야 합니다. <a href="#">OCE Toolkit으로 개발</a> 및 <a href="#">cec 명령행 유틸리티 사용</a>을 참조하십시오.</p> <p>복사할 사이트를 선택하고 마우스 오른쪽 단추 누르기 메뉴에서 <b>복사</b>를 선택하거나 작업 모음에서  을 누릅니다. <a href="#">사이트 복사</a>를 참조하십시오.</p>
사이트에서 템플릿 생성	<p>다른 사이트의 시작점으로 사용할 사이트가 있는 경우 해당 사이트에서 템플릿을 생성할 수 있습니다. 사이트를 선택하고 마우스 오른쪽 단추 누르기 메뉴에서 <b>템플릿 생성</b>을 선택하거나 작업 모음에서  을 누릅니다. <a href="#">사이트에서 템플릿 생성</a>을 참조하십시오.</p>

 주:

여러 저장소의 콘텐츠를 사용하는 사이트에서 템플릿을 생성하는 경우 OCE 툴킷을 사용하여 작업을 수행해야 합니다. [OCE Toolkit으로 개발](#) 및 [cec 명령행 유틸리티 사용](#)을 참조하십시오.

작업	설명
사이트 삭제 또는 복원	<p>적절한 권한이 있으면 사이트와 콘텐츠를 삭제할 수 있습니다. 사이트를 삭제하면 모든 미해결 업데이트, 사이트 페이지, 페이지 콘텐츠 및 내가 페이지에 추가한 이미지 자산을 비롯한 사이트 폴더의 모든 내용이 휴지통에 놓입니다.</p> <p>사이트를 직접 생성했거나(사이트 소유자) 누군가 나와 사이트를 공유하고 내게 제공자 또는 관리자 롤을 부여한 경우 사이트를 삭제하거나 복원할 수 있습니다.</p> <p>온라인 사이트는 삭제할 수 없습니다. 사이트를 오프라인으로 가져오려면 사이트 소유자이거나 사이트의 관리자 롤을 보유해야 합니다.</p> <p>또한 게시된 사이트를 삭제할 수 없습니다. 먼저 콘텐츠를 게시 취소해야 합니다. 사이트에 여러 저장소의 콘텐츠가 포함된 경우 각 저장소에서 연관된 콘텐츠를 게시 취소해야 합니다.</p> <p>사이트를 삭제하려면 선택 후 마우스 오른쪽 단추 누르기 메뉴에서 <b>삭제</b>를 선택하거나 작업 모음에서  을 누릅니다. 사이트와 모든 사이트 업데이트를 휴지통으로 이동할지 묻는 메시지가 표시됩니다. 다음 이벤트 전까지 삭제된 사이트가 휴지통에 보존됩니다.</p> <ul style="list-style-type: none"> <li>• 사이트를 복원합니다.</li> <li>• 사이트를 영구적으로 삭제합니다.</li> <li>• 휴지통 할당량에 도달합니다.</li> <li>• 서비스 관리자가 설정한 간격에 따라 자동으로 휴지통이 비워집니다. 기본값은 90일입니다.</li> </ul> <p>사이트를 복원하려면 [사이트] 메뉴에서 <b>휴지통</b>을 선택합니다. 목록에서 사이트를 선택하고 <b>복원</b>을 선택합니다.</p>
사이트에 멤버 추가	<p>관리자가 공유를 사용으로 설정한 경우 다른 Oracle Content Management 사용자와 사이트를 공유할 수 있으며 해당 사용자가 부여 받은 권한에 따라 Oracle Content Management의 사이트를 확인, 수정 또는 관리하도록 허용할 수 있습니다.</p> <p>서비스에 액세스할 수 있는 누구나 멤버로 간주됩니다. 사이트를 공유하는 경우 멤버가 사이트로 수행할 수 있는 작업과 없는 작업을 정의하는 롤을 지정합니다. 멤버는 지정된 롤에 따라서만 사이트를 사용할 수 있습니다.</p> <p>공유할 사이트를 선택하고 마우스 오른쪽 단추 누르기 메뉴에서 <b>멤버</b>를 선택하거나 작업 모음에서 <b>사이드바</b>를 누르고 <b>멤버</b>를 선택하여 멤버 사이드바를 엽니다. 멤버 사이드바가 열리면 <b>멤버 추가</b>를 누릅니다.</p> <p>사용자 이름 또는 전자메일 주소를 하나 이상 입력하고 다음 롤 중 하나를 지정합니다.</p> <ul style="list-style-type: none"> <li>• <b>열람자</b>: 열람자는 편집기에서 사이트를 볼 수 있지만 어떤 내용도 변경할 수 없습니다.</li> <li>• <b>다운로더</b>: 사이트의 경우 다운로더 롤은 열람자 롤과 동일한 권한을 제공하며 사용자는 사이트에서 새 템플릿을 생성할 수 있습니다.</li> <li>• <b>제공자</b>: 열람자와 동일한 권한 외에, 사이트를 편집하고 사이트 페이지를 삭제하며 사이트(오프라인일 경우)를 삭제할 수 있습니다.</li> <li>• <b>관리자</b>: 제공자와 동일한 권한 외에, 사용자를 추가하고 사용자 롤을 지정하며 온라인 사이트에 변경사항을 게시하고 사이트를 온라인 및 오프라인으로 전환할 수 있습니다. 사이트 생성자(소유자)에게는 자동으로 관리자 롤이 지정됩니다.</li> </ul>

 **주:**

여러 저장소에 액세스 권한이 있는 사이트를 공유하는 경우 기본 저장소만 공유됩니다. 추가 저장소는 개별적으로 공유되어야 합니다.




작업	설명
사이트 속성(설명, 로고 등) 변경	사이트 설명을 변경하고, 로고를 추가하고, 사이트 포함을 허용하고, vanity URL을 추가하고, 사이트에 대한 유용한 정보(사이트 URL, 사이트 소유자 및 기타 세부정보 등)를 확인하려면 사이트를 선택하고 마우스 오른쪽 단추 누르기 메뉴에서 속성을 선택하거나 작업 모음에서 사이드바를 누르고 속성을 선택합니다. <a href="#">사이트 설명, 로고 또는 포함 속성 변경</a> 을 참조하십시오.
사이트 번역	연관된 저장소가 허용하는 경우 사이트 파일을 익스포트하고 번역한 다음 번역된 파일을 임포트하여 엔터프라이즈 사이트를 다중 언어로 번역할 수 있습니다. <a href="#">사이트 번역</a> 을 참조하십시오.
검색 엔진 속성 설정	사이트 작성기에서 검색 엔진이 사이트 내용을 쉽게 식별하도록 키워드와 텍스트를 제공할 수 있습니다. <a href="#">검색 엔진 속성 설정</a> 을 참조하십시오.
사이트 설정 (즐거찾기 아이콘, 컨트롤러 파일 등) 사용자정의	사이트 작성기에서 여러 브라우저 및 플랫폼에 사용되는 사이트 아이콘을 지정하거나 링크 동작을 처리하는 컨트롤러 파일을 추가할 수 있습니다. <a href="#">사이트 설정 사용자정의</a> 를 참조하십시오.
Cobrowsing 사용	사이트 작성기에서 다른 사용자와 화면을 공유하거나 Cobrowsing 세션을 시작하는 데 사용할 수 있는 공동 작업 툴인 Oracle Cobrowse Cloud Service를 사용하여 설정할 수 있습니다. 예를 들어 고객이 주문을 진행하는 동안 담당자가 고객 화면을 볼 수 있도록 주문 양식에 이를 포함해야 할 수 있습니다. <a href="#">Cobrowse 통합 사용</a> 을 참조하십시오.
사이트에 Analytics 추적 추가	사이트 작성기에서 Google, Adobe 또는 Oracle Infinity와 같은 외부 분석 제공자와의 통합을 보다 간편하게 수행할 수 있도록 Web Analytics 추적을 위해 사이트에 JavaScript 추적 코드의 코드 조각을 추가할 수 있습니다. <a href="#">Analytics 추적 추가</a> 를 참조하십시오.
라이브 사이트 보안	사이트를 게시하여 온라인에서 사용할 수 있도록 설정하면 모두에게 공개됩니다. 그러나 사이트 소유자이거나 관리자 롤을 보유한 경우 등록된 사용자 또는 국한된 사용자로 라이브 사이트 사용을 제한할 수 있습니다. 사이트의 상태를 변경하려면 사이트 소유자이거나 관리자 롤을 보유해야 합니다.  보안할 사이트를 선택하고 마우스 오른쪽 단추 누르기 메뉴에서 속성을 선택하거나 작업 모음에서 사이드바를 누르고 속성을 선택한 다음 <a href="#">사이트 보안</a> 탭을 누릅니다. <a href="#">사이트 보안 변경</a> 을 참조하십시오.
사이트 재지정 추가	사이트 URL이 변경되면 재지정은 한 URL(소스)을 다른 URL(대상)로 전달합니다. 그러면 사용자 책갈피 및 검색 엔진 순위를 보존할 수 있습니다. <a href="#">사이트 재지정 추가</a> 를 참조하십시오.

## 사이트 설명, 로고 또는 포함 속성 변경

사이트를 생성하거나 업데이트할 때 대부분의 사이트 속성이 설정됩니다. 속성을 확인하면 사이트 URL, 사이트 소유자 및 기타 세부정보를 비롯하여 사이트에 대한 유용한 정보를 얻을 수 있습니다.

사이트 설명과 로고를 변경하고 vanity 사이트를 추가하고 내장 가능 사이트를 허용하려면 다음과 같이 하십시오.

1. 사이트를 선택하고 마우스 오른쪽 단추 누르기 메뉴에서 속성을 선택하거나 작업 모음에서 사이드바를 누르고 속성을 선택합니다.
2. 선택적 사이트 설명을 변경하려면 [사이트 속성](#) 탭을 누르고 설명을 입력하거나 변경합니다.
3. 사이트 URL을 직접 변경할 수 없지만, 사이트 URL을 선택하고 복사하여 문서, 프레젠테이션, 전자메일에 붙여넣으면 사이트에 액세스할 수 있습니다.

- 사이트가 온라인이면  을 눌러 온라인 사이트로 직접 이동합니다.
4. 사이트가 내장된 사이트로 사용되도록 허용하려면 **내장 가능 사이트** 옆에 있는 **예를** 토글합니다.
  5. 사이트 이미지를 변경하려면 **사이트 로고** 탭을 누르고 **변경**을 누릅니다. 사용할 이미지를 찾아 선택합니다. 이미지는 .png, .jpeg, .jpg 파일이어야 하고 4:3 (직사각형) 비율이어야 합니다. 최적 크기는 300x225 픽셀입니다. 더 작은 이미지는 왜곡될 수 있고, 더 큰 이미지는 성능에 영향을 미칠 수 있습니다.
  6. 마쳤으면 **닫기**를 누릅니다.

## 검색 엔진 속성 설정

검색 엔진이 사이트 내용을 식별할 수 있도록 키워드와 텍스트를 제공할 수 있습니다.

사이트 레벨과 페이지 레벨에서 검색 엔진 최적화(SEO) 설정을 정의할 수 있습니다. 아래 표에 설명된 대로 사이트 레벨 설정은 개별 페이지의 유사 설정을 보강하거나 무효화합니다.



옵션	사이트 레벨	페이지 레벨
설명 또는 페이지 설명	사이트 자체에 포함되지 않는 일반적인 사이트 정보를 제공합니다. 사이트의 각 페이지에 사이트 설명이 포함됩니다. 이 설명은 홈 페이지의 <b>페이지 설정</b> 을 사용하여 설정된 값이 없는 경우에만 홈 페이지에 대한 페이지 레벨 설명으로도 사용됩니다.	페이지 자체에 포함되지 않는 일반적인 페이지 정보를 제공합니다. 사이트의 모든 페이지에 포함된 사이트 설명에 페이지 설명이 추가됩니다.
키워드	사이트의 모든 페이지에 적용할 용어나 개념을 식별합니다. 이 값은 개별 페이지에 지정된 키워드에 추가됩니다.	개별 페이지에 적용할 용어나 개념을 식별합니다. 페이지 키워드는 페이지 텍스트에 없거나 이미지에 나타나는 용어나 개념을 식별하는 데 유용합니다.
머리글 또는 페이지 머리글	사이트에 분석/추적을 위한 스크립팅 또는 태그를 추가합니다. 사이트의 각 페이지에 사이트 머리글 내용이 포함됩니다.	사이트에 분석/추적을 위한 스크립팅 또는 태그를 추가합니다. 각 페이지에 포함된 사이트 머리글 내용에 페이지 머리글 내용이 추가됩니다.
바닥글 또는 페이지 바닥글	사이트에 분석/추적을 위한 스크립팅 또는 태그를 추가합니다. 사이트의 각 페이지에 사이트 바닥글 내용이 포함됩니다.	사이트에 분석/추적을 위한 스크립팅 또는 태그를 추가합니다. 각 페이지에 포함된 사이트 바닥글 내용에 페이지 바닥글 내용이 추가됩니다.

사이트 레벨에서 다음 옵션을 선택하는 경우 설정은 모든 페이지에 적용되고 개별 페이지의 설정을 무효화합니다. 사이트 레벨에서 옵션을 선택하지 않을 경우 개별적으로 옵션을 지정하는 페이지에만 옵션이 사용됩니다.

옵션	사이트 레벨	페이지 레벨
검색 엔진에서 숨기기	선택하는 경우 검색 엔진이 사이트 내 페이지의 콘텐츠를 인덱스화하지 않도록 모든 페이지에 NOINDEX 메타 태그가 추가됩니다. 이 경우 사이트와 모든 페이지가 웹 검색 결과에 노출되지 않습니다.	선택하는 경우 검색 엔진이 페이지의 콘텐츠를 인덱스화하지 않도록 현재 페이지에 NOINDEX 메타 태그가 추가됩니다. 이 경우 개별 페이지가 웹 검색 결과에 노출되지 않습니다.

옵션	사이트 레벨	페이지 레벨
검색 엔진에서 페이지 링크 숨기기	선택하는 경우 검색 엔진이 사이트 내 페이지의 링크를 따라가서 대상을 인덱스화하지 않도록 모든 페이지에 NOFOLLOW 메타 태그가 추가됩니다.	선택하는 경우 검색 엔진이 페이지의 링크를 따라가서 대상을 인덱스화하지 않도록 현재 페이지에 NOFOLLOW 메타 태그가 추가됩니다.
검색 엔진에서 페이지 설명 숨기기	선택하는 경우 검색 엔진이 검색 결과에서 페이지 다음에 설명(위에서 지정됨)을 포함하지 않도록 모든 페이지에 NOSNIPPET 메타 태그가 추가됩니다.	선택하는 경우 검색 엔진이 검색 결과에서 페이지 다음에 페이지 설명(위에서 지정됨)을 포함하지 않도록 현재 페이지에 NOSNIPPET 메타 태그가 추가됩니다.

검색 엔진 최적화(SEO) 설정을 변경하려면 다음과 같이 하십시오.

1. 편집을 위해 사이트를 엽니다.
2. 사이드바에서  을 누르고  SEO를 누릅니다.
3. 사이트에 대한 선택적 설명을 제공합니다. 사이트의 각 페이지에 사이트 설명이 포함됩니다.  
  
이 설명은 홈 페이지의 **페이지 설정**을 사용하여 설정된 값이 없는 경우에만 홈 페이지에 대한 페이지 레벨 설명으로도 사용됩니다.
4. 선택적으로, 검색 엔진이 사이트 내용을 식별할 수 있도록 콤마로 구분된 키워드를 지정합니다.  
  
사이트 키워드는 사이트의 모든 페이지에 적용할 용어나 개념을 식별합니다. 이 값은 **페이지 설정**을 사용하여 개별 페이지에 지정된 키워드에 추가됩니다.
5. 선택적으로, 사이트에 분석/추적을 위한 머리글 스크립팅 또는 태그를 추가합니다. 사이트의 각 페이지에 머리글 내용이 포함됩니다. 머리글에 사용할 코드가 제대로 작동하는지, 사이트에 보안 위험을 일으키지 않는지 검증합니다.
6. 선택적으로, 사이트에 분석/추적을 위한 바닥글 스크립팅 또는 태그를 추가합니다. 사이트의 각 페이지에 바닥글 내용이 포함됩니다. 바닥글에 사용할 코드가 제대로 작동하는지, 사이트에 보안 위험을 일으키지 않는지 검증합니다.
7. 이전 표에 설명된 대로, 선택적으로 검색 결과에 표시되지 않도록 정보를 제외하려면 옵션을 하나 이상 선택합니다.
8. 현재 업데이트에 보류 중인 변경사항을 모두 저장하려면 **저장**을 누릅니다.

## 검색 엔진 최적화에 대한 사전 렌더링 서비스 사용

Oracle Content Management로 구축된 사이트를 통해 Search Crawler 요청에 대한 응답으로 정적 HTML 페이지를 사전 렌더링할 수 있습니다.

사전 렌더링 서비스가 사용으로 설정되고 요청이 Search Crawler에서 온 경우 캐시에서 페이지가 검색됩니다.

- 페이지가 캐시에서 발견되고 페이지 저장 시간이 15일 미만인 경우 캐시에 저장된 페이지에 요청이 제공됩니다.
- 캐시의 페이지가 15일보다 오래되었거나 캐시에서 페이지를 찾을 수 없는 경우 Oracle Content Management에서 새 버전이 인출되며 캐시가 업데이트되고 사전 렌더링된 새 페이지가 제공됩니다.

- 페이지를 캐시 또는 Oracle Content Management에서 찾을 수 없을 경우 오류 페이지가 반환됩니다.

아직 캐시에 저장되지 않은 공용 사이트 또는 새 사이트에서 만료된 페이지는 하루에 한 번 렌더링됩니다. 페이지가 이미 캐시에 저장된 경우 캐시를 업데이트하지 않습니다. 사전 렌더링 서비스가 사용으로 설정된 경우 업데이트되고 재게시된 사이트에 대해 매시간 사전 렌더링된 캐시가 업데이트됩니다. 사이트 페이지를 변경할 때 해당 변경사항이 최적화에 어떤 방식으로 영향을 끼치는지 테스트하려는 경우 사이트 속성에서 수동으로 사전 렌더링된 캐시를 업데이트할 수 있습니다.

1. 측면 탐색에서 **사이트**를 누릅니다.
2. 사이트를 선택하고 마우스 오른쪽 단추 누르기 메뉴에서 **속성**을 선택하거나 작업 모음에서 **사이드바**를 누르고 **속성**을 선택합니다.
3. 속성 사이드바에서 **SEO**를 누릅니다.
4. 마지막 캐시 새로고침 날짜 및 시간이 나열됩니다. 캐시를 업데이트하려면 **지금 새로고침**을 누릅니다. 진행률 표시줄에 캐싱 상태가 표시됩니다.

#### 주:

페이지가 사이트 SEO 설정에서 검색 엔진으로부터 숨겨지도록 설정된 경우 해당 페이지는 사전 렌더링되거나 캐시에 저장되지 않습니다.

Oracle Content Management에서 사전 렌더링 서비스를 사용으로 설정하려면 다음과 같이 하십시오.

1. 측면 탐색의 [관리] 영역에서 **시스템**을 누릅니다.
2. 시스템 설정 메뉴에서 **사이트용 SEO**를 선택합니다.
3. **사용**을 누릅니다.
4. 필요한 추가 사용자-에이전트를 정의합니다.

**표 11-1 Oracle Content Management의 하드 코딩된 사용자-에이전트**

사용자-에이전트	사용자-에이전트	사용자-에이전트	사용자-에이전트
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	nuzzlel	Discordbot	Google Page Speed
Qwantify	pinterestbot		

표 11-2 config.cfg 파일의 PrerenderUserAgents 속성에서 추가로 사전 구성된 사용자-에이전트

사용자-에이전트	사용자-에이전트	사용자-에이전트	사용자-에이전트
AddSearchBot	AdIdxBot	AdsBot-Google	AdsBot-Google-Mobile-Apps
AppEngine-Google	Baidu-YunGuanCe	Bingbot	BingPreview
DuckDuckBot	DuckDuckGo-Favicons-Bot	endeca webcrawler	Exabot
Facebot	Feedfetcher-Google	FeedValidator	Fetch
FlipboardProxy	Google Favicon	Google Web Preview	Google-Adwords-Instant
Googlebot	Googlebot-Image	Googlebot-Mobile	Googlebot-News
Googlebot-Video	Google-PhysicalWeb	Google-Structured-Data-Testing-Tool	HubSpot,ia_archiver
Mediapartners-Google	MSNBOT	NetcraftSurveyAgent	nutch
Oracle Secure Enterprise Search	pinterest.com	PIs-Google	SEOkicks
seoscanners	Siteimprove.com	Slurp	Sogou web spider
VSE/1.0	W3C_CSS_Validator	W3C_I18n-Checker	W3C_Unicorn
W3C-checklink	W3C-mobileOK	Y!J	Yahoo Link Preview
Yahoo! Slurp	Yandex	YandexBot	YunGuanCe

## 정적 사이트 전달 옵션 설정

회사에서 컴파일된 사이트를 사용하는 경우 정적 사이트가 캐시되는 기간과, 사이트 컴파일 시 적응형 모바일 레이아웃이 지원되도록 호출할 모바일 사용자-에이전트를 제어할 수 있습니다.

- [게시할 때 자동 컴파일 사용](#)
- [컴파일된 사이트에 대한 기본 캐시 제어 헤더 무효화](#)
- [컴파일된 적응형 레이아웃을 지원하도록 모바일 사용자-에이전트 지정](#)

## 게시할 때 자동 컴파일 사용

Oracle Content Management에는 컴파일 서비스가 내장되어 있어서 추가 구성 없이 쉽게 사이트를 컴파일할 수 있습니다. 또는 서비스 관리자가 사이트 컴파일을 검증하거나 사용자정의 라이브러리를 사용하기 위해 테스트 목적상 수동으로 컴파일 서비스를 설정할 수 있습니다. 사용된 컴파일 서비스에 관계없이, 사이트가 게시되거나 재게시될 때 사이트를 컴파일하도록 선택할 수 있습니다.

1. Oracle Content Management에 로그인한 후 측면 탐색 메뉴에서 **사이트**를 누릅니다.
2. 수정할 사이트를 선택하고 **속성**을 누릅니다.
3. 속성 대화상자에서 **정적 전달**을 누릅니다.
4. **자동 컴파일 사용** 섹션에서 **게시 후 사이트 컴파일**을 사용으로 설정합니다.
5. 완료되면 **완료**를 누릅니다.

자동 컴파일 사용과 함께 사이트를 게시하거나 재게시할 때는 게시 상태가 **사이트** 페이지의 사이트 타일에서 추적 및 표시됩니다. 게시가 완료된 후에는 컴파일 프로세스가 사이트 속성 대화상자의 **정적 전달** 섹션에서 추적됩니다. 컴파일이 완료되면 사이트 속성 정적 전달 섹션에 마지막 컴파일 날짜 및 시간이 나열되고 컴파일 로그에 대한 다운로드 링크가 제공됩니다.

## 컴파일된 사이트에 대한 기본 캐시 제어 헤더 무효화

기본적으로 컴파일된 사이트는 사용자 브라우저에서 300초(5분) 동안 캐시됩니다. 서비스 관리자가 이 기본값을 변경할 수 있지만, 사이트 개발자는 사이트 속성에서 특정 사이트의 기본값을 무효화할 수 있습니다.

1. Oracle Content Management에 로그인한 후 측면 탐색 메뉴에서 **사이트**를 누릅니다.
2. 수정할 사이트를 선택하고 **속성**을 누릅니다.
3. 속성 대화상자에서 **정적 전달**을 누릅니다.
4. **캐싱 응답 헤더** 섹션에서 Cache-control: max-age=를 입력하고 사용자 브라우저에서 페이지가 캐시될 시간(초)의 숫자 값을 입력합니다. 예를 들어, Cache-control: max-age=600은 10분 동안 페이지를 캐시합니다.
5. 완료되면 **완료**를 누릅니다.

인스턴스가 Akamai를 사용하는 경우 기존 Akamai 설정을 유지하려면 **캐싱 응답 헤더** 섹션을 비워둡니다. 기존 Akamai 설정을 무효화하려면 Edge-Control: !no-store,max-age=1800,downstream-ttl=1800을 입력합니다. 여기서 굵게 표시된 항목은 기본 설정(초)입니다.

- !no-store는 이 설정이 속성에 해당하는 Akamai 캐싱 구성을 무효화해야 함을 나타냅니다.
- max-age는 Akamai가 이 페이지를 캐시해야 하는 기간을 결정합니다. 기본값은 1800초(30분)입니다. 이 시간 동안 Akamai는 Oracle Content Management에서 페이지를 요청하지 않고 페이지 요청을 이행합니다.
- downstream-ttl은 Akamai에게 클라이언트 브라우저에 대한 응답으로 "Cache-Control: max-age" 헤더를 보내도록 지시하고, 해당 브라우저는 정해진 시간 동안 페이지를 캐시합니다. 기본값은 1800초(30분)입니다.

## 컴파일된 적응형 레이아웃을 지원하도록 모바일 사용자-에이전트 지정

사이트를 컴파일할 때 적응형 레이아웃을 지원하도록 모바일 페이지를 생성할 수 있습니다. 서버가 사이트에 표준 컴파일 페이지 대신 모바일 페이지를 전달하도록 사용자-에이전트를 지정할 수 있습니다. 여기에 입력한 값은 브라우저가 전송한 사용자-에이전트 헤더와 일치시킬 때 대소문자 구분 없이 부분 문자열로 처리됩니다. Mobile 키워드는 모바일 장치에서 브라우저의 사용자-에이전트 문자열에 흔히 사용됩니다.

1. Oracle Content Management에 로그인한 후 측면 탐색 메뉴에서 **사이트**를 누릅니다.
2. 수정할 사이트를 선택하고 **속성**을 누릅니다.
3. 속성 대화상자에서 **정적 전달**을 누릅니다.
4. **모바일 사용자-에이전트** 섹션에서 모바일 페이지를 제공하려는 사용자 에이전트에 대한 사용자-에이전트 부분 문자열을 콤마로 구분된 목록으로 입력합니다. 부분 문자열의 일부가 브라우저의 사용자-에이전트 문자열과 일치하면 모바일 페이지가 제공됩니다.

## Vanity URL 지정 및 구성

사이트의 Vanity URL을 설정하면 사용자가 Oracle Content Management 인스턴스의 도메인이 아닌 지정된 도메인의 Oracle Content Management 사이트에 액세스할 수 있습니다. 예를 들어, 인스턴스 URL이 `https://myinstance.cec.ocp.oraclecloud.com/site/MyCustomerSite/`일 수 있지만, 보다 친숙한 URL(예: `https://www.example.com`)이 기억하기 쉽고 브랜딩에 더 적합하며 일반적으로 더 쉽게 사용할 수 있습니다. 또한 필요에 따라 Oracle Content Management 사이트는 사용자정의 경로(예: `https://www.example.com/store/`)로 호스트될 수도 있습니다.

Vanity 도메인을 사용하려면 몇 단계가 필요합니다.

- CDN(콘텐츠 전달 네트워크)을 사용합니다.
- 도메인 CNAME(표준 이름) 레코드가 CDN에 매핑되도록 DNS(도메인 이름 시스템)로 Vanity 도메인을 관리합니다.
- Vanity 도메인을 보호하는 CDN에서 적합한 인증서를 배치합니다.
- Vanity 도메인에서 요청을 수락하도록 Oracle Content Management 사이트를 구성합니다.
- Oracle Content Management로 요청이 제대로 경로 지정되도록 CDN을 구성합니다.

### 콘텐츠 전달 네트워크 사용

CDN(콘텐츠 전달 네트워크)은 웹 사이트 성능 및 보안 향상을 위해 전역적으로 분산된 서버의 플랫폼입니다. 콘텐츠에 대한 요청 성능을 최적화하는 동안 CDN은 사용자와 서버의 거리를 최소화합니다. CDN의 주요 목표는 사용자 환경을 개선하는 것이지만, 백그라운드 프로세스가 클린 상태가 아니더라도 사용자에게 클린 상태로 표시되도록 CDN을 통해 전송 중인 요청을 변경할 수 있습니다.

Vanity 도메인에서 Oracle Content Management 사이트의 호스트를 지원하려면 CDN을 사용한 구성을 통해 구성된 Vanity 도메인으로부터 오는 모든 요청이 처리되고, Oracle Content Management로 제대로 다시 경로 지정되고, 요청을 변경하여 Oracle Content Management에 의해 제대로 안전하게 처리되도록 해야 합니다.

### DNS(도메인 이름 시스템)로 도메인 관리

모든 도메인을 Oracle Content Management 사이트에 대한 Vanity 도메인으로 사용할 수 있습니다. Oracle Content Management 사이트에서 사용할 수 있도록 구성하기 전에 Vanity 도메인으로 사용되는 모든 도메인을 제어해야 합니다.

DNS의 제한사항으로 인해 `www`가 없는 루트 도메인(예: `example.com`) 또는 다른 하위 도메인(예: `store.example.com`)은 사용할 수 없습니다. DNS 및 CDN 제공자에게 루트 도메인 사용 가능 여부를 확인하십시오.

DNS는 경로 레벨이 아닌 도메인 레벨에서 작동하므로 Oracle Content Management가 도메인의 일부 경로를 호스트하고 다른 서비스가 나머지 경로를 호스트하도록 하려면 CDN이 경로 지정을 처리해야 합니다. DNS를 통해서만 도메인 및 하위 도메인 레벨에서 트래픽을 분리할 수 있습니다.

### 인증서 배치

CDN이 Vanity 도메인을 보호하는 인증서를 생성하고 호스트해야 합니다. 인증서는 단일 도메인, 다중 도메인 및 하위 도메인 또는 와일드 카드 문자 적용 하위 도메인(예: `*.example.com`)을 보호할 수 있습니다. Vanity 도메인에 대해 모든 조합이 수락될 수

있습니다. 보호되는 모든 도메인이 인증서 세부정보에 표시되므로, 해당 세부정보를 공개적으로 공유하지 않으려는 경우 별도의 인증서를 사용해야 합니다.

### 주:

인증서 생성 및 호스트 프로세스는 CDN에 따라 다른 경우가 많으므로 최상의 수행 방식을 지정해야 합니다.

## Vanity 도메인에 대한 Oracle Content Management 사이트 구성

Vanity 도메인을 사용할 때 Oracle Content Management 사이트가 제대로 로드되도록 하려면 먼저 Vanity 도메인을 사용 중인 사이트 URL을 Oracle Content Management에 알려야 합니다. 이 작업은 Oracle Content Management에서 사이트의 속성 탭을 통해 수행합니다.

1. Oracle Content Management의 측면 탐색에서 **사이트**를 누릅니다.
2. Vanity 도메인을 사용할 사이트를 선택하고 마우스 오른쪽 단추 누르기 메뉴 또는 작업 모음의 **사이드바**에서 **속성**을 선택합니다.
3. Vanity 도메인 필드에서 Vanity 도메인을 입력하고 **저장**을 누릅니다.

### 주:

Oracle Content Management가 도메인에서 요청 수락을 준비하는 데 1시간 이상 걸릴 수 있습니다. 언제든지 사이트 속성 패널에서 진행상황을 확인하십시오.

## 요청을 경로 지정하도록 콘텐츠 전달 네트워크 구성

Oracle Content Management가 제대로 구성되고 요청을 수락할 준비가 완료되면 Vanity 도메인에 대해 생성된 요청이 DNS를 통해 CDN로 경로 지정되며 CDN은 요청을 Oracle Content Management로 올바르게 전달해야 합니다. 예를 들어, 사이트 URL이 `https://myinstance.cec.ocp.oraclecloud.com/site/MyCustomerSite/`인 Oracle Content Management 사이트가 Vanity 도메인 `https://www.example.com/store`로 구성된 경우 다음 작업을 수행하도록 CDN을 구성해야 합니다.

- Vanity 도메인 `https://www.example.com/store` 인식
- Vanity 도메인을 사용하여 원래 Oracle Content Management 인스턴스 `https://myinstance.cec.ocp.oraclecloud.com/` 식별
- 특정 사이트에 대한 사이트 경로 `site/MyCustomerSite/` 추가
- 원래 Oracle Content Management 인스턴스에 전체 사이트 URL `https://myinstance.cec.ocp.oraclecloud.com/site/MyCustomerSite/` 전송
- Oracle Content Management가 요청을 수신하고 사용자 브라우저에 대한 요청을 충족하는 CDN에 응답하여 사용자에게 Vanity 도메인 `https://www.example.com/store`만 표시

CDN 구성 단계는 CDN에 따라 다른 경우가 많으므로 CDN 제공자와 협력하여 원래 Oracle Content Management 인스턴스에 도달하는 방식을 CDN에 알리는 원래 정의와 올바른 사이트 경로를 추가하는 사이트 경로가 제대로 구성되도록 하십시오.



 주:

요청에는 다음 경로가 포함되지 않아야 합니다.

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

해당 경로에 대한 요청에는 사이트 경로가 포함되지 않습니다. 이러한 요청이 제대로 처리되려면 원래 Oracle Content Management 인스턴스의 루트로 분석되어야 합니다.



## Cobrowse 통합 사용

Cobrowse 기능은 Oracle Cobrowse Cloud Service에서 사용되는 공동 작업 도구입니다.

먼저 서비스 관리자가 Cobrowse Cloud Service와의 통합을 수락된 통합으로 추가해야 합니다. *Oracle Content Management 통합 및 확장*의 Oracle Cobrowse Cloud Service와 통합을 참조하십시오.

Oracle Cobrowse Cloud Service 통합이 사용으로 설정되면 사이트에 맞게 기능을 구성한 다음 특정 사이트 페이지에 추가하여 사용할 수 있습니다.

사이트에서 Cobrowsing을 사용으로 설정하려면 다음을 수행합니다.

1. 편집을 위해 사이트를 엽니다.
2. 사이드바에서  을 누르고  을 누릅니다.
3. Cobrowse 섹션에서 **이 사이트에서 Oracle Cobrowse 사용으로 설정**을 선택합니다.
4. 사이트에 대한 시작 프로그램 스크립트를 입력합니다. Oracle Cobrowse Cloud Service 관리자는 Cobrowse 관리 콘솔에 액세스하여 적합한 시작 프로그램 Javascript 코드 조각을 가져올 수 있습니다. 시작 프로그램은 두 가지 유형이 있습니다.
  - 시작 지점 1: Cobrowse 단추가 페이지에 자동으로 추가됩니다.
  - 시작 지점 2: 페이지에 추가된 단추 및 인터페이스를 사용자가 정의할 수 있습니다.
5. 닫기를 누릅니다.

Cobrowse가 사이트에 대해 사용으로 설정된 다음 이를 페이지에 추가하거나 페이지에서 사용되는 방법을 사용자가 정의할 수 있습니다. 자세한 내용은 [페이지에서 Cobrowse 사용](#)을 참조하십시오.

### 보안 사이트 및 사이트 작성기 테스트에 Cobrowse 사용

Cobrowse에는 인스턴트(ICB) 및 고급(ACB)의 두 가지 모드가 있습니다. 보안 사이트에 대해 ICB 모드로 Cobrowsing을 사용하거나 개발 중인 사이트를 미리 보려면 추가 구성이 필요합니다. 이 구성은 Cobrowse 관리 콘솔에서 수행됩니다. [독립형 Cobrowse 사용의 에이전트 콘솔에 로그인](#)을 참조하십시오.

[사용자정의 함수] 필드에서 에이전트가 게시된 보안 사이트의 비밀번호 보호 리소스를 볼 수 있도록 허용하는 함수를 추가합니다.

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

모든 게시된 보안 사이트에 대해 이 기능을 사용하고 개발 중인 사이트를 보거나 미리 보려면 다음 코드를 추가합니다.



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

보호되는 리소스 구성은 새로운 Cobrowse 기능입니다. Cobrowse 페이지 마스킹과 동일한 와일드 카드 문자 URL 패턴을 사용합니다. 자세한 내용은 *Cobrowse Deployment and Use Guide*의 [Configure page masking](#)을 참조하십시오.

## Analytics 추적 추가

사이트 작성기에서 Google, Adobe 또는 Oracle Infinity와 같은 외부 분석 제공자와의 통합을 보다 간편하게 수행할 수 있도록 Web Analytics 추적을 위해 사이트나 페이지에 JavaScript 추적 코드의 코드 조각을 추가할 수 있습니다.

사이트에 Analytics 추적을 추가하려면:

1. 편집을 위해 사이트를 엽니다.
2. 사이드바에서  을 누르고  Analytics를 누릅니다.
3. Analytics 추적을 사용으로 설정하려면 스위치를 누릅니다.
4. **JavaScript 추적 코드 조각** 상자에서 새 코드 조각을 추가하거나 기존 스크립트를 편집합니다. 관리자가 해당 환경의 코드 조각을 제공했을 수 있습니다. 그러면 이 상자에 표시됩니다. 스크립트를 사용자정의하거나 고유 스크립트를 추가할 수 있습니다. 관리자가 제공한 코드 조각을 편집하면 스크립트가 수정되었음을 알리는 메시지가 표시됩니다. 사용자정의의 제거하려면 **최신 테넌트 스크립트로 복원**합니다.

Google Analytics 추적 코드 조각의 예는 다음과 같습니다.

```
<!-- 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>
```

이 변경사항을 저장 및 게시하며, 필요한 경우 사이트에 대한 분석을 수집하기 전에 사이트를 온라인으로 전환해야 합니다.

### 분석 데이터 보기

사이트를 게시하고 온라인으로 전환한 후에는 공급업체의 사이트(예: Google Analytics)에서 추적된 분석 데이터를 확인할 수 있습니다. Oracle Infinity 분석 추적에 코드 조각을 사용한 경우 Oracle Infinity 홈 페이지로 이동하고 분석을 눌러 데이터를 확인한 후 보고서를 선택 또는 생성합니다.

# 12

## 사이트 게시

사이트를 온라인으로 전환하고 오프라인으로 설정하며 사이트 변경사항 게시에 대해 알아보겠습니다.



- [사이트를 온라인으로 전환 또는 오프라인으로 설정](#)
- [사이트 변경사항 게시](#)

## 사이트를 온라인으로 전환 또는 오프라인으로 설정

사이트가 *온라인* 상태일 때 적절한 액세스 권한을 가진 사용자는 지정된 주소(URL)에서 표준 웹 브라우저로 사이트를 볼 수 있습니다. 사이트가 *오프라인*일 때 사이트를 공개로 이용할 수 없습니다. Oracle Content Management에서만 사이트를 볼 수 있습니다.



사이트를 온라인으로 전환할 수 있으려면 사이트가 먼저 게시되어 있어야 합니다. 사이트를 게시하려면 [사이트 변경사항 게시](#)를 참조하십시오.

오른쪽에 있는 상태 아이콘은 사이트가 온라인 또는 오프라인인지를 보여줍니다.

- 사이트가 이전에 게시된 적이 없으면 대시(-)가 표시됩니다.
- 사이트가 온라인이면 이 표시됩니다.
- 사이트가 오프라인이면 이 표시됩니다.

온라인 또는 오프라인 사이트에 액세스할 수 있는 사용자에 대한 자세한 내용은 [사이트 보안 이해](#)를 참조하십시오.

사이트의 상태를 변경하려면 사이트 소유자이거나 관리자 롤을 보유해야 합니다. 또는 사이트 제어가 사용으로 설정된 경우 사이트 공유 여부와 관계없이 사이트 관리자가 사이트의 상태를 변경할 수 있습니다.

1. [사이트] 페이지의 목록에서 사이트를 선택합니다.
2. 사이트를 온라인으로 전환하거나 오프라인으로 설정하려면 마우스 오른쪽 단추 누르기 메뉴에서 [온라인으로 전환](#) 또는 [오프라인으로 전환](#)을 선택합니다. 사이트 타일에서 을 눌러 사이트를 온라인으로 전환하거나, 사이트 타일에서 을 눌러 사이트를 오프라인으로 전환할 수도 있습니다.  
선택 확인 메시지가 표시됩니다.

사이트를 온라인으로 전환하면 완전히 렌더링된 HTML 버전의 사이트가 생성되고 Oracle Cloud 호스팅 위치로 복사됩니다. 온라인 사이트는 사이트 이름 아래에 URL이 표시됩니다. 기본 URL 형식은 다음과 같습니다.


```
https://service_name.identity_domain.sites.oraclecloud.com/site_name
```

사이트를 오프라인으로 가져오면 사이트와 해당 폴더 및 파일이 Oracle Cloud 호스팅 위치에서 제거됩니다.

## 사이트 변경사항 게시

사이트 변경사항을 게시하려면 사용자가 사이트 소유자이거나 사용자에게 관리자 롤이 있어야 합니다.

사이트 변경사항을 게시할 때는 다음 사항에 유의하십시오.

- 온라인 상태의 사이트에 대한 변경사항을 게시하면 사이트 액세스 권한이 있는 모든 사용자에게 커밋된 변경사항이 즉시 표시됩니다.
  - 사이트에 변경사항을 게시할 때 권장사항을 포함하여 사이트 채널에 대상으로 지정된 모든 자산을 게시하거나 사이트 페이지에 사용된 자산만 게시할 수 있습니다. 예를 들어, 사이트 채널에 대상으로 지정된 모든 자산을 게시하고 사이트에 직접적으로 속하지 않는 콘텐츠 항목을 참조하는 목록이 있으면 참조된 자산도 게시됩니다.
  - 다국어 사이트에 대한 변경사항을 게시하면 사이트의 연관된 지역화 정책에 정의된 언어의 번역만 게시됩니다.
  - 사이트의 연관된 지역화 정책에 더 이상 정의되지 않은 언어에 대한 번역이 이미 게시된 경우 번역된 콘텐츠는 게시된 사이트에서 제거됩니다.
  - 다국어 사이트에서 페이지를 삭제하면 사이트가 게시된 경우 번역된 페이지도 삭제됩니다.
1. 사이트 변경사항을 게시하려면 선택 후 마우스 오른쪽 단추 누르기 메뉴 또는 작업 모음에서 **게시**를 선택합니다.
  2. 사이트와 사이트 게시 채널에 대상으로 지정된 모든 자산을 게시할지, 또는 사이트와 사이트 페이지에 추가된 모든 자산을 게시할지 선택합니다.
  3. Oracle Content Management가 모든 사이트 문자열과 자산에 필요한 승인 및 번역이 있는지 검증합니다. 사이트의 연관된 지역화 정책에 필요한 번역이 없을 경우, 자산이 번역됨으로 표시되지 않은 경우 또는 자산에 승인이 필요하지만 자산이 승인됨으로 표시되지 않은 경우 사이트를 게시하지 못할 수 있습니다. 항목을 확장하여 추가 세부정보를 표시합니다. 문제가 있을 경우 수정한 후 게시를 다시 시도합니다. 모든 항목이 적합한 경우 **게시**를 누릅니다. 선택 확인 메시지가 표시됩니다.
  4. 사이트가 아직 온라인() 상태가 아닌 경우 사이트를 온라인으로 전환하여 사용자에게 사이트를 제공해야 합니다.

# 13

## 사이트 보안

사이트를 게시하고 온라인으로 제공하면 사이트에 액세스할 수 있는 사용자를 제어할 수 있습니다.

사이트를 보안하는 경우 지정된 룰에 따라 게시된(온라인) 사이트에 액세스할 수 있는 사용자 그룹을 지정합니다. 해당 룰은 서비스 인스턴스 관리자가 지정하는 서비스 차원의 룰입니다.

- [사이트 보안 이해](#)
- [사이트 보안 변경](#)

관리자인 경우 사이트 보안과 관련하여 추가 작업(예: 사용자정의 사인인 사용, 사이트 및 테마 공유 허용, 사이트, 템플릿 및 구성요소 생성 제한, 제어 및 기타 작업 사용)을 수행할 수 있습니다. *Oracle Content Management 관리*의 사이트 및 자산 설정 구성을 참조하십시오.

## 사이트 보안 이해

게시된(온라인) 사이트를 볼 수 있으며 사이트에서 보안 콘텐츠와 상호작용할 수 있고 게시되지 않은(오프라인) 사이트를 확인 및 편집할 수 있는 사용자를 제어할 보안을 적용할 수 있습니다.

### 사이트 보안

사이트를 게시하고 온라인으로 제공하면 사이트에 액세스할 수 있는 사용자를 제어할 수 있습니다. 시스템 및 사이트 관리자가 환경을 구성한 방식에 따라 사이트를 공개할 수도 있고, 사이트를 등록된 사용자로 제한할 수도 있고, 사이트를 특정 사용자로 제한할 수도 있습니다.

사이트 보안 또는 기타 설정을 변경하려면 사이트 소유자이거나 관리자 룰을 보유해야 합니다. 사인인 요구사항을 변경하려면 사이트가 오프라인 상태여야 합니다. 그러나 지정된 사용자나 사용자 룰을 변경하려면 사이트가 온라인 상태여도 됩니다. 사이트를 오프라인으로 가져오면 사이트와 해당 폴더 및 파일이 Oracle Cloud 호스팅 위치에서 제거됩니다.

사용 가능한 보안 옵션은 템플릿 정책(사이트 제어가 사용으로 설정된 경우) 또는 테넌트 정책(사이트 제어가 사용 안함으로 설정된 경우)에 따라 제한될 수 있습니다. [사이트 제어 이해](#)를 참조하십시오.

사이트를 보안하는 경우 지정된 룰에 따라 게시된(온라인) 사이트에 액세스할 수 있는 사용자 그룹을 지정합니다. 해당 룰은 서비스 인스턴스 관리자가 지정하는 서비스 차원의 룰입니다.

- **클라우드 사용자:** 인증된 사용자는 사용자 이름 및 비밀번호로 서비스 인스턴스에 사인인합니다. 여기에는 **방문자** 룰 또는 **사용자** 룰을 보유하거나 보유하지 않은 모든 인증된 사용자가 포함됩니다.
- **방문자:** 이 룰을 보유한 사용자만 사이트에 액세스할 수 있습니다. 예를 들어, 게시된 사이트를 확인할 수 있지만 해당 Oracle Content Management 인스턴스의 폴더 및 파일에 대한 액세스 권한을 가지지 않은 사용자에게 이 룰이 지정될 수 있습니다.

### 주:

여기에는 사이트 소유자가 아니거나 사이트가 명시적으로 공유되지 않은 경우 **사용자** 룰을 보유한 사용자가 포함되지 않습니다.

- **서비스 사용자:** 이 롤을 보유한 사용자만 사이트에 액세스할 수 있습니다. 예를 들어, 게시된 사이트를 확인할 수 있으며 해당 Oracle Content Management 인스턴스의 폴더 및 파일에 대한 액세스 권한을 가지는 사용자에게 이 롤이 지정될 수 있습니다.
- **특정 사용자:** 사이트 멤버로 추가하는 사용자만 게시된 사이트를 확인할 수 있습니다.

### 사이트 공유

*사이트 공유*를 사용하여 게시되지 않은 (오프라인) 사이트에 액세스할 수 있는 개별 사용자를 지정하고 해당 사용자가 부여된 권한에 따라 사이트를 확인, 수정 또는 관리하도록 허용합니다. 사이트 소유자인 경우 또는 사이트가 공유되었으며 관리자 롤이 부여된 경우 사이트를 공유할 수 있습니다.

#### 주:

사용자에게 공유 롤을 지정하면 해당 사용자의 보안 롤이 높아집니다. 예를 들어, **방문자** 롤을 보유한 사용자와 사이트를 공유하고 해당 사용자에게 제공자 롤을 부여하는 경우 해당 사용자는 오프라인 사이트를 수정할 수 있지만, **방문자** 롤을 보유한 다른 사용자는 온라인 사이트를 볼 수만 있습니다.

- **열람자:** 열람자는 편집기에서 사이트를 볼 수 있지만 어떤 내용도 변경할 수 없습니다.
- **다운로더:** 사이트의 경우 열람자 롤과 동일한 권한을 보유합니다.
- **제공자:** 열람자와 동일한 권한 외에, 사이트를 편집하고 사이트 페이지를 삭제하며 사이트(오프라인일 경우)를 삭제할 수 있습니다.
- **관리자:** 제공자와 동일한 권한 외에, 사용자를 추가하고 사용자 롤을 지정하며 온라인 사이트에 변경사항을 게시하고 사이트를 온라인 및 오프라인으로 전환할 수 있습니다. 사이트 생성자(소유자)에게는 자동으로 관리자 롤이 지정됩니다.

사이트를 생성하면 해당 사이트 이름으로 채널이 생성됩니다. 채널을 다른 사람과 공유하려면 사이트를 공유한 후 상대방에게 최소한 제공자 롤을 부여해야 합니다. 그래야만 해당 상대가 자산 게시에 채널을 사용할 수 있습니다. 사이트를 게시하려면 사용자에게 관리자 롤이 있어야 합니다.

### 구성요소 공유

일부 구성요소는 공유 리소스(예: 폴더, 파일 또는 대화)에 대한 액세스 권한을 제공합니다. *구성요소 공유*는 사이트 보안(게시된 사이트 볼 수 있는 사용자)과 리소스 공유(폴더와 파일, 대화를 보고 관련 작업을 수행할 수 있는 사용자)를 고려합니다.

예를 들어, 사이트에 문서 관리자 구성요소를 추가하는 경우 사이트의 모든 방문자가 폴더 콘텐츠를 확인할 수 있으며 롤 및 기타 권한에 따라 폴더의 요소를 추가, 수정 또는 삭제할 수도 있습니다.

일반 고려사항:

- 사이트 작성자는 보유한 액세스 권한에서 벗어나는 폴더 액세스 권한을 부여할 수 없습니다. 예를 들어, 작성자의 폴더 액세스 권한이 다운로더인 경우 제공자에게 사이트 방문자 권한을 부여할 수 없습니다.
- 구성요소에서 설정된 권한은 방문자의 권한을 보강할 수 있습니다. 예를 들어, 방문자의 폴더 권한이 열람자(또는 권한 없음)인 경우 문서 관리자 구성요소는 구성요소에서 선택된 롤에 따라 이보다 높은 권한을 부여할 수 있습니다. 이와 같이 향상된 권한은 구성요소 자체에서만 적합합니다.

- 사이트 방문자의 권한이 구성요소에 대해 지정된 권한보다 높은 경우 개별 권한이 구성요소에 대해 설정된 권한을 무효화합니다.
- 폴더에 대해 부여된 권한은 해당 폴더에 중첩된 폴더 및 파일에 적용됩니다.

공용 사이트의 경우:

- 보안 사이트에서만 대화 구성요소가 지원됩니다.
- 문서 관리자 구성요소는 기본적으로 모든 방문자에게 연관된 폴더에 대한 다운로드 권한을 제공합니다. 위에 나열된 지침에 따라 룰을 변경할 수 있으며 구성요소 자체의 설정을 통해 사용자에게 표시되는 옵션을 제한할 수 있습니다.
- 폴더 목록 및 파일 목록 구성요소는 모든 사용자에게 다운로드 액세스 권한을 부여합니다. 사용자는 룰에 관계없이 파일을 보고 다운로드할 수 있습니다.

### 보안 사이트 URL

사이트를 온라인으로 설정하면 완전히 렌더링된 HTML 버전의 사이트가 생성되고 Oracle Cloud 호스팅 위치로 복사됩니다. 온라인 사이트는 사이트 이름 아래에 URL이 표시됩니다.

비보안 사이트의 기본 URL 형식은 다음과 같습니다.

```
https://service_name.identity_domain.sites.oraclecloud.com/site_name
```

보안 사이트의 기본 URL 형식은 다음과 같습니다.


```
https://service_name.identity_domain.sites.oraclecloud.com/authsite/site_name
```

URL에 authsite를 추가합니다.

## 사이트 보안 변경

사이트를 게시하고 온라인으로 제공하면 사이트에 액세스할 수 있는 사용자를 제어할 수 있습니다. 시스템 및 사이트 관리자가 환경을 구성한 방식에 따라 사이트를 공개할 수도 있고, 사이트를 등록된 사용자로 제한할 수도 있고, 사이트를 특정 사용자로 제한할 수도 있습니다.

사이트 보안 또는 기타 설정을 변경하려면 사이트 소유자이거나 관리자 룰을 보유해야 합니다. 사인인 요구사항을 변경하려면 사이트가 오프라인 상태여야 합니다. 그러나 지정된 사용자나 사용자 룰을 변경하려면 사이트가 온라인 상태여도 됩니다. 사이트를 오프라인으로 가져오면 사이트와 해당 폴더 및 파일이 Oracle Cloud 호스팅 위치에서 제거됩니다. 사이트를 오프라인으로 설정하려면 [사이트] 페이지에서 사이트를 선택하고 마우스 오른쪽 단추 누르기 메뉴에서


**오프라인으로 전환**을 선택하거나 작업 모음에서  을 누릅니다. 선택 확인 메시지가 표시됩니다.

사이트 보안을 변경하려면 다음과 같이 하십시오.

1. [사이트] 페이지에서 사이트를 선택하고 마우스 오른쪽 단추 누르기 메뉴에서 **속성**을 선택하거나 작업 모음에서 누릅니다.
2. **사이트 보안** 탭을 누릅니다.

사이트 보안 탭의 사용 가능한 옵션은 시스템 관리자가 사이트 보안 설정을 구성한 방식 및 사이트 관리자가 이 사이트의 기반이 되는 템플릿에서 보안 설정을 구성한 방식(사이트 제어가 사용으로 설정된 경우)에 따라 달라집니다. [사이트 시작하기](#) 및 [사이트 제어 이해](#)를 참조하십시오.



3. 등록된 사용자가 사인인해야만 온라인 사이트를 볼 수 있도록 하려면 [로그인 필요] 옆에 있는 **예**를 누릅니다. 요구사항을 제거하고 온라인 사이트를 공개하려면 [로그인 필요] 옆에 있는 **아니오**를 누릅니다.
4. 온라인 사이트에 액세스할 수 있는 등록된 사용자 그룹을 선택합니다. 개별 그룹을 선택하려면 먼저 **클라우드 사용자**의 선택을 해제합니다.
  - **클라우드 사용자**: 인증된 사용자만 사이트에 액세스할 수 있습니다. 인증된 사용자는 사용자 이름 및 비밀번호로 도메인에 사인인합니다. 여기에는 **Oracle Content Management Cloud 방문자** 롤 또는 **Oracle Content Management Cloud 사용자** 롤이 있는 사용자가 포함됩니다.
  - **방문자**: 이 롤을 보유한 사용자만 사이트에 액세스할 수 있습니다. 여기에는 **Oracle Content Management Cloud 사용자** 롤이 있는 사용자는 포함되지 않습니다.
  - **서비스 사용자**: 이 Oracle Content Management 인스턴스에 사인인할 수 있는 사용자만 사이트에 액세스할 수 있습니다.
  - **특정 사용자**: 사이트에 액세스할 수 있는 개인을 지정합니다. **멤버 추가**를 누릅니다. 검색 필드에 사용자 이름 전체 또는 일부를 입력합니다. 표시되는 목록에서 사용자를 선택하고 반복하여 사용자를 더 추가합니다. 완료되면 **추가**를 누릅니다. 사용자를 제거하려면 사용자 이름 아래의 메뉴에서 **제거**를 누릅니다.
5. **저장**을 눌러 변경사항을 저장하고 창을 닫습니다.  
사이트가 오프라인이며 로그인해야 함을 알 수 있습니다.
6. 사이트를 온라인으로 전환하려면 마우스 오른쪽 단추 누르기 메뉴에서 **온라인으로 전환**을 선택하거나 작업 모음에서 을 누릅니다. **진행 확인**을 누르고 **확인**을 누릅니다.

사이트를 온라인으로 전환하면 완전히 렌더링된 HTML 버전의 사이트가 생성되고 Oracle Cloud 호스팅 위치로 복사됩니다. 온라인 사이트는 사이트 이름 아래에 URL이 표시됩니다. 비보안 사이트의 기본 URL 형식은 다음과 같습니다.

```
https://service_name-identity_domain.cec.ocp.oraclecloud.com/site/
site_name
```

보안 사이트의 기본 URL 형식은 다음과 같습니다.

```
https://service_name-identity_domain.cec.ocp.oraclecloud.com/site/
authsite/site_name
```

URL에 authsite를 추가합니다.

로그아웃 URL을 추가하고 링크, 단추 또는 메뉴에 표시되는 페이지로 구현할 수 있습니다. [단락](#), [단추](#) 및 [페이지 추가](#)를 참조하십시오.

로그아웃 URL의 형식은 다음과 같습니다.

```
https://service_name-identity_domain.cec.ocp.oraclecloud.com/cloudgate/
logout.html?postlogouturl=%2Fsite%2Fauthsite%2Fsite_name
```



postlogouturl은 위와 같이 인코딩된 형식이어야 합니다.

# 14

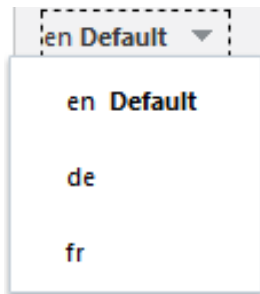
## 다국어 사이트 작업

다국어 대상 경험을 위한 사이트 번역에 대해 알아보겠습니다.

- [다국어 사이트 개요](#)
- [사이트 번역](#)
- [사이트 번역 작업 관리](#)

### 다국어 사이트 개요

지역화 정책에 지정된 모든 언어로 사이트를 번역할 수 있습니다. 사이트의 연관된 지역화 정책에 지정된 언어를 확인하려면 사이트 타일에서 언어 옆에 있는 아래쪽 화살표를 누릅니다. 지역화된 버전을 미리 보려면 언어를 선택하고 사이트를 엽니다.



#### 주:

Oracle Content Management Starter Edition을 통해서 는 다국어 사이트를 생성할 수 없습니다. 전체 기능과 무제한 사이트가 필요하면 Oracle Content Management Premium Edition으로 업그레이드하십시오.

번역에 대한 사이트를 선택하면 사이트 파일의 .zip 파일이 생성됩니다.

#### 주:

사이트에 여러 저장소의 자산이 포함된 경우 기본 저장소의 자산만 .zip에 포함됩니다.

.zip에는 다음 파일 및 폴더가 포함됩니다.

- assets 폴더 - 이 폴더는 전체 사이트 또는 대상으로 지정된 자산을 번역하도록 선택한 경우에만 존재합니다.
  - root 폴더

- \* 각 콘텐츠 항목의 <contentItem\_ID>.json 파일 - 콘텐츠 항목의 필드에 있는 이름, 설명 및 번역 가능한 문자열은 물론 편집하지 않아야 하는 콘텐츠 항목에 대한 추가 정보가 포함됩니다.
  - job.json - 번역 작업을 설명하는 파일입니다. 이 파일은 편집하지 마십시오.
- site 폴더 - 이 폴더는 전체 사이트 또는 사이트 콘텐츠를 번역하도록 선택한 경우에만 존재합니다.
  - root 폴더
    - \* 각 사이트 페이지의 <page\_ID>.json 파일 - 페이지에 있는 구성요소의 이름, 제목, 설명, 키워드, 머리글, 바닥글 및 번역 가능한 문자열은 물론 편집하지 않아야 하는 페이지에 대한 추가 정보가 포함됩니다. 사용자정의 구성요소가 있을 경우 번역 가능한 문자열을 사용하도록 구성되었을 수 있습니다. **다국어 사이트를 위한 번역 가능한 구성요소 개발**을 참조하십시오.
    - \* siteinfo.json - 설명, 키워드, 머리글 및 바닥글을 포함합니다.
    - \* structure.json - 탐색 및 사이트 구조를 포함합니다.
  - job.json - 번역 작업을 설명하는 파일입니다. 이 파일은 편집하지 마십시오.

## 사이트 번역

번역된 사이트의 페이지에 바로 자산이 표시되거나 콘텐츠 목록에 자산이 표시되며 해당 자산을 번역한 적이 있는 경우, 사이트와 동일한 언어로 자산이 표시됩니다. 또한 사이트와 별도로 콘텐츠 항목을 번역할 수 있습니다. 콘텐츠 항목 지역화를 참조하십시오.

여러 저장소의 자산이 포함된 사이트를 번역할 때는 기본 저장소의 자산만 번역 작업에 포함됩니다.

### 주:

Oracle Content Management Starter Edition을 통해서는 번역을 사용할 수 없습니다. 전체 기능과 무제한 사이트가 필요하다면 Oracle Content Management Premium Edition으로 업그레이드하십시오.

### 번역 작업 생성

수동 번역용 번역 패키지를 익스포트하거나 번역 커넥터를 사용하는 경우 번역 작업을 생성해야 합니다.

1. 번역할 사이트를 선택한 다음 **번역**을 누릅니다. **번역** 옵션을 보려면 **자세히**를 눌러야 할 수도 있습니다.
2. 익스포트하려는 사이트 파일(번역 작업)에 대한 정보를 입력한 다음 **생성**을 누릅니다.
  - 번역 작업에 대한 이름을 입력합니다.
  - 사이트가 번역될 대상 언어를 선택합니다.
  - 전체 사이트, 즉 사이트 콘텐츠(페이지, 구조 및 사이트 정보)와 대상으로 지정된 자산을 익스포트할지 아니면 사이트 콘텐츠 또는 대상으로 지정된 자산만 익스포트할지 선택합니다.
  - 번역 커넥터를 선택하거나 수동 번역용 번역 패키지를 익스포트하도록 선택합니다.

- 마쳤으면 **생성**을 누릅니다.

번역 작업이 완료되면 번역된 패키지가 임포트됩니다Oracle Content Management.

#### 수동으로 소스 언어 파일 번역

콘텐츠를 수동으로 번역하는 경우 번역 작업이 완료된 후 사이트 파일의 .zip 파일을 다운로드해야 합니다.

1. 배너에서 **번역 작업**을 누르고 번역 작업을 선택하고 **다운로드**를 누릅니다.
2. 대상 언어로 선택한 언어마다 *root* 폴더와 동일한 레벨에서 .zip 파일에 폴더(예: *de*, *es* 및 *fr*)를 생성합니다. 선택된 언어의 부분 집합을 번역할 수 있습니다. 예를 들어, 지금 독일어(*de*)로 번역하고 나중에 스페인어(*es*) 및 프랑스어(*fr*)로 번역할 수 있습니다.
3. *모든* .json 파일을 *root* 폴더에서 각 언어 폴더로 복사합니다.
4. 모든 .json 파일의 문자열을 해당 언어로 번역합니다. .json 파일에서 문자열을 삭제하지 말고 파일 이름을 바꾸지 마십시오.
5. *assets*(번역 작업에 자산이 포함된 경우) 폴더와 *site*(번역 작업에 사이트 콘텐츠가 포함된 경우) 폴더를 *job.json*, *root* 및 번역된 파일이 있는 모든 언어 폴더로 압축합니다.

#### 수동으로 번역된 패키지 임포트

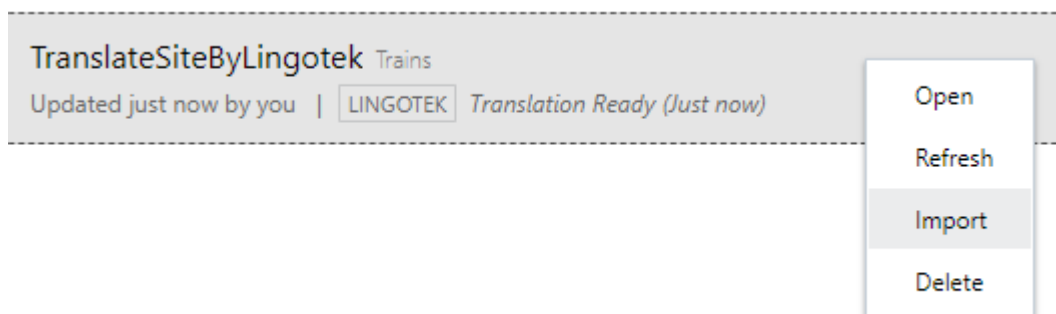
번역이 수동으로 또는 자동으로 수행되었는지 완성된 번역 패키지를 임포트해야 합니다.

1. **번역 작업** 페이지에서 **임포트**를 누릅니다.
2. **업로드**를 누르고 번역된 사이트 파일의 .zip 파일을 선택하고 **열기**를 누릅니다.
3. 업로드가 완료된 후 **확인**을 누릅니다.
4. Oracle Content Management는 작업에 정의된 번역이 .zip 파일에서 사용 가능한지 검증합니다. 번역 작업에 포함된 페이지 및 자산을 보려면 대화상자에서 링크를 누릅니다.
5. 번역을 임포트할 준비가 되면 **임포트**를 누릅니다. 임포트 상태가 배너 위에 나타납니다. **세부정보**를 눌러 작업 세부정보를 볼 수 있습니다.

#### 번역 커넥터에서 번역된 패키지 임포트

번역이 수동으로 또는 자동으로 수행되었는지 완성된 번역 패키지를 임포트해야 합니다.

1. ●●●를 누르고 **번역 작업**을 선택합니다.
2. 완성된 번역 작업을 마우스 오른쪽 단추로 누르고 **임포트**를 선택합니다.



3. Oracle Content Management는 작업에 정의된 번역을 검증합니다. 번역 작업에 포함된 페이지 및 자산을 보려면 대화상자에서 링크를 누릅니다.
4. 번역을 임포트할 준비가 되면 **임포트**를 누릅니다.

임포트 상태가 배너 위에 나타납니다. **세부정보**를 눌러 작업 세부정보를 볼 수 있습니다. 번역을 임포트한 후에는 사이트를 게시하고 사이트에서 번역이 제공되도록 사이트를 온라인으로 설정해야 합니다.

번역 패키지를 임포트한 후 번역 서비스에서 번역을 변경한 경우 **새로고침**을 눌러 번역을 업데이트할 수 있습니다.





번역 후 사이트를 변경하는 경우 새 문자열 또는 편집된 문자열을 번역해야 합니다. 사이트 작성기에서 사이트를 편집하는 경우 사이트의 기본 언어 버전을 편집합니다. 적용하는 모든 사이트 구조 변경사항(예: 구성요소 추가 또는 페이지 재배열)은 사이트의 지역화된 버전에서 복제됩니다. 그러면 새 번역 작업을 생성하여 업데이트된 문자열을 번역할 수 있습니다.

## 사이트 번역 작업 관리

번역에 대한 사이트를 선택하면 *번역 작업*이 생성됩니다. 그런 다음 번역용 파일을 다운로드하여 번역한 후 번역된 파일을 임포트할 수 있습니다.

번역 작업을 생성하려면 [사이트 번역](#)을 참조하십시오.

[번역 작업] 페이지는 모든 번역 작업과 해당 상태를 나열합니다.

-  (준비) - 사이트 파일의 .zip 파일을 다운로드할 준비가 되었습니다.
-  (진행 중) - .zip 파일이 다운로드되었습니다. 모든 대상 언어의 모든 번역이 성공적으로 임포트될 때까지 진행 중 상태로 유지됩니다.
-  (완료) - 이 작업에 대한 모든 대상 언어의 번역이 성공적으로 임포트되었습니다.
-  (실패) - 번역 작업을 실패했습니다. 배너 위에 작업 실패 이유에 대한 실패 메시지가 표시됩니다. 메시지를 다시 확인해야 하는 경우 번역 작업을 **다시 제출**할 수 있습니다. 문제를 해결한 다음 작업을 **다시 제출**합니다.

다음 작업을 수행할 수 있습니다.

- 번역 작업의 세부정보를 보려면 엽니다. 세부정보에는 소스 언어와 모든 선택된 대상 언어 및 해당 번역의 상태가 포함됩니다.
- 사이트 파일의 .zip 파일을 다운로드하려면 작업을 선택하고 **다운로드**를 누릅니다.
- 작업을 삭제하려면 해당 작업을 선택하고 **삭제**를 누릅니다.
- 번역을 임포트하려면 **임포트**를 누른 다음 **업로드**를 누르고 번역된 사이트 파일의 .zip 파일을 선택한 다음 **확인**을 누릅니다.  
Oracle Content Management는 작업에 정의된 모든 번역이 .zip 파일에서 사용 가능한지 검증합니다. 번역 작업에 포함된 사이트 페이지 및 자산을 보려면 대화상자에서 링크를 누릅니다. 번역을 임포트할 준비가 되면 **임포트**를 누릅니다.

## 번역용 로케일

번역할 항목을 제출할 때 대상 언어가 코드로 식별되므로 언어 서비스 제공자는 항목을 번역하고 반환할 언어를 알 수 있습니다. 예를 들어, **fr**은 프랑스어를 나타내고 **de**는 독일어를 나타냅니다.

이 코드는 더 많은 지역 방언으로 확장될 수 있습니다. 예를 들어, **de-LI**는 리히텐슈타인에서 사용되는 독일어 코드이고 **de-LU**는 룩셈부르크에서 사용되는 독일어 코드입니다. 그러나

언어 서비스 제공자가 지역 방언을 지원하지 않으면 제공된 코드는 2자 기본 언어로 잘립니다. 예를 들어, **de-LI** 및 **de-LU**의 경우 코드가 **de**로 잘립니다.

언어 서비스 제공자가 하나의 지역 방언을 지원하지만 전부 지원하지 않는 경우 대체될 수 있습니다. 예를 들어, **ms-BN**은 브루나이에서 사용되는 말레이어 코드인데, 언어 서비스 제공자가 이 방언을 지원하지 않으면 **ms-MY**(말레이시아에서 사용되는 말레이어 코드)와 같이 지원되는 방언으로 전환될 수 있습니다. 언어 서비스 제공자가 방언을 구별하지 않으면(예: 벨리즈에서 사용되는 영어인 **en-BZ**와 자메이카에서 사용되는 영어인 **en-JM**) 기본 언어로 잘립니다(이 경우 영어인 **en**).

## 번역용 사용자정의 로케일

조직의 요구사항에 따라 개발자가 사용자정의 로케일을 생성할 수 있습니다. 사용자정의 로케일 코드에는 기본 언어, 지역 방언 코드(해당되는 경우), 사용자정의 로케일을 지정하는 **x**, 조직에 필요한 사용자정의를 식별하는 기타 단어가 포함됩니다. 예를 들어, 영어의 사용자정의 로케일은 **en-JM-x-custom**과 비슷할 수 있습니다.

사용자정의 로케일은 조직마다 고유하기 때문에 기본 언어 및 지역 방언(언어 서비스 제공자가 지원하는 경우)으로 번역하도록 제출할 때 사용자정의 로케일 코드가 잘립니다. 위의 예에서 **en-JM-x-custom**은 사용자정의에 특정한 코드 부분은 없애고 **en-JM**으로 잘립니다. 또는 언어 서비스 제공자가 자메이카의 지역 방언 코드(JM)를 지원하지 않으면 기본 언어(**en**)로 잘릴 수 있습니다.

## URL 재지정용 로케일 별칭 설정


런타임 및 런타임 미리보기에서 사이트 URL에 사용될 로케일의 별칭을 쉽게 설정할 수 있습니다. 이는 조직에서 길이가 길고 URL에 복잡성을 추가하는 사용자정의 로케일을 사용하는 경우 특히 유용합니다.

예를 들어, **en-GB-x-cornish**로 정의된 사용자정의 로케일의 URL은 다음과 비슷합니다.

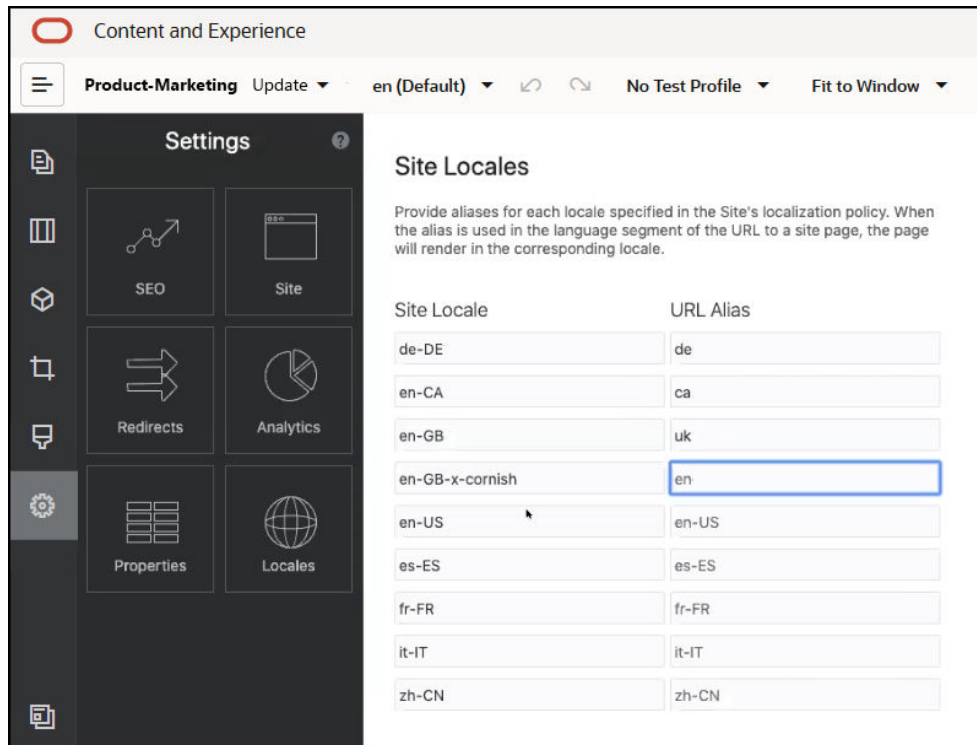
```
https://example.com/site/BlogSite/en-GB-x-cornish/home.html
```


별칭을 추가하면 더 간단한 URL로 재지정할 수 있습니다.

```
https://example.com/site/mysite/en/home.html
```

1. 편집을 위해 사이트를 엽니다.
2. 사이드바에서 을 누르고 로케일을 누릅니다.

사이트에서 사용된 모든 로케일 목록이 해당 URL 별칭 필드 옆에 표시됩니다.



3. 별칭이 필요한 각 사이트 로케일 옆에 별칭을 입력하고 단기를 누른 다음 **저장**을 누릅니다.
4. 런타임에 URL에 사용될 별칭을 미리 보려면  을 누릅니다.
5. 업데이트를 게시하는 경우 변경사항이 게시되어 사용할 수 있게 됩니다.



# 15

## 사이트 재지정 또는 URL 매핑 사용

웹 사이트를 재구조화하거나 이동할 때 사용자 요청을 이전 URL에서 현재 URL로 재지정할 수 있습니다. 30x 재지정을 URL에 지정하면 사이트 재설계 후에도 책갈피 또는 게시된 링크를 유지할 수 있습니다.

검색 엔진에서 평판 순위가 높은 페이지는 다른 인프라 기술에서 Oracle Content Management 호스트 사이트로 이동할 때 다른 URL로 이동할 수 있습니다. 재지정을 사용하면 사이트의 URL 구조를 재구성하고 검색 엔진 순위를 유지할 수 있습니다.

- [재지정 계획](#)
- [사이트 재지정 추가](#)
- [JSON 파일에 재지정 규칙 지정](#)
- [사이트에 재지정 규칙 파일 업로드](#)
- [사이트 URL 매핑](#)

### 재지정 계획

지정된 URL에 HTTP 30x 응답을 보내는 재지정을 지정할 수 있습니다. 추천한 재지정 중 하나와 요청이 일치하지 않으면 정규적인 URL 처리가 발생하고 일반적 방식으로 페이지가 반환됩니다.

재지정을 지정하는 JSON 파일을 생성하고 이 파일을 서버에 업로드할 수 있습니다. 서버는 수신 요청 URL을 처리할 때 JSON 파일을 사용합니다.

두 가지 종류의 재지정 규칙으로 수신 URL을 새 위치로 재지정할 수 있습니다.

- [단순 문자열-대-문자열 일치](#)
- [간소화된 와일드 카드 문자 일치](#)

### 단순 문자열-대-문자열 일치

단순 문자열 일치 및 교체를 위해 명시적 URL을 지정한 다음 대상 URL에 직접 매핑하여 각 URL을 재지정할 수 있습니다.

다음 표는 문자열-대-문자열 일치의 일부 샘플을 보여줍니다.

소스 URL	대상 위치 URL
/index.html	/home.htm
/products/widget	/items/knickknack
/index?page=widgets	/items/widgets

문자열-대-문자열 매핑은 이해하고 테스트하기 쉽습니다. 단순 문자열 일치와 맵 조회를 사용하여 빠르게 규칙이 평가됩니다.

그러나 URL 질의 매개변수와 관련해 유연성이 거의 없습니다. 정확히 일치해야 합니다. 여분의 URL 매개변수나 다른 순서의 매개변수가 있으면 규칙이 일치하지 않습니다.

## 간소화된 와일드 카드 문자 일치

간소화된 와일드 카드 문자 일치는 많은 URL을 규칙에 일치시키는 동시에, 결과를 얻는 데 필요한 정규 표현식 역추적 양을 제한할 수 있습니다.

정규 표현식은 작성하기 복잡하고 불량한 구조의 표현식은 무한정 시간 동안 평가될 수 있으므로(ReDoS) 두번째 유형의 규칙으로 간소화된 일치 방식을 사용할 수 있습니다. 와일드 카드 문자("\*")를 사용하여 수신 URL에서 0개 이상의 문자를 일치시키고 'wildcard' 키워드를 인덱스 값으로 사용하여 URL의 수신 부분을 재지정된 URL로 복사합니다.

다음 표는 간소화된 와일드 카드 문자 일치의 일부 샘플을 보여줍니다.

소스 URL	대상 위치 URL
/old/*	/new/<\$wildcard(1)\$>
/dispatch.asp?page=* & facet=Lang*	/page<\$wildcard(1)\$>/<\$wildcard(2)\$>

간소화된 와일드 카드 문자 일치는 단순 문자열-대-문자열 일치보다 URL 일치 능력이 더 강력하지만 무한한 정규 표현식 처리 없이 수행합니다. 구문은 간단합니다. URL에 패턴 일치를 사용하여 수많은 URL을 한 패턴에 수용할 수 있습니다.

와일드 카드 문자 규칙은 정규 표현식을 기반으로 하므로 단순 문자열 일치보다 평가 속도가 약간 느립니다. 수많은 규칙은 일반적 페이지 전달 성능에 성능 저하를 일으킬 수 있습니다.



## 사이트 재지정 추가

사이트 URL이 변경되면 재지정은 한 URL(소스)을 다른 URL(대상)로 전달합니다. 그러면 사용자 책갈피 및 검색 엔진 순위를 보존할 수 있습니다.

두 가지 유형의 재지정을 사용할 수 있습니다.

- 영구 재지정 - 301 HTTP 서비스 응답 코드 사용
- 임시 재지정 - 302 HTTP 서비스 응답 코드 사용

redirect.json 파일을 업로드하려면 다음과 같이 하십시오.

1. 편집을 위해 사이트를 엽니다.
2. 사이드바에서  을 누르고  재지정을 누릅니다.
3. 업로드할 파일 선택을 누르고 사용할 파일로 이동하여 선택한 다음 확인을 누릅니다.
4. 업데이트를 게시하는 경우 변경사항이 게시되어 사용할 수 있게 됩니다.

## JSON 파일에 재지정 규칙 지정

JSON 파일에 URL 재지정 규칙을 지정할 수 있습니다.

다음 형식을 JSON 파일에 사용하여 URL 재지정 규칙을 지정하십시오.

```
{
  "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 파일의 바깥쪽 포함 구조는 배열입니다. 배열에는 규칙 인스턴스가 포함됩니다.

"string" 규칙이 먼저 평가되고 이어서 "wildcard" 규칙이 순서대로 평가됩니다. 규칙 중 하나가 일치하면 후속 규칙 평가가 중단되고 해당하는 재지정이 생성됩니다.

각 규칙에는 다음 속성이 있습니다.

- "comment" 속성은 규칙 평가에 영향을 주지 않는 선택적 문자열입니다. 메모 또는 주석이 포함됩니다.
- "expression" 속성은 수신 사이트-상대 URL과 일치시키는 필수 문자열입니다. 와일드 카드 문자 규칙에서 별표(\*) 토큰은 0개 이상의 문자와 일치합니다.
- "location" 속성은 재지정 위치 또는 목적지를 나타내는 필수 문자열입니다. 재지정은 전체 또는 상대 URL일 수 있습니다.
- "code" 속성은 재지정을 실행할 때 사용할 HTTP 응답 코드를 제공하는 선택적 정수입니다. 값은 다음 정수 중 하나여야 합니다.
  - 301: 리소스가 영구적으로 이동되었음을 나타냅니다. "code" 속성이 생략된 경우 기본값입니다.
  - 302: 리소스가 일시적으로 이동되었음을 나타냅니다.
- "type" 속성은 재지정 규칙의 유형을 나타내는 선택적 문자열입니다. 값은 다음 문자열 중 하나여야 합니다.
  - "string"은 표현식이 전체 입력 URL과 정확히 일치하는 빠른 규칙을 지정합니다.
  - "wildcard"는 수많은 URL과 일치할 수 있는 와일드 카드 문자 규칙을 지정합니다. 속성이 생략된 경우 기본값입니다.

### 위치 토큰

위치 토큰을 사용하여 재지정 위치를 설계할 수 있습니다. 다음의 각 위치 토큰은 재지정을 지정하는 데 도움이 됩니다.

- <\$urlPath\$>: 일치 URL의 경로 부분입니다.
- <\$urlQueryString\$>: 일치 URL의 전체 URL 질의 문자열입니다.
- <\$urlQueryStringExcept(name1,name2)\$>: 일치 URL의 전체 URL 질의 문자열에서 명명된 매개변수를 뺀 값입니다.
- <\$wildcard(N)\$>: 일치 URL에서 일치 와일드 카드 문자의 1부터 시작하는 인덱스입니다. (정규 표현식의 \1..\9와 비슷합니다.)

- <\$name\$>: 명명된 질의 문자열 매개변수의 값입니다. 예를 들어, 입력 시 질의 문자열 msmith: ?page=42가 있으면 위치에 <\$page\$>를 사용하여 '42'를 위치에 넣을 수 있습니다.

### 제한사항

다음 제한사항은 redirects.json 파일 전체와 파일에 포함된 규칙에 적용됩니다.

- Oracle Content Management에서 허용된 최대 전체 파일 크기는 250KB입니다.
- redirects.json 파일의 최대 규칙 수는 1,000개입니다.
- 규칙의 최대 "expression" 길이는 1,000자입니다.
- 규칙의 최대 "location" 길이는 2,000자입니다.
- 와일드 카드 문자 규칙 표현식의 최대 '\*' 토큰 수는 10개입니다.

### 문자열 일치 예제

규칙:

```
{
    "type": "string",
    "expression": "/old/page.jsp?id=material&type=glass",
    "location": "/new/<$id$>.htm"
}
```

다음 URL은 규칙과 일치합니다.

```
/old/page.jsp?id=material&type=glass
```

- 결과 위치는 /new/material.htm입니다.
- 질의 문자열을 포함한 전체 URL이 일치합니다.
- <\$id\$>가 위치에 사용되지만 가능한 질의 문자열이 하나만 일치할 수 있으므로 이 예제에는 필요하지 않습니다. 위치는 /new/material.htm으로 작성되었을 수 있습니다.

다음 URL은 규칙과 일치하지 않습니다.

- /old/page.jsp  
(규칙 표현식은 일치시킬 질의 문자열을 제공합니다.)
- /old/page.jsp?id=material&type=glass&index=2  
(후보 URL의 추가 &index=2는 규칙 표현식과 정확히 일치하지 않습니다.)
- /old/page.jsp?type=glass&id=material  
(질의 문자열 매개변수의 순서는 "string" 규칙에서 일치해야 합니다.)

### 와일드 카드 문자 일치 예제

규칙:

```
{
    "type": "wildcard",
    "expression": "/old/*/pages/*?id=*&item=sheet-*",
```

```

        "location": "/new/<$id$>/<$wildcard(4)$>.html"
    }

```

다음 URL은 규칙과 일치합니다.

- /old/phones/android/pages/info.asp?id=XT1045&item=sheet-specs
  - 결과 위치는 /new/XT1045/specs.html입니다.
  - URL의 경로 부분이 일치하므로 일치 조건에 대해 질의 문자열도 검사됩니다.
  - 이 예제의 매개변수는 규칙 표현식과 매개변수 순서가 일치하지만 이것이 필수는 아닙니다.
- /old/phones/android/pages/info.asp?item=sheet-specs&id=XT1045
  - 결과 위치는 /new/XT1045/specs.html입니다.
  - URL의 경로 부분이 물음표(?) 앞의 규칙 표현식과 일치하므로 매개변수도 일치하는지 검사합니다.
  - 규칙 표현식에서 매개변수가 다른 순서로 나열되어 있지만 매개변수는 개별적으로 일치합니다.
- /old/phones/android/pages/info.asp?id=XT1045&item=sheet-specs&unrelated=thing
  - 결과 위치는 /new/XT1045/specs.html입니다.
  - URL의 경로 부분이 일치하므로 일치 조건에 대해 질의 문자열도 검사됩니다.
  - 후보 URL에 추가 &unrelated=thing 매개변수가 있지만 규칙 표현식에 명명된 질의 매개변수가 일치하므로 규칙이 일치하는 것으로 간주됩니다.
  - unrelated 매개변수는 위치에서 <\$unrelated\$>와 같이 토큰으로 사용 가능하며 규칙 일치에 기여하지 않더라도 thing 값을 갖습니다.

다음 URL은 일치하지 않습니다.

- /old/pages/info.jsp  
(URL의 경로 부분이 규칙 표현식의 경로 부분과 일치하지 않습니다.)
- /old/phones/android/pages/info.asp  
(URL의 경로 부분이 규칙 표현식의 경로 부분과 일치하지만, 규칙 표현식의 질의 매개변수가 일치하지 않습니다.)
- /old/phones/android/pages/info.asp?id=cellular  
(URL의 경로 부분이 규칙 표현식의 경로 부분과 일치하지만, 규칙 표현식의 모든 질의 매개변수가 일치하지는 않습니다.)

### 토큰 배열 정의

여러 vanity URL을 지원하는 재지정을 구성할 때 유용하도록 redirects.json 파일 내에 토큰 정의 배열을 생성할 수도 있습니다. 그러면 수신 URL의 특성에 따라 적절히 재지정할 수 있습니다.

다음 형식을 redirects.json 파일에 사용하여 재지정 규칙 URL에서 사용할 토큰을 정의하십시오.

```

{
    "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"
    }
  ],
}

```

tokenDefinitions에는 다음 속성이 있습니다.

- "token": 정의할 토큰의 이름입니다.
- "type": 다음 중 하나입니다.
  - "hostmatch" - 수신 URL의 호스트 값과 일치합니다.
  - "pathmatch" - 수신 URL의 경로명 값과 일치합니다.
  - "querymatch" - 수신 URL의 질의 값과 일치합니다.
- "expression": 일치에 사용되는 표현식입니다. 와일드 카드 문자가 지원됩니다.
- "value": 토큰에 사용되는 값입니다.
- "flags": flags 값이 caseinsensitive로 설정되지 않는 한, 기본적으로 표현식 일치는 대소문자를 구분합니다.

토큰 값을 계산할 때 tokenDefinitions 배열이 순서대로 열거됩니다. 첫번째 일치하는 정의가 사용됩니다. 토큰 정의가 토큰을 충족하지 않으면 대신 빈 문자열이 사용됩니다. 편의상 성능을 위해 자주 사용되는 토큰이 tokenDefinitions 목록의 맨 위에 놓여야 합니다.

tokenDefinitions에는 다음 제약 조건이 있습니다.

- 최대 250개의 토큰 정의를 생성할 수 있습니다.
- token 이름은 100자 미만이어야 합니다.
- expression은 최대 10개의 와일드 카드 문자를 포함할 수 있습니다.
- expression은 1000자 미만이어야 합니다.
- value는 1000자 미만이어야 합니다.

## 예제

예를 들어, 다음 redirects.json 파일이 있을 수 있습니다.

```
{
  "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"
    },
  ],
}
```

이 경우 규칙의 location 속성에 <\$pathPrefix\$> 토큰이 있습니다. pathPrefix 토큰은 tokenDefinitions 섹션에 정의됩니다. 수신 URL이 "vanity.com"과 일치하면 pathPrefix 값은 /fashion으로 설정됩니다. 이는 location 응답에 사용되며 결과적으로 /fashion/about/new-privacy-policy.html이 됩니다.

첫번째 vanity 도메인 URL이 http://example.com/legacy-privacy-policy.html이라고 가정해 보겠습니다. 이것은 첫번째이자 유일한 재지정 규칙과 일치합니다.

이 규칙에 대해 선언된 location은 <\$pathPrefix\$>/about/new-privacy-policy.html입니다. 이 상황에서 <\$pathPrefix\$> 토큰을 평가해야 합니다. 이를 위해 tokenDefinitions 배열이 열거되어 일치항목을 찾습니다.

첫번째 토큰 정의가 고려됩니다. token이 원하는 토큰이므로 더 평가됩니다. 표현식 vanity.com이 수신 URL의 example.com과 일치하지 않으므로 이 정의는 요구사항을 충족하지 않으며 열거가 계속됩니다.

이 시점에서 더 이상 토큰 정의가 없으므로 <\$pathPrefix\$> 토큰 값에 빈 문자열이 사용됩니다. 이 재지정으로 반환된 최종 위치는 /about/new-privacy-policy.html입니다.

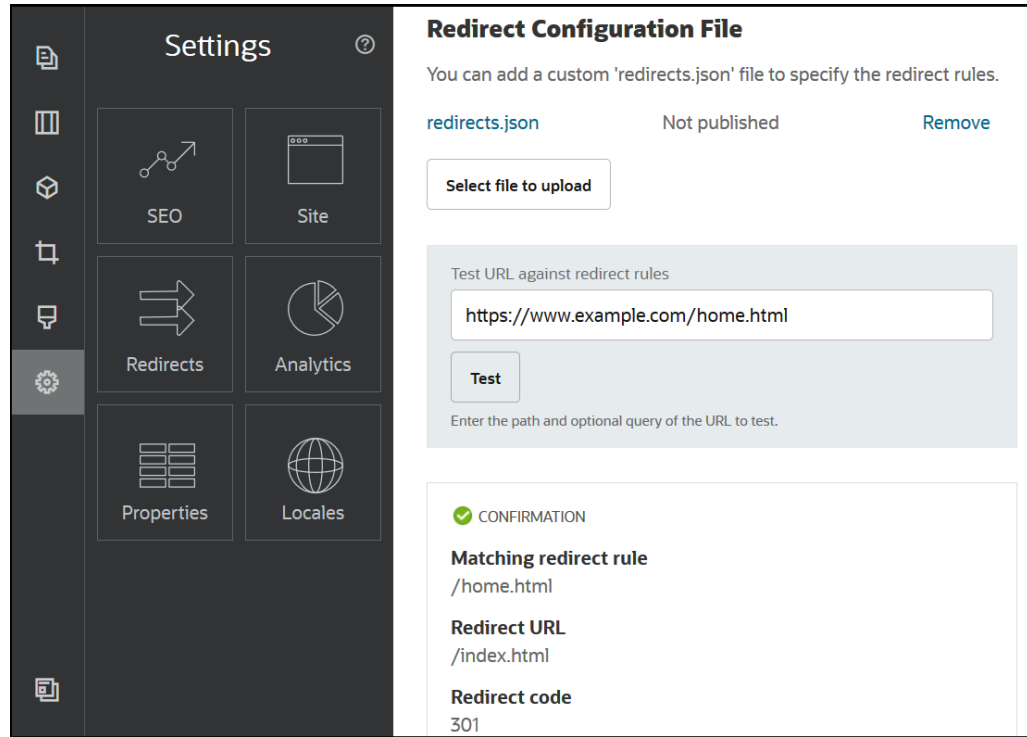
두번째 vanity 도메인 URL이 http://vanity.com/legacy-privacy-policy.html이라고 가정해 보겠습니다. 첫번째 URL과 마찬가지로 이 규칙에 대해 선언된 location은 <\$pathPrefix\$>/about/new-privacy-policy.html입니다. 이 상황에서 <\$pathPrefix\$> 토큰을 평가해야 합니다. 이를 위해 tokenDefinitions 배열이 열거되어 일치항목을 찾습니다.

첫번째 토큰 정의가 고려됩니다. 이전처럼 token이 원하는 토큰이므로 더 평가됩니다. 표현식 vanity.com이 수신 URL의 vanity.com과 일치하므로 이 정의는 요구사항을 충족하며 /fashion 값이 토큰 값으로 사용됩니다.

토큰의 일치항목을 찾았으므로 토큰 정의 배열의 열거가 정지되고 최종 위치는 /fashion/about/new-privacy-policy.html로 계산됩니다.

## 사이트 재지정 테스트



사이트를 편집할 때 설정 패널을 열고 재지정을 눌러 사이트 재지정을 테스트할 수 있습니다. 테스트할 URL을 입력하고 테스트를 누릅니다.



## 사이트에 재지정 규칙 파일 업로드

Oracle Content Management의 사이트에 재지정 규칙을 업로드할 수 있습니다.

redirect.json 파일을 사이트에 업로드하려면 다음과 같이 하십시오.

1. 편집을 위해 사이트를 엽니다.
2. 사이드바에서  을 누르고  을 누릅니다.
3. **업로드할 파일 선택**을 누르고 사용할 파일로 이동하여 선택한 다음 **확인**을 누릅니다.
4. 업데이트를 게시하는 경우 변경사항이 게시되어 사용할 수 있게 됩니다.

## 사이트 URL 매핑

Oracle Content Management를 사용하여 사이트를 생성하고 게시한 후 DNS(도메인 이름 시스템)를 구성하면 이 사이트에 등록된 도메인 이름(예: www.mysite.com)으로 액세스할 수 있습니다.

DNS(도메인 이름 시스템)는 도메인 이름을 사이트 위치 또는 표준 이름(CNAME)에 매핑하여 방문자가 웹 페이지를 찾을 수 있는 위치를 지정합니다.

도메인 이름을 매핑하려면 다음 항목이 필요합니다.

- Oracle Content Management 인스턴스의 URL입니다. 일반적으로 다음 형식입니다.



`service-tenant.documents.datacenter.oraclecloud.com`

- 도메인 이름 등록을 통해 등록된 도메인 이름  
예: `www.example.com`, `www.example.com/subdomain`과 같은 하위 도메인일 수도 있습니다.
- CDN(콘텐츠 전달 네트워크) 제공자에 대한 계정(예: Akamai)입니다. Oracle Content Management는 Akamai와의 통합을 제공합니다. 인스턴스에 대해 Akamai를 구성하려면 오라클 고객지원센터에 문의하십시오.

Oracle Content Management에서 제공한 Akamai가 아닌 고유의 CDN을 사용하려면 아래 단계를 수행하십시오.

각 DNS(도메인 이름 시스템) 제공자마다 다른 웹 인터페이스와 다른 CNAME 레코드 업데이트 단계를 사용합니다. 아래 단계는 필요한 정보와 일반적인 이행 단계를 제공합니다.

사이트 URL을 도메인 이름에 매핑하려면 다음과 같이 하십시오.

1. 도메인의 콘텐츠 전달 네트워크 제공자로부터 SSL(Secure Sockets Layer) 인증서를 요청합니다. 예: `https://www.example.com`
2. 다음과 같이 콘텐츠 전달 네트워크를 구성합니다.
  - a. 콘텐츠 전달 네트워크가 도메인으로 수신되는 모든 요청을 수락하고 보안 프로토콜(https)을 사용하여 해당 요청을 전달하도록 합니다.
  - b. 원점이 Oracle Content Management의 도메인을 가리키도록 합니다.

`service-tenant.documents.datacenter.oraclecloud.com`

3. 콘텐츠 전달 네트워크 제공자가 제공한 Edge Server에 도메인 이름을 매핑하기 위해 DNS 서버 영역 파일이 변경되도록 합니다.

*domain CNAME CDN Server*

4. 업데이트가 전달되기를 기다립니다. DNS 서비스에 따라 2시간에서 48시간까지 걸릴 수 있습니다.

변경사항이 전달된 후 도메인 이름을 사용하여 사이트에 액세스할 수 있습니다. 예를 들어, 다음과 같습니다.

`https://www.mysite.com/site_name`

기본적으로 Oracle Cloud REST API for Content Management 끝점은 사이트에 대해 제공된 표준 URL을 사용하는 경우 제공됩니다. 예를 들어, 폴더 및 파일 목록 구성요소는 REST API를 사용하여 폴더 및 파일 작업을 수행합니다. 사용자정의 URL을 사용하는 경우 도메인 이름이 포함된 끝점에 대한 액세스 권한을 가지는지 확인하십시오. 예를 들어, 다음과 같습니다.

`https://www.mysite.com/documents`

# 16

## 사이트 성능 향상

브라우저 캐시를 활용하여 브라우저에서 콘텐츠 전달 및 렌더링 성능을 향상시킬 수 있습니다. ATF(Above the Fold) 렌더링은 웹 사이트 렌더링을 향상시킬 수도 있습니다.

- 캐시를 활용하여 성능 향상
- ATF(Above the Fold) 렌더링

### 캐시를 활용하여 성능 향상

콘텐츠 항목, 디지털 자산 및 사이트를 제공하는 경우 브라우저에서 콘텐츠 전달 및 렌더링 성능을 향상하려면 방문자의 브라우저 캐시를 충분히 활용해야 합니다.

사이트, 테마, 콘텐츠 항목 및 디지털 자산은 방문자의 브라우저 캐시에 일정 시간 동안 캐싱됩니다. 사이트, 테마, 콘텐츠 항목 또는 디지털 자산이 업데이트되면 URL의 cache-buster 키가 변경되므로 브라우저는 다른 URL을 인출하여 새 항목을 가져와야 합니다.

캐시 키로 현재 리소스만 참조하여 브라우저 캐시 사용을 관리할 수 있습니다. 캐시 키는 URL에 포함되지만 흔히 있듯이 물리적 위치(폴더)가 아닌 논리적 요소입니다. 캐시 키를 변경해도 리소스를 찾기 위해 다른 물리적 위치를 가리키지 않으며, 단순히 현재 버전의 리소스를 인출하도록 서버에 알립니다.

CSS, JS 및 이미지 파일과 같은 정적 리소스도 있고 페이지 데이터, 사이트 데이터 및 콘텐츠 항목 데이터와 같은 동적 리소스도 있습니다. 웹 사이트 구축을 위한 다음 5가지 범주의 리소스가 있습니다.

- 제품 리소스 – 새 버전의 제품이 출시되거나 패치될 때마다 업데이트되는 제품의 일부인 리소스.
- 사이트 리소스 – structure.json, 페이지 데이터, 이미지와 같이 사이트의 일부인 리소스. 사이트가 게시될 때 업데이트됩니다. 컨트롤러는 다음 텍스트에서 설명합니다.
- 테마 리소스 – 레이아웃, CSS, 이미지와 같이 테마의 일부인 리소스. 테마가 게시될 때 업데이트됩니다.
- 구성요소 리소스 – 사용자정의 구성요소의 일부인 리소스. 여기에는 HTML, JS, CSS와 구성요소를 이루는 이미지 파일이 포함됩니다. 구성요소가 게시될 때 업데이트됩니다. 한 구성요소가 변경되어 재게시할 경우 캐시 키는 모든 구성요소에 대한 단일 키이므로 모든 구성요소에 대해 키가 변경됩니다.
- CaaS 리소스 – 콘텐츠 항목 및 디지털 항목을 제공하는 리소스. 콘텐츠 항목을 게시/재게시하거나 모음 대상이 변경될 때 업데이트됩니다.

다음 항목은 Oracle Content Management 런타임 캐싱과 사이트 작성기 캐싱을 설명합니다.

- 런타임 캐싱
- 사이트 작성기 캐싱

## 런타임 캐싱

런타임 동안 Oracle Content Management Cache-Control 헤더는 15일로 설정됩니다. 캐시 키가 모든 리소스의 URL에 추가됩니다.

URL이 동일하면 브라우저는 가능한 경우 로컬 캐시에서 리소스를 제공합니다. 리소스가 업데이트되면 캐시 키가 URL에서 업데이트되고, 브라우저는 서버에 새 요청을 보내고 로컬 캐시를 업데이트합니다.

캐시 키를 포함하는 컨트롤러도 1분 동안 캐싱됩니다. 이로 인해 업데이트된 캐시 키가 최대 1분 동안 보이지 않습니다.

런타임에 서버는 제품, 사이트, 테마, 구성요소 및 CaaS 리소스에 대한 최신 캐시 키와 함께 controller.html을 반환합니다. 다음과 같은 키 스크립트가 controller.html에 추가됩니다.

```
<script type="text/javascript">
    var SCSCacheKeys = {
        product: '123',
        site: '456',
        theme: '789',
        component: '012',
        caas: '345'
    };
</script>
```

이 키를 controller.js에서 사용하여 다음 표의 항목과 같은 URL을 구성합니다.

리소스 유형	예제
제품 리소스	<i>/sitePrefix/productCacheKey/_sitesclouddelivery/...</i>  <i>/mySite/_cache_947d/_sitesclouddelivery/</i>
테마 리소스	<i>/sitePrefix/themeCacheKey/_themesdelivery/themeName/...</i>
구성요소 리소스	<i>/sitePrefix/compCacheKey/_compdelivery/compName/...</i>
사이트 리소스	<i>/sitePrefix/siteCacheKey/content/...</i> <i>/sitePrefix/siteCacheKey/structure.json</i> <i>/sitePrefix/siteCacheKey/pages/100.json</i>

리소스 유형	예제
CaaS 리소스	<code>RegularCaaSUrl?cacheKey=caasCacheKey</code>

이와 같은 URL에 캐시 키를 삽입함으로써 Oracle Content Management는 URL을 효과적으로 변경하여 브라우저가 업데이트된 리소스(브라우저는 실제로 새 리소스라고 생각)를 강제 로드하도록 합니다.

#### 주:

보안 사이트의 경우 제품, 테마, 구성요소 리소스만 캐싱되고 사이트나 CaaS 콘텐츠는 캐싱되지 않습니다.

## 사이트 작성기 캐싱

사이트 작성기에서 정적 리소스는 15일 동안 캐싱됩니다.

사이트 작성기를 사용하는 경우 제품, 테마 및 구성요소 리소스가 캐싱됩니다. (사이트 및 CaaS 리소스는 캐싱되지 않습니다.) 사이트 작성기를 실행하거나 새로고침할 때 테마 및 구성요소 캐시 키가 재생성됩니다.

테마나 구성요소를 변경하고 이 변경사항을 사이트 작성기에 표시하려면 사이트 작성기를 새로고침(F5)해야 합니다.

## ATF(Above the Fold) 렌더링

ATF 렌더링은 웹 사이트가 실제보다 빠르게 로드되는 것처럼 보입니다. 목표는 먼저 페이지의 보이는 부분을 모두 렌더링하고, 그 다음 사용자가 아래로 스크롤하기 전에 처음에 보이지 않은 페이지의 나머지 부분을 렌더링하는 것입니다.

슬롯에 "Above the Fold"를 지정하면 탭에 아이콘이 표시됩니다.

이 새로운 방식으로 슬롯을 렌더링하려면 다음과 같이 `scs-atf`로 표시해야 합니다.

```
<div class="scs-slot scs-atf" id="headline"></div>
```

구성요소는 렌더링이 완료되면 렌더러에게 알려야 합니다. 미리 정의된 구성요소는 기본적으로 이를 수행합니다. 사용자정의 구성요소는 추가 호출을 실행하여 다음을 수행해야 합니다.

1. 렌더링이 완료될 때까지 기다릴 것을 렌더러에게 알립니다.
2. 완료되면 렌더러에게 알립니다.

1번의 경우 사용자정의 구성요소의 `appinfo.json` 파일에 다음 속성을 추가하십시오.

```
"initialData": {
  . . .
  "customRenderComplete": true,
  . . .
}
```

2번의 경우 구성요소의 `render.js` 파일에서 다음을 호출하여 완료되면 렌더러에게 알려야 합니다.

```
SitesSDK.setProperty('renderComplete', true);
```

ATF 슬롯의 모든 구성요소가 적시에 완료되었다고 보고하지 않으면 렌더러는 2초간 기다렸다가 페이지의 나머지 부분을 계속합니다. 이 시간이 충분히 길지 않다면 다음 전역 변수를 페이지 템플릿에 선언하여 시간을 늘릴 수 있습니다.

```
var SCSAtfPassTimeout = 3000;
```

### 주:

시간은 밀리초 단위이므로 이 예제는 시간 초과를 3초로 설정합니다.

API는 ATF 프로세스에 대한 진단 데이터를 제공합니다. 다음 메소드를 디버그 콘솔에서 호출하거나, 필요한 경우 페이지에서 액세스할 수 있습니다.

```
SCSRenderAPI.getRenderMetrics();
```

예를 들어, 다음과 같습니다.

```
{currentTime: 16243.400000000001, renderStartTime: 264.36,
atfPassEndTime: 306.535, mainPassStartTime: 316.475, mainPassEndTime:
331.385000000000005, ...}
```

1. atfComponentCount:13
2. atfPassEndTime:306.535
3. completionCount:23
4. completionRecords:Array(23)
  1. 0:{atf: true, componentId: "a7afdd33-3fbb-4329-bc1b-6be60056a995", time: 280.065}
  2. 1:{atf: true, componentId: "edfcfcb4-b0d3-422f-aa59-5c925bbbebee", time: 283.54}
  3. 2:{atf: true, componentId: "c1c3aec8-e52f-406c-8c29-ab69c05877ed", time: 283.560000000000006}
  4. 3:{atf: true, componentId: "b3a31dc6-62a1-44d9-9c80-bdb2c5bedaaa", time: 284.130000000000005}
  5. 4:{atf: true, componentId: "c05aala2-c11c-4ef5-9051-4799c5bee24a", time: 284.155000000000003}
  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

# IV부

## 사이트 개발

이 부분에서는 템플릿, 테마, 스타일, 구성요소 및 기타 유용한 항목을 이해하고 개발함으로써 사이트 개발을 시작하는 방법을 자세히 설명합니다. 다음 장이 포함됩니다.

- 디자인 및 스타일 사용자정의
- 배경 사용 이해
- 트리거 및 작업 설정
- 템플릿 개발
- 테마 개발
- 레이아웃 개발
- 구성요소 개발



# 17

## 디자인 및 스타일 사용자정의

Oracle Content Management의 각 테마에는 테마의 기본 사이트에서 사용된 페이지의 모양과 구성요소의 스타일 설정을 지정하는 디자인이 있어야 합니다.

- [디자인 정보](#)
- [디자인 파일](#)
- [대화 목록 스타일 사용자정의](#)
- [폴더 목록 및 파일 목록 스타일 사용자정의](#)
- [소셜 표시줄 아이콘 사용자정의](#)
- [Oracle Intelligent Advisor에 대한 인터뷰 스타일링 확장 구성](#)

### 디자인 정보

Oracle Content Management의 각 테마에는 테마에 사용된 페이지의 모양과 구성요소의 스타일 설정을 지정하는 디자인이 있어야 합니다.

사용자가 사이트를 생성할 때 기본적으로 테마가 포함된 템플리트를 선택해야 합니다. 사이트에 다른 테마를 생성하거나 사용하도록 선택할 수 있습니다.

테마를 생성할 때 페이지 레이아웃 디자인과 함께 사용자에게 제공할 각 구성요소 유형(단락, 제목, 이미지 등)에 대해 설정 패널에 표시할 사용 가능한 스타일을 지정해야 합니다. 디자인 내에 여러 스타일이 있어서 다양한 구성요소에 대한 기본 설정을 지정할 수 있습니다. 텍스트 글꼴 및 글꼴 크기, 이미지 프레임 등을 선택한 다음, 명명된 디자인으로 저장합니다. 디자인의 항목은 렌더링 시 페이지 레이아웃의 모양에 영향을 미칩니다.

*Building Sites with Oracle Content Management*의 사이트 페이지 작업을 참조하십시오.

### 디자인 파일

두 개의 파일 `design.json` 및 `design.css`가 테마의 기본 디자인 파일로 간주됩니다.

- `design.json`은 구성요소의 스타일을 지정합니다.
- `design.css`는 클래스 값(예: 색상 및 글꼴)에 대한 정의를 제공합니다.

이 파일은 테마 구조의 `/designs/default/` 디렉토리에 있습니다.

#### design.json 파일

`design.json` 파일의 구조는 다음과 같습니다.

```
{
  "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": []
    }
}
}
}

```

각 "styles":[] 항목은 특정 구성요소에 대한 스타일 목록을 포함할 수 있습니다. 예를 들어, 제목 구성요소는 다음 기본 스타일을 제공합니다.

```

"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"
}
]

```

 주:

구성요소에 대한 사용자정의 스타일을 생성하여 design.json의 스타일에 매핑하는 경우 시스템에서 제공한 전체 이름을 사용할 필요가 없습니다. 사용할 문자열을 지정하면 됩니다. 예를 들어, 이름에 "COMP\_STYLE\_BOX" 대신 간단히 "Box"를 사용하십시오. 그러면 "COMP\_STYLE\_BOX" 대신 "Box" 이름만 해당 구성요소에 대한 설정 패널의 스타일 탭에 목록으로 나타납니다.

다음과 같이 이름 값은 사용자 인터페이스에 표시할 실제 단어에 매핑됩니다.

```
"COMP_STYLE_FLAT": "Flat",
"COMP_STYLE_HIGHLIGHT": "Highlight",
"COMP_STYLE_DIVIDER": "Divider",
```

**design.css 파일**

design.css 파일은 클래스 값에 대한 정의를 제공합니다. 다음은 몇 가지 예제입니다.

```
.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; }
```

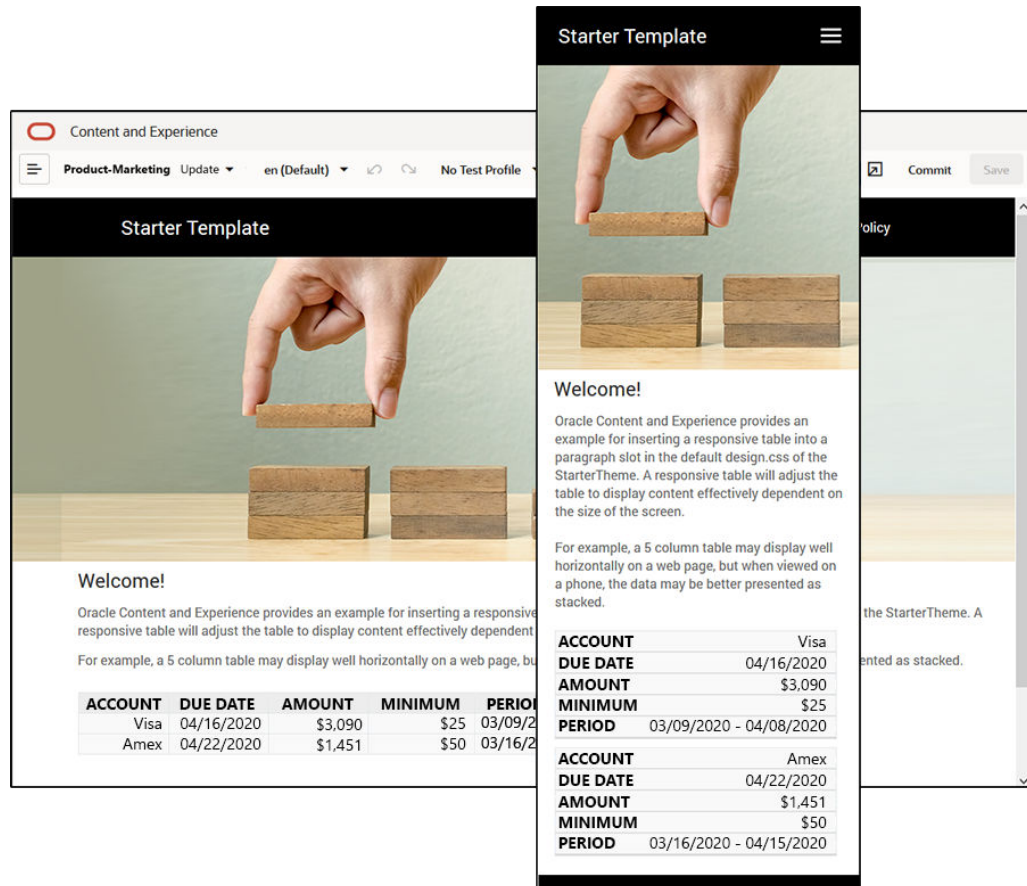
```
.scs-button-default-style .scs-button-button:hover {
  background: #f7f8f9;
  border: 1px solid #c4ced7;
  color: #0572ce;
  box-shadow: inset 0 1px 0 #f7f8f9;
  text-shadow: 0 1px 0 #f7f8f9; }
```

```
.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; }
```

## 반응형 테이블 디자인

Oracle Content Management는 모바일 장치에 표시될 때 행 데이터가 누적되도록 하는 단락 구성요소 내 반응형 테이블의 CSS 예를 제공합니다.

반응형 테이블은 화면 크기에 따라 효과적으로 콘텐츠가 표시되도록 테이블을 조정합니다. 예를 들어, 5열 테이블은 웹 페이지에서 가로로 잘 표시될 수 있지만, 전화에서 표시될 때는 데이터가 누적형으로 제공되는 것이 효과적일 수 있습니다. 반응형 테이블이 올바르게 작동하려면 머리글 행이 필요합니다.



다음과 같이 생성된 HTML에서는 값이 열 머리글 텍스트와 일치하는 각 테이블 셀에 추가된 data-label 속성이 있습니다.

```
<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>

```

속성이 각 셀에 있을 경우 아래 CSS 규칙을 적용할 때 TD는 서로의 위에 누적됩니다.

```

.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;
}

```

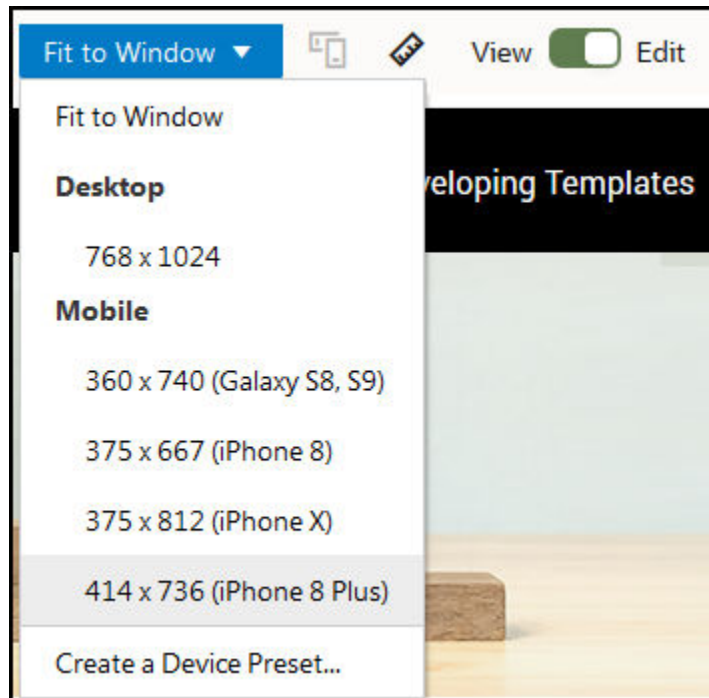
아래 매체 규칙은 화면 크기를 질의하며 화면 크기가 767픽셀 미만인 경우에만 적용됩니다.

```
@media screen and (max-width: 767px) {
```

CSS를 사용하여 삽입된 테이블이 반응형 테이블이 되도록 허용하는 코드 예는 제공된 StarterTheme의 기본 design.css에 있습니다. StarterTheme에서 사이트를 구축하는 경우 단락 슬롯에 삽입되는 테이블이 기본적으로 반응형 테이블로 설정됩니다. 테이블을 삽입하려면 다음과 같이 하십시오.

1. Oracle Content Management에서 사이트를 열고 **편집** 모드로 토글합니다.
2. 새 업데이트를 생성하거나 기존 업데이트를 선택하여 수정합니다.
3. 새 단락 구성요소를 페이지로 끌어와서 테이블을 삽입하려는 위치를 누르거나, 테이블을 삽입하려는 기존 단락을 누릅니다.
4. 서식 있는 텍스트 도구모음에서  을 누르고 테이블 속성을 설정합니다. 첫번째 행을 머리글로 선택하고, 사용이 예상되는 가장 작은 화면에서도 효과적으로 작동하도록 테이블 너비를 조정해야 합니다. 예를 들어, 전화에서 사이트를 표시해야 할 경우 기본 너비인 767픽셀은 너무 넓어서 전화 화면에서 완전히 표시되지 않을 수 있습니다. 테이블 너비를 보다 작은 크기(예: 300픽셀)로 설정하거나 너비를 100%로 설정하는 것이 좋습니다.

완료되면 **보기** 모드로 다시 토글하고 화면 너비가 767픽셀 미만인 보기 옵션을 선택하여 결과 미리보기를 수행합니다. 편집 모드에서는 테이블의 반응형 작동이 지원되지 않으므로 보기 모드여야 합니다.



다른 테마에서 사이트를 구축할 때 반응형 테이블을 사용하려면 StarterTheme design.css의 코드를 사용하려는 테마의 design.css 파일에 복사해야 합니다.

1. StarterTheme design.css의 코드를 복사하려면 Oracle Content Management의 측면 탐색에서 **개발자**를 누릅니다.
2. **모든 테마 보기**를 누릅니다.
3. **StarterTheme**를 선택하고 **열기**를 누릅니다.
4. 디자인을 눌러 폴더를 열고 **기본값**을 누릅니다.
5. StarterTheme에서 design.css 파일을 선택하고 **다운로드**를 누릅니다.
6. 텍스트 편집기에서 파일을 열고 *An example CSS of how to render a table responsively* 주석으로 시작되는 파일의 섹션을 찾습니다.
7. 다음 주석까지 코드를 선택하여 복사합니다.

```
/**
 * 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. 단계를 반복하여 수정할 템플리트의 `design.css` 파일을 다운로드하고 템플리트를 연 다음 복사된 코드를 파일에 붙여 넣습니다.
9. 변경사항을 저장하고 수정된 `design.css` 파일을 수정하려는 테마에 새 개정으로 업로드합니다.

## 대화 목록 스타일 사용자정의

`design.css` 파일에 선택기를 추가하여 대화 목록 구성요소의 스타일을 사용자정의할 수 있습니다.

다음 CSS 선택기를 사용하여 대화 목록 구성요소의 스타일을 사용자정의합니다.

선택기 이름	설명
<code>.scs-convo-list-cust .scs-convo-list-container</code>	구성요소의 가장 바깥쪽 DIV
<code>.scs-convo-list-cust .scs-convo-list-title</code>	선택할 때 목록에서 대화의 제목
<code>.scs-convo-list-cust .scs-convo-list-line-separator</code>	목록 제목과 목록 사이의 구분 기호
<code>.scs-convo-list-cust .scs-convo-list-convo-title</code>	목록에서 대화의 제목

선택기 이름	설명
.scs-convo-list-cust .scs-convo-list-convo-line-separator	각 대화 사이의 구분 기호
.scs-convo-list-cust .scs-convo-list-active	선택할 때 목록에서 대화의 제목
.scs-convo-list-cust .scs-convo-list-convo-posts	대화의 게시물 수
.scs-convo-list-cust .scs-convo-list-convo-unread	대화의 읽지 않은 메시지 수
.scs-convo-list-cust .scs-convo-list-convo-updated	대화의 최종 업데이트 날짜
.scs-convo-list-cust .scs-convo-list-no-convo-msg	목록이 비어 있을 때 메시지
.scs-convo-list-cust .scs-convo-list-no-auth-msg	사용자 권한 부여 없이 공용 사이트에 대화 목록을 렌더링할 때 메시지

*Building Sites with Oracle Content Management*의 스타일 및 서식 사용을 참조하십시오.

### 예제

이 샘플은 대화 목록의 글꼴, 스타일 및 색상 변경과 함께 사용자정의된 CSS 사용을 보여줍니다.



다음 코드는 샘플 생성에 사용된 사용자정의 CSS를 보여줍니다.

```
.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;
}

```

## 폴더 목록 및 파일 목록 스타일 사용자정의

design.css 파일에 선택기를 추가하여 폴더 목록 및 파일 목록 구성요소의 스타일을 사용자정의할 수 있습니다.

폴더 목록 구성요소를 사용하여 Oracle Content Management 계정에서 지정된 폴더 내에 폴더를 나열할 수 있습니다. 폴더 목록은 자동으로 페이지의 파일 목록 구성요소 및 문서 관리자와 통신하여 폴더 목록에서 선택된 폴더의 파일을 표시합니다.

파일 목록 구성요소를 사용하여 Oracle Content Management 계정에서 지정된 폴더의 파일 뷰를 제공할 수 있습니다. 파일 목록은 자동으로 페이지의 폴더 목록 구성요소와 통신하여 폴더 목록에서 선택된 폴더의 파일을 표시합니다.

### 폴더 목록 CSS 선택기

다음 CSS 선택기를 사용하여 폴더 목록 구성요소의 스타일을 사용자정의합니다.

선택기 이름	설명
.scs-folder-list-cust .scs-folder-list-container	구성요소의 가장 바깥쪽 DIV
.scs-folder-list-cust .scs-folder-list-folder-title	폴더 이름
.scs-folder-list-cust .scs-folder-list-line-separator	폴더 이름과 하위 폴더 목록 사이의 구분 기호
.scs-folder-list-cust .scs-folder-list-sub-folder-title	하위 폴더 이름

선택기 이름	설명
.scs-folder-list-cust .scs-folder-list-sub-folder-title-active	선택할 때 하위 폴더 이름
.scs-folder-list-cust .scs-folder-list-no-folder-msg	표시할 하위 폴더가 없을 때 메시지

### 파일 목록 CSS 선택기

다음 CSS 선택기를 사용하여 파일 목록 구성요소의 스타일을 사용자정의합니다.

선택기 이름	설명
.scs-file-list-cust .scs-file-list-container	구성요소의 가장 바깥쪽 DIV
.scs-file-list-cust .scs-file-list-folder-title	폴더 이름
.scs-file-list-cust .scs-file-list-line-separator	폴더 이름과 파일 목록 사이의 구분 기호
.scs-file-list-cust .scs-file-list-row	파일에 대한 정보가 포함된 행
.scs-file-list-cust .scs-file-list-left-col	구성요소의 왼쪽 섹션에 있는 파일의 작은 그림
.scs-file-list-cust .scs-file-list-mid-col	파일 이름, 설명, 최종 수정 및 크기가 포함된 구성요소의 중간 섹션
.scs-file-list-cust .scs-file-list-file-title	앱의 중간 섹션에 있는 파일 이름
.scs-file-list-cust .scs-file-list-file-desc	앱의 중간 섹션에 있는 파일 설명
.scs-file-list-cust .scs-file-list-file-lastModified	파일의 최종 수정
.scs-file-list-cust .scs-file-list-file-size	최종 수정에서 세로 구분 기호가 있는 파일의 크기
.scs-file-list-cust .scs-file-list-file-size-no-sep	세로 구분 기호가 없는 파일의 크기(최종 수정은 표시되지 않음)
.scs-file-list-cust .scs-file-list-right-col	앱의 오른쪽 섹션
.scs-file-list-cust .scs-file-list-file-download-icon	앱의 오른쪽 섹션에 있는 다운로드 아이콘
.scs-file-list-cust .scs-file-list-no-file-msg	표시할 파일이 없을 때 메시지

*Building Sites with Oracle Content Management*의 파일 목록 및 폴더 목록을 참조하십시오.

## 소셜 표시줄 아이콘 사용자정의

테마 기본 사이트의 소셜 표시줄에 사용할 사용자정의 소셜 아이콘을 생성할 수 있습니다.

사이트의 소셜 표시줄에 나타나는 소셜 아이콘은 사이트의 테마 디자인에 의해 결정됩니다. 사이트의 테마를 변경하면 테마와 함께 소셜 아이콘이 변경됩니다. Facebook, Twitter, LinkedIn, Google+, YouTube에 대한 공통 소셜 아이콘은 Oracle Content Management 테마에 포함되어 있습니다.

design.json 및 design.css 파일을 편집하여 소셜 아이콘 구성요소에 사용자정의의 소셜 아이콘을 추가할 수 있습니다.

### design.json 파일

design.json 파일에서 다음 샘플 코드와 같이 이름 및 클래스 구조를 사용하여 새 아이콘을 지정할 수 있습니다.

```
"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 파일

design.css 파일에서 다음 샘플 코드와 같이 이름 및 url 지정을 사용하여 새 아이콘을 추가할 수 있습니다.

```
.scs-facebook-icon {
  background-image: url("facebook.png"); }
.scs-twitter-icon {
  background-image: url("twitter.png"); }
.scs-linkedin-icon {
  background-image: url("linkedin.png"); }
```

## Oracle Intelligent Advisor에 대한 인터뷰 스타일링 확장 구성

기업 분위기로 모양을 통일하기 위해 OIA(Oracle Intelligent Advisor)(이전의 Oracle Policy Automation) 인터뷰를 스타일링할 수 있습니다.

design.css 파일에서 다음 CSS 클래스 선택기를 구성할 수 있습니다. 모든 클래스 선택기에는 접두어 "scs-opainterview-"가 붙습니다.

각 클래스는 지정된 구성요소의 모든 스타일을 정의합니다. 클래스는 완전히 제어되며 기존 OIA 스타일은 사용되지 않습니다.

선택기	적용 대상	설명
interview	interview	인터뷰 영역은 머리글, 바닥글 및 탐색 영역을 포함한 전체 인터뷰 콘텐츠로 구성됩니다.
interviewContent	인터뷰 콘텐츠	인터뷰 콘텐츠 영역에는 화면 제목과 컨트롤이 포함되지만 머리글, 바닥글 및 탐색 영역은 제외됩니다.
screenTitleBlock	화면 제목 블록	화면 제목 블록은 화면 제목을 포함하는 영역과 함께 해당 행에 포함된 다른 위젯(예: 화면 드롭다운 목록 및/또는 다음 및 뒤로 단추)으로 구성됩니다.
screenTitle	화면 제목	화면 제목 영역은 화면 제목을 포함하는 영역입니다.
nextButton	다음 단추	다음 단추입니다.
backButton	뒤로 단추	뒤로 단추입니다.
restartButton	재시작 단추	재시작 단추입니다.
exitButton	종료 단추	종료 단추입니다.
header	header	머리글 영역입니다.
footer	footer	바닥글 영역입니다.
question	질문 텍스트	질문 텍스트의 스타일입니다.
control	컨트롤의 컨테이너	컨트롤을 포함하는 요소의 스타일입니다.
label	레이블 컨트롤	레이블 컨트롤의 스타일입니다.
controlError	모든 컨트롤	오류 텍스트 컨테이너의 스타일입니다.
controlErrorText	모든 컨트롤	오류 텍스트 범위의 스타일입니다.
textInput	단일 행 텍스트 상자, 비밀번호 및 마스크	텍스트 입력 컨트롤의 스타일입니다.
textAreaInput	다중 행 텍스트 상자	다중 행 텍스트 입력 컨트롤의 스타일입니다.
calendarInput	달력	달력 입력 컨트롤의 스타일입니다. 달력 아이콘의 색상을 변경할 수 있는 iconColor 필드와 달력 아이콘을 표시해야 하는지 여부를 나타내는 keepIcon 필드를 지원합니다.
dropDownInput	드롭다운 목록	드롭다운 목록 입력 컨트롤의 스타일입니다.
filterDropDownInput	필터링된 드롭다운 목록	필터링된 드롭다운 목록 입력 컨트롤의 스타일입니다. 드롭다운 화살표의 색상을 변경할 수 있는 iconColor 필드를 지원합니다.
listInput	고정 목록	고정 목록 입력 컨트롤의 스타일입니다.
radioInput	라디오 단추	라디오 단추 입력 컨트롤의 스타일입니다. borderColor 및 fillColor 옵션으로 스타일링할 수 있습니다. iconType 속성은 사용된 아이콘의 유형을 변경할 수 있습니다. 현재 지원되는 대체 옵션은 'tick' 및 'fill' 뿐입니다.

선택기	적용 대상	설명
checkboxInput	체크박스	체크박스 입력 컨트롤의 스타일입니다. borderColor 및 fillColor 옵션으로 스타일링할 수 있습니다. iconType 속성은 사용된 아이콘의 유형을 변경할 수 있습니다. 현재 지원되는 대체 옵션은 'square' 및 'fill' 뿐입니다.
autocompleteInput	사용자정의 검색	customSearch 확장을 사용할 때 autocomplete 필드의 스타일입니다.
captchaInput	CAPTCHA 입력 필드	사용자가 CAPTCHA를 입력하는 입력 필드의 스타일입니다.
signatureInput	서명 컨트롤	서명 컨트롤의 스타일입니다. 서명 펜 잉크를 변경할 수 있는 추가 inkColor 필드를 지원합니다.
explanationHeader	설명 컨트롤	설명 컨트롤에 대한 최상위 레벨의 확장 가능한 헤더의 스타일입니다.
explanationText	설명 컨트롤	확장된 설명 컨트롤 텍스트의 스타일입니다.
signatureClearButton	서명 컨트롤	서명 컨트롤에서 지우기 단추의 스타일입니다.
uploadAddButton	업로드 컨트롤	업로드 컨트롤에서 추가 단추의 스타일입니다.
entityRemoveButton	엔티티 수집 컨트롤	엔티티 수집 컨트롤입니다.

### design.css에 정의된 스타일 확장과 예제

```
.scs-opainterview-interviewContent {
  background-color: beige;
}
.scs-opainterview-screenTitleBlock {
  background-color: bisque;
}
.scs-opainterview-screenTitle {
  font-style: italic;
  font-size: 20px;
}
.scs-opainterview-nextButton {
  color: darkgreen;
}
.scs-opainterview-backButton {
  color: crimson;
}
.scs-opainterview-question {
  color: green;
}
.scs-opainterview-control {
  background-color: cornflowerblue;
}
.scs-opainterview-label {
  color:aqua;
}
.scs-opainterview-textInput {
  color: red;
  cursor:crosshair;
}
```

```
}  
.scs-opainterview-radioInput {  
    background-color: pink;  
}  
.scs-opainterview-checkboxInput {  
    cursor: pointer;  
}
```

# 18

## 배경 사용 이해

사이트 페이지와 페이지상 개별 슬롯에 대해 배경색과 이미지를 지정할 수 있습니다. 페이지, 슬롯, 구성요소의 배경은 서로 위에 층층이 쌓입니다. 예를 들어, 슬롯의 배경을 지정하면 페이지에 지정된 배경 위에 쌓입니다.

- [배경 및 테마 정보](#)
- [배경이 구현되는 방법](#)
- [설정이 저장되는 위치](#)

## 배경 및 테마 정보

배경 기능은 주로 사이트 페이지 및 슬롯에 사용하기 위한 것입니다. 페이지 및 슬롯에 설정된 배경은 테마 레벨에서 개발자가 아닌 사용자가 구성한 것입니다.

테마 슬롯에서 배경 효과를 구성할 수 있지만, 이 효과는 사이트를 편집할 때 페이지에 설정된 배경 효과를 무효화할 수 있습니다.

배경 기능은 테마 또는 구성 테마 파일을 변경하지 않습니다. 테마는 페이지의 배경 설정에 의해 변경될 수 없으며, 사이트의 동일한 레이아웃에 기반한 다른 페이지는 소스 페이지의 배경 설정을 상속하지 않습니다. 테마를 통해 배경을 추가하거나 변경하려면 테마를 업데이트해야 합니다.

사용자는 사이트의 모든 페이지와, 동일한 레이아웃을 사용하는 다른 페이지에서 다양한 배경을 구성할 수 있습니다. 이 작업은 테마 내에서 불가능합니다. 테마가 페이지의 배경을 지정할 수 있지만 사용자의 배경 설정이 이 지정을 무효화할 수 있습니다. (배경 설정은 특정 페이지에만 무효화를 적용하며 어떤 식으로든 테마 자체를 수정하지 않습니다.)

사용자가 사이트 페이지 및 슬롯에 설정할 배경 스타일을 테마가 무효화하지 않도록 주의하십시오. 여러 가지 방법으로 발생할 수 있습니다.

- <body> 및 슬롯에서 요소 기반 "style" 속성을 사용할 때 사이트의 페이지 배경 설정이 테마에 의해 무효화될 수 있습니다.
- 테마 CSS(중속형 스타일시트) 파일에서 테마 배경 스타일을 "!important"로 표시할 때 사이트의 페이지 배경 설정이 테마에 의해 무효화될 수 있습니다.

*Building Sites with Oracle Content Management*의 배경 또는 테마 변경을 참조하십시오.

## 배경이 구현되는 방법

페이지 및 슬롯 배경 기능을 구현하기 위해 Oracle Content Management는 페이지 <head>에 CSS 스타일시트를 동적으로 생성합니다.

스타일 선택기는 페이지 배경 설정을 위한 태그 기반 선택기(body)입니다. 슬롯 배경 설정을 위한 ID 기반 선택기의 경우 슬롯 ID가 사용됩니다.

예를 들어, 페이지의 배경색을 설정하면 다음 CSS 마크업이 페이지 <head>에 생길 수 있습니다.

```
body
{
```

```

        background-color: #fa7c9d;
    }

```

마찬가지로, 슬롯의 배경 이미지를 설정하면 다음 CSS 마크업이 페이지 <head>에 생길 수 있습니다.

```

#PageFooter
{
    background-image: url("footer_image.png");
}

```

이 구현은 본문 태그 또는 슬롯 요소의 "style" 속성에 직접 지정된 스타일이 스타일시트의 <head> 코드에 구성된 설정을 무효화할 수 있음을 의미합니다.

### ! 중요사항:

테마 개발자는 요소 기반 스타일로 배경 설정을 무효화하지 않도록 주의해야 합니다.

*Building Sites with Oracle Content Management*의 배경 또는 테마 변경을 참조하십시오.

## 설정이 저장되는 위치

페이지를 렌더링할 때 배경 설정은 페이지 <head> 코드의 "style" 태그에 동적으로 작성됩니다.

배경 설정은 페이지 모델 파일(예: <pageid>.json)에 지속됩니다. 특히 페이지 배경 설정은 `properties.styles` 섹션에 저장되고 슬롯 배경 설정은 `slots[<slot_id>].styles` 섹션에 저장됩니다.

배경 설정은 페이지 JSON 파일, 구체적으로 이 대표 샘플에 표시된 "styles" 내에 저장됩니다.

```

{
  "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-ad71-51f2fa465cef\">
</div>
</div>
<div class=\"scs-col\" style=\"width: 50%;\">
<div id=\"95eb0fd6-bcfc-4e5e-ba67-a5c8c5d9c315\">
</div>
</div>
</div>\",
  "styles": [
    "background-image: url([!--$SCS_CONTENT_URL--]/
oracle-cloudworld.jpg)",
    "background-position: center",
    "background-size: cover",
    "background-repeat: no-repeat",
    "background-origin: padding-box",
    "background-clip: border-box",
    "background-color: transparent"
  ]
},
"componentInstances":
{
  "dedda3a8-615d-44ad-ad71-51f2fa465cef":
  {
    "type": "scs-title",
    "data": {
      "alignment": "fill",
      "backgroundColor": "",
      "borderColor": "#808080",
      "borderRadius": 0,
      "borderStyle": "none",
      "borderWidth": 1,
      "fontColor": "#333333",
      "fontFamily": "'Helvetica Neue', Helvetica, Arial, sans-
serif",
      "fontSize": 24,
      "marginBottom": 5,
      "marginLeft": 5,
      "marginRight": 5,
      "marginTop": 5,
      "styleClass": "",
      "useStyleClass": "true",
      "userText": "<div>My Test Title</div>\n",
      "width": 400
    }
  },
}
```

```
"95eb0fd6-bcfc-4e5e-ba67-a5c8c5d9c315":  
{  
  "type": "scs-image",  
  "data": {  
    "styleClass": "",  
    "useStyleClass": "true",  
    "imageUrl": "[!--$SCS_CONTENT_URL--]/example.jpg",  
    "defaultImageUrl": "/components/comp/images/  
default_image.png",  
    "style": "",  
    "imageWidth": 0,  
    "borderStyle": "none",  
    "borderWidth": 1,  
    "borderColor": "black",  
    "borderRadius": 0,  
    "altText": "My Image",  
    "title": "My Title",  
    "caption": "My Caption",  
    "imageAlignment": "center",  
    "imageHref": "",  
    "imageTarget": "_self",  
    "marginTop": 0,  
    "marginRight": 0,  
    "marginBottom": 0,  
    "marginLeft": 0,  
    "linkType": "scs-link-no-link"  
  }  
}
```

# 19

## 트리거 및 작업 설정

한 구성요소 내의 트리거가 다른 구성요소에 작업을 호출하도록 구성요소(인라인 프레임으로 렌더링된 구성요소 포함) 간의 통신을 구성할 수 있습니다.

- [트리거 및 작업 정보](#)
- [트리거 설정](#)
- [작업 설정](#)

### 트리거 및 작업 정보

한 구성요소 내의 트리거가 다른 구성요소에 작업을 호출하도록 구성요소(인라인 프레임으로 렌더링된 구성요소 포함) 간의 통신을 구성할 수 있습니다.

트리거는 Oracle Content Management 구성요소 상호 통신의 일부입니다. 모든 구성요소는 원하는 수의 트리거를 발생시킬 수 있습니다. 구성요소는 트리거에 대한 페이로드를 제공할 수 있으며, 이는 트리거가 발생할 때 실행되는 작업으로 전달됩니다. 각 트리거에 대해 실행할 작업을 선택할 수 있습니다. 함께 작동하도록 제작된 구성요소들은 사용자 상호작용 없이 자동으로 트리거를 발생시켜 다른 구성요소에 작업을 실행할 수 있습니다.

기본 프로세스는 다음과 같습니다.

1. 트리거 등록
2. 트리거 발생
3. 작업 등록
4. 작업을 실행하여 설정 확인

예를 들어, 단추 구성요소를 사용하여 페이지 구성요소 표시/숨기기 및 메시지 표시와 같은 작업을 하나 이상 수행할 수 있습니다. 한 구성요소에 사무실 위치 목록을 포함하고 목록의 위치를 누르면 다른 구성요소에 위치 세부정보가 표시됩니다.

사용자정의된 구성요소의 경우 트리거 및 작업은 구성요소 등록 데이터의 일부이며 구성요소 구현에는 속하지 않습니다. 등록 데이터에 있는 "triggers": [], 및 "actions": [], 항목은 구성요소가 지원하는 트리거 및 작업 목록을 포함합니다. 실제 구문은 로컬 및 원격 구성요소와 동일하며 위치와 검색 방법만 다릅니다.

*Building Sites with Oracle Content Management*의 트리거 및 작업 사용을 참조하십시오.

### 트리거 설정

한 구성요소에는 다른 구성요소에 작업을 실행할 트리거가 포함될 수 있습니다. 구성요소에서 발생시킬 트리거를 등록해야 합니다.

구성요소는 트리거에 대한 페이로드를 제공하며, 이는 트리거가 발생할 때 실행되는 작업으로 전달됩니다. 각 트리거에 대해 실행할 작업을 선택할 수 있습니다. 함께 작동하도록 제작된 구성요소들은 사용자 상호작용 없이 자동으로 트리거를 발생시켜 작업을 실행할 수 있습니다.

## 트리거 등록

사용자정의 구성요소의 경우 트리거는 구성요소 등록 데이터의 일부로 등록됩니다. 트리거를 추가하려면 구성요소가 지원하는 각 트리거로 "triggers" 속성 배열을 업데이트합니다. 또한 트리거가 지원하는 페이로드를 지정해야 사용자 인터페이스를 생성하여 사용자가 페이로드 내의 값을 작업에서 지원하는 속성에 매핑할 수 있습니다.

1. appinfo.json 파일을 편집하고 "triggers":[], 항목을 검토합니다.

```
"triggers": [{
  "triggerName": "helloWorldWhoAreYou",
  "triggerDescription": "Show Who I Am",
  "triggerPayload": [{
    "name": "whoAreYou",
    "displayName": "Who I Am"
  }]
}],
```

2. 파일을 사이트 서버와 동기화합니다.

이 샘플 트리거 항목에서 triggerName("helloWorldWhoAreYou")을 정의했습니다. 이름 값은 고유해야 합니다. 그런 다음 사용자 인터페이스 대화상자에서 트리거를 표시할 때 사용할 설명("Show Who I Am")을 제공했습니다. 마지막으로, 트리거에 대한 단일 값 페이로드를 정의했습니다. 사용자는 이 페이로드의 항목을 선택하여 작업의 필드에 매핑할 수 있습니다.

트리거가 등록되면 구성요소의 설정 패널에서 링크 탭으로 이동할 때 트리거를 보고 선택할 수 있어야 합니다.

## 트리거 발생

트리거는 언제든지 구성요소에 의해 발생할 수 있습니다. 일반적으로 트리거는 단추 누르기 또는 테이블의 행 선택과 같은 사용자 상호작용에 의해 발생합니다. 구성요소는 REST API 호출로 인해 데이터가 변경될 때와 같은 조건에 준하여 트리거를 발생시킬 수 있습니다. 트리거가 발생할 때 원하는 수의 작업을 실행할 수 있습니다.

다음은 트리거 발생 방법의 예입니다.

1. render.js 파일을 편집하고 Sites SDK를 호출하여 트리거를 발생시키는 JavaScript 함수를 viewModel 객체에 추가합니다.

```
self.raiseTrigger = function (triggerName) {
  SitesSDK.publish(SitesSDK.MESSAGE_TYPES.TRIGGER_ACTIONS, {
    'triggerName': 'helloWorldWhoAreYou',
    'triggerPayload': { "whoAreYou": "This is " + self.whoAreYou()
+ "!" }
  });
};
```

2. 사용자 인터페이스에 항목을 추가하여 트리거를 발생시키는 함수를 호출하고(-edit template.html) </div> 앞에 단추를 놓습니다.

```
<button data-bind="click raiseTrigger">Who Am I?</button>
```

3. render.js 파일을 Oracle Content Management 인스턴스 서버와 동기화하거나 업로드합니다.

ViewModel 객체에서 단추를 누를 때 호출되는 JavaScript 함수를 생성했습니다. 이 함수는 Sites SDK를 호출하여 이 트리거 "helloWorldWhoAreYou"에 대해 정의된 모든 작업을 트리거합니다. 또한 단일 필드 "whoAreYou"가 있는 triggerPayload를 통과합니다. "helloWorldWhoAreYou" 및 "whoAreYou" 값은 이전 단계에서 트리거를 등록할 때 입력한 값과 일치합니다.

### 주:

작업을 실행할 때 미리 정의된 순서는 없습니다. 각 작업은 나열된 순서대로 호출되지만, 다음 작업을 호출하기 전에 현재 작업이 완료되기를 기다리지는 않습니다. 작업이 비동기로 호출되는 경우 다음 작업을 실행하기 전에 현재 작업이 완료되지 않을 수 있습니다.

## 작업 설정

작업 등록을 활용하도록 구성요소를 설정하여 구성요소 내의 작업을 실행할 페이지 위에 놓을 수 있습니다.

### 작업 등록

작업은 트리거가 발생할 때 구성요소에서 호출됩니다. 구성요소는 원하는 수의 작업을 등록하고 작업이 지원하는 페이로드를 정의할 수도 있습니다. 사용자가 작업을 선택하면 페이로드를 채워서 작업으로 전달할 수 있습니다.

트리거 등록과 마찬가지로, 테마의 appinfo.json 등록 데이터에 구성요소가 지원하는 작업을 등록할 수 있습니다.

다음은 작업 등록 방법의 예입니다.

1. 구성요소의 appinfo.json 파일을 편집하고 "actions":[], 항목을 업데이트합니다.

```
"actions": [{
  "actionName": "helloWorldChangeWhoIAm",
  "actionDescription": "Change Who I Am",
  "actionPayload": [{
    "name": "whoAreYou",
    "description": "Who are you?",
    "type": {
      "ojComponent": {
        "component": "ojInputText"
      }
    },
    "value": ""
  }]
}]
```

2. 작업이 등록되면 구성요소의 설정 패널에 있는 링크 탭에서 트리거를 누를 때 호출되는 작업 대화상자에서 볼 수 있습니다.

### 작업 실행

작업이 등록되면 구성요소 내의 작업을 실행할 페이지 위에 구성요소를 놓을 수 있습니다. 구성요소가 작업을 실행하려면 EXECUTE\_ACTION 메시지를 수신해야 합니다. 이 메시지에는 작업으로 전달된 페이로드도 포함되며, 여기서 예상 값을 추출해야 합니다.

예를 들어, EXECUTE\_ACTION 메시지를 수신하려면 `render.js` 파일을 편집하고 다음 항목으로 `ViewModel` 객체를 업데이트합니다.

```
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);
  }
};
```

작업을 실행하는 JavaScript 함수가 생성되고 EXECUTE\_ACTION 메시지가 발생할 때마다 Sites SDK를 사용하여 함수를 호출합니다.

EXECUTE\_ACTION 메시지가 발생할 때마다 작업이 호출되며, 구성요소는 처리하도록 설계된 작업만 처리할 수 있습니다. 이를 위해 작업 이름을 검사하여 처리할 수 있는 작업인지 확인해야 합니다.

작업 페이로드는 값의 배열입니다. 예제에서는 값이 배열의 첫번째 항목이라고 가정합니다. 일반적으로 배열에서 원하는 페이로드 값을 찾아야 합니다.

 주:

작업 리스너는 콜백이기 때문에 함수를 실행할 때 `ViewModel`에 액세스할 수 있도록 JavaScript 클로저를 사용하거나 함수를 적절히 바인딩해야 합니다.

# 20

## 템플릿 개발

템플릿 패키지에는 사이트 개발 버전, 테마와 페이지 레이아웃, 스타일 및 탐색, 사이트에 사용된 관련 구성요소가 포함됩니다. Oracle Content Management는 사이트 생성에 사용되며 사용자정의 템플릿 생성의 시작점으로 사용 가능한 일련의 템플릿을 제공합니다.

- [템플릿 정보](#)
- [기본 템플릿 구조](#)
- [템플릿 생성](#)
- [템플릿 익스포트](#)
- [템플릿 임포트](#)
- [스타터 템플릿 작업](#)
- [Bootstrap 또는 웹 사이트 디자인 템플릿에서 사이트 템플릿 생성](#)
- [Developer Cloud Service로 템플릿 개발](#)

## 템플릿 정보

템플릿은 샘플 페이지와 콘텐츠가 담긴 사이트, 테마와 스타일, 탐색, 이미지 자산은 물론 연관된 구성요소까지 사용자가 웹 사이트 생성을 시작할 때 필요한 모든 것을 포함합니다.

Oracle Content Management는 사이트 생성에 사용할 수많은 템플릿을 제공합니다. 이 템플릿은 일반적으로 서비스를 초기화할 때 관리자에 의해 설치됩니다. *Oracle Content Management 관리*의 사이트 설정 구성을 참조하십시오.

새 사이트를 생성할 때마다 템플릿을 선택해야 합니다. 템플릿은 사이트 및 구성요소를 테마와 결합하여 파트너 포털 또는 마케팅 캠페인과 같은 기능이나 솔루션을 추진할 수 있습니다.

미리 정의된 템플릿 목록은 [템플릿 이해](#)를 참조하십시오.

개발자와 사용자 모두 새 템플릿을 생성하고 기존 템플릿을 수정 후 바꿀 수 있지만, 개발자의 주요 업무 중 하나는 새 템플릿을 디자인하는 것입니다. 이 프로세스는 기본적으로 다음 단계로 구성됩니다.

1. JET 스타터 템플릿과 같은 기존 템플릿을 복사하여 새 템플릿을 생성합니다. 템플릿과 연관된 테마도 얻을 수 있습니다.

예를 들어, 다음과 같습니다.

```
cec create-template My_JET_Template -f JETStarterTemplate
```

2. .zip 파일의 템플릿을 개발 환경으로 익스포트합니다.
3. 템플릿 패키지의 파일을 열고 내용을 변경합니다.
4. 수정된 템플릿 패키지를 .zip 파일로 생성합니다.
5. Oracle Content Management 인터페이스를 사용하여 새 템플릿을 인스턴스로 임포트합니다.

6. 다른 사람이 사용할 수 있도록 템플릿을 공유합니다.

다른 방법으로, Oracle Content Management 인터페이스에서 페이지 레이아웃 및 자산을 추가/수정하여 템플릿을 수정하고, 사이트 구조를 확장하고, 사이트 페이지에 구성요소를 추가하고, 이 템플릿의 테마를 사용하는 모든 사이트에 표시할 시드 콘텐츠를 추가할 수 있습니다.

템플릿 관리를 참조하십시오.

## 기본 템플릿 구조

템플릿의 기본 구조에는 사이트(자산, 레이아웃, 페이지, 콘텐츠), 관련 테마 및 사용자정의 구성요소가 포함됩니다.

사이트를 생성할 때 템플릿을 선택하여 사이트 구조와 초기 콘텐츠, 테마와 디자인 및 레이아웃 사양, 사용자정의 구성요소를 제공해야 합니다.

기본 폴더 및 파일을 보여주는 다음 예제와 같이 템플릿은 특정 구조로 구성됩니다.

```

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

```

구성요소와 테마의 폴더 및 파일은 이 설명서의 다른 곳에서 설명합니다. [구성요소 개발 정보](#) 및 [기본 테마 구조](#)를 참조하십시오.

### 설명:

- JET 스타터 템플릿 또는 스타터 템플릿(테마의 사본 사용)을 사용하지 않는 한, 일반적으로 테마는 템플릿 간에 공유됩니다. [스타터 템플릿 작업](#)을 참조하십시오.
- 테마는 더 이상 사이트를 포함하지 않습니다. 사이트 폴더 및 파일은 `/template` 폴더에 있습니다.

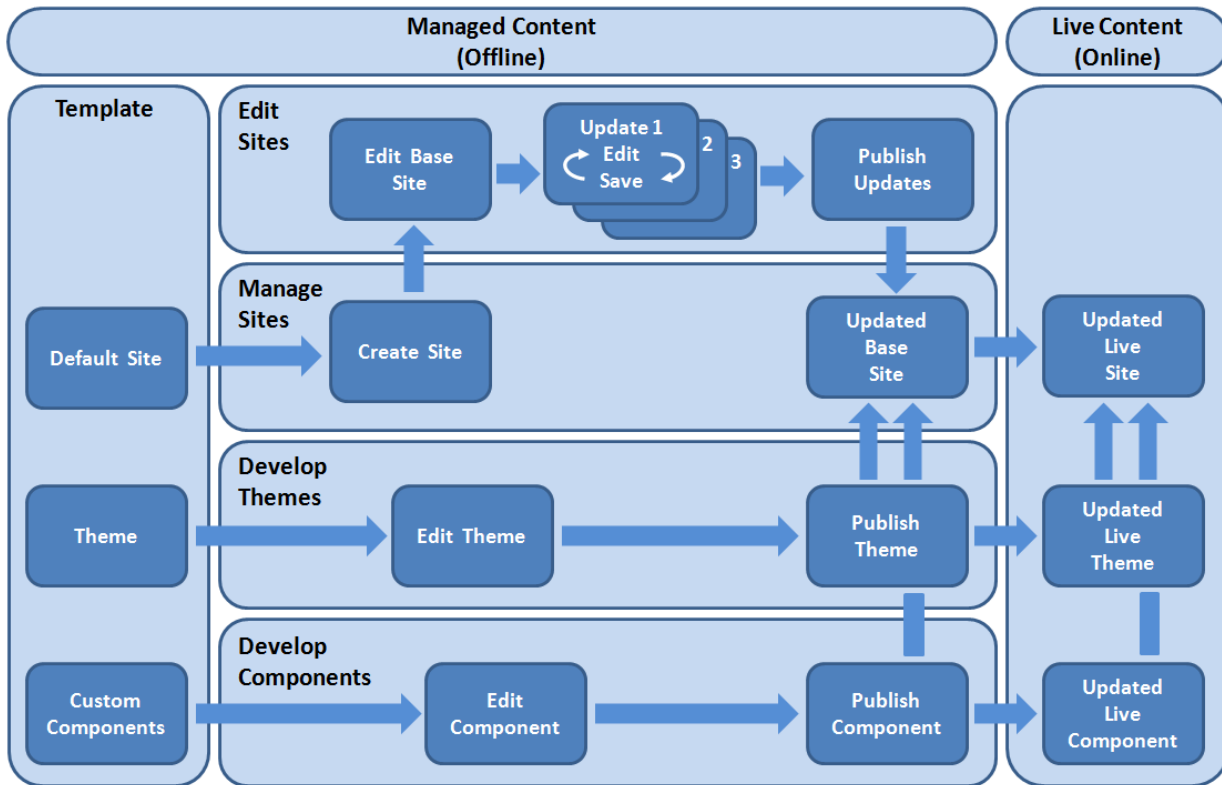
`template_name/template` 폴더에는 사이트의 폴더 및 파일이 포함됩니다.

- `assets`: 사용자 인터페이스에서 템플릿 세부정보 페이지에 표시되는 이미지를 포함합니다.
- `content`: 사이트에서 사용된 관리 콘텐츠를 포함합니다.
- `layouts`: 지금은 사용되지 않습니다.
- `pages`: 모든 페이지 JSON 파일과 데이터를 포함합니다. `nnn.json` 형식(여기서 `nnn`은 페이지 ID)을 사용합니다.
- `variants`: 사이트의 모든 업데이트에 대한 세부정보를 포함합니다.
- `_folder.json`: 사이트 작성자, 사이트 이름, 항목 GUID, 간단한/자세한 사이트 설명과 같은 템플릿의 메타데이터를 포함합니다.
- `_folder_icon.png`: 사용자 인터페이스에서 사이트를 나타냅니다.
- `componentsused.json`: (사용되지 않음) 사이트 내에 사용된 사용자정의 구성요소를 기록합니다. 역 호환성을 위해서만 유지보수됩니다.
- `controller.html`: 브라우저에 사이트를 표시하는 키 코드를 포함합니다. 이 파일을 변경하려면 오라클은 인터페이스의 사이트 설정을 통해 작업을 수행할 것을 권장합니다. 오프라인으로 파일을 수정할 수 있습니다. [컨트롤러 파일 사용자정의](#)를 참조하십시오.
- `siteinfo.json`: 사이트의 다른 메타데이터와 함께 사이트 이름 및 관련 테마 이름을 식별합니다. 이 파일은 수정하지 마십시오.
- `structure.json`: 사이트의 페이지 계층(상위 및 하위 페이지)을 정의합니다. Render API를 사용하여 사이트의 탐색을 설정할 때 트리 구조를 그릴 수 있습니다. [사이트 탐색](#) 및 [Render API 참조](#)를 참조하십시오.

## 템플릿 생성

다른 사이트의 시작점으로 사용할 사이트가 있는 경우 해당 사이트에서 템플릿을 생성할 수 있습니다. 기존 템플릿을 복사하고 사본을 변경하여 새 템플릿을 생성할 수도 있습니다.

기존 사이트에서 템플릿을 생성하는 경우 새 템플릿은 사이트 복사본을 기본 사이트로 사용합니다. 템플릿은 사이트에 사용된 테마와 사이트 페이지에 사용된 사용자정의 구성요소를 참조합니다. 테마와 사용자정의 구성요소는 템플릿에 복사되지 않지만, 사이트와 동일한 방법으로 참조됩니다. 템플릿에는 템플릿 생성 당시 사용된 사이트가 반영됩니다. 템플릿 생성에 사용된 사이트에 추가 변경사항이 발생할 경우 템플릿과 함께 저장된 사이트에 반영되지 않습니다.



기존 템플릿을 복사하고 사본 이름을 바꿔서 새 템플릿을 생성하는 경우 사본을 변경하십시오. 템플릿을 복사할 때 템플릿 공유 정보는 복사되지 않습니다.

템플릿, 테마, 구성요소, 사이트 또는 사이트 페이지에 다음 이름을 사용하지 마십시오: authsite, content, pages, scsTEMPLATE\_\*, \_comps, \_components, \_compsdelivery, \_idcSERVICE, \_sitesCLOUD, \_sitesCLOUDdelivery, themes, themesdelivery. 사이트 페이지에는 다음 이름을 사용할 수 있지만 템플릿, 테마, 구성요소 또는 사이트에는 사용하지 마십시오: documents, sites.

고유의 사용자정의 템플릿(사이트 및 테마 포함)을 생성하려면 사이트의 기본 요소와 관련 테마가 포함된 JET 스타터 템플릿 또는 Oracle Content Management에서 제공한 스타터 템플릿을 사용하는 것이 가장 좋습니다. 스타터 템플릿에는 사이트 페이지에 작성된 정보 및 지침이 있어서 사용자정의 템플릿에서 사이트와 테마를 배치하고 디자인하는 방법을 살펴볼 수 있습니다.

템플릿 관리 및 스타터 템플릿 작업을 참조하십시오.

## 템플릿 익스포트

템플릿을 익스포트하고 오프라인에서 수정한 후 새 템플릿으로 임포트하거나 기존 템플릿을 바꿀 수 있습니다. 또한 템플릿을 익스포트하여 다른 Oracle Content Management 인스턴스로 이동하고 거기서 임포트할 수도 있습니다.


템플릿을 익스포트할 때 반드시 Oracle Content Management의 폴더에 단일 .zip 파일로 템플릿을 복사해야 합니다. 폴더에서 직접 템플릿 패키지를 다운로드하여 압축을 풀고 개별 파일로 작업할 수 있습니다. 템플릿 파일 작업이 완료되면 템플릿 패키지가 포함된 .zip 파일을 생성하여 Oracle Content Management로 임포트하고 원본 템플릿을 겹쳐쓰거나 새로 생성합니다.



주:

템플릿을 익스포트할 때 템플릿 공유 정보는 포함되지 않습니다.

템플릿을 익스포트하려면 다음과 같이 하십시오.

1. Oracle Content Management 측면 탐색에서 **개발자**를 누릅니다.  
개발자 페이지가 표시됩니다.
2. **모든 템플릿 보기**를 누릅니다.  
기존 템플릿 목록이 표시됩니다.
3. 템플릿을 선택하고 마우스 오른쪽 단추 누르기 메뉴에서 **익스포트**를 선택하거나 작업 모음에서  을 누릅니다.
4. 폴더로 이동하거나, **생성**을 누르고 이름과 선택적 설명을 제공한 후 **생성**을 눌러 새 폴더를 생성합니다.  
폴더를 열려면 폴더 아이콘 또는 폴더 이름을 누릅니다.
5. 관련 폴더의 체크박스를 눌러 폴더를 선택하고 **확인**을 누릅니다.  
템플릿 패키지 파일이 템플릿 이름에 .zip 확장자가 붙은 형태로 선택한 폴더에 생성됩니다.

## 템플릿 임포트

템플릿을 익스포트하고 오프라인에서 수정한 후 새 템플릿으로 임포트하거나 기존 템플릿을 바꿀 수 있습니다. 또한 템플릿을 익스포트하여 다른 Oracle Content Management 인스턴스로 이동하고 거기서 임포트할 수도 있습니다.

템플릿을 익스포트할 때 반드시 Oracle Content Management의 폴더에 단일 .zip 파일로 템플릿을 복사해야 합니다. 폴더에서 직접 템플릿 패키지를 다운로드하여 압축을 풀고 개별 파일로 작업할 수 있습니다. 템플릿 파일 작업이 완료되면 템플릿 패키지가 포함된 .zip 파일을 생성하여 Oracle Content Management로 임포트하고 원본 템플릿을 겹쳐쓰거나 새로 생성합니다.

템플릿 패키지를 임포트하려면 다음과 같이 하십시오.

1. Oracle Content Management 측면 탐색에서 **개발자**를 누릅니다.

- 개발자 페이지가 표시됩니다.
2. 모든 템플리트 보기를 누릅니다.  
기존 템플리트 목록이 표시됩니다.
  3. 생성을 누르고 **템플리트 패키지 임포트**를 선택합니다.
  4. 템플리트 패키지를 업로드한 경우 템플리트 패키지가 포함된 폴더로 이동합니다.  
폴더를 열려면 폴더 아이콘 또는 폴더 이름을 누릅니다.  
템플리트 패키지를 아직 업로드하지 않은 경우:
    - a. 템플리트 패키지를 업로드할 폴더로 이동하거나, **새로 만들기**를 누르고 이름과 선택적 설명을 제공하고 **생성**을 눌러서 폴더를 생성합니다.
    - b. **업로드**를 누릅니다.
    - c. 템플리트 패키지를 찾아 선택하고 **열기**를 누릅니다.  
진행률 표시줄에 파일 이름 및 업로드 상태가 표시됩니다.
  5. 파일 이름 옆에 있는 체크박스를 눌러 템플리트 패키지를 선택하고 **확인**을 누릅니다.  
임포트된 템플리트 콘텐츠와 기존 템플리트, 테마 또는 사용자정의 구성요소 간에 충돌이 없으면 템플리트, 관련 테마와 사용자정의 구성요소에 대한 새 Oracle Content Management 폴더가 생성됩니다.
  6. 템플리트, 테마 또는 사용자정의 구성요소 이름/ID가 존재하면 충돌을 해결하라는 메시지가 표시됩니다.  
충돌 특성에 따라 새 템플리트, 테마 또는 사용자정의 구성요소를 생성하는 옵션이 제시되거나, 어떤 경우 기존 템플리트, 테마 또는 사용자정의 구성요소를 임포트 버전으로 겹쳐쓸 수 있습니다.

## 스타터 템플리트 작업

Oracle Content Management에서 제공한 스타터 템플리트의 사본을 사용하여 새 사이트를 테마 및 사용자정의 구성요소와 함께 생성합니다.

스타터 템플리트는 기본 사이트, 레이아웃, 탐색, 샘플 콘텐츠, 테마, 관련 콘텐츠 항목 등 사용자정의 웹 사이트를 구축할 때 필요한 모든 것을 하나의 패키지로 수집합니다. Oracle Content Management에서 제공한 다른 템플리트와 달리, 스타터 템플리트는 새 사이트와 새 테마 생성을 위한 간편한 기본 프레임워크를 제공합니다.

### 주:

일반적으로 테마는 템플리트 간에 공유되지만 스타터 템플리트는 테마의 사본을 사용합니다.

스타터 템플리트 StarterTemplate 및 JETStarterTemplate는 관리자가 서비스 설치 및 구성 중에 템플리트를 사용으로 설정할 때 Oracle Content Management의 다른 템플리트와 함께 제공됩니다. Oracle JET(JavaScript 확장 툴킷) 스타터 템플리트에는 템플리트의 최신 JET 스타일이 있으며 일부 페이지 콘텐츠를 JET 구성요소로 병합하고 JET 기반 템플리트 및 사이트 구축을 위한 스타터 구성요소를 제공합니다.

### 기본 프로세스

다음은 스타터 템플리트 작업의 기본 프로세스입니다.

1. 새 사이트를 생성하고 스타터 템플리트를 선택합니다. 새 스타터 테마가 새 사이트와 함께 생성됩니다.

 주:

신중하게 사이트 이름을 선택하십시오. 사이트에 제공한 이름은 새 테마 이름으로 복제되며, 사이트를 게시한 후 사용자에게 테마가 표시됩니다. 테마 이름은 생성된 후 변경할 수 없습니다.

스타터 사이트 및 테마에는 사이트 및 테마 개발을 시작할 때 필요한 폴더 및 파일 집합이 있습니다.

2. 새 테마를 데스크톱과 동기화합니다. 데스크톱에서 작업하면서 레이아웃 및 정적 자산으로 테마를 확장하고 사용자정의할 수 있습니다. 이것은 테마의 사본이고 템플리트에 특정한 이름을 사용하므로 소스 테마에 영향을 주지 않고 테마를 변경할 수 있습니다.
3. 사이트 작성기에서 사이트를 엽니다. 사이트 작성기를 사용하여 사이트 구조에 추가하고, 구성요소 및 상호작용을 페이지에 추가하고, 새 템플리트의 기본 사이트에 포함될 콘텐츠를 추가할 수 있습니다. 스타터 템플리트에 제공된 사이트 콘텐츠를 재사용하거나 제거할 수 있습니다.
4. 사이트 및 테마가 준비되면 Oracle Content Management 인스턴스와 변경사항을 동기화하고 선택한 사이트에서 새 템플리트를 생성합니다.
5. 다른 사람이 사용할 수 있도록 템플리트를 공유합니다.

#### 스타터 템플리트를 사용하여 템플리트 생성

스타터 템플리트를 사용하려면 다음과 같이 하십시오.

1. Oracle Content Management 측면 탐색에서 **사이트**를 누릅니다.  
기존 사이트 목록이 표시됩니다.
2. **생성**을 누릅니다.
3. [사이트 생성] 대화상자에서 사이트의 기초로 사용할 스타터 템플리트를 선택합니다.  
새 사이트는 스타터 템플리트와 함께 제공된 테마를 사용하지만 새 사이트 이름에 맞게 이름이 바뀝니다.
4. 대화상자에서 사이트의 이름을 입력합니다. 이 이름은 사이트 URL에 사용됩니다. 문자, 숫자, 밑줄(\_) 및 하이픈(-)을 사용할 수 있습니다. 공백을 입력하면 자동으로 밑줄로 바뀝니다.  
템플리트, 테마, 구성요소, 사이트 또는 사이트 페이지에 다음 이름을 사용하지 마십시오: authsite, content, pages, scsTEMPLATE\_\*, \_comps, \_components, \_compsdelivery, \_idcservice, \_sitescloud, \_sitesclouddelivery, \_themes, \_themesdelivery. 사이트 페이지에는 다음 이름을 사용할 수 있지만 템플리트, 테마, 구성요소 또는 사이트에는 사용하지 마십시오: documents, sites.

 주:

Oracle Content Management 사이트 URL의 경로는 대소문자를 구분합니다. 질의 또는 단편 문자열의 대소문자는 개발자가 사용자정의 코드로 관리합니다.

```
https:[//host[:port]][/]path[?query][#fragment]
      \_____/      \_____/
      Location      Data
```

5. 선택적으로 사이트에 대한 설명을 입력합니다.
6. 준비되면 **생성**을 누릅니다.

진행률 표시줄에 새 사이트 이름 및 생성 상태가 표시됩니다. 사이트가 생성되면 사이트 목록에 이름이 나타납니다. 초기 상태는 오프라인입니다.

새로 생성된 사이트를 목록에서 빨리 찾으려면 **최종 업데이트** 순으로 목록을 정렬합니다. 방금 생성한 사이트가 목록의 맨 위에 나타납니다.

 주:

직접 생성한 사이트에 대한 관리자 롤이 자동으로 지정됩니다.

7. 데스크톱 앱을 사용하여 테마를 데스크톱과 동기화합니다.  
이제 테마의 폴더 계층과 파일이 표시되어야 합니다.
8. 스타터 테마에는 다음과 같은 최소한의 폴더 및 파일 집합이 있습니다.

```
theme
  assets
    css
      main.css
    js
      topnav.js
  designs
    default
      design.css
      design.json
      facebook.png
      googleplus.png
      linkedin.png
      twitter.png
      youtube.png
      x-close.png
  layouts
    index.html
  resonsepages
    404.html
  _folder.json
  _folder_icon.jpg
```


```
components.json
viewports.json
```

x-close.png 파일에는 쿠키 동의 팝업에 대한 기본 닫기 아이콘이 있습니다.

/layouts 폴더에는 다음 콘텐츠와 함께 스타터 페이지 레이아웃 파일(index.html)이 있습니다.

- 파일을 페이지 레이아웃으로 사용할 수 있는 HTML 태그 집합.
- 테마를 데스크톱과 동기화하는 방법, 새 페이지 레이아웃을 추가하는 방법, 페이지 레이아웃에 구성요소를 추가하는 방법, 새 페이지 레이아웃을 사용하여 사이트 계층을 구축하는 방법과 같은 지침 텍스트가 시드된 단일 슬롯.
- 렌더러 API JavaScript 함수 및 객체를 사용하는 방법의 예를 제공하는 간단한 JavaScript 탐색 파일, 사이트 계층을 순회하고 필요한 HTML 마크업을 생성하여 사이트 내 탐색을 허용하려면 렌더링된 API가 필요합니다.

기본 테마 구조도 참조하십시오.

9. 테마 수정을 마쳤으면 테마 폴더 및 파일을 Oracle Content Management 인스턴스와 동기화합니다. 테마 변경사항이 사이트에서 어떻게 보이고 작동하는지 확인하려면 사이트 작성기에서 사이트를 엽니다. 이것은 반복적 과정일 것입니다.
10. 사이트를 보거나 수정하려면 사이트를 선택하고 마우스 오른쪽 단추 누르기 메뉴에서 열기를 누르거나 작업 모음에서  을 누릅니다.
11. 사이트를 변경할 수 있도록 편집기 모드를 편집으로 토글합니다. 스타터 테마에서 제공되는 페이지 레이아웃을 사용하여 기존 페이지를 수정하고 새 페이지를 추가할 수 있습니다.

#### 주:

페이지 레이아웃 구조 및 사용법에 익숙하다면 스타터 템플릿에서 제공한 원치 않는 섹션을 삭제하고, 원하는 섹션의 레이아웃을 새 레이아웃 중 하나로 전환할 수 있습니다.

12. 사이트 변경을 마쳤으면 사이트를 저장하고 게시를 눌러 기본 사이트에 업데이트를 병합합니다.
13. 사이트를 선택하고 사이트에 기반한 템플릿을 생성합니다. 새 템플릿의 자산과 테마를 끌어옵니다.
14. 템플릿을 사용하려는 멤버와 템플릿을 공유합니다.

## Bootstrap 또는 웹 사이트 디자인 템플릿에서 사이트 템플릿 생성

개방형 구조의 Oracle Content Management에서는 Foundation 또는 Bootstrap과 같은 다른 코딩 프레임워크에서 만든 작업을 사용할 수 있습니다. 약간의 변경으로 Bootstrap 템플릿을 테마로 전환하여 Oracle Content Management 템플릿의 일부로 만들 수 있습니다.

### 기본 프로세스

아래는 단계의 개요이며 다음 절에서 자세히 설명합니다.

#### 1. 필요 조건

2. 사이트 생성
3. 테마 폴더 동기화
4. 기본 테마 설정
5. 사이트 페이지 업데이트
6. 탐색 업데이트
7. 사이트 레이아웃 업데이트
8. 사이트 게시
9. 새 템플릿 생성

#### 필요 조건

- 로컬 컴퓨터와 폴더 및 파일을 동기화하기 위한 Oracle Content Management 데스크톱 앱을 설정하고 실행 중이어야 합니다.
- Bootstrap 템플릿 테마 폴더, 파일 및 콘텐츠를 로컬 컴퓨터로 다운로드하고 사용할 준비가 되어야 합니다.

#### 사이트 생성

Oracle Content Management 스타터 템플릿에서 사이트를 생성합니다.

1. 측면 탐색에서 **사이트**를 누릅니다.  
기존 사이트 목록이 표시됩니다.
2. **생성**을 누릅니다.
3. [사이트 생성] 대화상자에서 사이트의 기초로 사용할 **JET 스타터 템플릿** 또는 **스타터 템플릿**를 선택합니다.
4. 대화상자에서 사이트의 이름을 입력합니다. 이 이름은 사이트 URL에 사용됩니다. 문자, 숫자, 밑줄(\_) 및 하이픈(-)을 사용할 수 있습니다. 공백을 입력하면 자동으로 하이픈으로 바꿉니다.

템플릿, 테마, 구성요소, 사이트 또는 사이트 페이지에 다음 이름을 사용하지 마십시오: authsite, content, pages, scstemplate\_\*, \_comps, \_components, \_compsdelivery, \_idcservice, \_sitescloud, \_sitesclouddelivery, \_themes, \_themesdelivery. 사이트 페이지에는 다음 이름을 사용할 수 있지만 템플릿, 테마, 구성요소 또는 사이트에는 사용하지 마십시오: documents, sites.

#### 주:

Oracle Content Management 사이트 URL의 경로는 대소문자를 구분합니다. 질의 또는 단편 문자열의 대소문자는 개발자가 사용자정의 코드로 관리합니다.

```
https://[host[:port]][/path[?query][#fragment]]
      Location                               Data
```

5. 선택적으로 사이트에 대한 설명을 입력합니다.



## 6. 준비되면 생성을 누릅니다.

진행률 표시줄에 새 사이트 이름 및 생성 상태가 표시됩니다. 사이트가 생성되면 사이트 목록에 이름이 나타납니다. 초기 상태는 오프라인입니다.

새로 생성된 사이트를 목록에서 빨리 찾으려면 **최종 업데이트** 순으로 목록을 정렬합니다. 방금 생성한 사이트가 목록의 맨 위에 나타납니다.

## 테마 폴더 동기화

스타터 템플릿에서 사이트를 생성할 때 스타터 템플릿 테마의 사본이 생성되고 사이트 이름에 테마 이름이 붙은 형태로 이름이 지정됩니다. 예를 들어, `My_New_Site`의 테마는 `My_New_SiteTheme`입니다.

데스크톱 앱을 사용하여 사이트의 테마 폴더 및 파일을 로컬 컴퓨터와 동기화합니다. *Oracle Content Management*로 *문서 공동 작업*의 동기화 시작하기를 참조하십시오.

이제 로컬 데스크톱에 템플릿 테마 폴더 계층과 파일이 표시되어야 합니다. 예제는 다음과 같습니다.

```

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

```

## 기본 테마 설정

### 1. 다음 Bootstrap 파일을 로컬 데스크톱의 동기화된 테마 폴더로 복사하여 기존 파일을 덮어씁니다.

- html 파일이 `theme_name/layouts` 폴더로 이동
- css 파일이 `theme_name/assets/css` 폴더로 이동
- js 파일이 `theme_name/assets/js` 폴더로 이동
- image 파일이 `theme_name/assets/images` 폴더로 이동하며 배경, 바닥글, 사람 등의 이미지와 함께 하위 폴더로 그룹화될 수 있습니다.

2. layout 폴더의 html 파일을 수정하여 상대 경로를 업데이트하고 필요한 요소를 추가합니다. 일반적인 Bootstrap 테마의 경우 /assets 폴더에 대한 많은 상대 경로가 있으므로 테마 폴더를 가리키도록 수정해야 합니다.

다음을 사용하도록 css, js, images 폴더의 경로를 수정합니다.

```
_scs_theme_root_/assets/css/
_scs_theme_root_/assets/js/
_scs_theme_root_/assets/images/
```

### 주:

이 단계가 완료되면 \_scs\_theme\_root 부분은 테마가 사용되는 환경에 자동으로 조정됩니다.

3. 각 Oracle Content Management 레이아웃에 대한 세 가지 요구사항이 있습니다.
  - a. html 파일의 <head> 태그에 다음 태그를 포함합니다.
    - <!--\$SCS\_RENDER\_INFO-->
    - <!--\$SCS\_SITE\_HEADER-->
    - <!--\$SCS\_PAGE\_HEADER-->
  - b. 레이아웃 파일의 끝에 <body> 태그 바로 안에 renderer 스크립트를 포함합니다. 이 경로는 모두 사이트 작성기에서 런타임 환경에 자동으로 조정됩니다.


```
<script data-main="/_sitescloud/renderer/renderer.js" src="/_sitescloud/renderer/require.js"></script>
```


- c. renderer.js 파일을 포함시킨 후 다음 태그를 포함합니다.
  - <!--\$SCS\_PAGE\_FOOTER-->
  - <!--\$SCS\_SITE\_FOOTER-->




수정된 테마 파일이 Oracle Content Management 사이트와 동기화되었는지 확인합니다.

### 사이트 페이지 업데이트

사이트 작성기에서 사이트를 열면 처음에 기본 사이트에 있는 페이지가 보입니다. 원치 않는 페이지를 삭제하고 유지할 페이지의 레이아웃을 새 레이아웃 중 하나로 전환할 수 있습니다.

1. 측면 탐색에서 **사이트**를 누릅니다.  
기존 사이트 목록이 표시됩니다.
2. 사이트를 선택하고 마우스 오른쪽 단추 누르기 메뉴에서 **열기**를 선택하거나 작업 모음에서  을 누릅니다.
3. 업데이트에 대한 이름 및 선택적 설명을 입력한 다음 **생성**을 누릅니다.  
업데이트 이름에는 문자, 숫자, 밑줄(\_), 하이픈(-)을 사용할 수 있습니다. 공백을 입력하면 자동으로 하이픈으로 바뀝니다.

사이트에 대한 업데이트가 이미 있을 경우 목록에서 업데이트를 선택하고  을 누릅니다.

4. 사이트 작성기가 미리보기 모드로 열립니다. 내용을 변경하거나 사이드바의 탐색 옵션을 사용하려면 편집 스위치 가 편집으로 설정되어 있는지 확인합니다.
5. 특정 페이지를 편집하려면 사이드바의 사이트 트리를 사용하거나 사이트의 고유 탐색을 사용하여 페이지를 선택합니다.
6. 원치 않는 페이지를 제거하려면 페이지를 선택하고 을 누릅니다.
7. 새 페이지를 추가하려면 **페이지 추가**를 누릅니다. 사이트 트리에서 끌어 놓기로 페이지를 재배포할 수 있습니다.
8. 페이지와 연관된 레이아웃을 변경하려면 사이트 트리에서 페이지를 선택하고 을 눌러 페이지 설정을 표시합니다.  
**페이지 레이아웃** 필드로 이동하고 메뉴에서 다른 레이아웃을 선택합니다. 페이지 레이아웃의 개수와 유형은 사이트와 연관된 테마에 따라 달라집니다.
9. **저장**을 눌러 현재 업데이트에 변경사항을 저장합니다. 현재 업데이트에서 작업을 계속하거나, 필요한 경우 새 업데이트를 생성할 수 있습니다.

### 탐색 업데이트

사이트 미리보기로 탐색을 볼 때 레이아웃에 있는 Bootstrap 테마의 하드 코딩된 탐색 때문에 해당 탐색이 현재 계층과 일치하지 않습니다.

탐색을 업데이트하여 Bootstrap 테마의 하드 코딩된 코드를 사이트 계층에서 동적으로 생성된 코드로 바꿉니다.

테마 파일의 로컬 동기화 사본을 편집합니다.

1. 레이아웃에서 하드 코딩된 탐색 코드를 제거합니다. 다음은 일반적인 Bootstrap 테마에서 제거해야 하는 하드 코딩된 탐색의 예입니다. 다음은 로고가 있는 일반적인 머릿글 섹션으로, 페이지가 매우 좁을 때 표시되는 '햄버거' 메뉴의 '탐색 토글' 부분(반응형 부분)과 다른 페이지에 대한 하드 코딩된 페이지 탐색으로 구성됩니다.

```
<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. 사이트 구조 정보를 순회하고 탐색 코드를 생성하는 JavaScript 코드를 작성하고, 다음과 같이 레이아웃에 JavaScript 코드를 포함합니다.

```

<script type="text/javascript" src="_scs_theme_root_/assets/js/
navbar.js"></script>

```

3. 특정 CSS에서 예상한 마크업과 일치하도록 topnav.js 파일의 정확한 출력을 수정합니다.

### 사이트 레이아웃 업데이트

이 시점에서 사이트는 작동하지만 레이아웃에 편집 가능한 영역(슬롯)이 없습니다.

1. 레이아웃에서 DIV 요소를 찾거나 추가하여 슬롯으로 지정합니다.

슬롯은 클래스 속성에 "scs-slot" 값을 가진 레이아웃의 DIV 요소입니다. 각 슬롯에는 고유 id 속성이 있어야 합니다. 예를 들어, 다음과 같습니다.

```

<div id="slot-content1" class="scs-slot scs-responsive"></div>

```

브라우저 뷰포트의 크기에 맞게 자동으로 슬롯을 조정하려면 클래스 속성 "scs-responsive"를 포함합니다.

- 필요에 따라 이 단계를 반복하여 해당 레이아웃이나 다른 레이아웃에 추가 슬롯을 생성합니다.

### 사이트 게시

모든 변경사항을 완료하고 사이트에 저장한 후 사이트를 게시해야 합니다.

사이트를 게시하면 현재 업데이트의 모든 변경사항을 가져와서 기본 사이트에 병합함으로써 업데이트된 내용으로 새로운 기본 사이트를 만들 수 있습니다.

### 새 템플릿 생성

이제 기능적 사이트와 관련 테마가 있으므로 템플릿으로 만들어서 다른 사람이 사이트를 생성할 수 있도록 공유할 수 있습니다.

- Oracle Content Management 측면 탐색에서 **개발자**를 누릅니다.
- 모든 템플릿 보기**를 누릅니다.
- 생성**을 누르고 **기존 사이트에서**를 선택합니다.
- 스타터 템플릿과 Bootstrap 템플릿을 사용하여 생성한 새 사이트를 선택합니다.
- 새 템플릿의 이름을 입력하고 **생성**을 누릅니다.
- 다른 Oracle Content Management 인스턴스에서 사용하도록 템플릿을 패키징하려면 템플릿을 선택하고 **익스포트** 메뉴 옵션을 선택하여 다운로드할 수 있는 .zip 파일을 생성합니다.

## Developer Cloud Service로 템플릿 개발

Developer Cloud Service를 사용하여 Oracle Content Management용 템플릿을 개발할 수 있습니다.

다음 단계에 따라 Developer Cloud Service에서 템플릿을 개발하여 로컬에서 테스트하고 Oracle Content Management로 익스포트하십시오.

- [로컬 시스템에서 OCE Toolkit 설정](#).
- [Oracle Content Management용 Developer Cloud Service 콘솔에 사인인](#).
- [Developer Cloud Service에서 프로젝트 생성](#).
- [새 Git 저장소에서 프로젝트 코드에 OCE Toolkit 추가](#).
- [Developer Cloud Service에서 템플릿 생성](#).  
개발할 새 템플릿을 생성하거나 [Developer Cloud Service에서 기존 템플릿 복사](#) 또는 [Oracle Content Management에서 템플릿 임포트](#)를 수행할 수 있습니다.
- [로컬 테스트 하네스에서 템플릿 테스트](#).
- [변경사항 병합](#).
- [Developer Cloud Service에서 Oracle Content Management로 템플릿 익스포트](#).

## Oracle Content Management용 Developer Cloud Service 콘솔에 사인인

Developer Cloud Service 콘솔에서 Oracle Content Management용 사용자정의 구성요소 개발을 시작합니다.

Oracle Cloud 서비스 관리자는 [내 서비스 관리]를 사용하여 클라우드 서비스를 생성하고 관리할 수 있습니다. Oracle Content Management의 서비스 인스턴스 관리자이자 Standard Developer Service의 서비스 관리자인 경우 서비스를 설정하고 사용을 시작할 수 있습니다.

1. 계정에 제공된 정보를 사용하여 Oracle Cloud에 사인인합니다.
2. [내 서비스 관리]에 사인인하여 Oracle Content Management 인스턴스 및 Standard Developer Service를 생성하고 관리합니다.



3. 요청에 따라 Oracle Developer Cloud Service 전자메일을 확인합니다.
4. 서비스 구독 세부정보를 사용하여 Oracle Content Management 인스턴스를 설정하고 인스턴스의 Oracle Content Management URL로 이동합니다.
5. Standard Developer Service의 URL로 이동합니다.
6. Oracle Developer Cloud Service 계정에 사인인합니다.  
Developer Cloud Service URL에 액세스하여 콘솔에 사인인합니다.

## Developer Cloud Service에서 프로젝트 생성

"Content Experience Cloud" 프로젝트 템플릿을 사용하여 Developer Cloud Service에서 프로젝트를 생성하거나, 빈 Git 저장소로 프로젝트를 생성하고 Oracle Content Management 인스턴스에서 OCE Toolkit을 импорт할 수 있습니다.

- [Oracle Content Management 템플릿으로 Developer Cloud Service 프로젝트 생성](#)
- [Oracle Content Management에서 다운로드한 OCE Toolkit으로 Developer Cloud Service에서 프로젝트 생성](#)
- [새 Git 저장소에서 프로젝트 코드에 OCE Toolkit 추가](#)

## Developer Cloud Service에서 템플릿 생성

cec 명령행 유틸리티를 사용하여 사용 가능한 소스 템플릿에서 Oracle Content Management 템플릿을 생성할 수 있습니다.

cec create-template 명령을 사용하여 사용 가능한 소스 템플릿 중 하나에서 템플릿을 생성합니다. 명령행에 cec create-template -h를 입력하면 사용 가능한 소스 템플릿이 나타납니다.

다음은 템플릿 생성의 예입니다.

```
cec create-template CafeSupremoLite_yourname -f CafeSupremoLite
```

**Windows:** 이 명령은 템플릿을 생성하는 동안 외부 HTML WYSIWYG 편집기(예: `_scs_theme_root_`)에 렌더링할 테마의 심볼 링크를 생성합니다. Windows에서 심볼 링크를 생성하려면 일반적으로 관리 권한으로 명령행 유틸리티를 실행해야 합니다. 테마 편집을 위해 WYSIWYG 편집기를 사용하지 않는 경우 관리 권한으로 실행할 필요가 없으며 심볼 링크 생성 오류를 무시할 수 있습니다.

앞의 예제는 `CafeSupremoLite_yournameTheme` 템플릿을 생성하고 소스 코드가 `cec-components/src/main/`에 제공되도록 합니다. 다음 표는 Developer Cloud Service에서 사이트 템플릿을 생성한 후 소스 코드의 위치를 보여줍니다.

템플릿 소스 코드	테마	템플릿의 구성요소
<code>cec-components/src/main/templates</code>	<code>cec-components/src/main/themes</code>	<code>cec-components/src/main/components</code>

로컬 테스트 하네스는 구성요소도 보여주며, 템플릿 및 유형별로 필터링할 수 있습니다.

텍스트 또는 코드 편집기에서 테마 및 구성요소 파일을 편집할 수 있습니다. [로컬 테스트 하네스로 테스트](#)를 참조하십시오. 테마나 구성요소를 편집한 후 브라우저를 새로고침하여 변경사항을 확인합니다.

### ! 중요사항:

템플릿, 테마 및 구성요소의 소스 코드는 `src/main/`에 있습니다. `src/main` 외부 파일은 Oracle Content Management 로컬 서버 작동에 필요하므로 수정하면 안됩니다.

## Developer Cloud Service에서 템플릿 복사

Developer Cloud Service에서 기존 Oracle Content Management 템플릿을 복사할 수 있습니다.

`src/main/templates`에서 기존 템플릿 중 하나를 복사하려면 `cec copy-template` 명령을 사용합니다. 템플릿에 다른 저장소의 자산이 포함된 경우 선택적으로 저장소 매핑을 제공합니다. 그렇지 않으면 해당 자산이 복사되지 않습니다.

다음 예제는 `Temp1` 템플릿을 `Temp2`라는 새 템플릿으로 복사합니다.

```
cec copy-template Temp1 -n Temp2
```

## Developer Cloud Service로 템플릿 импорт

추가 개발을 위해 Oracle Content Management에서 Developer Cloud Service로 템플릿을 импорт할 수 있습니다.

Oracle Content Management 서버에서 생성된 템플릿 zip 파일이 있는 경우 테마나 구성요소 편집과 같은 추가 개발을 위해 해당 파일을 Developer Cloud Service로 임포트할 수 있습니다. 다음 명령을 사용하십시오.

```
cec import-template <location of the template zip file>
```

Oracle Content Management에서 zip 파일이 포함된 폴더를 지정합니다.

[템플릿 정보](#) 및 [템플릿 익스포트](#)를 참조하십시오.

## 변경사항 병합

로컬 시스템에서 구성요소, 템플릿 또는 콘텐츠 레이아웃을 생성하거나 소스 코드를 편집한 후에 신규/변경된 구성요소와 템플릿을 프로젝트 Git 저장소로 병합해야 합니다.

Git 저장소로 변경사항을 병합하려면 터미널 창에서 다음 명령을 순서대로 입력합니다.

```
cd cec-components git pull
git add .
git status
git commit -a -m "Your comments" git pull
git push
```

## Developer Cloud Service에서 템플릿 익스포트

Developer Cloud Service에서 템플릿 zip 파일을 익스포트하고 이 파일을 사용하여 Oracle Content Management에 사이트를 생성할 수 있습니다.

템플릿 개발이 완료되면 다음 명령을 실행하여 템플릿을 익스포트할 수 있습니다. 명령 응답은 Oracle Content Management에서 템플릿의 zip 파일이 생성된 위치를 알려줍니다.

```
cec export-template CafeSupremoLite_yourname
```

[템플릿 정보](#) 및 [템플릿 임포트](#)를 참조하십시오.



# 21

## 테마 개발

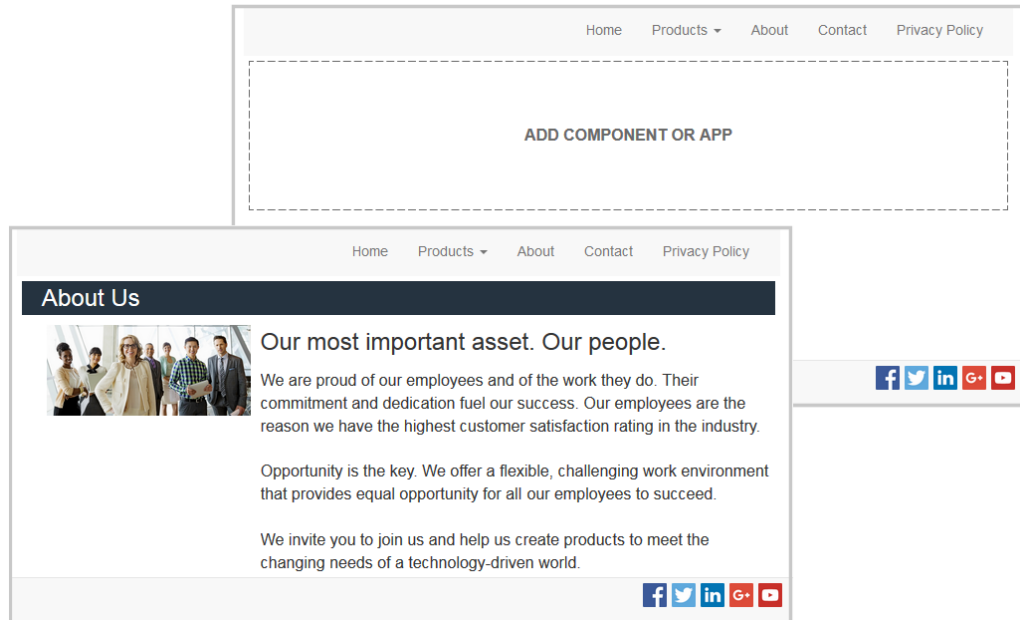
테마는 색상표, 글꼴 크기, 글꼴 유형, 페이지 배경 등 일반적인 사이트 모양(전체적 스타일)을 정의합니다. 테마는 사이트 내 페이지 간에 시각적 일관성을 제공합니다. 디자인과 샘플 콘텐츠를 지정하여 고유 테마와 테마 변형을 생성하고, 이를 사용하여 브랜드와 비전을 홍보하는 사이트를 생성할 수 있습니다.

- [테마 정보](#)
- [기본 테마 구조](#)
- [사이트 탐색](#)
- [테마 생성](#)
- [테마와 구성요소 연관](#)
- [사이트 렌더링 API](#)

## 테마 정보

테마는 콘텐츠, 모양, 동작 등 일반적인 사이트 모양을 정의합니다. 테마는 사이트 내 페이지 간에 시각적 일관성을 제공합니다.

새 테마를 디자인하려면 레이아웃, 스타일, 샘플 콘텐츠, 탐색과 함께 새 사이트의 시작점으로 작용할 모든 기본 정보를 지정해야 합니다. 테마 디자이너는 사이트의 모양과 동작에 대한 예상치를 설정합니다. 테마는 사용 방법을 염두에 두고 디자인해야 합니다. 예를 들어, 대부분 또는 모든 사용자는 모바일 장치로 사이트에 액세스할 것으로 예상됩니다. 많은 유사 사이트를 생성하려는 사용자가 있으면 사용자정의 테마를 디자인하는 것이 유용합니다. 테마 간에 공유할 수 있는 공통 패턴의 페이지 레이아웃을 사용하여 테마를 디자인할 수 있습니다.



테마에는 사이트 콘텐츠, 모양 및 동작을 디자인하는 데 사용되는 페이지 레이아웃이 포함됩니다. 디자인 및 설정을 변경하고 콘텐츠를 추가하여 나만의 스타일, 브랜드, 비전이 담긴 사이트를 생성할 수 있습니다.

테마는 다음을 포함합니다.

- 배경 이미지 또는 기타 페이지 레이아웃에 속하는 콘텐츠의 자산(이미지, JavaScript 파일 등)
- 사이트의 스타일 설정(CSS)
- 여러 페이지 레이아웃(HTML 파일)
- 사이트 탐색을 구성하는 코드(Javascript 파일)
- 구성요소와 함께 사용할 수 있는 기본 스타일 목록(design.css 및 design.json 파일에 지정됨)

테마에는 페이지 템플릿 중 하나에서 생성된 새 페이지를 채울 때 사용되는 *시드 데이터*가 포함될 수도 있습니다. 예를 들어, 사용자는 Products 섹션에 대한 새 페이지를 생성하고 new\_product.html이라는 페이지 레이아웃을 선택합니다. 테마에 new\_product-pageseed.json 파일이 포함되어 있으면 처음 생성할 때 새 페이지는 페이지 시드 파일의 콘텐츠로 채워집니다. 샘플 콘텐츠와 마찬가지로, 이 시드 데이터는 수정할 수 있으며 페이지 구축의 시작점으로만 제공됩니다.

테마와 연동하도록 고안된 구성요소의 부분 집합을 사용하는 테마를 생성할 수 있습니다. 사용자가 사이트에 해당 테마를 선택하면 해당 테마에 지정된 구성요소만 표시됩니다. [테마와 구성요소 연관](#)을 참조하십시오.

각 웹 사이트는 테마를 사용합니다. 템플릿에서 사이트를 생성할 때 템플릿에서 테마를 상속합니다. 언제든지 사이트에 대한 테마를 변경할 수 있습니다. Oracle Content Management는 시작할 때 사용할 수 있는 다양한 템플릿과 테마를 제공합니다.

사이트에 게시되지 않은 새로운 테마를 사용하는 경우, 처음 사이트를 온라인에 놓을 때 테마가 자동으로 사이트에 게시됩니다. 테마를 변경하고 온라인 사이트를 업데이트하여 변경사항을 표시하려면 테마를 명시적으로 게시해야 합니다. 테마 소유자나 관리자 권한을 가진 사용자만 테마를 명시적으로 게시할 수 있습니다.

 주:

테마에 변경사항을 게시하는 경우 테마를 사용하는 모든 온라인 사이트에 변경사항이 반영됩니다. 예를 들어, 테마에 지정된 기본 글꼴을 변경하고 테마를 게시하는 경우 테마를 사용하는 모든 사이트는 새로운 기본 글꼴을 사용합니다.

테마 관리를 참조하십시오.

## 기본 테마 구조

테마의 기본 구조에는 Oracle Content Management에 저장된 폴더에 지정된 디자인, 탐색 및 스타일이 포함됩니다. 테마는 사이트 템플릿의 일부입니다.

사용자가 새 사이트 생성을 위해 템플릿을 선택하면 관련 테마 데이터가 자동으로 로드됩니다. JET 스타터 템플릿 또는 스타터 템플릿을 사용하는 경우 테마가 참조되는 대신 자동으로 복사됩니다. 개발자가 스타터 템플릿을 사용하는 경우 고유의 테마 사본이 필요합니다.

기본 폴더 및 파일을 보여주는 다음 예제와 같이 테마는 특정 폴더 및 파일 구조로 구성됩니다.

```

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

```

어떤 폴더에는 다음 폴더를 비롯한 특정 유형의 정보가 포함됩니다.

- **assets:** JavaScript, CSS(종속형 스타일시트), 이미지 및 기타 레이아웃에서 참조되는 지원 파일입니다.
- **designs:** 구성요소의 스타일 옵션을 지정하는 데 사용되는 `design.css` 및 `design.json` 파일입니다.
- **layouts:** 사이트의 페이지를 표시하는 데 사용되는 페이지 템플릿용 HTML 파일입니다.

 주:

모든 HTML 파일에는 `<!DOCTYPE html>`과 비슷한 DOCTYPE 요소가 파일 시작 부분에 있어야 합니다.

- `publish`: 게시된 사이트에서 테마가 사용된 후에 이 디렉토리가 나열되고 파일 사본이 포함됩니다. 이 디렉토리는 데스크톱 앱을 사용하여 테마를 동기화한 경우 볼 수 있지만, 테마를 템플릿의 일부로 익스포트한 경우 포함되지 않습니다.
- `responsepages`: 오류 처리를 위한 특수 페이지입니다(404).

### 주:

사이트의 페이지가 오류 페이지로 플래그된 경우 테마의 404 오류 메시지가 무시되고 지정된 오류 페이지가 대신 사용됩니다.

- `viewport.json`: 테마의 뷰포트 설정을 지정합니다.
- `_folder.json`: 테마의 이름과 GUID를 지정합니다. 예를 들어, 다음과 같습니다.

```
{
  "themeName": "MarketingCampaignTheme",
  "itemGUID": "TB79D65F699B022AC4E11F4D4EE870070A1ADD86BBBB"
}
```

GUID는 테마를 처음 임포트하거나 복사할 때 Oracle Content Management에서 생성됩니다. 테마 이름은 테마를 생성할 때 테마 개발자가 지정합니다.

- `components.json`: (사용되지 않음) 테마 내에 사용된 사용자정의 구성요소를 기록합니다. 역 호환성을 위해서만 유지보수됩니다.

새 테마를 생성할 때 작업할 두 개의 키 파일이 있습니다. 이 파일은 구성요소의 스타일을 설정합니다.

- `design.css`
- `design.json`

탐색 정보를 하나의 JavaScript 파일(예: `nav.js` 파일)에 넣는 것이 좋습니다. 테마의 `/assets/js/` 폴더는 해당 파일을 보관하기 좋은 위치입니다.

## 사이트 탐색

사이트 계층은 사이트와 연관된 `structure.json` 파일에 저장됩니다. 계층은 메모리에 로드되고 페이지 컨텍스트에서 `SCS.structureMap` 객체로 사용할 수 있습니다.

사이트 작성기는 `structure.json` 파일을 읽고 사이트 작성기에서 사이트 트리를 그립니다. `structure.json` 파일에는 사이트 페이지의 코드가 포함됩니다. 예를 들어, 다음과 같습니다.

```
"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
  }

```

사이트 페이지 내에 탐색 JavaScript 코드가 있어야 해당 구조를 읽고 사이트의 탐색 링크를 그릴 수 있습니다. Oracle Content Management와 함께 제공된 템플리트에는 이 작동 방식을 시연하는 샘플 탐색 JavaScript 파일이 있습니다.

Oracle Content Management와 함께 제공된 일부 테마에서 사용되는 `topnav.js` 파일은 `SCS.structureMap` 객체를 Render API 호출(예: `SCSRenderAPI.getPageLinkData`)과 함께 사용하여 사이트 구조를 순회하고 페이지의 탐색 메뉴 렌더링에 필요한 HTML 마크업을 그리는 방법의 예를 보여줍니다. 다음은 샘플 `topnav.js` 파일의 코드입니다.

```

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();
            }
        }
    )
}
```

```
    );  
}
```

Render API 호출을 사용하여 사이트 편집 및 미리보기 모드와 게시된 온라인 사이트에서 작동할 탐색 링크를 생성할 수 있습니다. [Render API 참조](#)를 참조하십시오.

탐색 정보를 하나의 JavaScript 파일(예: `topnav.js`)에 넣는 것이 좋습니다. Oracle Content Management와 함께 제공된 샘플 테마에서 볼 수 있듯이, JavaScript 파일은 일반적으로 테마의 `/assets/js/` 폴더에 저장됩니다.


## 테마 생성

기존 테마를 복사하고 사본을 변경하여 새 테마를 생성할 수 있습니다. Bootstrap 콘텐츠를 임포트하여 새 테마에서 사용할 수도 있습니다.

### 주:

새 테마를 생성하든 기존 테마를 업데이트하든, 항상 테마의 사본을 만들어서 사본에 작업하십시오. 샘플 사이트 또는 실제 사이트 사본으로 테스트하여 올바르게 작동하는지 확인합니다. 기존 테마에 대한 변경사항은 수정된 테마가 게시된 후 즉시 테마를 사용하는 모든 사이트에서 구현됩니다.

### 테마 복사

1. 홈 페이지에서 **개발자**를 누릅니다.  
개발자 페이지가 표시됩니다.
2. **모든 테마 보기**를 누릅니다.  
기존 테마 목록이 표시됩니다. 보기 아이콘을 누르고 목록에서 옵션을 선택하여 테마가 표시되는 방법을 제어할 수 있습니다.
3. 테마를 선택하고 마우스 오른쪽 단추 누르기 메뉴에서 **복사**를 선택하거나 작업 모음에서  을 누릅니다.  
샘플 페이지 및 콘텐츠를 포함한 테마의 모든 폴더 및 파일이 복사됩니다.

### 주:

테마를 복사할 때 테마 공유 정보는 복사되지 않습니다.

4. 복사된 테마의 이름을 입력합니다. 다른 테마에 사용되는 이름은 사용할 수 없습니다.  
문자, 숫자, 밑줄(\_), 하이픈(-)을 이름에 사용할 수 있습니다. 공백을 입력하면 자동으로 하이픈으로 바뀝니다.  
템플릿, 테마, 구성요소, 사이트 또는 사이트 페이지에 다음 이름을 사용하지 마십시오: `authsite`, `content`, `pages`, `scstemplate_*`, `_comps`, `_components`, `_compsdelivery`, `_idcservice`, `_sitescloud`, `_sitesclouddelivery`, `_themes`, `_themesdelivery`. 사이트 페이지에는 다음 이름을 사용할 수 있지만 템플릿, 테마, 구성요소 또는 사이트에는 사용하지 마십시오: `documents`, `sites`.
5. 선택적으로 테마에 대한 설명을 입력합니다.

**6. 복사를 누릅니다.**

진행률 표시줄에 새 테마 이름 및 복사 상태가 표시됩니다. 테마가 복사되면 테마 목록에 이름이 나타납니다. 테마 목록에서 테마 이름을 눌러 테마를 구성하는 폴더 및 파일을 탐색할 수 있습니다.

- 7. Oracle Content Management 데스크톱 앱을 사용하여 테마 폴더 및 파일을 로컬 시스템과 동기화합니다.** 그러면 로컬 폴더를 찾아보고 파일로 직접 작업할 수 있습니다. 테마에 대한 변경사항은 자동으로 동기화됩니다. 즐겨찾는 HTML, 코드 또는 텍스트 편집 도구를 사용하여 내용을 변경할 수 있습니다.

**사용자정의 스타일을 사이트에 사용할 수 있는지 여부 설정**

테마 관리자는 테마와 어울리는 스타일을 지정하거나, 스타일을 사용자정의할 수 있습니다.

테마의 `components.json` 파일 설정은 사이트 작성기에서 사용자정의 스타일을 지정할 수 있는지 여부를 지정합니다. 사이트 제공자가 사이트를 구축할 때 스타일을 유지하고 다른 글꼴과 색상을 사용하지 않도록 제어하기 위한 것입니다.

사이트에서 사용자정의 스타일 사용을 허용하지 않으려면 다음과 같이 하십시오.

- 1. 구성요소의 설정 패널에서 사용자정의 옵션을 숨깁니다.**  
테마의 `components.json` 파일에서 다음 객체를 다른 구성요소 정의와 함께 추가합니다.

```
[
  {
    "showStyleClassOnly": "true"
  }
]
```

- 2. CKEditor의 도구모음 그룹 및 단추를 사용자정의하여 사용자정의 스타일을 제거합니다.**  
테마의 `components.json` 파일에서 CKEditor의 `toolbarGroups` 및 `removeButtons` 속성을 지정하여 Title 및 Paragraph 구성요소에 대한 그룹과 그룹 내의 단추를 사용자정의할 수 있습니다.

예를 들어, 다음 객체를 다른 구성요소 정의와 함께 추가하여 `styles` 그룹에서 스타일, 글꼴 및 색상을 제거합니다.

```
[
  {
    "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,BGCol
}
},
{
    "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,BGCol
    }
    }
    ]
}
]

```

- 스타일을 CKEditor로 붙여넣지 못하게 합니다. 붙여넣기하면 구성된 스타일 및 단추가 따라옵니다. 앞의 예제에서는 붙여넣은 데이터에서 Font가 제거됩니다.

사용자정의 스타일을 허용하지 않으면 다음 효과가 있습니다.

- 모든 내장 구성요소에서 스타일 탭에 (o) 사용자정의 옵션이 숨겨집니다.
- CKEditor에서 스타일 설정을 위한 도구모음 단추가 숨겨지고 글꼴 모음, 글꼴 색상 등의 지정이 무효화됩니다.
- 또한 CKEditor에서 붙여넣은 텍스트에 스타일 설정이 불가능해야 합니다. 굵은체 또는 기울임꼴은 허용됩니다.

### 테마에 대한 구성요소 및 섹션 레이아웃 숨기기

테마의 component.json 파일을 편집하여 사이트 개발자가 구성요소 및 섹션 레이아웃을 사용하지 못하도록 숨길 수 있습니다. 일관된 모양을 승격하기 위해 이 작업을 수행하고자 할 수 있습니다. [테마에 대한 구성요소 및 섹션 레이아웃 숨기기](#)를 참조하십시오.

### 사이트에서 테마 확인

테마 편집을 완료한 후 모든 부분이 계획대로 사이트와 함께 작동하는지 확인해야 합니다.

- 복사한 폴더 및 파일이 Oracle Content Management 데스크톱 앱과 동기화되었고 모든 수정사항이 저장되었는지 확인합니다.
- 사이트 작성기에서 사이트(테스트 또는 기존 오프라인 사이트 가능)를 열고 새 테마를 사용하도록 사이트를 전환합니다.
- 페이지를 추가하고 설정을 사용하고 이미지를 보고 탐색을 확인하고 그 밖에 새 테마에서 변경한 내용을 확인하여 사이트에서 테마를 테스트합니다.
- 모든 파일 및 링크를 포함한 모든 것이 올바르게 렌더링되는지 확인합니다.

5. 사이트가 테마 변경사항을 구현하도록 테마를 게시하고 런타임 환경에서 확인합니다.

테마 게시를 참조하십시오.

### Bootstrap 테마 사용

Oracle Content Management 테마와 Bootstrap 테마 간의 유사점 덕분에, 기존 Bootstrap 테마 페이지 및 콘텐츠를 변환하여 Oracle Content Management 테마에서 사용할 수 있습니다.

Bootstrap은 웹 사이트와 웹 애플리케이션 생성을 위한 무료 오픈 소스 도구 모음입니다. 인터페이스 구성요소 및 JavaScript 확장을 위한 HTML 및 CSS 기반의 디자인 템플리트를 포함합니다.

테마를 생성할 때와 동일한 지침을 따르지만, 선택한 편집기를 사용하여 코드 작업으로 Bootstrap 테마에서 원하는 페이지나 콘텐츠를 임포트하여 편집할 수 있습니다.

모든 페이지와 콘텐츠에 모든 Bootstrap 테마를 사용하지는 않습니다. 일반적으로 Oracle Content Management 테마에 사용할 테마의 일부를 선택하고 몇몇 페이지 템플릿과 그들 중 일부만 선택하게 됩니다. 예를 들어, Bootstrap 테마에는 머리글, 탐색 블록, 본문 블록, 바닥글 블록을 비롯한 여러 블록이 템플릿 내에 포함됩니다. Bootstrap 테마는 웹 서버 없이 파일 시스템에서 직접 볼 수 있는 파일 모음으로 제공되기 때문에 각 페이지 내에 많은 중복이 포함됩니다(관련된 페이지 어셈블리 엔진이 없기 때문). Oracle Content Management 테마에서 이 정보를 사용할 때 해당 부분을 가져와서 재사용 가능한 Oracle Content Management 페이지 템플릿에 추가한 다음, Oracle Content Management를 사용하여 해당 템플릿의 여러 페이지를 동적으로 어셈블해야 합니다.


재사용 가능한 템플릿에 정보를 추가하는 한 가지 예는 탐색 섹션입니다. 일반적인 Bootstrap 테마의 탐색은 모든 페이지에 중복되지만, Oracle Content Management 테마로 이동할 때는 JavaScript 코드를 사용하여 사이트 계층을 동적으로 순회하고 탐색 구조를 생성해야 합니다. 그런 다음 해당 스크립트를 모든 페이지에 포함시키면 모두 탐색을 얻고, 사이트에서 페이지를 추가하거나 제거할 때 해당 탐색이 조정됩니다.

Bootstrap 또는 웹 사이트 디자인 템플릿에서 사이트 템플릿 생성을 참조하십시오.

## 테마에 대한 구성요소 및 섹션 레이아웃 숨기기

기본적으로 모든 구성요소와 섹션 레이아웃이 사이트 개발자에게 제공됩니다. 특정 테마를 사용할 때 사용할 수 없는 구성요소 또는 섹션 레이아웃을 숨기려는 경우가 있을 수 있습니다. 예를 들어, 단추 구성요소 또는 슬라이더 섹션 레이아웃을 사용하지 않는 일관된 모양을 승격하려는 경우 테마의 components.json 파일을 편집하여 테마에서 숨길 수 있습니다.

테마에서 구성요소 및 섹션 레이아웃을 숨기려면 다음과 같이 하십시오.

1. Oracle Content Management의 왼쪽 탐색 메뉴에서 **개발자**를 누릅니다.
2. 개발자 페이지에서 **모든 테마 보기**를 누릅니다.
3. [테마] 페이지에서 수정할 테마를 선택하고 마우스 오른쪽 단추 누르기 메뉴에서 **열기**를 누르거나 작업 모음에서  을 누릅니다.
4. components.json 파일을 선택하고 **다운로드**를 누릅니다.
5. 텍스트 편집기에서 components.json 파일을 엽니다.
6. 객체를 추가하여 구성요소 유형 및 ID를 지정하고 속성 숨김을 true로 설정합니다.

예를 들어, 다음은 단추 구성요소를 숨기는 입력 예입니다.

```
"type": "scs-button",
"id": "scs-button",
"hidden": true
```

마찬가지로 다음은 슬라이더 섹션 레이아웃을 숨기는 입력 예입니다.

```
"type": "scs-sectionlayout",
"id": "scs-sl-slider",
"hidden": true
```

단추 구성요소와 슬라이더 섹션 레이아웃을 숨길 components.json 파일로의 전체 입력은 다음과 같습니다.

```
[
  {
    "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
      }
    ]
  }
]
```


- 원하는 변경사항을 적용한 후에는 components.json 파일을 새 버전으로 테마에 업로드합니다.

## 테마에 대해 구성요소 맞춤, 너비 또는 간격 옵션 숨기기

기본적으로 대부분의 구성요소에서는 구성요소의 설정 대화상자에서 맞춤, 너비 및 간격 옵션을 지정하여 사이트를 구축할 수 있습니다. 하지만 개발자는 제공자가 사이트 테마와 일치하지 않는 방식으로 콘텐츠를 렌더링하지 못하도록 방지하기 위해 사용자정의 구성요소에서 이러한 옵션을 숨겨야 할 수 있습니다. 예를 들어, 단추를 가운데 두고 간격을 30픽셀로 유지하는 일관된 모양을 프로모션하려는 경우에는 테마의 components.json 파일을 편집하여 설정에서 이를 변경하는 옵션을 숨길 수 있습니다.

또한 사용자정의 구성요소의 초기 기본값을 무효화해야 할 수도 있으며, 이것 역시 components.json 파일을 편집하여 수행될 수 있습니다. 초기 값에 대한 모든 변경사항은 편집된 components.json 파일이 새 버전으로 업로드된 다음에 페이지에 추가된 구성요소에만 적용됩니다.

테마에서 한 구성요소에 대한 맞춤, 너비 또는 간격의 설정 옵션을 숨기려면 다음과 같이 하십시오.

1. Oracle Content Management의 왼쪽 탐색 메뉴에서 **개발자**를 누릅니다.
2. 개발자 페이지에서 **모든 테마 보기**를 누릅니다.
3. [테마] 페이지에서 수정할 테마를 선택하고 마우스 오른쪽 단추 누르기 메뉴에서 **열기**를 누르거나 작업 모음에서  을 누릅니다.
4. components.json 파일을 선택하고 **다운로드**를 누릅니다.
5. 텍스트 편집기에서 components.json 파일을 엽니다.
6. 구성요소 유형 및 ID를 지정할 객체를 추가하고 숨기려는 속성을 true로 설정합니다. 예를 들어, 다음은 설정 대화상자에서 단추 구성요소의 맞춤, 너비 및 간격 속성을 숨기는 항목입니다.

```
"type": "scs-button",
"id": "scs-button",
"hideAlignmentAndWidth": true,
"hideSpacing": true,
```

단추 구성요소의 기본 맞춤 및 간격 속성을 무효화하려고 할 때 초기 값을 지정하는 방법을 보여주는 예는 다음과 같습니다.

```
"initialData": {
  "alignment": "center",
  "marginTop": 30,
  "marginBottom": 30
```

테마에서 사용되는 단추 구성요소의 설정 대화상자에서 기본값을 지정하고 맞춤, 너비 및 간격 옵션을 숨기기 위해 components.json 파일에 입력할 전체 항목은 다음과 같습니다.

```
[
  {
    "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
        }
    ]
}
]

```

- 원하는 변경사항을 적용한 후에는 components.json 파일을 새 버전으로 테마에 업로드합니다.

## 테마와 구성요소 연관

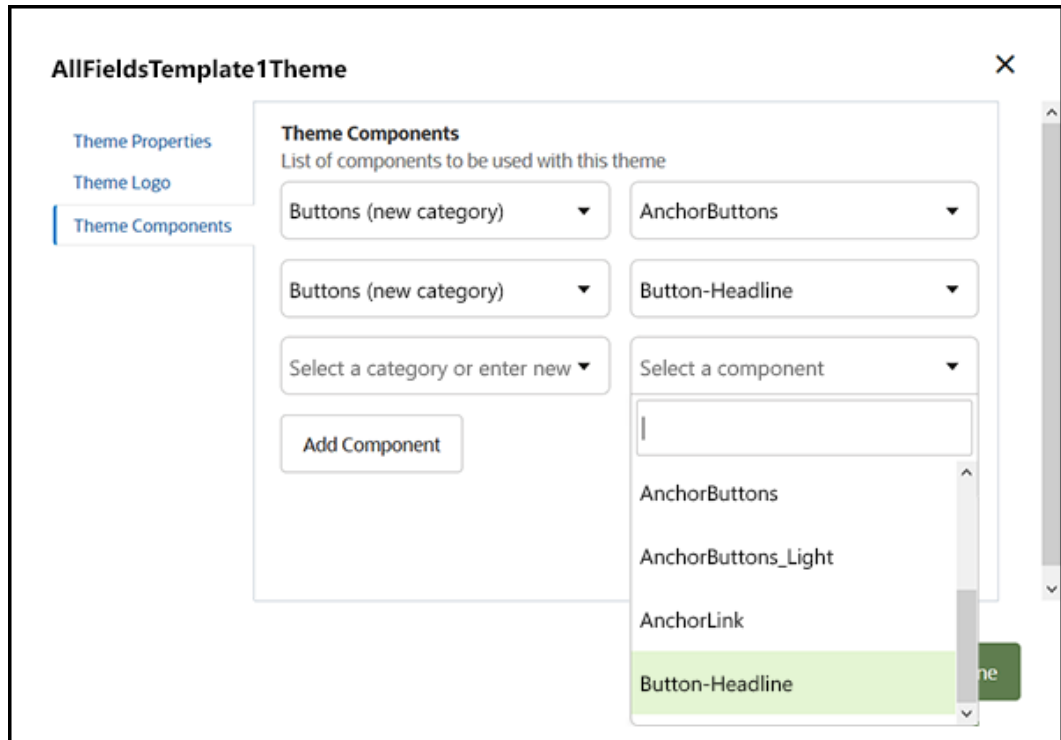
구성요소, 섹션 레이아웃 및 구성요소 그룹을 테마와 연관시켜서 특정 구성요소 부분 집합을 테마와 함께 사용할 수 있습니다.

개발자가 템플릿의 테마 및 구성요소를 생성하면 마케터는 이를 사용하여 조직의 제품을 홍보하는 사이트를 생성할 수 있습니다. 테마와 구성요소를 연관시키면 사용자가 사이트 작성기에서 **테마 구성요소**를 선택할 때 사용할 수 있습니다. **테마 구성요소** 탭에서 사용자는 테마와 연관된 구성요소만 볼 수 있습니다. 테마와 구성요소를 연관시키면 이 구성요소는 사이트에 사용되지 않더라도 사이트 템플릿과 함께 익스포트됩니다.

테마와 구성요소를 연관시키려면 다음과 같이 하십시오.

- Oracle Content Management의 왼쪽 탐색 메뉴에서 **개발자**를 누릅니다.
- 개발자** 페이지에서 **모든 테마 보기**를 누릅니다.
- [테마] 페이지에서 테마를 선택하고 마우스 오른쪽 단추 누르기 메뉴에서 **속성**을 누르거나 작업 모음에서 누릅니다.
- 테마 구성요소**를 눌러 **테마 구성요소** 탭을 엽니다.
- 테마와 연관시킬 하나 이상의 구성요소를 선택하고 구성요소를 범주화합니다.
  - 구성요소 선택** 드롭다운 목록에서 사용자정의 구성요소를 선택합니다.
  - 첫번째 선택한 구성요소에 대해 범주 이름을 입력하여 구성요소의 범주를 생성합니다.
  - 추가로 선택한 구성요소에 대해 기존 범주를 선택하거나 새 범주를 생성합니다.

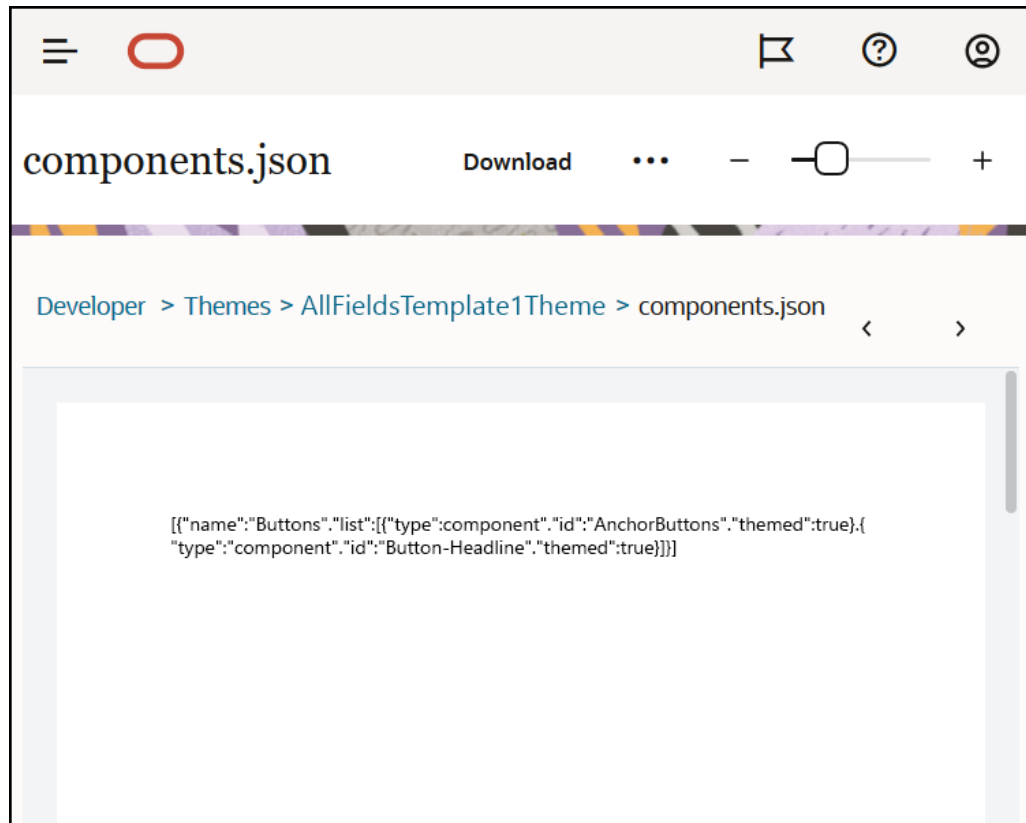
- d. 구성요소 추가를 누릅니다.



6. **저장**을 눌러 선택한 구성요소를 테마와 연관시킵니다.
7. **테마** 페이지에서 테마를 선택하고 **[속성]**을 눌러 **테마 구성요소** 탭을 다시 엽니다.
8. 선택한 구성요소가 저장되었는지 확인합니다.

연관된 구성요소를 테마에서 제거하려면 **테마 구성요소** 탭에서 구성요소 이름 옆에 있는 **X**를 누르고 **구성요소 제거** 대화상자에서 **예**를 누릅니다.

템플리트의 **세부정보** 페이지에서 템플리트가 사용 중인 테마와 연관된 구성요소 목록을 볼 수 있습니다. 테마와 연관된 구성요소에 대한 참조는 테마의 `components.json` 파일에 저장됩니다.



테마 연관을 구성한 후 Oracle Content Management에 템플리트를 배치하여 마케팅 팀과 공유할 수 있습니다. 마케터가 사이트에 해당 테마를 선택하면 사이트 작성기의 **테마 구성요소** 탭에서 테마에 지정한 구성요소만 볼 수 있습니다. **사용자정의** 및 **모두** 구성요소 목록에도 연관된 테마 구성요소가 포함됩니다.

#### OCE Toolkit에서 테마와 구성요소 연관

OCE Toolkit에서 테마와 구성요소를 연관하려는 경우 다음 **cec** 명령을 사용할 수 있습니다. 구성요소가 사이트 작성기의 **테마 구성요소** 탭과 **사용자정의** 및 **모두** 구성요소 목록에 나타납니다.

```
cec add-component-to-theme <component>
```

다음 **cec** 명령은 구성요소 연관을 테마에서 제거합니다.

```
cec remove-component-from-theme <component>
```

OCE Toolkit에 대한 자세한 내용은 OCE Toolkit으로 개발을 참조하십시오.

## 사이트 렌더링 API

Oracle Content Management(SCSRenderAPI)용 **사이트 렌더링 API**는 모든 Oracle Content Management 웹 페이지에서 제공되는 window-global 객체입니다. 주로 페이지의 슬롯 및 구성요소를 렌더링하며 테마 레이아웃에 있는 JavaScript 코드에 대한 인터페이스를 제공합니다. 개발자로 테마 또는 구성요소를 사용 중인 경우 해당 이벤트 및



함수를 찾을 수 있습니다. 런타임 SCSRenderAPI는 보기 및 미리보기 표시 모드를 렌더링하며, 디자인 타임 SCSRenderAPI는 탐색, 편집 및 주석 표시 모드를 렌더링합니다.

런타임에서 사이트 렌더링 API의 일반적인 용도 세 가지는 다음과 같습니다.

- 페이지의 슬롯에 구성요소 및 콘텐츠를 채웁니다.
- 페이지 레이아웃에서 발견된 JavaScript 코드로 생성된 정보 요청을 충족합니다.
- 페이지 렌더링 프로세스의 수명 주기 동안 이벤트를 발생시킵니다.

디자인 타임에서 사이트 렌더링 API의 일반적인 용도 네 가지는 다음과 같습니다.

- 페이지의 슬롯에 구성요소 및 콘텐츠를 채웁니다.
- 페이지 레이아웃에서 발견된 JavaScript 코드로 생성된 정보 요청을 충족합니다.
- 페이지 렌더링 및 편집 프로세스의 수명 주기 동안 이벤트를 발생시킵니다.
- 사이트 작성기와의 상호작용을 통해 페이지 편집 및 주석을 허용합니다.

사이트 렌더링 API를 Oracle Content Management 웹 페이지에서 로드하려면 테마 레이아웃에서 다음 스크립트 태그를 배치합니다.

```
<script data-main="/_sitescloud/renderer/renderer.js" src="/_sitescloud/renderer/require.js"></script>
```

일반적으로 레이아웃 아래쪽에 배치됩니다. 이 태그의 URL은 적절한 환경(디자인 타임 또는 런타임)에 맞게 자동으로 조정됩니다. SCSRenderAPI 객체는 비동기적으로 로드됩니다. 사용자정의 JavaScript 코드는 scsrenderstart 이벤트를 처리하여 사이트 렌더링 API의 가용성을 수신할 수 있습니다.

# 22

## 레이아웃 개발

레이아웃은 페이지에 콘텐츠를 배열하는 방법을 정의하며 Oracle Content Management 사이트에서 사용된 페이지의 HTML을 생성하는 데 사용됩니다.

- 레이아웃 정보
- 검색 엔진 최적화(SEO)
- components.json 파일 및 형식 이해
- 사이트 작성기에서 도구모음 그룹 사용자정의
- 슬롯의 구성요소 제한
- 레이아웃 콘텐츠를 편집 가능으로 만들기
- 섹션 레이아웃 생성
- 지연 로드를 지원하는 섹션 레이아웃 생성
- API를 사용한 사용자정의 섹션 레이아웃 개발
- 콘텐츠 레이아웃 개발

## 레이아웃 정보

레이아웃은 사이트 페이지에 콘텐츠를 배열하는 방법을 정의합니다. 각 레이아웃은 다른 개수의 명명된 슬롯을 포함할 수 있습니다. 슬롯은 페이지 너비에 걸친 영역으로, 하나 이상의 콘텐츠 유형을 포함할 수 있습니다.

모든 테마는 여러 페이지 레이아웃을 갖습니다. [테마 정보](#)를 참조하십시오.

사이트에 페이지를 추가할 때 해당 페이지에 사용할 레이아웃을 선택합니다. 각 레이아웃에는 콘텐츠를 끌어 놓을 수 있는 슬롯이라는 페이지 영역이 있습니다. 이 슬롯에 어떤 콘텐츠를 담을지는 완전히 당신에게 달려 있습니다. 제목, 텍스트, 구분선부터 멀티미디어, 갤러리, 소셜 미디어에 이르기까지 어떤 것도 가능합니다.

레이아웃에는 적합한 HTML 생성자뿐만 아니라 Oracle Content Management 렌더러가 이해하는 특수 마크업이 포함됩니다. 표준 준수 모드로 페이지를 렌더링하도록 브라우저를 구성하려면 레이아웃이 DOCTYPE 문으로 시작해야 합니다. 예: `<!DOCTYPE html>` 특정 구성요소가 최상의 결과를 얻으려면 이 명령문이 필요합니다.

이 샘플 코드는 최소 레이아웃을 보여줍니다.

```
1 <!DOCTYPE html>
3 <head>
4     <meta http-equiv="X-UA-Compatible" content="IE=edge">
5
6     <script src="/_themes/[--$SCS_THEME_NAME--]/assets/js/topnav.js"></script>
7     <link rel="stylesheet" type="text/css"
8         href="/_themes/[--$SCS_THEME_NAME--]/assets/css/main.css">
9     <link rel="stylesheet" type="text/css"
```

```

10             href="/_themes/[!--$SCS_THEME_NAME--]/designs/
[!--$SCS_DESIGN_NAME--]/design.css">
11
12         <!--$SCS_RENDER_INFO-->
13         <!--$SCS_SITE_HEADER-->
13         <!--$SCS_PAGE_HEADER-->
14     </head>
15     <body>
16         <div id="topNavigation"></div>
17         <div id="mainContentSlot" class="scs-slot scs-responsive"></
div>
18
19         <script data-main="/_sitescloud/renderer/renderer.js"
20             src="/_sitescloud/renderer/require.js"></script>
21         <!--$SCS_SITE_FOOTER-->
22     </body>
23 </html>

```

브라우저에서 페이지를 렌더링할 때 다양한 토큰이 확장됩니다.

- [!--\$SCS\_THEME\_NAME--]

현재 사이트에 선택된 테마의 이름으로 확장됩니다. 이 토큰을 사용하는 URL은 현재 테마를 참조하므로 이 토큰을 사용하면 테마를 복사할 수 있습니다.
- [!--\$SCS\_DESIGN\_NAME--]

현재 사이트에 선택된 디자인의 이름으로 확장됩니다. 그러면 테마 내의 여러 디자인에서 레이아웃이 사용될 수 있습니다.
- [!--\$SCS\_RENDER\_INFO--]

해당 페이지의 페이지 계층 및 구성요소 렌더링 정보를 보유한 스크립트 태그로 확장됩니다. 레이아웃의 <head> 섹션에 놓여야 합니다.
- [!--\$SCS\_SITE\_HEADER--]

검색 엔진 최적화(SEO) 속성의 머리글 필드에 지정된 사이트 머리글 값으로 확장됩니다. 모든 페이지에 놓일 사이트 수준 마크업을 여기에 입력할 수 있습니다. [검색 엔진 속성 설정](#)을 참조하십시오.
- [!--\$SCS\_PAGE\_HEADER--]

사이트 작성기의 [페이지 설정] 속성의 [페이지 머리글] 필드에 있는 페이지 머리글 값으로 확장됩니다. 이 페이지에 놓일 페이지 특정 마크업을 여기에 입력할 수 있습니다.
- [!--\$SCS\_SITE\_FOOTER--]

SEO(검색 엔진 최적화) 속성의 바닥글 필드에 있는 사이트 바닥글 값으로 확장됩니다. 모든 페이지에 놓일 사이트 수준 마크업을 여기에 입력할 수 있습니다. [검색 엔진 속성 설정](#)을 참조하십시오.

#### 주:

토큰은 [!--\$ 및 --] 대신 구분자로 <!--\$ 접두어와 --> 접미어를 사용할 수도 있습니다.

사이트가 온라인일 때 링크의 다음 토큰은 사용 중인 컨텍스트를 인지하는 실제 값으로 바뀝니다. 그러면 사이트를 편집할 때와 게시된 사이트가 온라인일 때 링크가 작동할 수 있습니다.

- `/_sitescloud/`가 `/_sitesclouddelivery/`로 바뀜
- `/_themes/`가 `/_themesdelivery/`로 바뀜

슬롯은 클래스 속성에 "scs-slot" 값을 가진 레이아웃의 DIV 요소입니다. 슬롯은 사용자가 사이트 콘텐츠를 채울 구성요소를 추가할 수 있는 곳입니다. "scs-slot" 클래스 속성을 지정하여 여러 DIV 요소를 슬롯으로 지정할 수 있습니다. 각 슬롯에는 고유 id 속성이 있어야 합니다.

#### 주:

슬롯은 중첩될 수 없지만 CSS를 사용하여 슬롯을 겹치도록 제안할 수 있습니다. 이를 위해 구성요소 그룹 또는 섹션 레이아웃을 사용하십시오. [섹션 레이아웃 생성](#)을 참조하십시오.

추가로 "scs-responsive" 클래스 속성값을 가진 슬롯은 브라우저 뷰포트의 해상도가 변경될 때 콘텐츠를 새로고침합니다. 그러면 대형 데스크톱 디스플레이 또는 소형 모바일 장치에 반응하여 슬롯을 렌더링할 수 있습니다.

마지막 `<script>` 태그(예제 19행)는 Oracle Content Management 렌더링 코드를 로드합니다. 이 코드는 페이지에 구성요소를 그리는 역할을 하며 사용자정의 코드로 Render API에 액세스할 수도 있습니다. 이 `<script>` 태그가 없으면 레이아웃에 기반한 페이지를 사이트 작성기에서 변경할 수 없습니다.

## 검색 엔진 최적화(SEO)

검색 엔진이 사이트 내용을 식별할 수 있도록 키워드를 제공할 수 있습니다.

### SEO 설정

검색 엔진 최적화(SEO) 설정은 사이트 레벨과 페이지 레벨에서 정의됩니다. SEO 텍스트는 모든 미리 정의된 템플릿에 바닥글로 편입됩니다.

[검색 엔진 속성 설정](#)을 참조하십시오.

### 사이트 방문자용 쿠키

사이트 방문자는 청구 목적으로 각 브라우저와 각 사이트에 하나씩 SEO용 쿠키를 사용합니다. 각 쿠키는 매시간 또는 24시간 후에 갱신해야 합니다.

각 사이트에는 방문자에게 쿠키를 알리는 팝업이 나타나야 합니다.



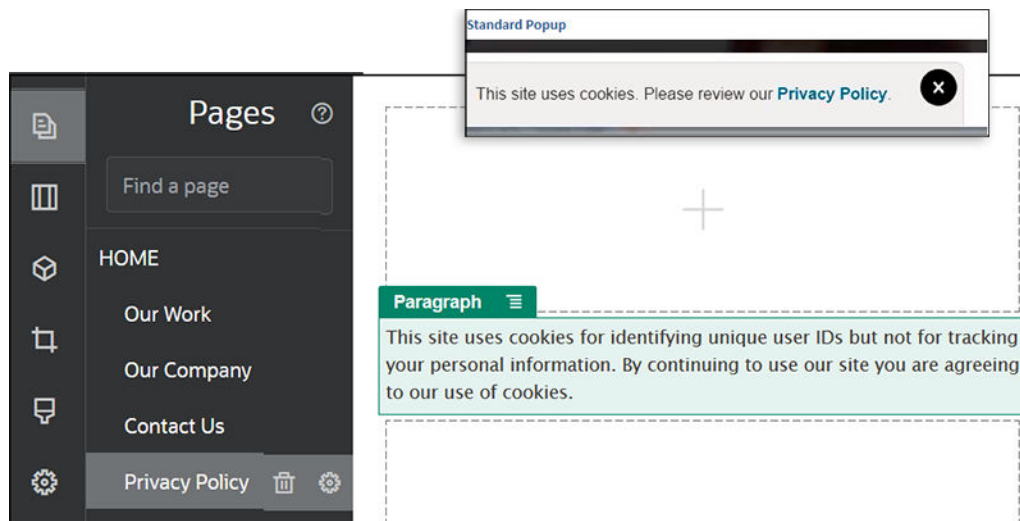
팝업에는 다음 정보가 포함되어야 합니다.

- SEO 머리글 및 바닥글 텍스트
- 텍스트가 선택되어 링크될 특정 ID를 가진 바닥글의 DIV

```
<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>
```

### 개인정보 보호정책 페이지

권고 텍스트가 있는 개인정보 보호정책 페이지로 텍스트가 링크됩니다.



### 검색 엔진 최적화용 Inner HTML

사이트 작성기에서 구성요소 데이터를 저장할 때 구성요소가 런타임에 렌더링된 경우 생성될 Inner HTML을 저장할 수 있습니다. 이 Inner HTML은 페이지 데이터에 저장되므로 페이지를 렌더링할 때 구성요소 대신 페이지에 데이터를 삽입할 수 있습니다. 이는 페이지 표현 초기에 발생하므로 검색 엔진이 성공적으로 페이지 콘텐츠를 검색할 가능성이 높아집니다.

## components.json 파일 및 형식 이해

테마에는 사용된 구성요소를 지정하는 `/ThemeName/components.json`에 위치한 `components.json` 파일이 있어야 합니다.

이 파일은 적합한 JSON을 포함해야 하며 최소한 파일에는 빈 JSON 배열 `[]`이 포함되어야 합니다.

`components.json` 파일 구문은 모든 로컬 구성요소를 나열하며 구성요소 범주화를 완전히 지원합니다. (원격 구성요소는 구성요소 카탈로그에 등록됩니다.)

어떤 구성요소가 테마 레벨 `components.json` 파일에 추가되든지(없음 포함), Oracle Content Management는 사용자에게 제공되는 기본 구성요소 집합을 채웁니다. 이 기본 집합은 소스 코드에 정의됩니다. 다음 목록은 인라인 프레임으로 렌더링된 구성요소 및 시드된 구성요소를 보여줍니다. 또한 서비스 레벨에 등록되어 인스턴스의 사용자에게 제공되는 원격 구성요소는 사이트 작성기에서 사용할 수 있습니다.

다음 로컬 구성요소는 Oracle Content Management에 포함되어 있습니다.

이름	유형	ID
제목	scs-title	scs-title
단락	scs-paragraph	scs-paragraph
이미지	scs-image	scs-image
갤러리	scs-gallery	scs-gallery
갤러리 그리드	scs-gallerygrid	scs-gallerygrid
문서	scs-document	scs-document
단추	scs-button	scs-button
맵	scs-map	scs-map
구분선	scs-divider	scs-divider
스페이스	scs-spacer	scs-spacer
YouTube	scs-youtube	scs-youtube
소셜 표시줄	scs-socialbar	scs-socialbar
비디오	scs-video	scs-video
기사(사용자정의 구성요소)	scs-component	scs-comp-article
헤드라인(사용자정의 구성요소)	scs-component	scs-comp-headline
이미지 및 텍스트(사용자정의 구성요소)	scs-component	scs-comp-image-text

이같이 인라인 프레임으로 렌더링된 구성요소는 Oracle Content Management에 포함되어 있습니다. 등록된 원격 구성요소는 포함되지 않습니다.

이름	유형	ID
대화	scs-app	대화
문서 관리자	scs-app	문서 관리자
폴더 목록	scs-app	폴더 목록
파일 목록	scs-app	파일 목록
Facebook 좋아요	scs-app	Facebook 좋아요
Facebook 추천	scs-app	Facebook 추천
Twitter 팔로우	scs-app	Twitter 팔로우
Twitter 공유	scs-app	Twitter 공유

### 일반 형식

components.json 파일의 일반 형식은 다음과 같습니다.

- 구성요소의 속성은 각 구성요소 내에 지정됩니다. 최상위 레벨 "components" 또는 "apps" 속성은 사용되지 않습니다.
- 각 구성요소에는 "type" 속성이 있습니다. 구성요소는 특정 값만 가질 수 있습니다(모든 가능한 값은 기본 구성요소의 테이블에 나열됨).
- 각 구성요소에는 "id" 속성이 있으며 고유해야 합니다. 이 속성은 동일한 "type"의 구성요소를 구별하는 데 사용됩니다. 이전에는 앱에 "appName" 속성이 있었습니다. "appName"은 "id" 속성을 사용할 수 없을 때 여전히 작동하지만 "appName" 속성은 사용되지 않습니다.
- 각 구성요소에는 사용자 인터페이스의 표시 이름인 "name" 속성이 있습니다. 폴백 값이 지정되지 않은 경우, 구성요소의 경우 값은 해당하는 기본 구성요소의 이름이고 원격 구성요소의 경우 값은 ID입니다.

다음은 components.json 파일의 예입니다.

```
[
  {
    "name": "COMP_CONFIG_TEXT_CATEGORY_NAME",
    "list": [
      {
        "type": "scs-title",
        "id": "my-headline",
        "name": "My Headline",
        ...
      },
      {
        ...
      },
      ...
    ],
  },
  {
    "name": "My own category name",
    "list": [ ... ]
  }
]
```

일반 구조는 범주 객체의 JSON 배열입니다. 각 범주 객체에는 "name" 속성과 "list" 속성이 있습니다. "name" 속성은 지역화된 문자열에 매핑되는 키일 수 있습니다. 이 기본 범주가 충분하지 않으면 고유의 범주 이름(지역화되지 않음)을 제공할 수 있습니다. 다음 표는 사용 가능한 기본 범주와 해당 키를 나열합니다.

키	범주 이름(영어)
COMP_CONFIG_CONTENT_CATEGORY_NAME	콘텐츠
COMP_CONFIG_CUSTOM_CATEGORY_NAME	사용자정의
COMP_CONFIG_MEDIA_CATEGORY_NAME	매체
COMP_CONFIG_SOCIAL_CATEGORY_NAME	소셜
COMP_CONFIG_TEXT_CATEGORY_NAME	텍스트

각 범주 객체의 "list" 속성은 구성요소 객체의 배열을 포함합니다. 각 구성요소 또는 객체에는 "type" 및 "id" 속성이 있어야 합니다. 기타 속성은 선택사항입니다.

- "type" 속성은 기본 구성요소에 있는 유형 중 하나와 같아야 합니다. "type"이 아직 없으면 구성요소가 표시되지 않습니다.
- "id" 속성은 구성요소 간에 고유해야 합니다. "id"가 이미 존재하면 구성요소가 표시되지 않습니다.
- "name" 속성은 사용자 인터페이스에서 구성요소의 표시 이름입니다. 앱(지금 원격 구성요소)에 대한 이전 "appName" 속성을 대체합니다.
- 다른 모든 속성은 이전 릴리스와 동일하게 취급됩니다.

### components.json에 새 구성요소 추가

기본 구성요소 수정은 허용되지 않습니다. 그러나 기존 기본 구성요소를 기반으로 새 구성요소를 생성할 수 있습니다. 예를 들어, 기본 텍스트를 설정하는 "scs-title" 구성요소를 기반으로 새 구성요소를 생성할 수 있습니다. 새 구성요소 추가에 필요한 최소 작업은 "type" 및 "id" 속성을 지정하는 것입니다.

- "type"은 기본 구성요소에 있는 유형 중 하나와 같아야 합니다. "type"이 아직 없으면 구성요소가 표시되지 않습니다.
- "id"는 구성요소 간에 고유해야 합니다. "id"가 이미 존재하면 구성요소가 표시되지 않습니다.

다음은 새 제목 구성요소를 추가하는 코드의 예입니다. 이 구성요소는 기본 제목 구성요소와 함께 표시됩니다.

```
[
  {
    "name": "COMP_CONFIG_TEXT_CATEGORY_NAME",
    "list": [
      {
        "type": "scs-title",
        "id": "my-headline"
      }
    ]
  }
]
```



다음은 표시 이름 및 기본 텍스트와 함께 새 제목 구성요소를 추가하는 코드의 예입니다.

```
[
  {
    "name": "COMP_CONFIG_TEXT_CATEGORY_NAME",
    "list": [
      {
        "type": "scs-title",
        "id": "my-headline",
        "name": "My Headline",
        "initialData": {
          "userText": "This is a second title component"
        }
      }
    ]
  }
]
```

제목 구성요소는 기본 제목 구성요소의 모든 속성을 기반으로 가져와서 그 위에 테마 레벨 수정사항을 적용하는 방식으로 새 구성요소를 생성합니다.

### 역 호환성

이전 형식의 components.json 파일을 아직 읽을 수 있습니다.

- 최상위 레벨 "components" 또는 "apps" 속성이 있는 파일.
- 파일에 "apps" 속성이 있으면 이 속성 아래의 사용자 정의 원격 구성요소가 계속 로드됩니다.
- 파일에 최상위 레벨 "apps" 속성이 있으면 그 아래에 나열된 원격 구성요소는 "scs-app" 유형으로 간주됩니다.
- "appName" 속성이 있으면 "id"를 "appName" 값으로 설정합니다. 표시 이름은 "name" (지정된 경우)과 동일하거나 "id" 값으로 풀백됩니다.

## 사이트 작성기에서 도구모음 그룹 사용자정의

사용자정의 구성요소의 경우 사이트 작성기 도구모음에 표시된 미리 정의된 도구모음 그룹의 복사본을 사용자정의할 수 있습니다.

제한된 집합을 가지고 사용자정의 태그에 사용할 수 있는 미리 정의된 구성요소의 고유한 버전을 생성할 수 있습니다. 사용자정의 요소는 미리 정의된 구성요소에 영향을 미치지 않습니다.

제한된 집합을 가지고 사용자정의 태그에 사용할 수 있는 미리 정의된 구성요소의 고유한 버전을 생성할 수 있지만, 미리 정의된 구성요소에는 영향을 미치지 않습니다.

도구모음 그룹은 제목 또는 단락 구성요소를 눌러 편집할 때 도구모음에 표시되는 항목을 정의합니다. 제목 또는 단락 구성요소에서 지원되는 항목을 제거하고 재정렬할 수 있지만 추가 플러그인은 허용되지 않습니다.

사이트 작성기에서 도구모음 그룹을 사용자정의하려면 CKEditor의 toolbarGroups 구성에 사용한 것과 동일한 구문을 사용해야 합니다. <https://ckeditor.com/latest/samples/toolbarconfigurator/index.html#basic>을 참조하십시오.

제목 및 단락 구성요소는 다음 그룹을 지원합니다.

- "basicstyles" - 굵은체/기울임꼴/밑줄로 제한됨 <https://ckeditor.com/cke4/addon/basicstyles>
- "styles" - 글꼴 스타일 [https://docs.ckeditor.com/ckeditor4/latest/guide/dev\\_styles.html](https://docs.ckeditor.com/ckeditor4/latest/guide/dev_styles.html)
- "colors" - 텍스트 및 배경색 [https://docs.ckeditor.com/ckeditor4/latest/guide/dev\\_colorbutton.html](https://docs.ckeditor.com/ckeditor4/latest/guide/dev_colorbutton.html)
- "undo" - CKEditor에서 현재 항목의 실행 취소/재실행
- "links" - 사용자정의 플러그인 링크 대화상자
- "paragraph" - 글머리표/번호 목록 및 들여쓰기 지원
  - "list"
  - "indent"
- "align" - 왼쪽/오른쪽/가운데
- "cleanup" - 선택된 텍스트의 스타일 제거

단락 구성요소는 이미지 및 테이블 삽입 플러그인도 지원합니다.”

- "insert"
  - "image"
  - "table"

또한 행 구분 기호 항목을 사용할 수 있습니다.

- "/"

 주:

도구모음 그룹 구성에서 다른 값을 설정하면 Site Builder 도구모음이 생성되기 전 해당 값이 제거됩니다. "extraPlugins"는 제공할 수 없습니다. "name"/"groups" 구성만 지원됩니다. 모든 "items" 항목은 무시됩니다.

예를 들어, 사용자가 글꼴, 색상, 스타일 또는 크기를 정의하지 못하게 하려면 다음과 같이 도구모음 구성을 업데이트할 수 있습니다. "id" 값의 경우 미리 정의된 값과 다른 사용자정의 값을 지정해야 합니다.

```
[{
  "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;"
    }
  }
}]

```

## 검증

components.json 파일을 앞의 코드로 바꾸고 사이트를 편집하십시오(이미 편집 중인 경우 브라우저 새로고침). 이 시점에 제목 구성요소를 편집할 때 더 이상 선택할 글꼴 스타일 또는 색상이 표시되지 않습니다. 단락 구성요소에는 계속 표시되며 사용 가능한 글꼴 크기 목록은 16, 24, 48로 제한됩니다.

## 기본 도구모음 그룹

제목 및 단락에 대한 기본 도구모음 그룹은 다음과 같습니다.

- 제목

```

[ {
  "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"]
  }
]

```

- 단락

```

[ {
  "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"]
}
]

```

## 슬롯의 구성요소 제한

모든 레이아웃 슬롯에 대해 슬롯에서 허용되는 구성요소에 대한 특정 제한을 지정할 수 있습니다.

슬롯의 구성요소를 제한하면 허용되지 않는 구성요소를 끌어오는 사용자에게 경고 메시지가 표시되고 해당 슬롯에 구성요소를 추가/이동할 수 없습니다.

이 제한을 구성하려면 테마의 레이아웃(예: 레이아웃 파일 `themes\theme_name\layouts\oneslot.htm`)을 편집하고 슬롯 DIV에 사용자정의 데이터 속성을 추가하십시오.

다음은 사용자정의 속성의 형식입니다. 이전 버전과 주요 차이점은 사용자가 구성요소의 id만 지정해야 한다는 것입니다. 이전에 구성요소를 제한하려면 사용자가 상세 "`<type><id>`" 구문(아직 지원됨)을 사용해야 했습니다.

```
data-allowed-items='["<id>:", "<type>:", "<type>:<id>", ...]'
data-disallowed-items='["<id>:"<type>", "<type>:<id>", ...]
```

 주:

`data-allowed-items` 및 `data-disallowed-items` 값은 큰 따옴표(") 또는 작은 따옴표(')를 사용할 수 있습니다. 다음 예제에서 `data-allowed-items`는 JSON 배열 주위에 작은 따옴표를 사용하고 `data-disallowed-items`는 JSON 배열 주위에 큰 따옴표를 사용합니다.

```
<div id="slot101"
  class="scs-slot"
  data-allowed-items='["scs-app", "scs-title"]'
  data-disallowed-items="['File List', 'scs-map']">
</div>
```

다음 표는 Oracle Content Management와 함께 제공된 구성요소와 각각의 ID를 나열합니다. 이름 지정 충돌을 방지하려면 사용자정의된 (로컬 또는 원격) 구성요소 ID 앞에 `scs-`를 붙이거나 이 표에 나열된 유형이나 ID를 사용하지 *마십시오*.

이름	유형	ID
문서 관리자	scs-app	문서 관리자
Facebook 좋아요	scs-app	Facebook 좋아요
Facebook 추천	scs-app	Facebook 추천
파일 목록	scs-app	파일 목록
폴더 목록	scs-app	폴더 목록
Twitter 팔로우	scs-app	Twitter 팔로우
Twitter 공유	scs-app	Twitter 공유
단추	scs-button	scs-button
기사(사용자정의 구성요소)	scs-component	scs-comp-article
헤드라인(사용자정의 구성요소)	scs-component	scs-comp-headline
이미지 및 텍스트(사용자정의 구성요소)	scs-component	scs-comp-image-text
구성요소 그룹	scs-componentgroup	scs-componentgroup
콘텐츠 검색	scs-contentsearch	scs-contentsearch

이름	유형	ID
콘텐츠 목록	scs-contentlist	scs-contentlist
콘텐츠 위치 표시자	scs-component	scs-contentplaceholder
콘텐츠 항목	scs-component	scs-contentitem
구분선	scs-divider	scs-divider
문서	scs-document	scs-document
갤러리	scs-gallery	scs-gallery
갤러리 그리드	scs-gallerygrid	scs-gallerygrid
이미지	scs-image	scs-image
맵	scs-map	scs-map
단락	scs-paragraph	scs-paragraph
소셜 표시줄	scs-socialbar	scs-socialbar
스페이스	scs-spacer	scs-spacer
제목	scs-title	scs-title
YouTube	scs-youtube	scs-youtube

사용자는 로컬 또는 원격 구성요소를 생성할 수 있습니다. 이 샘플에 제공된 이름 (My\_Local\_Component)은 이 구성요소가 슬롯 내에 제한되도록 지정할 때 사용할 수 있는 ID입니다.

### Create Local Component ✕

Provide a name for your component

My\_Local\_Component

Use only letters, numbers, hyphens, and underscores in component names.

Provide a description for your component (optional)

Add an optional description for this component

Component Type

Default ▼

Cancel
Create

## 레이아웃 콘텐츠를 편집 가능으로 만들기

레이아웃에 특정 텍스트 또는 이미지 콘텐츠를 구성하여 사용자가 레이아웃에 기반한 페이지 작업을 수행할 때 편집 가능하게 만들 수 있습니다.

이 기능은 Bootstrap 테마를 포함한 모든 테마에서 사용할 수 있습니다. 수정사항은 자동으로 원래 페이지의 스타일로 간주됩니다.

레이아웃의 HTML 태그 <p>, <h1> ~ <h6>, <div> 및 <img>에 간단한 마크업을 추가할 수 있습니다.

텍스트 및 이미지 형식 지정 옵션은 다음과 같습니다.

- 텍스트: 사용자가 굵은체, 기울임꼴, 밑줄 및 링크를 지정하도록 허용합니다.
- 이미지: 사용자가 이미지 파일에 링크를 지정하도록 허용합니다. 제목의 속성(사용자가 이미지 위로 커서를 가져갈 때 표시되는 내용)을 변경하고 접근성을 위해 대체 텍스트를 추가합니다.

텍스트 또는 이미지 요소를 사용자가 편집할 수 있도록 수정하려면 다음 두 가지 작업이 필요합니다.

1. `scs-editable`을 class에 추가합니다.
2. 고유 id 속성을 추가합니다.

레이아웃이 수정된 후에는 해당 레이아웃에 기반한 모든 페이지(새 페이지도)에서 기능을 사용할 수 있습니다.

페이지를 복사하여 붙여넣으면 수정사항이 새 페이지로 복사됩니다.

### 제목 태그 수정

다음은 사용자가 편집할 수 있도록 레이아웃의 제목 태그를 수정하는 방법의 예입니다.

1. 레이아웃 파일을 로컬 데스크톱과 동기화하거나 HTML 소스 파일에서 <h1> 헤더를 편집합니다.
2. `scs-editable`을 class에 추가하고 제목 코드 행에 `id="test-heading"` 속성을 추가하여 편집기에 첨부되도록 합니다. 예를 들어, 다음과 같습니다.

```
<h1 class="brand-heading scs-editable" id="test-heading">Sample
Heading Value</h1>
```

3. 파일을 저장합니다.
4. Oracle Content Management와 동기화하고 브라우저를 다시 로드합니다.

사용자가 제목 위로 커서를 가져갈 때 콘텐츠를 편집할 수 있음을 나타내는 검정 테두리가 제목 주위에 나타나야 합니다. 사용자가 제목을 누르면 테두리가 녹색이 되어 지금 사용자가 제목 콘텐츠를 편집할 수 있음을 나타냅니다.

5. 테마가 게시되지 않은 경우 게시합니다. 게시된 경우 브라우저를 새로고침할 때 변경사항이 나타납니다.

테마가 게시된 후에는 사이트 사용자가 헤더를 누르고 사이트 작성기에서 편집할 수 있습니다.

`scs-editable` 요소에 대한 의사 구성요소가 생성되어 사이트 작성기에서 변경하고 페이지 데이터로 저장할 수 있습니다. 런타임에 페이지를 렌더링하기 전에 컨트롤러가 `scs-editable` 태그를 사이트 작성기에서 설정한 값으로 바꿉니다.

### 이미지 태그 수정

사용자가 편집할 수 있도록 레이아웃의 이미지 태그를 수정하는 절차는 텍스트 사례와 비슷합니다.

1. scs-editable을 class에 추가합니다.
2. 고유 이미지 id를 추가합니다.

사용자는 이미지를 누르고 다른 이미지를 사용하도록 속성을 변경합니다.

다음은 사용자가 편집할 수 있는 이미지의 샘플 코드입니다.

```

```

## 섹션 레이아웃 생성

섹션 레이아웃을 생성하여 사이트 페이지의 슬롯 내에 콘텐츠를 배열합니다.

엔터프라이즈 사용자는 개발자가 제공한 섹션 레이아웃을 기반으로 사이트에 콘텐츠 항목을 배열할 수 있습니다. 기본 레이아웃에서 새 섹션 레이아웃을 생성할 수 있습니다.

섹션 레이아웃을 익스포트하고 오프라인에서 수정한 후 새 섹션 레이아웃으로 임포트하거나 기존 섹션 레이아웃을 바꿀 수 있습니다. 섹션 레이아웃을 개별적으로 익스포트하거나 템플릿 패키지(사용자정의 구성요소와 레이아웃 포함)의 일부로 익스포트합니다.


다음과 같이 미리 정의된 섹션 레이아웃을 사용할 수 있습니다.

- 가로
- 2열
- 3열
- 세로
- 탭
- 슬라이더

아무것도 생성하지 않고 사이트 작성기에서 바로 사용할 수 있습니다.

이 섹션 레이아웃의 파일에는 섹션 레이아웃 파일 구조에 대한 세부정보가 포함된 주석이 있습니다. 주석을 보려면 미리 정의된 섹션 레이아웃을 기반으로 새로 생성한 후, 다음 절차에 설명된 대로 편집을 위해 새 레이아웃을 익스포트할 수 있습니다.

섹션 레이아웃을 생성하려면 다음과 같이 하십시오.

1. 홈 페이지에서 **개발자**를 누릅니다.
2. **모든 구성요소 보기**를 누릅니다.
3. 오른쪽의 **생성** 드롭다운 메뉴에서 **섹션 레이아웃 생성**을 선택합니다.
4. **섹션 레이아웃 생성** 대화상자에서 섹션 레이아웃 구성요소에 대한 이름과 설명을 제공합니다.
5. 편집을 위해 섹션 레이아웃을 익스포트하려면 선택 후 마우스 오른쪽 단추 누르기 메뉴에서 **익스포트**를 누르거나 작업 모음에서  을 누릅니다.
  - a. 기존 폴더로 이동하거나, **생성**을 눌러 새 폴더를 생성하고 이름과 선택적 설명을 제공합니다.
  - b. 폴더 옆에 있는 체크박스를 선택하고 **확인**을 누릅니다.



- c. 폴더의 아이콘 또는 이름을 눌러 엽니다.

레이아웃 패키지 파일이 섹션 레이아웃 이름에 .zip 확장자가 붙은 형태로 선택한 폴더에 생성됩니다. 개발 환경으로 파일을 다운로드하여 파일을 편집합니다.

[API를 사용한 사용자정의 섹션 레이아웃 개발](#)에서 섹션 레이아웃 API에 대한 정보를 찾을 수 있습니다.

6. 수정된 파일을 새 섹션 레이아웃으로 импорт하거나 기존 섹션 레이아웃을 바꿉니다.
  - a. 홈 페이지에서 **콘텐츠** 탭을 누르고 **문서**를 누릅니다.
  - b. 익스포트할 때와 동일한 폴더 및 파일 이름을 포함한 **.zip** 확장자 파일로 수정된 섹션 레이아웃을 폴더에 업로드합니다.
  - c. **개발자** 페이지에서 **모든 구성요소 보기**를 누릅니다.
  - d. **생성** 메뉴에서 **구성요소 импорт**를 선택합니다.
  - e. 수정된 섹션 레이아웃을 포함한 업로드된 zip 파일 옆에 있는 체크박스를 선택하고 **확인**을 누릅니다.

수정된 섹션 레이아웃이 선택한 폴더로 импорт됩니다.

또한 섹션 레이아웃을 익스포트하여 다른 Oracle Content Management 인스턴스로 복사/이동하고 거기서 импорт할 수도 있습니다.

## 지연 로드를 지원하는 섹션 레이아웃 생성

콘텐츠 목록은 추가 구성요소가 질의될 때 섹션 레이아웃을 호출할 수 있습니다.

콘텐츠 목록을 렌더링할 때 반환되는 모든 콘텐츠 항목을 렌더링하기 위해 섹션 레이아웃을 선택할 수 있는 옵션이 있습니다. 이를 통해 테이블, 슬라이더 또는 8열 레이아웃과 같이 콘텐츠 항목마다 다른 레이아웃을 생성할 수 있습니다. 이 사용자정의 섹션 레이아웃은 더 고급 페이지 매김 기능에 참여할 수도 있습니다.

콘텐츠 목록은 다음과 같은 페이지 매김을 지원합니다.

- **페이지 매김**
- **스크롤 시 로드**
- **누를 때 로드**

표준 페이지 매김 기능의 경우 섹션 레이아웃이 아무것도 수행할 필요가 없습니다. 사용자가 다음 페이지를 누르면 다음 항목 집합으로 다시 렌더링됩니다. 그러나 **스크롤 시 로드** 및 **누를 때 로드**의 경우 섹션 레이아웃을 다시 렌더링하지 않고 추가 구성요소가 섹션 레이아웃에 추가됩니다. 주로 처음  $n$ 개 항목을 로드하고 사용자가 페이지 아래로 스크롤할 때 다음 항목 집합을 인출/렌더링하는 무한 스크롤 모델에 사용됩니다. **스크롤 시 로드** 및 **누를 때 로드**를 지원하려면 사용자정의 섹션 레이아웃에서 다음을 수행해야 합니다.

1. `render.js: addComponent()` API를 구현합니다. 이것은 섹션 레이아웃에 추가될 각각의 새 구성요소와 함께 호출됩니다.

```
// 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: 다음 항목을 포함하여 섹션 레이아웃이 addComponent() API를 지원함을 콘텐츠 목록에 알립니다.

```
"contentListData": {
  "addComponent": true
},
```

appInfo.json이 업데이트된 후에 사용자가 설정 패널에서 이 섹션 레이아웃을 선택하고 페이지 매김 화면으로 이동하면 누를 때 로드 및 스크롤 시 로드 옵션이 표시됩니다.

## API를 사용한 사용자정의 섹션 레이아웃 개발

렌더링 API 및 편집 API가 포함된 섹션 레이아웃 API를 사용하여 Oracle Content Management에서 사용자정의 섹션 레이아웃을 개발할 수 있습니다.

스타터 파일을 보려면 [섹션 레이아웃 생성](#)을 참조하십시오. 섹션 레이아웃의 스타터 파일에는 섹션 레이아웃 파일 구조에 대한 세부정보가 포함된 주석이 있습니다.

### 렌더링 API

render.js 모듈에서 로드된 렌더링 API는 사이트 작성기에서 런타임에 사용됩니다.

렌더링 API	설명	입력 매개변수	반환 결과
(생성자)	섹션 레이아웃 렌더링 모듈을 초기화합니다.	다음 속성을 포함하는 JavaScript 객체: <ul style="list-style-type: none"> <li><b>sectionlayoutData (객체):</b> 페이지 모델이 있는 섹션 레이아웃 데이터입니다.</li> <li><b>componentId (문자열):</b> 섹션 레이아웃의 componentId 값입니다 (일반적으로 GUID).</li> <li><b>renderMode (문자열, 선택사항):</b> 렌더링 작업의 렌더링 모드입니다.</li> <li><b>customSettingsData (객체):</b> sectionLayoutData에 있는 customSettingsData의 복사본입니다.</li> </ul>	섹션 레이아웃 렌더링 API가 초기화됩니다.
render	하위 구성요소에 대한 컨테이너 DIV를 포함하여 섹션 레이아웃에 적합한 DOM 요소를 페이지로 내보냅니다.	<b>container (요소):</b> 섹션 레이아웃의 마크업이 렌더링되는 DOM 요소입니다.	이 메소드가 반환된 후 하위 구성요소가 렌더링됩니다. 하위 div[id] 요소를 찾아서 하위 구성요소를 식별할 수 있습니다.

렌더링 API	설명	입력 매개변수	반환 결과
<b>addComponent</b>	콘텐츠 목록 구성요소와 함께 사용되어 하위 구성요소를 섹션 레이아웃에 동적으로 추가합니다. 이 함수는 선택사항입니다.	<b>container (요소):</b> 새 구성요소가 렌더링되는 DOM 요소입니다. <b>componentId (문자열):</b> 섹션 레이아웃에 추가할 새 구성요소의 ID입니다.	이 메소드가 반환된 후 <code>componentId</code> 입력과 ID가 일치하는 요소가 렌더링됩니다.

### 편집 API

섹션 레이아웃과 연관된 `appinfo.json` 파일에서 `hasEditHandlers` 속성이 `true`로 설정된 경우 `edit.js` 모듈이 로드됩니다.

편집 API는 사이트 작성기에서 사용됩니다.

생성자를 제외한 이 모듈의 모든 함수는 선택사항입니다.

편집 API	설명	입력 매개변수	반환 결과
<b>(생성자)</b>	섹션 레이아웃 편집 모듈을 초기화합니다.	다음 속성을 포함하는 JavaScript 객체: <ul style="list-style-type: none"> <li><b>componentId (문자열):</b> 섹션 레이아웃의 <code>componentId</code> 값입니다(일반적으로 GUID).</li> </ul>	섹션 레이아웃 편집 API가 초기화됩니다.

편집 API	설명	입력 매개변수	반환 결과
<b>getCapabilities</b>	섹션 레이아웃의 편집 기능을 설명하는 객체를 반환합니다.	<p>섹션 레이아웃의 편집 기능을 설명하는 JavaScript 객체입니다.</p> <p>입력 시 기본 기능이 함수에 제공됩니다. 함수는 필요에 따라 Capabilities 객체를 수정할 수 있습니다.</p> <p>Capabilities 객체는 다음 기능을 포함할 수 있습니다.</p> <ul style="list-style-type: none"> <li>• <b>title (문자열):</b> 사용자에게 표시할 섹션 레이아웃의 제목입니다.</li> <li>• <b>settingsTitle (문자열):</b> 설정 패널 대화상자에 표시할 제목입니다.</li> <li>• <b>hasSettings (부울):</b> 섹션 레이아웃이 설정 패널을 지원하는지 나타냅니다.</li> <li>• <b>allowMove (부울):</b> 섹션 레이아웃에서 하위 항목을 이동할 수 있는지 나타냅니다.</li> <li>• <b>allowDelete (부울):</b> 섹션 레이아웃에서 하위 항목을 삭제할 수 있는지 나타냅니다.</li> <li>• <b>isHidden (부울):</b> 섹션 레이아웃이 사용자 옵션에 응답하여 현재 숨겨져 있는지 나타냅니다.</li> <li>• <b>dropTarget (부울):</b> 섹션 레이아웃이 끌어 놓기 작업의 대상인지 나타냅니다.</li> <li>• <b>customMenuOptions (배열):</b> 섹션 레이아웃 컨텍스트 메뉴에 추가할 사용자정의 메뉴 옵션입니다. 각 메뉴 옵션 객체에는 다음 속성이 있습니다. <ul style="list-style-type: none"> <li>- <b>label (문자열):</b> 메뉴 항목의 표시 텍스트입니다.</li> <li>- <b>action (함수):</b> 메뉴 항목을 누를 때 호출할 함수입니다.</li> <li>- <b>disabled (부울):</b> 메뉴 항목이 사용 안함 상태로 표시되어야 하는지 나타냅니다.</li> <li>- <b>icon (문자열):</b> 메뉴 항목의 레이블 옆에 표시할 URL입니다. (이 속성은 나중에 사용하도록 예약되어 있습니다.)</li> <li>- <b>checkmark (부울):</b> 메뉴 항목의 레이블 옆에 선택 표시가 있어야 하는지 나타냅니다.</li> </ul> </li> </ul>	(객체): 섹션 레이아웃의 기능입니다.

편집 API	설명	입력 매개변수	반환 결과
<b>getCaptionContent</b>	UI 요소로 나타나는 섹션 레이아웃 표시 이름을 반환합니다.	없음.	(문자열): 섹션 레이아웃의 표시 이름입니다.
<b>filterCapabilities</b>	사용자에게 메뉴를 표시하기 전에 섹션 레이아웃에서 <b>Capabilities</b> 객체를 수정할 수 있습니다. 이 API를 사용하여 메뉴 옵션을 조정하거나 제거할 수 있습니다. ( <b>getCapabilities</b> 도 참조하십시오.)	섹션 레이아웃의 편집 기능을 설명하는 JavaScript 객체입니다. 입력 시 기본 기능이 함수에 제공됩니다.	(객체) 섹션 레이아웃의 기능입니다.
<b>onDragOver</b>	끌어 놓기 작업 중 호출되어 끌어온 항목을 섹션 레이아웃에 놓을 수 있는지 나타냅니다.	<b>eventObject (이벤트 객체):</b> 끌기 이벤트에 대한 정보를 보유한 이벤트 객체입니다. <b>dataTransfer (DataTransfer 객체):</b> 섹션 레이아웃 위로 끌어올 항목에 대한 정보를 보유한 DataTransfer 객체입니다.	(부울) 끌어온 항목을 섹션 레이아웃에서 수락할 수 있는지 나타내는 값입니다. 섹션 레이아웃이 끌어온 항목을 수락할 수 있으면 <i>true</i> 를 반환하고, 그렇지 않으면 <i>false</i> 를 반환합니다.
<b>onDrop</b>	끌어 놓기 작업의 놓기 부분 중 호출되어 끌어온 항목을 섹션 레이아웃 안에 놓아야 하는지 나타냅니다.	<b>eventObject (이벤트 객체):</b> 놓기 이벤트에 대한 정보를 보유한 이벤트 객체입니다. <b>dataTransfer (DataTransfer 객체):</b> 섹션 레이아웃에 놓을 항목에 대한 정보를 보유한 DataTransfer 객체입니다.	(부울) 놓기 작업이 섹션 레이아웃에서 처리되었는지 나타내는 값입니다. <i>true</i> 를 반환하면 기본 논리가 무시됩니다.
<b>onAddComponent</b>	끌어 놓기 작업이 섹션 레이아웃에서 항목을 추가했음을 섹션 레이아웃에 알립니다.	<b>eventObject (이벤트 객체):</b> 끌기 이벤트에 대한 정보를 보유한 이벤트 객체입니다. <b>dataTransfer (DataTransfer 객체):</b> 섹션 레이아웃에 놓을 항목에 대한 정보를 보유한 DataTransfer 객체입니다. <b>componentId (문자열):</b> 새로 추가된 항목의 <i>componentId</i> 값입니다.	섹션 레이아웃 통지입니다.

편집 API	설명	입력 매개변수	반환 결과
<b>onMoveComponent</b>	끌어 놓기 작업이 섹션 레이아웃에서 항목을 이동했음을 섹션 레이아웃에 알립니다.	<b>eventObject (이벤트 객체):</b> 끌기 이벤트에 대한 정보를 보유한 이벤트 객체입니다. <b>dataTransfer (DataTransfer 객체):</b> 섹션 레이아웃에 놓을 항목에 대한 정보를 보유한 DataTransfer 객체입니다. <b>componentId (문자열):</b> 이동된 항목의 componentId 값입니다.	섹션 레이아웃 통지입니다.
<b>getSettingsData</b>	설정 패널을 표시하기 전에 섹션 레이아웃에서 설정 데이터를 변경할 수 있습니다.	<b>settingsData (객체):</b> 섹션 레이아웃에 대해 계산된 기본 설정 데이터입니다.	(객체) 섹션 레이아웃에 대한 설정 데이터입니다.
<b>updateSettings</b>	설정 패널을 닫은 후에 섹션 레이아웃에서 해당 설정을 변경할 수 있습니다. 이 API는 페이지 모델에 설정이 저장되기 직전에 호출됩니다.	<b>parameters (객체):</b> 설정 패널에서 반환된 원시 매개변수 객체입니다. <b>sectionLayoutData (객체):</b> 저장될 섹션 레이아웃 데이터입니다. 기본 데이터는 parameters (객체)에서 생성되어 이 매개변수의 함수로 전달됩니다.	(객체) 페이지 모델에 저장할 섹션 레이아웃 데이터입니다.
<b>dispose</b>	편집 모듈이 메모리를 비우고 이벤트를 분리하며 편집 처리기와 연관된 리소스 할당을 해제할 수 있습니다. 이 API는 실행 취소/재실행 작업의 경우처럼 섹션 레이아웃을 완전히 다시 그려야 할 때 호출됩니다.	없음.	섹션 레이아웃 다시 그리기가 사용으로 설정됩니다.

## 콘텐츠 레이아웃 개발

콘텐츠 레이아웃을 사용하여 사용자는 사이트 페이지에서 사용된 콘텐츠 목록 또는 콘텐츠 위치 표시자 구성요소를 통해 콘텐츠 항목의 데이터를 볼 수 있습니다. 콘텐츠 유형에 대해 여러 콘텐츠 레이아웃을 생성하여 다양한 뷰를 생성하거나 콘텐츠 항목의 다른 부분을 나타낼 수 있습니다.

예를 들어, Blog-Post 콘텐츠 유형은 Blog-Post 콘텐츠가 사용될 방법과 위치에 따라 다른 콘텐츠 레이아웃이 필요할 수 있습니다. 사이트의 홈 페이지는 Blog-Post 항목의 목록을 표시할 수 있지만, 홈 페이지에서 블로그 게시물을 누르면 세부정보 페이지에 해당 블로그 게시물에 대한 세부정보가 표시될 수 있습니다.

홈 페이지에 구성된 콘텐츠 목록은 Blog-Post-Summary 콘텐츠 레이아웃을 항목 뷰로 사용하여 Blog-Post 콘텐츠 유형의 항목을 나열합니다.

세부정보 페이지는 콘텐츠 위치 표시자에 Blog-Post-Header 콘텐츠 레이아웃을 사용하여 헤더 이미지 및 제목을 보여줍니다. 2열 섹션 레이아웃은 Blog-Post-Content 및 Blog-Post-Author 콘텐츠 레이아웃을 사용하여 두 개의 콘텐츠 위치 표시자를 70% 대 30% 너비로 묶습니다. 보시다시피, 4개의 다른 콘텐츠 레이아웃이 동일한 콘텐츠 유형 시각화에 사용됩니다.

다음 두 가지 방법 중 하나로 콘텐츠 레이아웃을 생성할 수 있습니다.

- Oracle Content Management에서 **개발자 > 모든 구성요소 보기 > 생성 > 콘텐츠 레이아웃 생성**을 선택합니다.
- OCE Toolkit 프로젝트에서 `cec create-contentlayout` 명령을 사용합니다.

Oracle Content Management는 콘텐츠 유형에 대한 기본 콘텐츠 레이아웃을 생성합니다. 기본 콘텐츠 레이아웃을 수정하려면 다음 파일을 편집할 수 있습니다.

- `assets/layout.html`  
이 파일을 편집하여 HTML 뷰를 변경합니다.
- `assets/design.css`  
이 파일을 편집하여 콘텐츠 레이아웃을 스타일링합니다.
- `assets/render.js`  
이 파일을 편집하여 `layout.html`에서 사용된 데이터를 변경하거나 콘텐츠 레이아웃에 동적 동작을 추가합니다.

콘텐츠 레이아웃 구성요소는 Oracle Content Management 서버에서 콘텐츠 항목을 렌더링합니다. 대부분의 자산은 Oracle Content Management 서버에 저장됩니다. 때때로 스타일링 목적의 배경 이미지와 같이 콘텐츠 레이아웃 자체에서 로컬로 제공되는 정적 자산을 사용하려는 경우가 있습니다. 예를 들어, 다음 콘텐츠 레이아웃에서 `images/background.jpg`의 절대 URL을 `render.js`에 생성하고 `layout.html`에서 사용할 수 있습니다.

절대 URL을 생성하는 가장 쉬운 방법은 내장 `requirejs` 파일을 사용하는 것입니다. 다음 코드와 같이 'require'를 종속성으로 정의하고 `require.toURL()`을 사용하여 URL을 생성합니다.

```
define([
  'require',
  'jquery',
  'mustache',
  'text!./layout.html',
  'css!./design.css'
], function (require, $, Mustache, templateHtml, css) {
  'use strict';

  var imageURL = req.toUrl('./images/background.jpg');
```

다음 항목은 콘텐츠 레이아웃을 개발하는 방법을 설명합니다.

- [Oracle Content Management로 콘텐츠 레이아웃 생성](#)
- [콘텐츠 레이아웃에 레이아웃 뷰 전달](#)
- [API를 사용한 사이트 세부정보 페이지 URL 생성](#)

- Developer Cloud Service로 로컬에서 콘텐츠 레이아웃 개발
- 콘텐츠 목록 질의에서 매크로 확장
- 강력한 콘텐츠 레이아웃 개발
- 샘플 블로그 템플릿 생성
- 콘텐츠 레이아웃 매핑을 템플릿에 추가
- 로컬 테스트 하네스로 콘텐츠 레이아웃 테스트
- 콘텐츠 레이아웃이 있는 템플릿을 Oracle Content Management로 임포트

#### 관련 항목

콘텐츠 레이아웃 관리 방법에 대한 자세한 내용은 [사용자정의 구성요소 및 레이아웃 관리](#)를 참조하십시오.

디지털 자산 사용과 사이트의 기타 콘텐츠 항목에 대한 자세한 내용은 *Managing Assets with Oracle Content Management*의 자산 사용 및 디지털 자산 관리를 참조하십시오.

## Oracle Content Management로 콘텐츠 레이아웃 생성

사이트 페이지에 표시될 때 콘텐츠 항목의 필드를 배치하기 위한 콘텐츠 레이아웃을 생성합니다. 콘텐츠 항목을 페이지에 추가할 때 선택한 콘텐츠 레이아웃을 사용합니다.

엔터프라이즈 사용자는 개발자가 제공한 콘텐츠 유형 및 레이아웃을 기반으로 콘텐츠 항목을 생성하고 사용할 수 있습니다. 기본 레이아웃에서 새 콘텐츠 레이아웃을 생성할 수 있습니다. 콘텐츠 유형과 연관된 여러 콘텐츠 레이아웃을 사용하면 사이트 디자이너가 콘텐츠를 변경하지 않고도 콘텐츠 항목을 다양한 컨텍스트로 표시할 수 있습니다.

콘텐츠 목록 구성요소에서 콘텐츠 레이아웃을 사용할 경우 콘텐츠 항목당 한번씩 콘텐츠 레이아웃이 반복됩니다. 그러면 콘텐츠 레이아웃은 섹션 레이아웃에 따라 배열됩니다.

콘텐츠 레이아웃을 익스포트하고 오프라인에서 수정한 후 임포트하여 기존 콘텐츠 레이아웃을 바꿀 수 있습니다.

콘텐츠 레이아웃을 생성하려면 다음과 같이 하십시오.

1. Oracle Oracle Content Management 홈 페이지의 측면 탐색에서 **개발자**를 누릅니다.  
개발자 페이지가 표시됩니다.
2. **모든 구성요소 보기**를 누릅니다.
3. 오른쪽의 **생성** 드롭다운 메뉴에서 **콘텐츠 레이아웃 생성**을 선택합니다.
4. **콘텐츠 레이아웃 생성** 대화상자에서 레이아웃을 사용할 콘텐츠 유형을 선택하고, 표시할 필드를 선택하고, 콘텐츠 항목 및 콘텐츠 목록 구성요소를 페이지에 추가할 때 사이트 생성자가 사용자정의 설정 및 스타일을 추가할 수 있도록 하려면 **사이트에서 사용하는 경우 사용자정의 설정에 대한 지원 추가**를 사용으로 설정합니다.



### 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

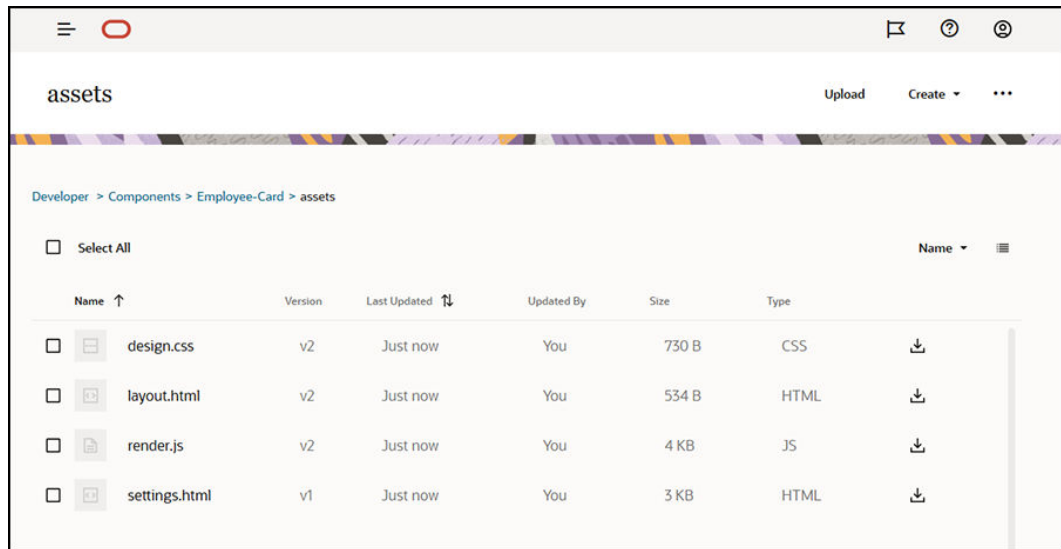
5. 콘텐츠 레이아웃 구성요소에 대한 이름과 설명을 제공하고 **생성**을 누릅니다. 콘텐츠 레이아웃이 구성요소에 추가됩니다.

 **주:**

영숫자, 하이픈 및 밑줄만 콘텐츠 레이아웃 제목에 적합합니다.

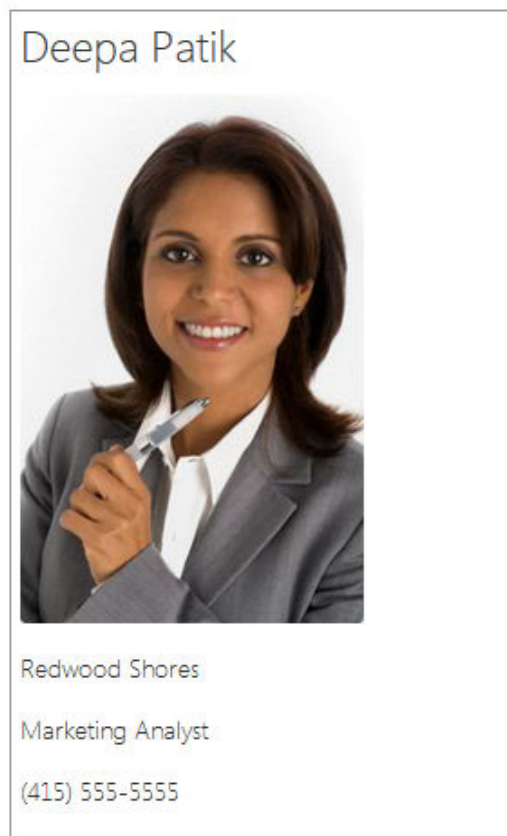
6. 편집을 위해 콘텐츠 레이아웃을 익스포트하려면 선택 후 마우스 오른쪽 단추 누르기 메뉴에서 **익스포트**를 누르거나 작업 모음에서  을 누릅니다.
  - a. 기존 폴더로 이동하거나 **생성**, **폴더**를 차례로 선택하여 새 폴더를 생성하고 이름과 선택적 설명을 제공합니다.
  - b. 폴더 옆에 있는 체크박스를 선택하고 **확인**을 누릅니다.
  - c. 폴더의 아이콘 또는 이름을 눌러 엽니다.

레이아웃 패키지 파일이 콘텐츠 레이아웃 이름에 `.zip` 확장자가 붙은 형태로 선택한 폴더에 생성됩니다. 개발 환경으로 파일을 다운로드하여 파일을 편집합니다.

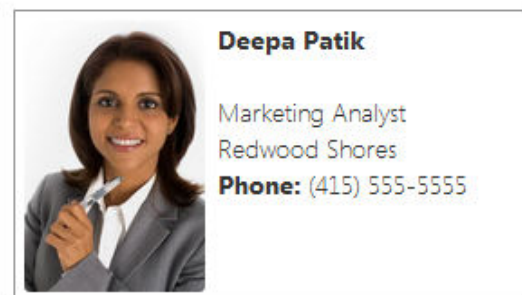


이 파일은 Employee-Card 콘텐츠 레이아웃을 사용하는 콘텐츠 항목의 필드 레이아웃을 제어합니다. 사이트에서 사용하는 경우 사용자정의 설정에 대한 지원 추가를 사용으로 설정한 경우, 단일 콘텐츠 항목의 기본 렌더링을 표시할 수 있도록 제공하는 **settings.html**이라는 추가 파일도 생성됩니다.

### Default Layout



### Card Layout



7. design.css, layout.html 및 render.js 파일을 편집하여 원하는 콘텐츠 레이아웃을 연습합니다.

예를 들어, 다음 파일은 Employee-Card 콘텐츠 레이아웃을 지정합니다.

- a. design.css 파일을 편집합니다.

```
.scs-tile-layout {
    font-family: 'Helvetica Neue', 'Segoe UI', sans-serif-
regular, Helvetica, Arial;
    font-size: 16px;
    margin: 0px;
    padding: 0px;
    font-style: normal;
    color: #333;
}

.scs-tile-layout li {
    list-style: none;
    font-size: 14px;
    font-style: normal;
    font-variant-caps: normal;
    font-weight: 200;
    margin: 0px;
}

.scs-tile-layout-img-container {
    height: 150px;
    width: 100px;
    float: left;
    margin: 0em 0.5em 0em 0em;
    padding: 0px;
    border-radius: 3px;
    overflow: hidden;
    position: relative;
}

.scs-tile-layout-img {
    position: absolute;
    left: -100%;
    right: -100%;
    top: -100%;
    bottom: -100%;
    margin: auto;
    height: 100%;
    min-width: 100%;
}

.scs-tile-layout p {
    margin: 0px;
}
```

- b. layout.html 파일을 편집합니다.

```
{{#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. render.js 파일을 편집합니다.

```

/* 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;
});

```

render.js 및 기타 파일 편집에 대한 자세한 내용은 [구성요소 개발](#)을 참조하십시오.

8. 수정된 파일을 임포트하여 기존 콘텐츠 레이아웃을 바꿉니다.
  - a. 홈 페이지에서 **문서**를 누릅니다.
  - b. 익스포트할 때와 동일한 폴더 및 파일 이름을 포함한 **.zip** 확장자 파일로 수정된 콘텐츠 레이아웃을 폴더에 업로드합니다.  
새 콘텐츠 레이아웃으로 임포트하려면 `_folder.json`에서 콘텐츠 레이아웃의 GUID를 변경해야 합니다.
  - c. 홈 페이지에서 **개발자**를 누릅니다.  
**개발자** 페이지가 표시됩니다.
  - d. 모든 **구성요소 보기**를 누릅니다.
  - e. **생성** 메뉴에서 **구성요소 임포트**를 선택합니다.

- f. 수정된 구성요소를 포함한 업로드된 zip 파일 옆에 있는 체크박스를 선택하고 **확인**을 누릅니다.

수정된 콘텐츠 레이아웃이 구성요소로 импорт됩니다.

또한 콘텐츠 레이아웃을 익스포트하여 다른 Oracle Content Management 인스턴스로 복사/이동하고 거기서 импорт할 수도 있습니다.

## 콘텐츠 레이아웃에 레이아웃 뷰 전달

콘텐츠 레이아웃을 개발할 때 콘텐츠 레이아웃이 사이트에서 사용되는 경우 기본 구성요소의 속성을 가져올 수 있습니다.

이 특정 용례의 경우 콘텐츠 목록에서 콘텐츠 유형에 대해 선택된 콘텐츠 레이아웃 뷰를 가져올 수 있습니다. 그런 다음 범주 유형에 따라 구성요소 렌더링 방법을 변경할 수 있습니다. 이 속성에 액세스할 수 없으면 사실상 동일한 두 개의 콘텐츠 레이아웃을 생성해야 합니다.

contentLayoutCategory 속성은 콘텐츠 항목 및 콘텐츠 목록 구성요소를 위해 렌더링된 콘텐츠 레이아웃에 사용할 수 있습니다. 다음과 같이 Sites SDK를 통해 이 속성에 액세스할 수 있습니다.

```
scsData.SitesSDK.getProperty('contentLayoutCategory', function
(layoutCategory)
    { console.log(layoutCategory);});
```

Oracle Content Management SDK를 참조하십시오.

이 속성은 Oracle Content Management 사이트에 있는 콘텐츠 항목 및 콘텐츠 목록에서 콘텐츠 레이아웃이 렌더링되는 경우에만 사용할 수 있습니다. 콘텐츠 레이아웃이 타사 앱에서 렌더링되는 경우 사용할 수 없습니다.

## API를 사용한 사이트 세부정보 페이지 URL 생성

고유 질의를 통해 콘텐츠 레이아웃의 콘텐츠 항목 목록을 렌더링하는 경우 콘텐츠 항목에 대한 세부정보 페이지의 링크를 생성할 수 있습니다. Sites SDK SCSRenderAPI.getPageLinkData API를 사용하여 **사이트 세부정보 페이지 URL**을 생성할 수 있습니다.

detailPageId 속성은 콘텐츠 항목 또는 콘텐츠 목록에서 선택된 세부정보 페이지의 ID입니다. 값이 설정되지 않은 경우 반환 값은 isDetailPage 속성이 설정된 SiteStructureMap의 첫번째 페이지입니다.

이 속성에 액세스하려면 Sites SDK를 사용할 수 있습니다. 이 속성은 콘텐츠 레이아웃이 Oracle Content Management 사이트에 사용되는 경우에만 사용할 수 있습니다. 콘텐츠 레이아웃을 생성할 때 인수 중 하나로 전달된 scsData를 통해서만 액세스할 수 있습니다. 예를 들어, 다음과 같습니다.

```
scsData.SitesSDK.getProperty('detailPageId', function (detailPageId)
{ console.log(detailPageId);});
```

detailPageId가 있으면 이를 사용하여 세부정보 페이지의 링크를 생성할 수 있습니다.

SCSRenderAPI에는 getPageLinkData() 함수가 있어서 pageId 및 추가 옵션을 취하고 이 옵션을 통과하는 페이지에 필요한 URL을 생성합니다. 이 함수의 서명은 다음과 같습니다.

```
SCSRenderAPI.getPageLinkData(pageId,
options);
```

다음 매개변수가 있습니다.

- **pageId:** Sites SDK detailPageId 속성에서 반환된 detailPageId와 동일합니다.
- **options:**
  - contentType
  - contentId
  - contentName

반환 값은 다음 속성을 가진 객체입니다.

- hideInNavigation
- href
- href

다음 예제는 이 모두를 종합한 것입니다.

```
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);
});
```

인쇄 출력: "/sites/{site}/{detailPageName}/{contentType}/{contentId}/{contentSlug}"

pageId가 세부정보 페이지가 아닌 경우 콘텐츠 값은 URL에 추가되지 않습니다.

Oracle Content Management SDK를 참조하십시오.

## Developer Cloud Service로 로컬에서 콘텐츠 레이아웃 개발

OCE Toolkit을 통해 Developer Cloud Service를 사용하여 Oracle Content Management에 대해 로컬에서 콘텐츠 레이아웃을 생성, 편집, 구성 및 테스트할 수 있습니다.

다음 단계에 따라 Developer Cloud Service로 콘텐츠 레이아웃을 개발할 준비를 합니다.

1. 로컬 시스템에서 OCE Toolkit 설정.
2. Oracle Content Management용 Developer Cloud Service 콘솔에 사인인
3. Developer Cloud Service에서 프로젝트 생성.
4. 새 Git 저장소에서 프로젝트 코드에 OCE Toolkit 추가.
5. 콘텐츠 레이아웃 생성

다음 항목은 OCE Toolkit을 사용하여 Developer Cloud Service로 로컬에서 콘텐츠 레이아웃을 개발하는 방법에 대해 설명합니다.

- Developer Cloud Service로 콘텐츠 레이아웃 생성
- RequireJS 모듈 정의
- 생성자 함수 매개변수 구성
- 콘텐츠 레이아웃 렌더링
- Mustache 템플릿에서 콘텐츠 레이아웃 편집
- 동적 DOM 조작 추가
- design.css 파일에서 스타일 정의
- 참조 항목 가져오기
- 미디어 URL 가져오기
- 트리거 발생
- 검색 질의를 사용하여 검색 페이지로 이동
- 매크로 확장 및 서식 있는 텍스트 렌더링
- 세부정보 페이지에 링크

Developer Cloud Service 템플릿으로 콘텐츠 레이아웃을 개발한 후에 프로젝트의 Git 저장소와 변경사항 병합을 수행합니다.

## Developer Cloud Service로 콘텐츠 레이아웃 생성

cec create-contentlayout 명령을 사용하여 Developer Cloud Service 프로젝트에서 콘텐츠 유형에 대한 콘텐츠 레이아웃을 생성할 수 있습니다.

Developer Cloud Service 프로젝트에서 콘텐츠 레이아웃을 생성하려면 다음과 같이 하십시오.

1. 터미널 창에서 cec-components 디렉토리로 이동합니다.
2. cec create-contentlayout을 입력하여 명령 옵션 및 예제를 확인합니다.

```
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. 다음 명령을 입력하여 서버에서 사용 가능한 콘텐츠 유형을 확인합니다.

```
cec list-server-content-types
```

4. "Usage"에 표시된 대로, Oracle Content Management 서버의 콘텐츠 유형 또는 `cec-components/src/main/templates` 아래 템플릿의 콘텐츠 유형에 대한 콘텐츠 레이아웃을 생성할 수 있습니다. 예를 들어, 다음 명령은 서버의 유형에 대한 콘텐츠 레이아웃을 생성합니다.

```
cec create-contentlayout Blog-Post-Overview-Layout -c Blog-Post -r
```

다음 파일을 편집하여 콘텐츠 레이아웃을 수정할 수 있습니다.

- `assets/layout.html`  
이 파일은 HTML 뷰를 지정합니다. [Mustache 템플릿에서 콘텐츠 레이아웃 편집을 참조하십시오.](#)
- `assets/design.css`  
이 파일은 콘텐츠 레이아웃의 스타일을 지정합니다. [design.css 파일에서 스타일 정의를 참조하십시오.](#)
- `assets/render.js`  
이 파일은 `layout.html`에서 사용된 데이터를 지정하며 콘텐츠 레이아웃에 동적 동작을 추가할 수 있습니다. [RequireJS 모듈 정의를 참조하십시오.](#)

사용자정의 설정을 허용하는 레이아웃을 생성한 경우(레이아웃을 생성할 때 `-a` 옵션 사용), 단일 콘텐츠 항목의 기본 렌더링을 표시할 수 있도록 제공하는 `settings.html`이라는 추가 파일도 생성됩니다.

## RequireJS 모듈 정의

`render.js` 파일에서 RequireJS 모듈을 정의할 수 있습니다. 사이트는 JQuery, Mustache, RequireJS 텍스트 플러그인 및 RequireJS CSS 플러그인과 같은 종속성을 로드합니다.

```
define([
  'jquery',
  'mustache',
  'text!./layout.html',
  'css!./design.css'
], function ($, Mustache, templateHtml, css) {
```

Mustache 템플릿 시스템을 사용하여 레이아웃을 렌더링할 수 있습니다.

콘텐츠 레이아웃의 `assets/render.js` 파일에는 다음 속성이 있습니다.

- RequireJS 모듈이어야 합니다.
- JavaScript 생성자 함수를 반환해야 합니다. 사이트는 매개변수 객체를 전달하여 생성자 함수를 호출합니다. 매개변수 객체에는 콘텐츠 항목 데이터와 레이아웃 렌더링에 필요한 API가 있습니다.
- 이 생성자 함수에는 콘텐츠 레이아웃 렌더링을 처리하는 `render(parentObj)` 메소드가 있어야 합니다. `render()` 메소드로 전달된 `parentObj` 객체에 콘텐츠 레이아웃 DOM 객체를 추가해야 합니다.
- RequireJS 모듈은 JQuery, Mustache, RequireJS 텍스트 플러그인 및 RequireJS CSS 플러그인을 포함한 종속성을 사용할 수 있습니다. 이 종속성은 사이트에 의해 로드됩니다. 다른 라이브러리도 사용할 수 있습니다.

## 생성자 함수 매개변수 구성

사이트에서 생성자 함수의 새 인스턴스를 생성할 때 콘텐츠 레이아웃 개발을 돕기 위해 `contentItemData`, `scsData` 및 `contentClient`가 포함된 매개변수를 전달합니다.

다음은 생성자 함수의 예제 코드입니다.

```
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;
```

생성자 함수 매개변수에는 다음 객체가 포함됩니다.

- **params.contentItemData:** 이름, 설명, ID 및 데이터를 비롯한 콘텐츠 항목을 포함합니다. 예를 들어, 콘텐츠 항목의 `'blogpost_title'` 필드는 `params.contentItemData.data['blogpost_title']`을 사용하여 액세스할 수 있습니다.
- **params.scsData:** 이 객체는 생성자가 사이트 내에서 호출될 때 정보를 전달합니다. 이 객체는 타사 애플리케이션에서 렌더링된 콘텐츠 레이아웃에는 존재하지 않습니다. 이 객체에는 Sites SDK 객체, 트리거를 발생시키는 `contentTriggerFunction` 메소드 및 세부정보 페이지 링크가 포함됩니다.

- **params.contentClient:** Content SDK에서 생성된 `contentClient` 객체로, 콘텐츠 레이아웃을 호출하는 데 사용됩니다. 따라서 콘텐츠 서버에 적합한 매개변수로 구성됩니다. 콘텐츠 서버에 추가 호출을 실행해야 하는 경우 직접 생성하는 대신 이 `contentClient` 객체를 사용할 수 있습니다. 이 객체에는 콘텐츠의 클라이언트 API가 포함됩니다. API에서 콘텐츠 항목과 해당 콘텐츠 유형을 질의, 검색하고 가져올 수 있습니다. 다른 헬퍼 API도 사용할 수 있습니다. 예를 들어, `expandMacros()`는 서식 있는 텍스트에 사용된 매크로를 확장합니다.

## 콘텐츠 레이아웃 렌더링

`ContentLayout`의 `render(params)` 메소드는 템플릿에서 콘텐츠 레이아웃을 렌더링합니다. `Mustache` 템플릿이 콘텐츠 레이아웃에 기본적으로 사용되지만, 원하는 템플릿 기술을 사용할 수 있습니다.

`ContentLayout`의 `render(params)` 메소드는 다음 코드를 사용하여 데이터와 함께 템플릿을 렌더링할 수 있습니다.

```
try {
  // Mustache
  template = Mustache.render(templateHtml, content);
  if (template) {
    $(parentObj).append(template);
  }

  // Dynamic DOM Manipulation can be done here

} catch (e) {
  console.error(e.stack);
}
```

`params.contentItemData`에서 생성된 `content` 객체에 필요한 데이터를 추가할 수 있습니다. 오라클은 `params.scsData`의 속성을 이 객체로 병합하여 템플릿에서도 사용할 수 있도록 권장합니다. 렌더링된 템플릿은 `render()` 메소드로 전달된 상위 객체에 추가되어야 합니다.

## Mustache 템플릿에서 콘텐츠 레이아웃 편집

기본 콘텐츠 레이아웃이 포함된 `assets/layout.html` 파일에서 기본 `Mustache` 템플릿을 편집할 수 있습니다.

블로그 제목을 렌더링하는 간단한 템플릿은 다음과 같습니다.

```
<h1>
  {{blogTitle}}
</h1>
```

## 동적 DOM 조작 추가

`Mustache.render()`가 호출되고 템플릿이 상위 객체에 추가된 후 `render.js`에 동적 DOM 조작을 추가할 수 있습니다.

예를 들어, 스타일을 동적으로 변경하면서 리스너를 첨부할 수 있습니다.

```
// Dynamic DOM Manipulation can be done here
$('h1').click(function (event) {
    alert('Title is : ' + $(this).text());
});
```

## design.css 파일에서 스타일 정의

design.css 파일에서 콘텐츠 레이아웃 템플릿에 사용된 모든 스타일을 정의할 수 있습니다.

design.css 파일은 RequireJS CSS 플러그인을 사용하여 모듈 정의에 로드됩니다.

## 참조 항목 가져오기

다른 콘텐츠 유형을 참조하는 참조 데이터 필드를 사용하여 콘텐츠 유형에 대한 참조 항목을 가져올 수 있습니다.

예를 들어, Blog-Post 콘텐츠 유형의 Author 필드는 Author 콘텐츠 유형을 참조합니다. Blog-Post의 콘텐츠 레이아웃에서 contentClient.getItems()는 현재 Blog-Post 항목과 Author 참조 항목의 세부정보를 연관시킵니다.

```
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
});
```

## 미디어 URL 가져오기

contentClient.getRenditionURL()을 사용하여 이미지와 같은 디지털 자산의 기본 표현을 가져올 수 있습니다.

```
blogHeaderImage = contentClient.getRenditionURL({
    'id': data['blog-post_header_image'].id
}),
```

thumbnail과 같은 기타 표현이 필요한 경우 contentClient.getItems()를 사용하여 디지털 자산을 가져와서 item.data.renditions.default 및 item.data.renditions.thumbnail을 참조할 수 있습니다.

## 트리거 발생

scsData.contentTriggerFunction(payload)를 사용하여 콘텐츠 레이아웃에서 트리거를 발생시킬 수 있습니다.

다음은 작성자 이름을 누를 때 트리거를 발생시키는 Author 콘텐츠 레이아웃의 예입니다.

```

{{#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}}

```

페이로드는 현재 선택된 작성자에 대한 검색 질의로, 페이지의 다른 콘텐츠 목록 항목이 수신할 수 있습니다.

## 검색 질의를 사용하여 검색 페이지로 이동

일반적인 용례는 콘텐츠 레이아웃 안의 링크를 누를 때 동적 검색 질의를 사용하여 검색 페이지로 이동하는 것입니다.

예를 들어, 콘텐츠 레이아웃에서 "이 작성자의 다른 기사" 링크를 누르고 검색 페이로드를 전달할 때 "작성자"라는 검색 페이지로 이동한다고 가정해 보겠습니다. 다음 코드가 이 작업을 수행합니다. 전역 객체 SCS 및 SCSRenderAPI는 사이트 페이지 내에 실행될 때 콘텐츠 레이아웃에서 사용할 수 있습니다.

```

$($('.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));
```

동일한 콘텐츠 레이아웃을 동일한 페이지에서 여러 번 사용하려는 경우 `$('.more-from-author').click(...)`과 같은 클래스 선택기가 아닌 CSS 선택기에서 고유 ID를 사용하는 것이 좋습니다.

예를 들어, 다음과 같습니다.

```
template.html
  <div id="{{navigateId}}">...</div>

render.js
  content.navigateId = this.scsData.id + 'detailTrigger';
  $('#'+ navigateId).click(...)
```

## 매크로 확장 및 서식 있는 텍스트 렌더링

콘텐츠 항목의 서식 있는 텍스트는 디지털 이미지를 포함할 수 있습니다.

콘텐츠 레이아웃에서 이 서식 있는 텍스트를 올바르게 렌더링하기 위해 서식 있는 텍스트 필드는 `contentClient.expandMacros()` API를 사용합니다. 서식 있는 텍스트 안의 디지털 자산에 대한 모든 참조가 해결됩니다.

```
data["starter-blog-post_content"] =
  contentClient.expandMacros(data["starter-blog-post_content"]);
```

렌더링을 위해 `Mustache`를 사용할 경우 서식 있는 텍스트에 HTML이 있으므로 서식 있는 텍스트 값을 렌더링하려면 `{{{ }}}`를 사용해야 합니다. `{{{ }}}`가 변수 주위에 사용될 때 `Mustache`는 HTML을 이스케이프 처리하지 않습니다.

## 세부정보 페이지에 링크

세부정보 페이지 링크는 `scsData.detailPageLink`를 통해 사용할 수 있습니다.

예를 들어, 블로그 제목을 누를 때 블로그 세부정보를 표시하는 세부정보 페이지로 이동하려면 다음과 같이 세부정보 페이지 링크를 사용할 수 있습니다.

```
<a href="{{scsData.detailPageLink}}">
  <h2 class="post-title">
    {{blogTitle}}
  </h2>
</a>
```

## 콘텐츠 목록 질의에서 매크로 확장

콘텐츠 목록 질의에서 페이지를 실행할 때 계산되는 속성 값을 정의하여 최근에 업데이트된 콘텐츠를 표시할 수 있습니다.

사이트 내 구성요소의 속성은 대부분 정적입니다. 사용자는 구성요소의 속성 중 하나에 대해 고정 문자열 또는 값을 선택하거나 입력하며, 페이지가 실행되는 시기나 위치에 관계없이 변경되지 않습니다. 그러나 페이지를 실행할 때 계산되는 속성 값을 정의할 수 있습니다. 이는 콘텐츠

질의에서 최근에 업데이트된 콘텐츠를 표시하는 데 유용합니다. 사용자는 "최근 3일 이내"와 같은 날짜를 입력할 수 있습니다.

Mustache JS 확장을 여러 속성에 삽입할 수 있습니다. 이 문자열에서 참조된 값은 페이지를 실행할 때 실행되는 모델에서 파생됩니다. 미리 정의된 모델은 Content REST API 호출용으로 형식 지정된 날짜를 처리합니다. 이 모델을 추가 값으로 확장하여 사용자 요구사항을 충족시킬 수 있습니다.

속성에 입력할 수 있는 문자열의 예는 다음과 같습니다.

```
Content List component:
  Additional Query String property:
    updatedDate gt "{{#content.date}}today - 3 days{{/
content.date}}"
```

이 Mustache 날짜 입력은 런타임에 평가되므로 실행 시간에 따라 반환 값이 변경됩니다(즉, updatedDate gt "2220181002060000000"으로 확장됨). 이와 같이 사용자는 미리 정의된 값을 입력할 필요 없이 복잡한 날짜 문자열을 구성할 수 있습니다.

### 지원되는 구성요소 속성

다음 속성은 Mustache JS 템플리트 구문을 지원합니다.

- 콘텐츠 목록
  - 추가 질의 문자열
  - 예: updatedDate gt "{{#content.date}}today - 3 days{{/content.date}}"
- 제목/단락/텍스트
  - CKEditor를 통해 입력한 서식 있는 텍스트
  - 예: "Content REST API format for date: {{#content.date}}now{{/content.date}}"

#### 주:

Mustache 템플리트에 대한 사용자정의 모델이 없으면 제목/단락/텍스트 확장은 별로 쓸모가 없습니다. 그러나 추가 질의 문자열에 입력한 내용은 편집-보기 간을 전환할 때 평가되어 즉시 표시되므로 검증에 매우 유용합니다.

### 지원되는 구성요소 구문

content.date 객체가 기본적으로 지원됩니다. 두 가지 주요 매개변수로 today 및 now를 취합니다.

today 값은 현재 브라우저 시간을 가져와서 오늘날 자정으로 변환한 후 해당 값을 UTC 시간으로 변환합니다.

- {{#content.date}}today{{/content.date}}는 오늘날 자정에 브라우저 값이 UTC 값으로 변환되어 Content REST API 날짜 형식이 지정된 형태로 확장됩니다. 예를 들어, 다음과 같습니다.

```
2220181008065959999
```

- 다음과 같이 보강할 수 있습니다.

```
today +/- [day | week | month | year]
```

- today 값은 더하기 또는 빼기 시 다르게 작동합니다. 빼기 시에는 오전 시간을 사용합니다. 더하기 시에는 자정 시간을 사용합니다. 예를 들어, 다음과 같습니다.
  - `{{#content.date}}today - 1 day{{/content.date}}`는 하루 시작에 어제로 확장됩니다.
  - `{{#content.date}}today + 2 days{{/content.date}}`는 자정에 내일 모레로 확장됩니다.

now 값은 현재 브라우저 시간을 가져와서 아무 조정 없이 UTC 시간으로 변환합니다.

- `{{#content.date}}now{{/content.date}}`는 현재 브라우저 시간이 UTC 값으로 변환되어 Content REST API 날짜 형식이 지정된 형태로 확장됩니다.
- now는 hour로 보강할 수도 있습니다. 그러면 다음과 같습니다.

```
now +/- [hour | day | week | month | year]
```

- 예를 들어, 다음과 같습니다.
  - `{{#content.date}}now + 2 hours{{/content.date}}` - 지금부터 2시간 후가 UTC 시간으로 변환되어 Content REST API 날짜 형식이 지정된 형태
  - `{{#content.date}}now - 1 day{{/content.date}}` - 이 브라우저 시간의 어제가 UTC 시간으로 변환되어 Content REST API 날짜 형식이 지정된 형태

#### 지원되는 구성요소 구문 사용

추가 질의 문자열에서 매크로 확장을 사용하려면 최근 3주간 모든 항목을 반환한다고 가정할 때 다음을 입력하십시오.

```
updatedAt gt "{{#code.date}}today - 3 weeks{{/code.date}}"
```

날짜만 반환되므로 Content REST API 호출에 작동하려면 정적 값을 입력할 때와 마찬가지로 질의 문자열을 작성할 때 따옴표가 추가됩니다.

#### MustacheJS

구문은 [mustache.github.io/mustache.5.html](http://mustache.github.io/mustache.5.html)에서 Mustache JS 템플릿 페이지를 참조하십시오.

문자열을 확장할 때 실행되는 Mustache 인스턴스에 한 가지 변경사항이 있었습니다. Mustache는 `{{ }}`를 사용하는 text 확장과 `{{{ }}`를 사용하는 html 확장을 모두 제공합니다. 둘 사이의 차이점은 텍스트 확장은 문자열에 HTML 인코딩을 한다는 것입니다. 즉, 값이 `a < b`로 확장되면 결과는 `a &lt; b`가 됩니다. 이것은 URL에 문자열을 작성하려는 것이 아닙니다. 사용자에게 HTML 확장을 사용하도록 지시할 수 있지만, 이는 오버헤드일 뿐이며 `{{{ }}`를 사용해야 하는 이유를 설명하는 등의 더 많은 문제가 발생합니다.

이 문제를 피하기 위해 Mustache는 `{{ }}`를 사용할 때 값을 이스케이프하지 않도록 설정되었습니다. 즉, `{{ }}` 및 `{{{ }}`가 똑같이 작동합니다. 또한 필요한 경우 사용자 연습용으로 결과 인코딩을 남겨 둡니다.



## OOTB Mustache 모델

Mustache는 확장을 위해 템플릿에 모델을 적용해야 합니다. 앞의 예제에서 `{{#content.date}}`는 이미 기본적으로 정의되어 있지만, 개발자가 `{{person}}`과 같은 새 속성을 추가할 수 있습니다. 사용자가 모델에 없는 Mustache 템플릿에 값을 입력하면 결과는 빈 문자열이 됩니다. 따라서 `Hello {{person}}`의 경우 개발자가 모델에 `person`을 추가하지 않는 한 단지 `Hello` 로 확장됩니다.

사용된 모델 객체는 `SCSMacros`라는 전역 객체입니다. 개발자는 자유롭게 이 객체에 추가 항목을 추가할 수 있습니다. 템플릿을 평가할 때 객체가 Mustache로 전달됩니다.

미리 정의된 모델 객체는 현재 `content.date` 객체만 지원합니다.

```
{
  content: {
    date: <lambda implementation>
  }
}
```

## 사용자정의 Mustache 모델

개발자의 요구사항에 따라 지원되는 객체를 개선할 수 있습니다. 따라서 `lastTwoDays` 객체를 도입하여 `{{lastTwoDays}}`까지만 확장을 간소화할 수 있습니다.

앞의 예제에서 `Hello {{person}}`과 같은 것을 지원하도록 모델을 확장하려면 `SCSMacros`에 `person` 객체를 추가해야 합니다. 페이지를 실행하기 전에 이 작업을 수행해야 합니다. 시작 부분에 스크립트 태그를 추가하여 페이지 레이아웃 내에서 수행할 수 있습니다. 예를 들어, 다음과 같습니다.

```
<script type="text/javascript">
window.SCSMacros = window.SCSMacros || {}; // define/get the SCSMacros
object
window.SCSMacros.person = "World";
</script>
```

이와 같이 변경하면 `Hello {{person}}` 템플릿이 `Hello World`로 확장됩니다.

`Hello {{#person}}personId{/person}`과 같이 객체에 값을 전달하려면 `mustache lambda`를 구현하고 구현 내에서 값을 래핑하고 확장해야 합니다.

예를 들어, 다음과 같습니다.

```
<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>
```

이와 같이 변경하면 Hello `{{#person}}111{{/person}}` 템플릿이 Hello Small로 확장되고 Hello `{{#person}}222{{/person}}` 템플릿이 Hello Big으로 확장됩니다.

### 주:

Mustache 확장은 동기적으로 실행됩니다. 비동기 값을 검색해야 하는 경우 Mustache 확장을 실행하기 전에 모델 내에서 값을 확인해야 하며, 맞춤형 구현이 가능하지만 현재 지원되지 않습니다.

## 강력한 콘텐츠 레이아웃 개발

콘텐츠 레이아웃은 Content REST 호출에서 얻은 세 가지 유형의 응답 데이터에 대해 견고해야 합니다.

- 콘텐츠 항목: `expand=all` 매개변수가 있으면 참조가 확장되고 큰 텍스트 필드를 얻습니다.
- 콘텐츠 항목: `expand=all`이 없으면 참조가 확장되지 않지만 큰 텍스트 필드를 얻습니다.
- 콘텐츠 질의는 참조가 확장되지 않거나 큰 텍스트 필드가 없습니다.

성능 향상을 위해 Oracle Content Management 릴리스를 기준으로 자산 데이터를 검색하는 Content REST 호출에 더 이상 `expand=all` 매개변수가 포함되지 않습니다. `expand` 매개변수는 Oracle Content Management 응답에서 현재 항목뿐만 아니라 모든 참조된 항목을 드릴다운하여 검색하도록 지시합니다. `expand` 매개변수를 통해 검색된 데이터에 의존하는 사용자정의 콘텐츠 레이아웃을 업데이트하여 검색된 데이터에 참조된 필드 값이 포함되지 않은 사례를 처리해야 합니다.

## 콘텐츠 항목 렌더링

콘텐츠 레이아웃은 콘텐츠 항목 렌더링에 사용됩니다. 콘텐츠 항목 데이터를 수신하고 HTML로 렌더링하여 페이지에 삽입합니다.

기본적으로 콘텐츠 레이아웃은 JavaScript 기술로 구현할 수 있더라도 Mustache 템플릿을 활용하여 콘텐츠 항목을 렌더링합니다. Mustache 템플릿이 렌더링하려면 특정 형식의 데이터가 필요합니다. 콘텐츠 레이아웃 `render.js` 파일은 템플릿으로 전달되는 모델이 해당 형식과 일치하는지 확인해야 합니다.

여러 사용 사례에서 콘텐츠 레이아웃이 렌더링됩니다.

- Oracle Content Management 자산 관리 UI에서 사용될 경우, 사용자가 변경사항을 저장하기 전에 미리 볼 수 있도록 데이터가 "편집됨" 상태일 수 있습니다.
- Oracle Content Management 사이트의 콘텐츠 목록 또는 콘텐츠 항목에서 사용될 경우, 실행 중인 사이트에 대한 추가 정보로 데이터가 보강됩니다.
- Content SDK의 `contentClient.renderLayout()` 호출에서 사용될 경우, Content SDK 사용자는 원하는 데이터를 콘텐츠 레이아웃으로 직접 전달합니다.

성능상 단일 질의(모든 필요한 데이터를 반환할 수 있음)와 다중 질의(빠른 초기 질의로 가능한 빠른 아웃라인을 렌더링하고 이어서 후속 질의를 통해 영역을 채움) 생성 간에 일반적인 장단점이 있습니다. 어떤 모델을 선택할지는 데이터 및 사용 사례에 따라 다릅니다.

또한 콘텐츠 레이아웃으로 전달된 데이터는 데이터 검색 방법에 따라 형식이 다를 수 있습니다. 예를 들어, 확장 매개변수와 함께 콘텐츠 REST를 사용하는 경우 다른 콘텐츠 항목에 대한 필드 참조도 개별적으로 또는 그룹으로 반환될 수 있습니다. 참조된 콘텐츠 항목이 포함되지 않을 경우 추가 REST 호출을 수행해야 합니다.

모든 사례를 처리하려면 콘텐츠 레이아웃 개발자는 수신된 데이터 형식에 대해 유연하게 대처해야 합니다. 또한 필요한 경우 개발자는 추가 데이터를 인출하여 렌더링 템플릿에 필요한 형식으로 데이터를 강제 변환할 수 있습니다.

## 콘텐츠 레이아웃을 위한 데이터 구조 표준화

콘텐츠 레이아웃 개발자는 콘텐츠 레이아웃이 수신하는 데이터 구조를 표준화해야 합니다.

모든 데이터가 있으면 콘텐츠 레이아웃이 간단히 구성요소를 렌더링할 수 있습니다. 모든 데이터가 없으면 콘텐츠 레이아웃이 추가 질의를 실행해야 할 수도 있습니다. 모든 경우 콘텐츠 레이아웃은 특정 데이터 형식을 가정해서는 안되며, 대신 렌더링할 형식으로 데이터를 강제 변환해야 합니다.

필요한 모든 데이터가 있는지 확인해야 합니다. 데이터가 없으면 추가 질의를 실행해야 합니다. 다음 필드는 잠재적으로 데이터에서 누락될 수 있습니다.

- 참조된 필드의 "fields" 항목
- 큰 텍스트 필드

콘텐츠 레이아웃은 특정 콘텐츠 유형에 맞게 설계되기 때문에 콘텐츠 레이아웃 개발자는 필요한 필드 목록을 알고 있습니다. 각 필드에 대해 콘텐츠 레이아웃이 렌더링할 수 있도록 데이터를 인출해야 합니다. 두 가지 옵션이 있습니다. 하나는 누락된 데이터를 인출한 후 완전한 데이터로 렌더링하는 것이고, 다른 하나는 즉시 렌더링한 후 누락된 데이터를 인출하여 공간을 채우는 것입니다.

### 옵션 1: 누락된 데이터를 인출한 후 완전한 데이터로 렌더링

Promise를 생성하여 필요한 데이터를 검색한 후 모든 Promise가 반환될 때 렌더링을 계속합니다.

예를 들어, 다음 콘텐츠 유형과 해당 필드가 있습니다.

- starter-blog-author
  - 필드
    - \* starter-blog-author\_name - 텍스트 필드
    - \* starter-blog-author\_bio - 텍스트 필드
- starter-blog-post
  - 필드
    - \* starter-blog-post\_title - 텍스트 필드
    - \* starter-blog-post\_content - 큰 텍스트 필드
    - \* starter-blog-post\_author - starter-blog-author 항목에 대한 참조

콘텐츠 레이아웃은 다음 템플릿을 가지고 필요한 필드 값을 렌더링합니다.

```

{{#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}}

```

콘텐츠 레이아웃은 다음 질의를 통해 데이터를 호출할 수 있습니다.

- "expand"를 사용한 항목 질의 - 모든 제공된 데이터
  - /content/published/api/v1.1/items/{id}?expand=fields.starter-blog-post\_author&channelToken=8dd714be0096ffaf0f7eb08f4ce5630f
  - 템플릿의 모든 값을 성공적으로 채우는 데 필요한 데이터 형식입니다. 다른 질의 중 하나를 사용할 경우 데이터를 인출한 후 해당 형식으로 변환하는 추가 작업이 필요합니다.
  - "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": "..."
  }
}

```
- "expand"가 없는 항목 질의 - 누락된 참조 항목 필드 "**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 질의 - 누락된 큰 텍스트 필드 "**starter-blog-post\_content**", 누락된 참조 항목 필드 "**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"
  }
}
```

이 질의로 일관되게 렌더링할 수 있으려면 콘텐츠 레이아웃의 `render.js`에서 모든 참조된 필드가 확장되고 큰 텍스트 필드가 있는지 확인해야 합니다.

그렇지 않으면 다시 질의하여 데이터를 수정한 후 완전한 데이터로 렌더링해야 합니다.

샘플 `render()` 함수:

```
render: function (parentObj) {
  var self = this,
      template,
      contentClient = self.contentClient,
      content = self.contentItemData;

  var getRefItems = function (contentClient, ids) {
    // Calling getItem() with no "ids" returns all items.
    // If no items are requested, just return a resolved Promise.
    if (ids.length === 0) {
      return Promise.resolve({});
    } else {
      return contentClient.getItems({
        "ids": ids
      });
    }
  };

  var fetchIDs = [], // list of items to fetch
      referredFields = ['starter-blog-post_author'], // names of
reference fields
      largeTextFields = ['starter-blog-post_content'], // large text
fields in this asset
      fieldsData = content.fields;
  // See if we need to fetch any referenced fields
  referredFields.forEach(function (fieldName) {
    if(fieldsData[fieldName] && fieldsData[fieldName].fields) {
      // got data already, nothing else to do
    } else {
      // fetch this item
      fetchIDs.push(fieldsData[fieldName].id);
    }
  });

  // See if we need to fetch any large text fields
  for(var i = 0; i < largeTextFields.length; i++) {
    if(!fieldsData[largeTextFields[i]]) {
      // need to fetch this content item directly to get all the
large text fields

```

```

        fetchIDs.push(content.id);
        break;
    }
}
// now we have the IDs of all the content items we need to fetch, get
them all before continuing
getRefItems(contentClient, fetchIDs).then(function (referenceData) {
    var items = referenceData && referenceData.items || [];

    // add the data back in
    items.forEach(function (referencedItem){
        // check if it's the current item
        if(referencedItem.id === content.id) {
            // copy across the large text fields
            largeTextFields.forEach(function (fieldName) {
                fieldsData[fieldName] = referencedItem.fields[fieldName];
            });
        } else{
            // check for any referenced fields
            for (var i = 0; i < referedFields.length; i++) {
                if(referencedItem.id === fieldsData[referedFields[i]].id)
                {
                    // copy across the fields values
                    fieldsData[referedFields[i]].fields =
referencedItem.fields;
                    break;
                }
            }
        }
    });

    // now data is fixed up, we can continue as before
    try{
        // Mustache
        template = Mustache.render(templateHtml, content);

        if(template) {
            $(parentObj).append(template);
        }

    } catch (e) {
        console.error(e.stack);
    }
});
}

```

### 옵션 2: 즉시 렌더링한 후 누락된 데이터를 인출하여 공란 채우기

표시되지 않을 항목을 분리했다가 두번째 패스로 렌더링하면 성능을 향상시킬 수 있습니다. 두 개의 Mustache 템플릿이 필요합니다. 첫번째는 초기 렌더링을 수행하고, 데이터가 완료되면 두번째 렌더링을 사용하여 "구멍"으로 남겨둔 부분을 채웁니다.

이를 위해 "구멍"마다 별도의 템플릿을 사용하거나 모델이 실제 값이 아닌 템플릿 매크로를 반환하는 방법으로 다중 패스를 지원하도록 Mustache 템플릿을 설정해야 합니다. 어느 경우든

데이터가 검색될 때까지 이 구멍을 "숨겼다" 채우고 적절한 UI 애니메이션으로 표시해야 페이지가 너무 많이 "튀어오르는" 것을 피할 수 있습니다.

## 샘플 블로그 템플릿 생성

BlogTemplate 샘플은 콘텐츠 레이아웃 기능을 시연합니다.

Developer Cloud Service에서 템플릿을 생성하고, 템플릿에서 콘텐츠 레이아웃을 검사하고, 검색 기능을 테스트할 수 있습니다.

1. BlogTemplate 유형의 템플릿을 생성합니다.

```
cec create-template MyBlogTemplate -f BlogTemplate
```

BlogTemplate를 복사하여 새 템플릿 MyBlogTemplate가 생성됩니다.

2. <http://localhost:8085/>를 열고 **템플릿, MyBlogTemplate**를 차례로 누릅니다.
3. 블로그 게시물 목록이 나타납니다. 그 중 하나를 누릅니다. 세 가지 다른 콘텐츠 레이아웃으로 렌더링할 수 있는 세부정보 페이지로 이동합니다.
4. **이 작성자의 다른 기사**를 누릅니다. 검색 페이지로 이동합니다.
5. 작성자 이름을 누릅니다. 해당 콘텐츠 레이아웃에 트리거가 발생하고 오른쪽에 기사가 표시됩니다.
6. 홈 페이지로 다시 이동하여 검색을 테스트합니다.

## 콘텐츠 레이아웃 매핑을 템플릿에 추가

콘텐츠 레이아웃을 생성한 후 콘텐츠 레이아웃 매핑을 추가하여 Developer Cloud Service 프로젝트에서 로컬 템플릿에 추가할 수 있습니다.

cec add-contentlayout-mapping 명령을 사용하여 템플릿에 대한 콘텐츠 레이아웃 매핑을 추가합니다. 콘텐츠 레이아웃의 기반이 되는 콘텐츠 유형(-c)과 매핑 대상인 템플릿(-t)가 필요합니다. 예를 들어, 다음과 같습니다.

```
cec add-contentlayout-mapping Blog-Post-Detail-Layout -c Blog-Post -t
BlogTemplate
```

기본 콘텐츠 레이아웃 매핑은 데스크톱의 기본 스타일입니다. -s <layoutstyle> 옵션을 지정하여 Overview 또는 Details 등의 다른 레이아웃 스타일을 이름별로 지정할 수 있습니다.

```
cec add-contentlayout-mapping Blog-Post-Detail-Layout -c Blog-Post -t
BlogTemplate -s Details
```

-m 옵션을 사용하여 모바일용 매핑을 설정할 수도 있습니다.

```
cec add-contentlayout-mapping Blog-Post-Detail-Layout -c Blog-Post -t
BlogTemplate -m
```

## 로컬 테스트 하네스로 콘텐츠 레이아웃 테스트

콘텐츠 유형 및 콘텐츠 레이아웃 매핑을 템플릿에 추가한 후 로컬 테스트 하네스로 콘텐츠 레이아웃을 테스트할 수 있습니다.

[로컬 테스트 하네스로 테스트](#)를 참조하십시오.

## 로컬 테스트 하네스로 테스트

사용자정의 구성요소, 템플릿 및 콘텐츠 레이아웃을 Oracle Content Management로 임포트하기 전에 로컬 테스트 하네스에서 실행하십시오.

로컬 테스트 하네스를 시작하려면 다음과 같이 하십시오.

1. 터미널 창에 `cd cec`를 입력합니다.
2. `cec develop &` 또는 `cec develop --server <server-name> &`를 입력합니다.
3. `http://localhost:8085`에서 브라우저를 열어 로컬 테스트 하네스에서 실행 중인 구성요소, 템플릿 및 콘텐츠 레이아웃을 확인합니다.
4. 다음 디렉토리에서 구성요소, 템플릿, 테마 등을 찾을 수 있습니다.
  - `cec/src/main/components`
  - `cec/src/main/templates`
  - `cec/src/main/themes`

## 콘텐츠 레이아웃이 있는 템플릿을 Oracle Content Management로 임포트

콘텐츠 레이아웃을 개발 및 테스트한 후 Developer Cloud Service 프로젝트에서 콘텐츠 레이아웃이 포함된 템플릿을 익스포트하여 Oracle Content Management로 템플릿을 임포트할 수 있습니다.

[템플릿 익스포트](#)를 참조하십시오.



# 23

## 구성요소 개발

Oracle Content Management 사이트에서 사용할 구성요소를 개발할 수 있습니다.

- [구성요소 정보](#)
- [구성요소 개발 정보](#)
- [구성요소 생성](#)
- [Developer Cloud Service로 사용자정의 구성요소 개발](#)
- [다국어 사이트를 위한 번역 가능한 구성요소 개발](#)
- [설정 패널이 있는 H1 구성요소 빌드](#)
- [HTML 템플릿을 사용하여 간단한 구성요소 생성](#)
- [로컬 구성요소와 원격 구성요소 비교](#)
- [구성요소 설정 렌더링](#)
- [로컬 구성요소 구현](#)
- [구성요소에 대한 스타일 클래스](#)
- [내장 구성요소 스타일링 방법](#)
- [구성요소 속성 설정](#)
- [인라인 프레임으로 렌더링된 구성요소](#)
- [인라인 프레임으로 렌더링된 구성요소에 대한 인스턴스 ID 및 구조 정보](#)
- [원격 구성요소에 대한 보안](#)
- [원격 구성요소 등록](#)
- [구성요소 삭제](#)
- [Sites SDK](#)

## 구성요소 정보

구성요소는 사이트의 페이지에 추가할 수 있는 특정 유형의 콘텐츠입니다. Oracle Content Management 구성요소에는 단락, 제목, 이미지, 구분선과 같은 항목이 포함됩니다.

Oracle Content Management는 다음 유형의 구성요소를 지원합니다.

- 로컬 구성요소: Oracle Content Management에 파일이 저장됩니다.
- 원격 구성요소: 원격 서버에 파일이 저장됩니다.

로컬 구성요소는 페이지 내에 직접 렌더링하거나 페이지의 인라인 프레임으로 렌더링하도록 설정할 수 있습니다. 원격 구성요소는 항상 인라인 프레임으로 렌더링됩니다.

Oracle Content Management는 각 템플릿(테마 및 사이트도 포함)와 함께 기본 구성요소 집합을 제공합니다. 새 구성요소를 생성하고 이 구성요소 내에 기본 구성요소를 사용할 수도 있습니다. 사이트에 구성요소를 추가했다면 페이지 콘텐츠, 글꼴과 글꼴 크기, 이미지 프레임과 배치 및 기타

스타일을 지정하여 요구사항에 맞게 구성요소의 속성 설정을 편집할 수 있습니다. 편집 가능한 설정은 구성요소 유형에 따라 다릅니다.

해당 구성요소는 Oracle Content Management와 함께 제공됩니다.

이름	유형	ID
제목	scs-title	scs-title
단락	scs-paragraph	scs-paragraph
텍스트	scs-text	scs-text
이미지	scs-image	scs-image
갤러리	scs-gallery	scs-gallery
갤러리 그리드	scs-gallerygrid	scs-gallerygrid
문서	scs-document	scs-document
단추	scs-button	scs-button
맵	scs-map	scs-map
구분선	scs-divider	scs-divider
스페이스	scs-spacer	scs-spacer
YouTube	scs-youtube	scs-youtube
소셜 표시줄	scs-socialbar	scs-socialbar
기사(사용자정의 구성요소)	scs-component	scs-comp-article
헤드라인(사용자정의 구성요소)	scs-component	scs-comp-headline
폴더 목록	scs-app	폴더 목록
파일 목록	scs-app	파일 목록
문서 관리자	scs-app	문서 관리자
Facebook 좋아요	scs-app	Facebook 좋아요
Twitter 팔로우	scs-app	Twitter 팔로우
Twitter 공유	scs-app	Twitter 공유
Facebook 추천	scs-app	Facebook 추천
이미지 및 텍스트(사용자정의 구성요소)	scs-component	scs-comp-image-text

*Building Sites with Oracle Content Management*의 페이지 콘텐츠 배열을 참조하십시오.

## 구성요소 개발 정보

고유의 사용자정의 구성요소를 개발하면 원하는 페이지 기술을 사용하여 사이트 페이지 안에 포함할 수 있는 복합 요소를 개발할 수 있습니다. 사실상 Oracle Content Management와 함께 제공된 구성요소 목록을 확장할 수 있습니다.

모든 등록된 구성요소 항목을 포함하는 Oracle Content Management의 폴더인 구성요소 카탈로그에 모든 등록된 구성요소 항목이 저장됩니다.

템플릿, 테마, 구성요소, 사이트 또는 사이트 페이지에 다음 이름을 사용하지 마십시오: authsite, content, pages, scs-template\_\*, \_comps, \_components, \_compsdelivery, \_idcservice, \_sitescloud, \_sitesclouddelivery, \_themes, \_themesdelivery. 사이트

페이지에는 다음 이름을 사용할 수 있지만 템플릿, 테마, 구성요소 또는 사이트에는 사용하지 마십시오: documents, sites.

### 구성요소 유형

구성요소 카탈로그는 다음 유형의 구성요소를 지원합니다.

- 로컬 구성요소
- 인라인 프레임으로 렌더링된 로컬 구성요소
- 원격 구성요소

구성요소 유형은 구성요소 폴더의 확장 속성 "xScsAppType"으로 저장됩니다. 적합한 값이 따릅니다.

유형	설명
로컬 구성요소	모든 종속성이 로컬로 예상됩니다.
인라인 프레임으로 렌더링된 로컬 구성요소	구성요소가 Oracle Content Management와 동일한 도메인에서 서비스됩니다.
원격 구성요소	구성요소가 원격 위치에서 서비스됩니다.

### 구성요소 파일 구조

각 유형의 구성요소는 구현 방법에 따라 구성요소 카탈로그에 생성될 때 다른 파일 집합을 갖습니다.

#### 로컬 구성요소:

```
/Components/component-name
  appinfo.json
  _folder_icon.jpg
  assets
    settings.html
    render.js
```

#### 인라인 프레임을 사용하는 로컬 구성요소:

```
/Components/component-name
  appinfo.json
  _folder_icon.jpg
  assets
    settings.html
    render.js
  js
    sites.min.js
    knockout.min.js
    jquery.min.js
```

#### 원격 구성요소:

```
/Components/component-name
  appinfo.json
```

```
_folder_icon.jpg
keys.json
```

## 폴더 메타데이터

구성요소 등록 데이터는 폴더 메타데이터에 저장됩니다. 다음 속성은 구성요소와 해당 유형을 고유하게 식별하는 데 사용됩니다.

속성	설명
app name	구성요소 이름을 지정하는 폴더의 이름입니다.
app description	폴더의 설명입니다.
app guid	모든 구성요소는 GUID와 연관되며 확장 속성 xScsItemGUID로 저장됩니다. GUID는 구성요소를 생성할 때 서버에서 생성됩니다.
app type	속성은 구성요소 유형을 지정합니다. 확장 속성 xScsApType으로 저장됩니다.

### 주:

확장 속성 xScsAppIconUrl로 저장된 imageUrl 속성은 더 이상 사용되지 않습니다.

## appinfo.json 파일

각 구성요소 유형에 대한 appinfo.json 등록 파일에는 폴더 메타데이터로 제공되지 않는 데이터만 포함됩니다. 폴더 메타데이터에 정의된 구성요소 속성은 appinfo.json 파일과 중복되지 않습니다.

### 로컬 구성요소:

```
{
  "settingsData": {
    "settingsHeight": 80,
    "settingsRenderOption": "dialog",
    "settingsWidth": 300,
    "componentLayouts": [],
    "triggers": [],
    "actions": []
  },
  "initialData": {
    "customSettingsData": []
  }
}
```

### 인라인 프레임으로 렌더링된 로컬 구성요소:

```
{
  "endpoints": {
    "settings": {
      "height": "300",
```

```

        "width": "400"
      }
    }
    "initialData": {
      "customSettingsData": {}
    }
  }
}

```

#### 원격 구성요소:

```

{
  "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": {}
  }
}

```

로컬 구성요소에는 등록된 "url" 값이 없습니다. 콘텐츠를 편집할 수 있는 특정 파일을 사용하지만 위치나 이름은 변경할 수 없습니다.

- 로컬 구성요소는 assets/render.js 및 assets/settings.html 파일을 사용합니다.
- 인라인 프레임으로 렌더링된 로컬 구성요소는 assets/render.html 및 assets/settings.html 파일을 사용합니다.
- 원격 구성요소는 지정된 "url" 값을 사용합니다.

## 구성요소 생성

Oracle Content Management에서 사용할 사용자정의된 구성요소를 생성할 수 있습니다.

Oracle Content Management는 웹 사이트 구축에 사용할 많은 미리 정의된 구성요소를 제공하지만, 개발자는 동일한 기능에 액세스할 수 있는 사용자정의 구성요소를 생성할 수 있습니다. 이러한 구성요소 유형에는 로컬, 인라인 프레임으로 렌더링된 로컬 및 인라인 프레임으로 렌더링된 원격이 포함됩니다.

#### 로컬 구성요소

로컬 구성요소를 생성할 때 고유의 구성요소 생성을 위한 기초로 사용할 수 있는 전기능의 샘플 구성요소가 제공됩니다. 구성요소의 Knockout 버전을 생성하려면 기본값을 선택하고, Mustache 기반 버전을 생성하려면 템플릿을 선택하십시오. *Building Sites with Oracle Content Management*의 로컬 구성요소 또는 레이아웃 생성을 참조하십시오.

### 인라인 프레임이 있는 로컬 구성요소

인라인 프레임이 있는 로컬 구성요소를 생성할 때 고유의 구성요소 생성을 위한 기초로 사용할 수 있는 전기능의 샘플 구성요소가 제공됩니다. 로컬 구성요소를 생성할 때와 동일한 지침을 사용하고 **Sandbox 적용** 옵션을 선택하여 로컬로 저장된 구성요소의 인라인 프레임 버전을 생성합니다. [인라인 프레임으로 렌더링된 구성요소](#)를 참조하십시오.

### 원격 구성요소

인라인 프레임을 사용하는 원격 구성요소를 생성할 때는 **Sandbox 적용** 옵션을 선택합니다.

생성된 파일을 원격 서버에 복사하고 원격 구성요소를 등록합니다. 게시된 사이트에 포함하기 전에 구성요소를 테스트합니다.

[인라인 프레임으로 렌더링된 구성요소](#) 및 [원격 구성요소 등록](#)을 참조하십시오.

## Developer Cloud Service로 사용자정의 구성요소 개발

Oracle Developer Cloud Service는 Oracle Content Management용 템플리트, 테마 및 사용자정의 구성요소 개발을 돕습니다.

Oracle Content Management와 Developer Cloud Service 통합은 템플리트 및 구성요소 개발 도구와 함께 템플리트를 제공합니다. 또한 시작용 샘플 단위 테스트를 제공합니다. 통합에는 템플리트와 구성요소 개발을 돕는 Git 저장소 및 도구뿐만 아니라 템플리트, 테마 및 사용자 구성요소의 빠르고 반복적인 개발을 위한 로컬 테스트 하네스가 포함됩니다.

Developer Cloud Service는 다음 작업을 도울 수 있습니다.

- 템플리트, 테마, 구성요소의 로컬 개발 및 테스트에 Oracle Content Management 인스턴스를 사용할 수 있도록 로컬 개발 환경을 설정합니다.
- 샘플 또는 스타터에서 템플리트와 구성요소를 생성하고 테스트 하네스에서 실행하고 탐색하며 Developer Cloud Service 환경에서 템플리트, 테마 및 구성요소를 개발합니다.
- 소스 관리 및 추가 개발을 위해 Oracle Content Management에서 생성된 템플리트나 구성요소를 Developer Cloud Service 환경으로 импорт합니다.
- Developer Cloud Service 환경에서 템플리트나 구성요소를 익스포트하고 웹 사이트에서 사용하도록 Oracle Content Management로 импорт합니다.
- 기존 구성요소 복사
- 단위 테스트 쓰기
- 구성요소 최적화
- Oracle Content Management에 구성요소 배치

다음 항목은 Oracle Content Management용 개발을 위해 Developer Cloud Service 통합을 사용하는 방법에 대해 설명합니다.

- [Oracle Content Management용 사용자정의 구성요소 개발](#)
- [성능 향상을 위해 구성요소 최적화\(축소\)](#)
- [연속 통합 작업 실행](#)

## Oracle Content Management용 사용자정의 구성요소 개발

Developer Cloud Service와 로컬 시스템을 사용하여 Oracle Content Management용 사용자정의 구성요소를 개발합니다.

다음 항목은 Developer Cloud Service를 사용한 Oracle Content Management 구성요소 개발 및 테스트 단계에 대해 설명합니다.

1. 로컬 시스템에서 OCE Toolkit 설정
2. Oracle Content Management용 Developer Cloud Service 콘솔에 사인인
3. Developer Cloud Service에서 프로젝트 생성
4. 새 Git 저장소에서 프로젝트 코드에 OCE Toolkit 추가
5. 사용자정의 구성요소 개발
6. 로컬 테스트 하네스로 테스트
7. 단위 테스트 쓰기 및 실행

### 사용자정의 구성요소 개발

cec 명령행 유틸리티를 사용하여 새 구성요소를 생성하고 로컬로 구성요소를 개발한 다음, Oracle Content Management로 구성요소를 익스포트합니다.

#### 구성요소 생성

사용자정의 구성요소를 생성하고 로컬로 개발하려면 다음 명령을 사용하십시오.

```
cec create-component <component-name> -f <source>
```

*source*에 대해 다음 값 중 하나를 선택합니다.

- 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

예제:

```
cec create-component MyLocalComponent1 -f local
```

구성요소는 Git 저장소의 `cec-components/src/main/components` 아래에 생성됩니다.

`src/main/components` 디렉토리는 Sample-To-Do 구성요소와 함께 시드됩니다. 직접 생성한 모든 구성요소는 이 디렉토리에 있습니다.

### 구성요소 복사

`cec copy-component` 명령을 사용하여 Developer Cloud Service에 구성요소를 복사할 수 있습니다.

```
cec copy-component <source> [<destination>]
```

이 명령은 `<source>`라는 기존 구성요소를 `<destination>`으로 복사합니다.

### Oracle Content Management에 구성요소 익스포트 또는 배치

로컬 서버에서 구성요소를 개발하고 테스트한 후 다음 명령을 사용하여 구성요소를 익스포트할 수 있습니다. 구성요소 zip 파일이 생성됩니다. 이 구성요소 zip을 Oracle Content Management로 수동으로 임포트할 수 있습니다.

```
cec export-component <component name>
```

또는 다음 명령을 사용하여 Developer Cloud Service에서 Oracle Content Management로 직접 구성요소를 배치할 수 있습니다.

```
cec deploy <component name>
```

`cec deployAll` 명령은 `src/main/components`의 모든 구성요소를 배치합니다.

### Developer Cloud Service로 구성요소 임포트

Oracle Content Management 서버에서 생성된 구성요소 ZIP 파일이 있는 경우 추가 개발을 위해 해당 파일을 Developer Cloud Service로 임포트할 수 있습니다. 다음 명령을 사용하십시오.

```
cec import-component <location of the component zip file>
```

#### ! 중요사항:

구성요소의 소스 코드는 `src/main/components`에 있습니다. `src/main/components` 외부 파일은 Oracle Content Management 로컬 서버 작동에 필요하므로 수정하면 안됩니다.



## 단위 테스트 쓰기 및 실행

샘플 단위 테스트로 시작해서 단위 테스트를 작성하고 사용자정의 Oracle Content Management 구성요소에 대해 실행합니다.

### 샘플 단위 테스트로 시작

Oracle Content Management용 Developer Cloud Service 샘플에는 단위 테스트 작성을 돕는 다음 파일이 포함됩니다.

- **src/test/unit:** Sample-To-Do 구성요소에 대한 단위 테스트를 포함합니다. RequireJS 모듈을 포함하는 구성요소 JavaScript 코드에 대한 단위 테스트 작성 샘플로 활용됩니다. JavaScript 단위 테스트를 위한 Mocha 및 Chai 프레임워크가 사용됩니다.
- **index.html:** 단위 테스트를 실행합니다. Mocha, Chai 및 기본 테스트 파일 test-main.js를 로드합니다.
- **test-main.js:** 단위 테스트 모듈 Sample-To-Do-Test를 로드하고 Mocha 단위 테스트를 실행합니다.
- **Sample-To-Do-Test.js:**
  - 실제 테스트를 정의합니다. RequireJS를 사용하여 구성요소 소스 코드 components/Sample-To-Do/assets/render.js를 로드합니다.
  - add(), delete(), title(), placeholder() 메소드를 테스트합니다.

### 구성요소에 대한 단위 테스트 쓰기

사용자정의 구성요소에 대한 각 단위 테스트를 작성하려면 다음 단계를 따르십시오.

1. Sample-To-Do-Test.js와 비슷한 단위 테스트를 작성합니다.
2. 작성한 테스트를 test-main.js에 로드합니다.

### 로컬로 단위 테스트 실행

단위 테스트를 로컬로 실행하려면 다음 단계를 따르십시오.

1. Git 저장소를 로컬로 복제합니다.
2. 로컬 시스템의 터미널 창에서 npm install을 입력합니다.
3. npm start를 입력합니다.
4. npm test를 입력하거나 브라우저에서 http://localhost:8085/unit/를 엽니다.

## 성능 향상을 위해 구성요소 최적화(축소)

Developer Cloud Service로 JavaScript 코드를 축소하여 구성요소의 성능을 향상하고 다운로드 크기를 줄입니다.

### 구성요소 최적화

구성요소 소스 코드 최적화(선택사항)의 예로 Sample-Text-With-Image 구성요소가 제공됩니다. RequireJS 최적기를 사용하여 소스 코드를 축소하고 결합할 수 있습니다. 구성요소를 축소하면 JavaScript 코드에 불필요한 공간이 제거되어 성능을 향상하고 다운로드 크기를 줄일 수 있습니다. 최적화에는 다음 작업이 관여합니다.

- JavaScript 축소

- CSS 압축
- JavaScript, HTML 및 CSS를 단일 파일 `render.js`로 결합

이 최적화는 다운로드 크기를 줄이고 성능을 향상시킵니다.

#### 구성요소에 대해 최적화 사용으로 설정

Sample-Text-With-Image 구성요소에서 최적화는 `render.js`에 종속성으로 표시된 모든 파일(`template.html`, `data-defaults.js`, `design.css`)을 축소하고 단일 파일로 `render.js`에 결합합니다. 이미 최적화를 지원하는 다른 구성요소는 다음과 같습니다.

- Sample-Folder-List
- Sample-File-List
- Sample-Documents-Manager
- Sample-Process-Start-Form
- Sample-Process-Task-List
- Sample-Process-Task-Details

구성요소에 대해 최적화를 사용으로 설정하려면 다음 단계를 따르십시오(필수).

1. Sample-Text-With-Image 구성요소의 `gulpfile.js`를 해당 구성요소로 복사합니다.
2. `render.js` 종속성을 위해 `gulpfile.js`의 모듈 이름을 수정합니다. `gulpfile.js`의 주석을 참조합니다.

#### 주:

기존 구성요소에서 `assets` 아래에 `css` 폴더가 있는 경우 최적화 중에 오류가 발생하지 않도록 `css` 폴더 이름을 `styles`로 바꾸십시오.

## 연속 통합 작업 실행

연속 통합 작업을 실행하여 사이트를 최신으로 유지할 수 있습니다.

Developer Cloud Service에서 "Getting Started with CEC Custom Components Development" 위키 항목을 참조하십시오. 연속 통합을 위한 빌드 작업을 구성할 수 있습니다.

## 다국어 사이트를 위한 번역 가능한 구성요소 개발

사용자정의 구성요소의 개발자는 사용자정의 구성요소 내에서 다국어(MLS) 사이트용 페이지 번역에 참여시킬 문자열을 지정할 수 있습니다.

이를 위해 `customSettingsData` 객체에 데이터를 저장할 때 최상위 레벨에 `nls` 속성을 도입해야 합니다.

예를 들어, 다음과 같습니다.

```
SitesSDK.setProperty('customSettingsData', {
  'nls': {
    linkText: 'More...'
  }
});
```

```
});
```

번역 작업이 생성되면 Oracle Content Management는 customSettingsData 객체의 최상위 레벨 속성을 확인하고 페이지의 각 사용자정의 구성요소 인스턴스에 대해 전체 nls 객체를 익스포트합니다. 번역자가 이 값을 번역한 다음, 사이트로 다시 임포트할 수 있습니다.

사이트 번역이 임포트된 후에 번역된 로케일을 위한 올바른 버전의 nls 객체가 customSettingsData 객체에 반환됩니다.

예를 들어, 사이트를 프랑스어로 번역하고 프랑스어 로케일로 페이지를 렌더링한 경우 사용자정의 구성요소로 전달된 customSettingsData 객체에서 nls 객체의 값이 다음과 같이 업데이트됩니다.

```
{
  'nls': {
    'linkText': 'Plus...'
  }
},
```

customSettingsData에서 nls 객체의 형식은 이름/값 쌍으로 제한되어야 합니다. 이것은 번역에 도움이 되며 번역된 로케일로 페이지를 렌더링할 때 번역된 값을 사이트의 기본 값에 올바르게 적용할 수 있습니다.

## 설정 패널이 있는 H1 구성요소 빌드

간단한 HTML 템플릿과 CSS를 사용한 최소 Oracle Content Management 구성요소를 생성할 수 있습니다. 이 H1 구성요소에는 간단한 설정 패널과 design.json에 테마 항목이 있어서 다른 Oracle Content Management 사용자가 편집기에서 구성요소를 사용할 때 세 가지 내장 스타일 중에서 선택할 수 있습니다.

새 구성요소를 생성하면 기본적으로 작동할 시드 파일 집합을 얻습니다. 시드 파일은 제품 내의 구성요소 기능을 대부분 처리합니다. 시드 코드를 변경하여 고유의 구성요소를 생성하면 시드 코드의 작은 부분 집합만으로 최종 결과를 성취할 수 있습니다.

다음 5단계로 설정 패널이 있는 H1 구성요소를 빌드할 수 있습니다.

1. 새 로컬 구성요소 생성
2. 기본 H1 구성요소 빌드
3. 구성요소에 대한 CSS 추가
4. 설정 패널을 추가하여 머리글 텍스트 변경
5. 다른 사람이 H1 구성요소 스타일을 선택하도록 테마 업데이트

### 새 로컬 구성요소 생성

Oracle Content Management에서 즉시 페이지 위에 놓을 수 있는 로컬 구성요소를 생성합니다. 새 구성요소 생성을 위한 시작점입니다.

새 로컬 구성요소를 생성하려면 다음과 같이 하십시오.

1. 측면 탐색에서 **개발자**를 누르고 **모든 구성요소 보기**를 누릅니다.
2. **생성 > 로컬 구성요소 생성**을 선택합니다.

- 이름(예: H1\_Component)과 선택적 설명을 입력합니다.
- 생성을 눌러 새 구성요소를 생성합니다.

구성요소를 성공적으로 생성했으므로 이제 구성요소 페이지의 구성요소 목록과 생성한 사이트에 대한 **추가 > 사용자정의** 구성요소 팔레트에서 볼 수 있습니다. 다음 단계에 따라 구성요소 생성을 검증하십시오(체크포인트 1).

- 시드된 템플릿을 사용하여 새 사이트를 생성합니다. 예를 들어, StarterTemplate 템플릿을 사용하여 ComponentTest라는 사이트를 생성합니다.
- 편집** 옵션을 선택하고 사이트의 업데이트를 생성하여 편집기에서 엽니다.
- 직접 생성한 사이트 내의 페이지를 편집합니다.
- 왼쪽 막대에서 **추가 단추(+)**를 누르고 사용자정의 구성요소 목록에 대한 **사용자정의**를 선택합니다.
- 사용자정의 구성요소 팔레트에서 **H1\_Component**를 선택하고 페이지 위에 놓습니다. 이제 생성한 로컬 구성요소에 대한 기본 렌더링이 표시되어야 합니다.
- 구성요소의 컨텍스트 메뉴를 선택합니다.
- 드롭다운 메뉴에서 **설정**을 선택합니다.  
설정을 변경하여 시드된 구성요소 렌더링이 어떻게 변경되는지 확인할 수 있습니다. 시드 파일을 수정하여 새 사용자정의 구성요소를 생성할 수 있습니다.

## 기본 H1 구성요소 빌드

시드 파일에서 대부분의 내용을 제거하여 H1 구성요소를 생성할 수 있습니다. viewModel을 생성할 때 시드된 머릿글 텍스트가 표시됩니다. 나중에 구성요소의 설정과 스타일을 제공할 수 있습니다.

로컬 구성요소의 구조를 검토하려면 다음과 같이 하십시오.

- Oracle Content Management 데스크톱 동기화 앱을 사용하여 구성요소를 찾아 파일 시스템과 동기화합니다.
  - 최신 버전의 데스크톱 동기화 앱에서 **동기화 시작** 또는 **동기화할 폴더 선택** 옵션을 선택합니다.
  - 데스크톱 동기화 앱이 없는 경우 Oracle Content Management의 **구성요소** 탭에서 구성요소를 선택하고 드릴다운하여 파일을 볼 수 있습니다.
- 구성요소 아래의 파일을 나열하면 다음 파일을 볼 수 있습니다.
  - assets 폴더의 구성요소 파일:
    - render.js
    - settings.html
  - appinfo.json: 구성요소 설명이 있는 JSON 파일.  
[구성요소 개발 정보](#)를 참조하십시오.
  - folder\_icon.jpg: 구성요소 카탈로그에 표시되는 아이콘.

H1 구성요소를 빌드하려면 다음과 같이 하십시오.

1. appinfo.json 파일을 열고 내용을 다음 행으로 바꿉니다.

```
{
  "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. 즐겨찾는 텍스트 편집기에서 assets 폴더의 render.js 파일을 엽니다.
3. render.js 내용을 다음 행으로 변경합니다.

```
/* 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 recieve SETTINGS_UPDATED
    events

    SitesSDK.subscribe(SitesSDK.MESSAGE_TYPES.SETTINGS_UPDATED, $.proxy(this.updateSettings
```

```

//
// 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. assets 폴더에서 구성요소의 간단한 HTML 템플릿으로 사용될 새 파일 render.html을 생성합니다.
5. render.html 파일에서 다음 내용을 사용합니다.

```

<h1 data-bind="text: headingText()">
</h1>

```

구성요소 assets 폴더에 이제 세 개의 파일이 있습니다.

- render.html
- render.js
- settings.html

새 H1 구성요소를 페이지에 추가합니다(체크포인트 2).

## 구성요소에 대한 CSS 추가

구성요소에 대한 기본 스타일을 제공할 CSS를 추가할 수 있습니다.

CSS를 추가하려면 다음과 같이 하십시오.

1. 다음 내용과 함께 구성요소의 assets 폴더에 design.css 파일을 추가합니다.

```

.h1-component-default-style .scs-component-content {
font-family: "Helvetica Neue", "Helvetica", "Arial", sans-serif;
font-size: 24px;
color:red;
font-weight: normal; }

```

2. appinfo.json에 추가하여 구성요소 스타일링에 사용할 스타일 클래스 접두어를 선언합니다. styleClassName으로 h1-component를 추가하면 구성요소를 페이지 위에 놓을 때 기본

스타일은 `h1-component-default-style`이 됩니다. `appinfo.json`의 새로운 내용은 다음과 같습니다.

```
{
  "id": "h1-component-id",

  "settingsData": {
    "settingsHeight": 90,
    "settingsWidth": 300,
    "settingsRenderOption": "inline",
    "componentLayouts": [ ],
    "triggers": [ ],
    "actions": [ ]
  },
  "initialData": {
    "componentId": "h1-component-id",
    "styleClassName": "h1-component",
    "customSettingsData": {
      "headingText": "Heading 1"
    },
    "nestedComponents": [ ]
  }
}
```

3. `render.js`를 업데이트하여 CSS 파일을 로드합니다. 2행(`define(['knockout', 'jquery', 'text!./render.html'], function(ko, $, template) {})`을 다음 행으로 바꾸십시오.

```
define(['knockout', 'jquery', 'text!./render.html', 'css!./
design.css'], function(ko, $, template, css) {
```

#### 주:

`appinfo.json` 파일을 변경했으므로 브라우저를 새로고침하고 페이지에 다시 구성요소를 추가하여 변경사항을 적용해야 합니다.

이제 구성요소가 `appinfo.json` 파일에서 기본 데이터를 선택하는지 확인합니다.

## 설정 패널을 추가하여 머리글 텍스트 변경

`settings.html` 파일을 업데이트하여 H1 구성요소의 텍스트 설정에 사용할 수 있는 설정 패널을 제공합니다.

설정 패널을 추가하여 머리글 텍스트를 변경하려면 다음과 같이 하십시오.

1. 다음 내용을 포함하도록 `settings.html` 파일을 업데이트합니다.

```
<!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);
        }
      });
    };
  </script>
</body>
</html>
```

```

    }
  }, self);
};

// apply the bindings
ko.applyBindings(new SettingsViewModel());
</script>
</body>

```

2. 사이트 작성기에서 구성요소를 선택하고 **설정**을 누릅니다.

설정 패널은 크기가 작고, 구성요소 설정 패널의 상단에 포함되도록 설정됩니다. "머리글 텍스트"라는 섹션을 찾습니다.

머리글 텍스트를 변경할 수 있는지 확인합니다(체크포인트 4).

설정 패널에서 머리글 텍스트를 변경하면 구성요소가 업데이트되어 새 텍스트를 표시합니다.

## 다른 사람이 H1 구성요소 스타일을 선택하도록 테마 업데이트

구성요소의 스타일을 테마에 등록하면 다른 사용자가 설정 패널 또는 스타일 탭에서 구성요소에 제공한 스타일 간을 전환할 수 있습니다.

다른 사용자가 구성요소 스타일을 선택하도록 테마를 업데이트하려면 다음과 같이 하십시오.

1. 구성요소 design.css 파일에 더 많은 스타일을 추가합니다. appinfo.json에 정의된 대로 구성요소의 등록된 styleClassName이 각 스타일 앞에 접두어로 붙습니다. 이 구성요소의 경우 접두어는 h1-component입니다.

두 개의 스타일 h1-component-gothic-style 및 h1-component-courier-style이 추가되었습니다.

이제 design.css의 새로운 내용은 다음과 같습니다.

```

.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. 테마의 design.json 파일에 스타일을 등록합니다. 사이트에 사용 중인 테마에서 이 파일을 찾을 수 있습니다. designs 폴더의 테마 파일로 드릴다운하고 이어서 defaults 폴더로 드릴다운하여 구성요소의 섹션을 design.json에 추가합니다.

다음의 굵은체 텍스트는 추가할 항목의 예입니다.

```

"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": [

```

추가된 design.json 코드 조각의 이름("Plain", "Courier", "Gothic")이 스타일 탭의 구성요소 설정 패널에 아래와 같이 나타납니다. 선택 시 구성요소에 해당 스타일("h1-component-default-style", "h1-component-courier-style", "h1-component-gothic-style")이 각각 적용됩니다.

## HTML 템플리트를 사용하여 간단한 구성요소 생성

Mustache JS 템플리트와 함께 HTML 및 CSS를 사용하는 Oracle Content Management 구성요소를 생성하여 설정 패널에 입력한 데이터를 기반으로 콘텐츠를 렌더링할 수 있습니다.

제공한 설정 패널에서 HTML 템플리트에 렌더링할 데이터를 편집할 수 있습니다.

이 절차의 예제는 Mustache JS 템플리트를 사용합니다.

다음 2단계로 HTML 템플리트를 사용하여 간단한 구성요소를 생성할 수 있습니다.

1. HTML 구성요소 생성
2. HTML 구성요소에 사용자 제공 데이터 추가

## HTML 구성요소 생성

Oracle Content Management 구성요소 카탈로그에 새 로컬 구성요소를 생성한 다음, assets 폴더에 파일을 추가하고 변경하여 HTML 구성요소를 빌드할 수 있습니다.

HTML 구성요소를 생성하고 빌드하려면 다음과 같이 하십시오.

1. 구성요소를 생성하고 HTML Component라는 이름을 지정합니다.  
새 로컬 구성요소 생성을 참조하십시오.
2. HTML 구성요소의 assets 폴더에 mustache.min.js 파일을 추가합니다.
3. 구성요소 본문의 HTML을 포함하는 render.html이라는 새 파일을 assets 폴더에 생성합니다. 이 예제의 경우 다음 내용으로 render.html 파일을 생성합니다.

```
<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. 구성요소의 CSS인 design.css라는 새 파일을 assets 폴더에 생성합니다. 이 예제의 경우 다음 행을 design.css 파일의 내용으로 추가합니다.

```
.wrapper {
  text-align: center;
}
.box {
  display: inline-block;
  position: relative;
  width: 200px;
  height: 200px;
  padding: 0px 10px 10px 10px;
```

```

background: transparent;
border-width:1px;
border-style:solid;
border-radius: 5px;
border-color:#CCCCCC;
z-index: 0;
margin: 2px 2px 2px 2px;
transition: all .15s ease-in-out;
}
.box:hover {
background: #9CC;
z-index: 100;
transform: scale(1.2,1.2);
box-shadow: 0 5px 10px 0 rgba(0,0,0,.2);
}
.title {
color:red;
}
.text {
color:#555555;
}

```

5. assets 폴더의 render.js 파일을 열고 내용을 다음으로 변경합니다. 앞의 단계에서 어떤 HTML 및 CSS를 사용했든 관계없이, 다음 render.js 파일은 HTML 및 CSS를 페이지에 렌더링합니다.

```

/* globals define */
define(['jquery', './mustache.min', 'text!./render.html', 'css!./
design.css'], function($, Mustache, template, css) {
  'use strict';

  // -----
  // Create a Mustache-based component implementation
  // -----
  var SampleComponentImpl = function(args) {
    this.SitesSDK = args.SitesSDK;

    // Initialize the custom component
    this.createTemplate(args);
    this.setupCallbacks();
  };
  // create the template based on the initial values
  SampleComponentImpl.prototype.createTemplate = function(args) {
    // create a unique ID for the div to add, this will be passed to the
callback
    this.contentId = args.id + '_content_' + args.viewMode;
    // create a hidden custom component template that can be added to the
DOM
    this.template = '<div id="' + this.contentid + '">' +
template +
'</div>';
  };
  SampleComponentImpl.prototype.updateSettings = function(settings) {
    if (settings.property === 'customSettingsData') {
      this.update(settings.value);
    }
  };
}

```

```

    }
  };
  SampleComponentImpl.prototype.update = function(data) {
    this.data = data;
    this.container.html(Mustache.to_html(this.template, this.data));
  };
  //
  // SDK Callbacks
  // setup the callbacks expected by the SDK API
  //
  SampleComponentImpl.prototype.setupCallbacks = function() {
    //
    // callback - render: add the component into the page
    //
    this.render = $.proxy(function(container) {
      this.container = $(container);

      this.SitesSDK.getProperty('customSettingsData', $.proxy(this.update, this));
    }, this);
    //
    // callback - SETTINGS_UPDATED: retrieve new custom data and re-
    render the component
    //

    this.SitesSDK.subscribe(this.SitesSDK.MESSAGE_TYPES.SETTINGS_UPDATED, $.proxy(this.update,
    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;
});

```

HTML 구성요소가 빌드되었는지 확인합니다(체크포인트 1).

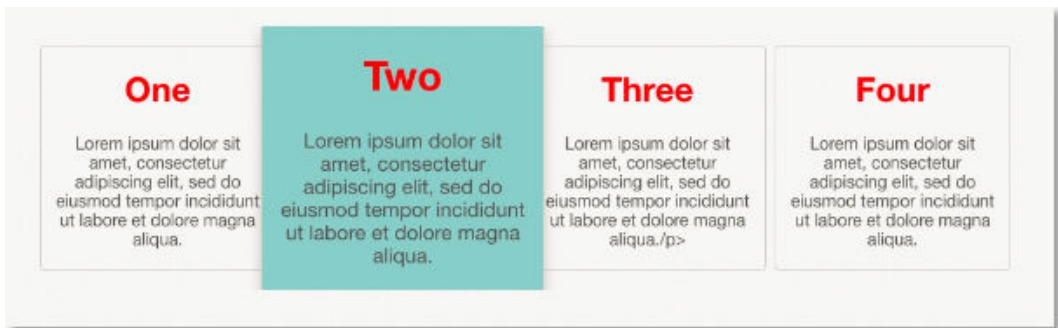
1. 구성요소의 `assets` 폴더에 다음 5개 파일이 있는지 확인합니다.
  - `design.css`
  - `mustache.min.js`
  - `render.html`
  - `render.js`

- settings.html
2. 테스트 사이트의 페이지에 새 HTML 구성요소를 추가합니다. 편집기에서 편집 및 미리보기 모드로 다음과 같이 페이지에 구성요소가 표시되어야 합니다.

편집 모드



미리보기 모드



## HTML 구성요소에 사용자 제공 데이터 추가

HTML 구성요소에 정의한 모든 데이터 요소에 대한 필드를 제공하는 **사용자정의 설정 패널**을 추가할 수 있습니다. 사용자는 **사용자정의 설정 패널**에서 페이지에 구성요소 인스턴스의 값을 채울 수 있습니다.

HTML 구성요소에 사용자 제공 데이터를 추가하려면 다음과 같이 하십시오.

1. render.html 파일에서 HTML 템플릿을 업데이트하여 원하는 데이터 요소를 포함합니다. 다음 예제는 Mustache JS 템플릿 언어를 사용하므로 다음과 같이 `{{ 및 }}` 구문을 사용하여 데이터 요소를 추가해야 합니다.

```
<ul class="wrapper">
  <li class="box">
    <h1 class="title">{{title1}}</h1>
    <p class="text">{{text1}}</p>
  </li>
  <li class="box">
    <h1 class="title">{{title2}}</h1>
    <p class="text">{{text2}}</p>
  </li>
</ul>
```

```

<li 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. HTML 구성요소에서 appinfo.json 파일의 내용을 변경하여 다음 데이터 필드에 대한 기본값을 제공합니다.

```

{
  "id": "html-component-id",

  "settingsData": {
    "settingsHeight": 600,
    "settingsWidth": 300,
    "settingsRenderOption": "dialog",
    "componentLayouts": [],
    "triggers": [],
    "actions": []
  },
  "initialData": {
    "styleClassName": "html-component",
    "customSettingsData": {
      "title1": "One",
      "title2": "Two",
      "title3": "Three",
      "title4": "Four",
      "text1": "Lorem ipsum dolor sit amet, consectetur adipiscing
elit, sed do eiusmod tempor incididunt ut labore et dolore magna
aliqua.",
      "text2": "Lorem ipsum dolor sit amet, consectetur adipiscing
elit, sed do eiusmod tempor incididunt ut labore et dolore magna
aliqua.",
      "text3": "Lorem ipsum dolor sit amet, consectetur adipiscing
elit, sed do eiusmod tempor incididunt ut labore et dolore magna
aliqua.",
      "text4": "Lorem ipsum dolor sit amet, consectetur adipiscing
elit, sed do eiusmod tempor incididunt ut labore et dolore magna
aliqua."
    },
    "nestedComponents": []
  }
}

```

3. HTML 템플릿 render.html 파일에서 데이터 값을 찾는 사용자정의 설정 패널을 추가하여 사용자가 편집기에서 변경할 설정을 생성합니다. settings.html 파일의 내용을 다음 행으로 변경합니다.

```

<!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>

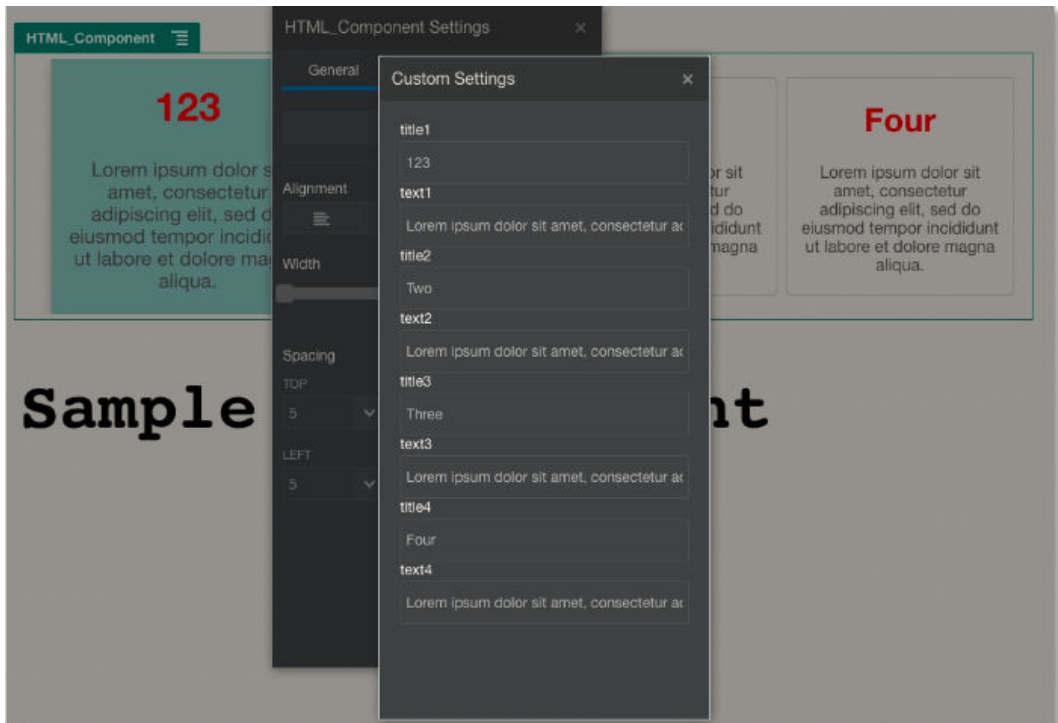
```

**주:**

appinfo.json 파일을 변경했으므로 브라우저를 새로고침하고 페이지에 다시 구성요소를 추가하여 변경사항을 적용해야 합니다.

이제 구성요소가 appinfo.json 파일에서 기본 데이터를 선택하고 **사용자정의 설정** 패널에서 값을 변경할 수 있는지 확인합니다(체크포인트 2).

1. appinfo.json 파일을 변경한 후에 반드시 브라우저를 새로고침하고 페이지에 다시 구성요소를 추가하여 변경사항을 적용하십시오.
2. HTML 구성요소에서 **사용자정의 설정** 패널을 열고 데이터를 입력합니다.



3. Oracle Content Management 구성요소가 Mustache JS 템플릿과 함께 HTML 및 CSS를 사용하여 **사용자정의 설정** 패널에 입력한 데이터를 기반으로 콘텐츠를 렌더링하는지 확인합니다.

## 로컬 구성요소와 원격 구성요소 비교

로컬 구성요소와 원격 구성요소는 다르게 구현됩니다.

다음 표는 구성요소 구현 방법의 차이점을 나열합니다. 로컬 구성요소를 사용할지, 인라인 프레임으로 렌더링된 원격 구성요소를 사용할지 결정하는 데 도움이 됩니다.

로컬 구성요소 구현	원격 구성요소 구현
복잡한 콘텐츠 중심의 사용자 인터페이스를 Oracle Content Management로 통합	애플리케이션 논리를 Oracle Content Management로 통합

로컬 구성요소 구현	원격 구성요소 구현
JavaScript를 실행하여 페이지에 콘텐츠 렌더링	<iframe> HTML 태그를 사용하여 페이지에 콘텐츠 렌더링
JavaScript 기술 스택 사용	JavaScript뿐만 아니라 모든 기술 사용 가능
Oracle Content Management JavaScript 스택을 활용하여 Oracle Content Management 구성요소 재사용 가능	Oracle Content Management JavaScript 스택과 통합되지 않음
Oracle Content Management 서버에 저장된 / assets 문서에서 자산 종속성 추가	모든 자산 액세스에 HTTPS URL과 해당 인증서 필요
템플릿으로 호스트되고 Oracle Content Management에 의해 테마와 독립적으로 게시됨	URL 끝점을 위해 호스팅된 중간 계층 서버 필요
페이지 내에서 코드가 실행되므로 구성요소 코드가 망가지면 페이지가 손상될 수 있음	페이지가 인라인 프레임 콘텐츠와 독립적으로 렌더링되므로 구성요소가 망가져도 페이지가 손상되지 않음
구성요소 렌더링 속도가 느리면 페이지 렌더링이 느려질 수 있음	페이지가 인라인 프레임 콘텐츠와 독립적으로 로드되므로 페이지가 가능한 빨리 로드되고 이어서 인라인 프레임 콘텐츠가 로드됨

 주:

로컬 구성요소와 원격 구성요소의 렌더링은 다르지만 사이트 작성기의 설정 패널 구현은 동일합니다. 두 유형의 구성요소에 대한 설정 패널은 인라인 프레임을 사용하여 렌더링되고, 동일한 JavaScript SDK를 사용하여 경계를 넘는 교차 도메인 통신이 가능합니다.

## 구성요소 설정 렌더링

구성요소 설정을 렌더링하려면 구성요소 설정 URL 및 구성요소 설정 렌더링 옵션을 사용할 수 있습니다.

### 구성요소 설정 URL

구성요소 설정 URL은 인라인 프레임을 사용하여 렌더링되고 매개변수와 함께 호출되어 실제 구성요소의 특정 설정이 페이지 위에 놓이도록 합니다. 설정 URL의 형식은 다음과 같습니다.

```
{Component Settings URL}?instance=<app-instance>&width=<width>&currCompId=<id of the app associated with the settings panel>&locale=<locale>
```

이름	유형	설명
Component Settings URL	URL	구성요소의 구성요소 설정 URL
width	숫자	설정 인라인 프레임의 너비(픽셀)
currCompId	문자열	설정 패널에서 편집한 구성요소의 현재 구성요소 ID

이름	유형	설명
locale	문자열	호스트 사이트의 현재 로케일(사이트 작성기). 형식은 <code>&lt;language&gt;_&lt;dialect&gt;</code> 입니다. 예: <code>En_us</code> .

### 로컬 구성요소 설정 렌더링 옵션

로컬 구성요소의 설정 패널에서 인라인 프레임의 크기와 복잡성에 따라 인라인 프레임을 렌더링하는 세 가지 옵션이 있습니다. 각 옵션은 `settingsRenderOption` 속성에 지정됩니다.

#### 주:

이 옵션은 로컬 구성요소에만 사용할 수 있습니다. 원격 구성요소 설정 패널은 항상 대화상자에 렌더링됩니다.

- inline
  - 사용자가 입력할 몇 가지 작은 속성이 있는 경우에만 이 옵션을 사용합니다.
  - 설정 패널로 이동하는 일반 단추 대신 **일반 사항** 탭에 인라인 프레임이 삽입됩니다.
  - 가장 적은 사용자 누르기 횟수만으로 가장 통합된 솔루션을 제공하지만 공간이 제한적입니다.
- panel
  - 속성 목록이 더 길지만 표준 설정 패널의 300 픽셀 내에 충분히 표시할 수 있는 경우 이 옵션을 사용합니다.
  - 인라인 프레임이 부로 슬라이딩되고 사용자를 [일반 사항] 탭으로 되돌리는 [뒤로] 단추가 나타납니다.
  - 페이지의 인라인 프레임과 상호작용할 수 있는 통합된 솔루션을 제공합니다.
- dialog
  - 더 일반적인 설정 레이아웃을 처리하기 위한 기본 모드로, 복잡한 사용자 인터페이스가 필요합니다.
  - 페이지의 모달 대화상자에 인라인 프레임을 표시합니다.

## 로컬 구성요소 구현

구성요소 인스턴스는 구성요소 팩토리 객체에 의해 생성됩니다.

구성요소 등록 파일의 구성요소 팩토리 객체 항목은 `initData.componentFactory`입니다. RequireJS가 구성요소 팩토리 로드를 위해 사용하는 JavaScript 파일을 참조합니다. `componentFactory` 인터페이스를 구현하는 JavaScript 객체를 반환해야 합니다.

구성요소 팩토리 객체는 각 구성요소 인스턴스를 생성하는 API `customComponentFactory.createComponent(args)`를 구현해야 합니다.

`args`에는 다음 설정이 포함됩니다.

- SitesSDK: Oracle Content Management Sites SDK입니다.

- `id`: 페이지에 추가된 구성요소의 고유 ID(GUID)입니다.
- `viewMode`: 페이지가 렌더링되는 현재 모드입니다. 페이지를 편집할 때는 "편집"입니다. 페이지를 미리보기할 때는 "탐색"입니다. 런타임에 사이트가 게시될 때 값이 정의되지 않습니다. 각 모드에 어떤 기능을 노출해야 하는지에 따라 다른 구현을 제공할 수 있습니다. 예를 들어, 페이지가 편집 모드로 실행할 때는 링크가 활성이 아니어야 합니다.

### 주:

구성요소에 JQuery 또는 Knockout을 사용하라는 요구사항은 없지만, 중첩 구성요소와 같은 Oracle Content Management 기능을 활용하려면 Oracle Content Management에서 제공한 Knockout 버전을 사용해야 합니다. 이 Knockout 버전에는 다른 곳에 제공되지 않는 확장된 구성요소 등록 및 처리기가 있습니다.

구성요소 자체의 경우 구성요소를 인스턴스화할 때 SDK가 전달되므로 구성요소가 페이지 수명 주기와 통신할 수 있습니다. 페이지 수명 주기 기능은 구성요소에 의해 구현되고 Oracle Content Management에 의해 호출되어 페이지에 구성요소를 렌더링할 수 있습니다.

필수 및 선택적 API가 구성요소 구현을 위해 제공됩니다.

#### 필수 API

`customComponent.render(container)`: 제공된 DOM 컨테이너 요소에 구성요소를 삽입하도록 요청합니다.

- `container`: 사용자정의 구성요소 HTML에 대한 DOM 컨테이너 요소입니다.

#### 선택적 API

`customComponent.dispose()`: 구성요소가 페이지에서 제거될 때 호출됩니다. 구성요소가 더 이상 필요하지 않은 리소스를 제거할 수 있는 기회를 제공합니다.

## 구성요소에 대한 스타일 클래스

사용자가 구성요소에 적용할 수 있는 정의된 스타일 목록을 생성할 수 있습니다.

구성요소에 미리 정의된 스타일 목록이 있으면 테마의 디자인 파일을 통해 Oracle Content Management에서 제공한 구성요소의 스타일 클래스를 정의할 때와 동일한 모델을 따릅니다. 구성요소의 `appinfo.json` 파일에서 사용자정의 스타일 클래스의 이름을 지정합니다.

`design.css` 및 `design.json` 파일에서 추가 스타일을 정의합니다. `json` 파일은 사용자 인터페이스에 표시되는 이름을 실제 기본 `css` 클래스 이름에 매핑하고, `css` 파일은 각 스타일 클래스에 대한 세부정보를 제공합니다.

`design.json` 파일의 구성요소 구조는 다음과 같습니다.

```
{
  "componentStyles": {
    "scs-image": {
      "styles": []
    }
  }
}
```

```

    },
    "scs-map": {
      "styles": []
    },
    "scs-title": {
      "styles": []
    },
    "scs-paragraph": {
      "styles": []
    },
    "scs-txt": {
      "styles": []
    },
    "scs-divider": {
      "styles": []
    },
    "scs-button": {
      "styles": []
    },
    "scs-app": {
      "styles": []
    },
    "scs-spacer": {
    },
    "scs-gallery": {
      "styles": []
    },
    "scs-youtube": {
      "styles": []
    },
    "scs-socialbar": {
      "styles": []
    },
    "scs-document": {
      "styles": []
    }
  }
}

```

각 "styles":[] 항목은 특정 구성요소에 대한 스타일 목록을 포함할 수 있습니다. "name"은 내장된 지역화 문자열에 대한 참조이거나, 사용하도록 지정된 값일 수 있습니다. 예를 들어, 제목 구성요소는 다음 기본 스타일을 제공합니다.

```

{
  "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"
    }
}
}

```

다음과 같이 이름 값은 사용자 인터페이스에 표시할 실제 단어에 매핑됩니다.

```

"COMP_STYLE_FLAT": "Flat",
"COMP_STYLE_HIGHLIGHT": "Highlight",
"COMP_STYLE_DIVIDER": "Divider",

```

css 파일은 클래스 값에 대한 정의를 제공합니다.

```

.scs-title-default-style {
  color: #333333;
  display: block;
  font-family: "Helvetica Neue", "Helvetica", "Arial", sans-serif;
  font-size: 24px;
  font-weight: normal;
}

```

예를 들어, 테마의 design.json 파일에서 components.json 파일에 정의한 initialData.componentId 값을 기반으로 구성요소에 항목을 추가할 수 있습니다.

```
"componentId": "news-article"
```

design.json 파일의 해당 항목은 다음과 같습니다.

```

"componentStyles": {
  "news-article": {
    "styles": [{
      "name": "News Article 1",
      "class": "news-article-default-style"
    },
    {
      "name": "News Article 2",
      "class": "news-article-style-1"
    }
  ]
},

```

design.css 파일의 해당 항목은 다음과 같습니다.

```

.news-article-default-style .scs-image {...}
.news-article-style-1 .scs-image {...}

```

## 내장 구성요소 스타일링 방법

Oracle Content Management에서 내장 스타일을 무효화-확장하여 나만의 모양으로 내장 구성요소 스타일을 생성합니다.



내장 구성요소는 다음 두 곳에서 시각적 스타일을 얻습니다.

- `comp.css` - 각 구성요소의 기본 모양을 지정하는 내장 CSS 파일
- `design.css` - 사이트에 사용 중인 테마의 일부인 CSS 파일

`design.css` 파일에서 내장 `comp.css` 스타일을 무효화-확장하여 나만의 모양을 만들 수 있습니다. 테마에서 `design.css` 파일은 `designs/default` 디렉토리에 있습니다.

다음 항목은 모든 내장 구성요소에 공통적인 `comp.css` 파일의 클래스를 설명하고 테마 정의에 대한 개요를 제공합니다.

- [구성요소 스타일링 기본사항](#)
- [구성요소별 스타일링](#)
- [구성요소 속성 설정](#)

## 구성요소 스타일링 기본사항

모든 내장 사이트 구성요소는 유사한 CSS 클래스 구조를 공유합니다.

각 구성요소는 다음 세 가지 CSS 클래스가 가장 바깥쪽 `<div>` 요소에 적용됩니다.

```
scs-component scs-type design-style
```

`type`은 구성요소 유형(예: `image`, `gallery` 또는 `divider`)입니다. `design-style`은 테마 파일에 정의된 대로 구성요소에 선택된 스타일 클래스입니다.

### 테마 스타일 기본사항

테마의 `design.json` 파일은 각 구성요소 유형(예: `button` 또는 `image`)에 적용할 수 있는 모든 스타일(`frame`, `shadow`, `highlighted` 등)을 나열합니다. 각 스타일에는 표시 이름과 클래스 이름이 있습니다. 표시 이름은 **설정** 패널의 **스타일** 탭에 표시됩니다. 클래스 이름은 테마의 `design.css` 파일에서 CSS 선택기를 나타냅니다. 예를 들어, `button` 구성요소의 항목은 다음과 같습니다.

```
"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"
  }
  ]
},
```

내장 구성요소의 이름은 번역되므로 리소스 번들에서 스타일 이름을 가져오는 키가 표시됩니다. 페이지에 `button` 구성요소를 추가하고 **설정 > 스타일** 패널에서 **단순** 스타일을 선택하면

design.json 파일은 표시 이름 **단순**(키 COMP\_STYLE\_SIMPLE)을 클래스 이름 scs-button-style-3과 연관시킵니다. button은 다음 클래스로 렌더링됩니다.

```
scs-component scs-button scs-button-style-3
```

주어진 구성요소에 선택한 스타일이 없으면 기본 스타일인 scs-type-default-style이 사용됩니다. 앞의 예제에서 button은 다음 클래스로 렌더링됩니다.

```
scs-component scs-button scs-button-default-style
```

### scs-component-content 스타일

모든 내장 구성요소에 대해 이전에 언급한 scs-component <div> 안에 CSS 클래스 scs-component-content와 함께 content <div>가 있습니다. 다시 말해, 다음과 같습니다.

```
scs-component scs-type design-style
scs-component-content
```

design.css 파일에서 scs-component-content 클래스는 구성요소 주위의 "상자" 스타일링에 사용되기도 합니다(예: 테두리 또는 그림자 적용).

내장 comp.css 파일에서 공통 scs-component-content 클래스는 다른 CSS 속성 중에서 position:relative 및 display:inline-block으로 정의됩니다.

scs-component-content는 각 구성요소 주위의 "상자" 스타일링에 유용하지만, 구성요소를 완전히 스타일링하려면 구성요소별 클래스가 필요합니다. [구성요소별 스타일링](#)을 참조하십시오.

## 구성요소별 스타일링

이미지, 단추, 문서, 단락, 제목, 맵과 기타 구성요소에 특정 스타일을 적용할 수 있습니다.

### 이미지 구성요소

이미지 구성요소는 scs-component-content 클래스 아래에 다음 CSS 클래스 구조를 갖습니다.

```
scs-image-container
scs-image-link
scs-image-image
scs-image-caption
```

scs-image-image 클래스는 <img> 태그 자체에 적용됩니다. 캡션이 있는 경우 scs-image-caption 클래스는 캡션 스타일링에 사용됩니다.

scs-image-link 클래스는 이미지에 첨부된 링크가 있는 경우에만 제공됩니다. 이 클래스나 scs-image-container 클래스는 일반적으로 사용자정의 스타일링이 필요하지 않습니다.

기본적으로 이미지 캡션은 이미지 하단에 펼쳐진 반투명 오버레이로 렌더링됩니다.



```
.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;  
}
```

이미지 상단에 캡션을 배치하고 색상을 변경하려면 `design.json` 파일에서 이미지 구성요소에 다른 스타일을 추가하고 `design.css` 파일에서 CSS를 정의합니다.



```
.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;  
}
```

### 단추 구성요소

단추 구성요소는 다음 클래스 구조를 갖습니다.

```
scs-button-button  
  scs-button-text
```

`scs-button-button` 클래스는 단추 스타일의 누르기 가능한 `<div>`입니다. `scs-button-text` 클래스는 단추 안의 텍스트 스타일링에 사용됩니다.

예를 들어, `design.json` 파일에 다른 스타일을 추가하고 `design.css` 파일에서 CSS를 정의하여 단추 구성요소의 모양 변경을 테스트합니다.



```
.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;
}
```

## 문서

문서 구성요소는 다음 클래스 구조를 갖습니다.

```
scs-document-container
  scs-document-cap
    scs-document-title
  scs-document-desc
```

`scs-document-container` 클래스는 문서 뷰어를 래핑하며 일반적으로 스타일링하지 않습니다.

## 갤러리

갤러리 구성요소에는 기본 JSSOR 슬라이더 구성요소를 래핑하는 단일 클래스가 있습니다.

```
scs-gallery-container
```

JSSOR 슬라이더는 스타일링 가능한 여러 클래스를 사용합니다.

```
jssorb14 (navigator)
jssora02l (left arrow)
jssora02r (right arrow)
jssort07 (thumbnails)
```

## 갤러리 그리드

갤러리 그리드 구성요소에 사용되는 클래스는 설정 패널에서 선택한 레이아웃과 자르기에 따라 다릅니다.

```
scs-gallerygrid-container scs-gallerygrid-layout
  scs-gallerygrid-cell
    scs-image (multiple)
```

갤러리 그리드에 선택된 자르기 및 레이아웃 설정에 따라 `layout` 값은 늘이기, 자르기, 맞춤 또는 흐름이 됩니다.

`scs-gallerygrid-cell` 클래스는 열 레이아웃에만 제공됩니다.

## 소셜 표시줄

소셜 표시줄 구성요소는 다음 클래스 구조를 갖습니다.

```
scs-socialbar-container
  scs-socialbar-icon
```

`scs-socialbar-icon` 클래스는 소셜 표시줄의 각 `<img>` 태그에 적용됩니다.

## 단락

단락 구성요소에는 실제 단락 텍스트를 래핑하는 단일 클래스만 있습니다.

```
scs-paragraph-text
```

예를 들어, 단락 구성요소에 입력한 텍스트에 `engraved-text-on-metal` 효과를 주려면 `design.json` 파일에 다른 스타일 클래스를 추가하고 `design.css` 파일에서 CSS를 정의합니다.



```
.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;  
}
```

또는 화려하게 보이려면 다음 예제와 같이 사용하십시오.



```
.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;
}

```

## 제목

제목 구성요소에도 실제 텍스트를 래핑하는 단일 클래스만 있습니다.

```
scs-title-text
```

## 맵

맵 구성요소에는 맵 표현을 래핑하는 단일 클래스가 있습니다.

```
scs-map-content
```

이 클래스는 일반적으로 스타일링하지 않습니다.

## 사용자정의 로컬 구성요소

사용자정의 로컬 구성요소에는 실제 구성요소를 래핑하는 단일 클래스만 있습니다.

```
scs-custom-component-wrapper
```

사용자정의 로컬 구성요소에 대한 사용자정의 뷰를 렌더링할 때 사용할 CSS 스타일을 완전히 제어할 수 있습니다. 로컬 구성요소는 인라인으로 렌더링됩니다. 즉, 테마 또는 design.css 파일에 정의된 CSS 스타일을 직접 적용할 수 있습니다.

## 사용자정의 원격 구성요소

사용자정의 원격 구성요소에는 iframe을 래핑하는 단일 클래스만 있습니다.

```
scs-app-iframe-wrapper
```

사용자정의 원격 구성요소에 정의된 CSS 스타일을 적용하는 것 외에, Sites SDK를 활용하여 호스트 사이트에서 design.css 파일을 인출할 수 있습니다.

```

// 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);
    }
  }
});

```



따라서 구성요소 스타일을 호스트 스타일에서 상속하도록 만들 수 있습니다.

### 구분선

구분선 구성요소에 대한 구성요소별 클래스는 없지만 <hr> 태그 자체를 스타일링할 수 있습니다.

예를 들어, 점선 구분선을 생성할 수 있습니다.

```
.design-style .scs-divider hr {
border-top: 1px dotted #333333;
}
```

### 비디오, YouTube, 스페이서


비디오, YouTube 또는 스페이서 구성요소에 대한 구성요소별 클래스는 없습니다.

## 구성요소 속성 설정

사이트에서 사용할 구성요소 속성을 구성할 수 있습니다.

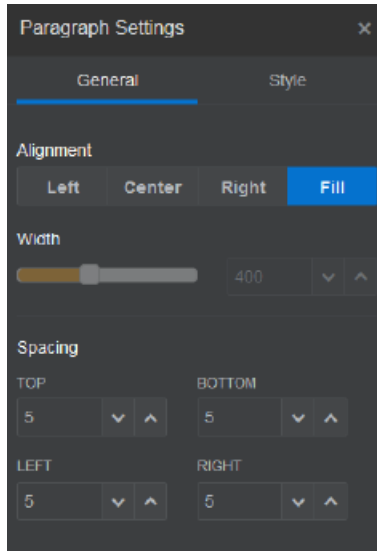
구성요소는 웹 페이지의 개별 부분으로 텍스트, 제목, 이미지, 단추, 구분선, 맵, 갤러리, 비디오 등을 포함합니다. 테마의 디자인을 생성할 때 각 구성요소 유형에 대한 기본 설정도 지정해야 합니다. 각 구성요소에는 크기, 맞춤, 간격, 색상, 테두리 등 구성요소의 모양과 동작을 정의하는 설정이 있습니다. 설정은 구성요소에 따라 다릅니다. 사이트에 새 테마로 제공된 구성요소 속성을 사용자가 변경할 수 있는지 여부를 선택할 수도 있습니다.


예를 들어, 다음 단계는 단락 구성요소의 설정을 구성할 수 있는 방법을 보여줍니다.

1. 개발 사이트가 편집 모드로 열린 상태에서 단락 구성요소가 있는 페이지를 선택하거나 단락 구성요소를 추가합니다.
2. 단락 구성요소를 누르고 구성요소 모서리에서  을 누른 다음 **설정**을 선택합니다. 구성요소의 옵션을 선택할 수 있는 **설정** 패널이 표시됩니다.

#### 주:

설정 옵션은 각 구성요소 유형에 특정합니다. 로컬 또는 원격 사용자정의 구성요소로 작업하는 경우 **사용자정의 설정** 링크가 표시됩니다.



3. 예를 들어 맞춤, 너비, 간격, 글꼴, 스타일, 색상, 크기 등의 설정을 변경할 수 있습니다. 구성요소 모양이 새 설정으로 변경됩니다.
4. 완료되면  을 누릅니다. 설정이 페이지에 적용됩니다.

## 인라인 프레임으로 렌더링된 구성요소

인라인 프레임으로 렌더링된 구성요소는 외부 서버의 구성요소(일명 **원격 구성요소**)를 등록 및 추가하고 로컬 구성요소를 생성할 때 인라인 프레임 옵션을 선택하면 Oracle Content Management 페이지에 지정할 수 있습니다. 이 유형의 구성요소는 소셜 구성요소 또는 계산대 카드 구성요소를 추가하는 등 사이트 기능을 확장할 수 있습니다.

원격 구성요소의 경우 끝점 URL을 지정해야 합니다. 인라인 프레임으로 렌더링된 로컬 구성요소의 경우 URL은 구성요소 카탈로그의 구성요소 이름에서 파생됩니다.

- 끝점 URL: 위젯 콘텐츠가 이 URL에서 인출되어 인라인 프레임 내에 포함됩니다.
- 설정 URL: 이 URL은 인라인 프레임으로 렌더링되어 페이지 위에 놓인 구성요소를 구성합니다.

구성요소는 정적 또는 동적 데이터 시각화를 렌더링하고 사이트 기능을 확장하는 폼 또는 기타 대화식 사용자 인터페이스를 표시할 수 있습니다. [구성요소 설정 렌더링](#)을 참조하십시오.

### Oracle Content Management에서 제공하는 구성요소

인라인 프레임으로 렌더링된 구성요소 집합은 Oracle Content Management와 함께 제공됩니다.

이름	유형	ID
폴더 목록	scs-app	폴더 목록
파일 목록	scs-app	파일 목록
문서 관리자	scs-app	문서 관리자
Facebook 좋아요	scs-app	Facebook 좋아요

이름	유형	ID
Twitter 팔로우	scs-app	Twitter 팔로우
Twitter 공유	scs-app	Twitter 공유
Facebook 추천	scs-app	Facebook 추천

### 구성요소 등록

인라인 프레임으로 렌더링된 원격 구성요소를 사이트에서 사용하기 전에 HTTPS를 사용하여 끝점 URL을 등록해야 합니다. 이 정보는 구성요소 카탈로그에 저장됩니다. 끝점은 URL이 인라인 프레임으로 표시되도록 해야 합니다. 헤더에 X-Frame-Options="sameorigin"을 설정하지 마십시오. 로컬 구성요소의 경우 파일이 Oracle Content Management 서버에 저장되므로 이 끝점 기준이 자동으로 충족됩니다.

인라인 프레임으로 렌더링된 구성요소가 등록되면 구성요소를 나타내는 새 GUID가 생성됩니다. 해당 구성요소가 Oracle Content Management 인스턴스에 여러 번 등록된 경우 각각의 구성요소 등록을 나타내는 여러 GUID를 얻습니다. 구성요소를 등록할 때 GUID에 대해 Oracle Content Management에 저장된 것은 설명뿐입니다. 구성요소는 여전히 원격 끝점에서 실행됩니다. [원격 구성요소 등록](#)을 참조하십시오.

인라인 프레임으로 렌더링된 구성요소가 등록되면 인스턴스 ID도 생성됩니다. 이 인스턴스 ID는 특정 Oracle Content Management 테넌트에 등록된 구성요소를 나타냅니다. 동일 구성요소가 동일한 Oracle Content Management 테넌트 내에 두 번 이상 등록되거나 다른 Oracle Content Management 테넌트에 등록되면 다른 인스턴스 ID를 갖습니다.

### 원격 구성요소 설정 지속성

인라인 프레임으로 렌더링된 구성요소의 설정 URL이 설정 패널 대화상자에 렌더링되면 인스턴스 ID와 구성요소 ID가 모두 제공됩니다. 그러면 구성요소는 인스턴스 ID 및 구성요소 ID별로 인덱스화된 자체 설정을 고유 서버에 지속하도록 선택할 수 있습니다. 다른 방법으로, Oracle Content Management Sites SDK를 사용하여 구성요소 ID에 대해 최대 1.5KB의 JSON 데이터를 사이트 페이지의 페이지 모델에 저장할 수 있습니다.

인라인 프레임으로 렌더링된 구성요소에 대해 Sites SDK를 사용하여 설정을 지속하면 두 가지 이점이 있습니다.

- 구성요소가 페이지 버전, 페이지 업데이트 및 사이트 게시 모델에 쉽게 참여할 수 있습니다.
- 구성요소가 백엔드 시스템에서 실행되는 것이 아닌, 브라우저에서 실행되는 HTML 끝점으로 구성될 수 있습니다.

Oracle Content Management SDK를 참조하십시오.

## 인라인 프레임으로 렌더링된 구성요소에 대한 인스턴스 ID 및 구조 정보

구성요소 인스턴스 ID는 사이트 내에 인라인 프레임으로 렌더링된 구성요소의 고유 식별자입니다.

사용자가 인라인 프레임으로 렌더링된 구성요소를 구성요소 카탈로그에서 사이트 페이지 위에 끌어 놓으면 Oracle Sites Cloud Service에 프로비전 호출을 발생시켜 새로운 고유 구성요소 인스턴스 ID를 생성합니다. 이 ID는 고유한 것으로 보장되며 해당 서비스에 프로비전된 모든 구성요소 인스턴스는 동일한 인스턴스 ID를 얻습니다.

구성요소 인스턴스 ID에는 인라인 프레임으로 렌더링된 구성요소의 사용 및 설정을 보안할 수 있는 추가 정보가 있어서 구성요소는 설정 업데이트가 신뢰된 곳에서 온다고 확인할 수 있습니다.

인스턴스 ID 매개변수를 통해 개발자는 사이트를 식별하고 발신자를 인증할 수 있습니다. 구성요소 보안 키를 사용하여 생성된 디지털 서명을 확인하여 호출자가 인증됩니다. 보안 키는 구성요소 등록 프로세스 중에 생성됩니다.

구성요소 인스턴스는 ':' 구분자로 분리된 두 부분(데이터 및 구조)으로 구성됩니다.

#### 구성요소 인스턴스: 데이터

인라인 프레임으로 렌더링된 구성요소에 대한 인스턴스의 데이터 부분은 Base64 JSON 인코딩 문자열입니다. 다음은 JSON 문자열의 구조입니다.

```
{
  "instanceid": "BBDC7614F693B75110D811E6C0B77C935FAEC5112E5E",
  "permissions": "",
  "entitlements": "",
  "signdate": "1435426735293",
  "sitedomain": "service1-tenant4.localhost"
}
```

필드 이름	설명
instanceid	Oracle Content Management 테넌트에서 인라인 프레임으로 렌더링된 구성요소의 고유 식별자입니다.
signdate	서명 생성 날짜입니다.
sitedomain	Oracle Content Management 인스턴스의 도메인 이름입니다.
permissions	사이트 멤버의 권한 집합입니다. 편집 모드에서 "SITE_OWNER" 값을 갖고, 그렇지 않으면 값이 없습니다.
entitlements	사이트 소유자가 구매한 프리미엄 기능 목록입니다.

#### 구성요소 인스턴스: 서명

구성요소 인스턴스의 데이터 부분은 APP\_SECRET\_KEY에 의해 서명되기 전에 직렬화됩니다. 이 보안 키는 구성요소를 등록하는 동안 생성되어 개발자에게 표시되어야 합니다. 다음과 같이 보안 키로 구성요소 인스턴스의 데이터 부분(직렬화된 JSON 구조)의 해시를 생성하여 서명이 계산됩니다.

```
$signature = HMAC (serialized JSON structure, APP_SECRET_KEY)
```

서명 생성에 사용된 해시 알고리즘은 SHA256입니다. 다음과 같이 직렬화된 JSON 구조와 생성된 서명 구성요소가 연결되어 토큰이 형성됩니다.

```
$instance = {base64encoded serialized JSON structure}.
{base64encoded $signature}
```

예제:

```
//base64 encoded serialized object          //signature
eyJpbmN0YW5jZWlkIjojQTRGOTE3REY5OTZEN0Q3ODBCMjUzODZFOTFEMDA3ODJGMjVBRjY2Rjc3OTIiLCJzaWduZGZG
BHczdbVZx2VH8=
```

## 원격 구성요소에 대한 보안

Oracle Content Management에서 타사 개발자는 사용자정의 구성요소를 Oracle Content Management 플랫폼으로 통합할 수 있지만 원격 서버에 저장됩니다.

각 원격 구성요소는 설정 및 렌더링 끝점이 Oracle Content Management에 등록되어야 합니다. 끝점 외에도 개발자는 등록된 구성요소에 고유한 보안 키를 제공해야 합니다.

Oracle Content Management는 등록된 구성요소의 끝점을 호출하여 사이트 페이지에 콘텐츠를 실현합니다. 해당 끝점은 공용 인터넷에 노출되므로 개발자는 등록된 원격 구성요소의 끝점이 Oracle Content Management에서 호출되는지 확인해야 합니다. 호출자의 진위를 확인하기 위해 서명한 토큰이 등록된 URL 끝점으로 전달됩니다. 등록 프로세스 중에 제공된 원격 구성요소의 보안 키를 사용하여 서명한 토큰에 포함된 디지털 서명을 확인하여 발신자가 인증됩니다.

토큰 형식은 다음과 같습니다.

```
{base64 encoded serialized JSON data}.{base64 encoded signature}
```

등록된 앱 끝점으로 전달된 샘플 토큰은 다음과 같습니다.

```
eyJpbmN0YW5jZWlkIjojQTRGOTE3REY5OTZEN0Q3ODBCMjUzODZFOTFEMDA3ODJGMjVBRjY2Rjc3OTIiLCJzaWduZGZG
BHczdbVZx2VH8=
```

토큰은 ':' 구분자로 분리된 두 개별 부분(데이터 및 서명)으로 구성됩니다.

일반적 지침상, 개발자는 구성요소의 등록된 끝점에 액세스를 부여하기 전에 항상 편집 또는 미리보기 모드에서 토큰을 인증해야 합니다. 또한 설정 끝점에서 발신자를 인증하는 동안 개발자는 항상 토큰의 **permissions** 필드에서 `SITE_OWNER` 값을 찾아야 합니다. 토큰의 **permissions** 필드는 편집 모드에서만 `SITE_OWNER` 값을 보여줍니다. 편집 세션 중에 생성된 토큰은 절대 페이지 모델에 지속되지 않으며 런타임 토큰으로 전환되어 **permissions** 필드에 `NULL` 값을 갖습니다.

### 데이터

인스턴스의 데이터 부분은 Base64 JSON 인코딩 문자열입니다. 다음은 JSON 문자열의 구조입니다.

```
{
  "instanceid": "BBDC7614F693B75110D811E6C0B77C935FAEC5112E5E",
  "permissions": "",
  "entitlements": "",
  "signdate": "1435426735293",
  "sitedomain": "service1-tenant4.localhost"
}
```

필드 이름	설명
instanceid	Oracle Content Management 테넌트에 대한 구성요소의 고유 식별자입니다.

필드 이름	설명
signdate	서명 생성 날짜입니다.
sitedomain	Oracle Content Management 인스턴스의 도메인 이름입니다.
permissions	사이트 멤버의 권한 집합입니다. 편집 모드에서 "SITE_OWNER" 값을 갖고, 그렇지 않으면 값이 없습니다.
entitlements	사이트 소유자가 구매한 프리미엄 기능 목록입니다.

## 서명

원격 구성요소 인스턴스의 데이터 부분은 APP\_SECRET\_KEY에 의해 서명되기 전에 직렬화됩니다. 이 보안 키는 구성요소를 등록하는 동안 생성되어 개발자에게 표시되어야 합니다. 다음과 같이 보안 키로 구성요소 인스턴스의 데이터 부분(직렬화된 JSON 구조)의 해시를 생성하여 서명이 계산됩니다.

```
$signature = HMAC (serialized JSON structure, APP_SECRET_KEY)
```

서명 생성에 사용된 해시 알고리즘은 SHA256입니다. 다음과 같이 직렬화된 JSON 구조와 생성된 서명 구성요소가 연결되어 토큰이 형성됩니다.

```
$instance = {base64encoded serialized JSON structure}.
{base64encoded $signature}
```

# 원격 구성요소 등록

사이트에서 원격 구성요소를 사용하기 전에 Oracle Content Management에 등록해야 합니다.

타사 원격 구성요소와 자체 개발한 구성요소를 등록할 수 있습니다.

Oracle Content Management 인스턴스에서 사용하도록 원격 구성요소를 등록하려면 다음과 같이 하십시오.

1. 개발자를 누르고 모든 구성요소 보기를 누릅니다.
2. 생성을 누르고 원격 구성요소 등록을 선택합니다.
3. [원격 구성요소 등록] 창에서 다음 정보를 입력하거나 선택합니다.
  - **이름:** 사용자에게 표시되는 구성요소의 이름입니다.
  - **설명:** 사용자에게 표시되는 구성요소의 설명입니다.
  - **구성요소 URL:** 페이지에 구성요소 콘텐츠를 렌더링하기 위해 iframe에 사용되는 끝점입니다. HTTPS여야 합니다.
  - **설정 URL:** 페이지에 추가된 원격 구성요소 설정을 렌더링하기 위해 iframe에 사용되는 끝점입니다. HTTPS여야 합니다.
  - **설정 너비:** 구성요소 설정 패널의 기본 너비(픽셀)를 설정합니다.
  - **설정 높이:** 구성요소 설정 패널의 기본 높이(픽셀)를 설정합니다.
  - **키:** 원격 구성요소와 연관된 192비트 AES 키로, 구성요소를 프로비전할 때 서명된 해시 토큰을 생성하는 데 사용됩니다. 구성요소를 암호화하고 구성요소 설정을 안전하게 읽고 쓸 수 있도록 합니다.

#### 4. 등록을 누릅니다.

원격 구성요소가 생성되면 구성요소 목록에 이름이 나타납니다. 구성요소 목록에서 구성요소 이름을 눌러 구성요소 등록에 사용되는 파일을 탐색할 수 있습니다.

동일한 Oracle Content Management 인스턴스에서 생성된 사이트에서 사용되는 카탈로그에 구성요소 등록 정보가 저장되지만, 구성요소는 원격 서비스로 남아 있습니다.

구성요소 소유자의 경우 구성요소 아이콘이 사이트 작성기의 사용자정의 구성요소 패널에 구성요소에 지정된 이름으로 추가됩니다. 다른 사용자와 구성요소를 공유하면 사이트 작성기의 사용자정의 구성요소 패널에서 구성요소를 볼 수 있습니다.

## 구성요소 삭제


적절한 권한이 있으면 더 이상 사용할 수 없도록 구성요소 관리자에서 구성요소를 삭제할 수 있습니다. 구성요소를 삭제하면 구성요소 폴더와 모든 관련 폴더 및 파일이 휴지통으로 이동합니다.

구성요소를 직접 생성했거나(구성요소 소유자) 누군가 나와 구성요소를 공유하고 내게 관리자 롤을 부여한 경우 구성요소 관리자에서 구성요소를 삭제할 수 있습니다.

### 주:

휴지통에 있는 사이트/업데이트를 포함해 모든 사이트/업데이트에서 사용 중인 구성요소는 삭제할 수 없습니다.

구성요소를 삭제하려면 다음과 같이 하십시오.

1. 홈 페이지에서 **개발자**를 누릅니다.
2. **모든 구성요소 보기**를 누릅니다.  
현재 등록된 구성요소가 표시됩니다.
3. 구성요소 이름을 선택하고 마우스 오른쪽 단추 누르기 메뉴에서 **삭제**를 선택하거나 작업 모음에서  을 누릅니다.  
작업 확인 메시지가 표시됩니다.
4. 삭제 작업을 확인하려면 **예**를 누릅니다. 삭제 작업을 정지하려면 **아니오**를 누릅니다.  
삭제를 확인하면 구성요소와 모든 관련 폴더 및 파일이 휴지통으로 이동합니다.

다음 이벤트 전까지 삭제된 구성요소 폴더가 휴지통에 보존됩니다.

- 폴더를 복원합니다.
- 폴더를 영구적으로 삭제합니다.
- 휴지통 할당량에 도달합니다.
- 서비스 관리자가 설정한 간격에 따라 자동으로 휴지통이 비워집니다. 기본값은 90일입니다.

## Sites SDK

Oracle Content Management용으로 개발된 구성요소는 사이트에 구성요소로 렌더링되고 사이트의 특정 페이지 아무 곳이나 끌어 놓을 수 있습니다.

Sites SDK는 구성요소와 페이지 간의 모든 통신을 처리합니다.

- `Sites.Settings.getProperty(propertyName, callbackFunction)`: 사용자정의 구성요소 인스턴스에 대해 요청한 속성을 검색하도록 콜백을 제공합니다.
- `Sites.Settings.setProperty(propertyName, propertyValue)`: 사용자정의 구성요소 인스턴스에 대해 요청한 속성을 저장합니다.

모든 사용자정의 구성요소 속성에 대해 `Sites.Settings.getProperty` 또는 `Sites.Settings.setProperty`를 사용한 다음, 메시지 이벤트 수신을 위해 `SitesSDK.publish` 및 `SitesSDK.subscribe`를 사용하십시오.

[Sites SDK 참조](#)를 참조하십시오.



# 24

## 컨트롤러 파일 사용자정의

Oracle Content Management 사이트에서 컨트롤러 파일은 사이트의 각 페이지를 표시하기 위해 사용됩니다. 브라우저가 웹 페이지에 대해 요청을 전송하면 서버가 컨트롤러 파일 복사본으로 응답합니다. 요청된 모든 페이지에 대해 컨트롤러 파일의 동일한 복사본이 서버에서 제공됩니다.

- [컨트롤러 파일 정보](#)
- [기본 controller.html 파일 수정](#)
- [SCS 객체 정보](#)
- [사용자정의하지 않아야 하는 컨트롤러 파일 섹션](#)
- [사용자정의 컨트롤러 파일 이식성 허용을 위한 토큰 사용](#)
- [사용자정의 컨트롤러 파일 샘플](#)

## 컨트롤러 파일 정보

컨트롤러 파일은 페이지의 남은 부분에 대해 렌더링 시퀀스를 동적으로 시작하는 작은 HTML 페이지입니다. 사이트의 모든 페이지 동작에 영향을 주도록 사용자정의를 적용할 수 있는 첫번째 위치입니다.

컨트롤러 파일의 기본 작업은 컨트롤러 JavaScript의 실행 환경을 제공하고 호스트하는 것입니다. 그런 후 컨트롤러 JavaScript가 페이지를 로드하고 표시합니다. 컨트롤러 파일의 사용자정의는 컨트롤러 JavaScript의 작업을 무효화하고 이에 영향을 주는 기능을 제공합니다.

## 기본 컨트롤러 파일

Oracle Content Management 사이트를 처음 생성하면 기본 컨트롤러 파일인 controller.html이 사이트와 연관됩니다.

```
<!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>
```

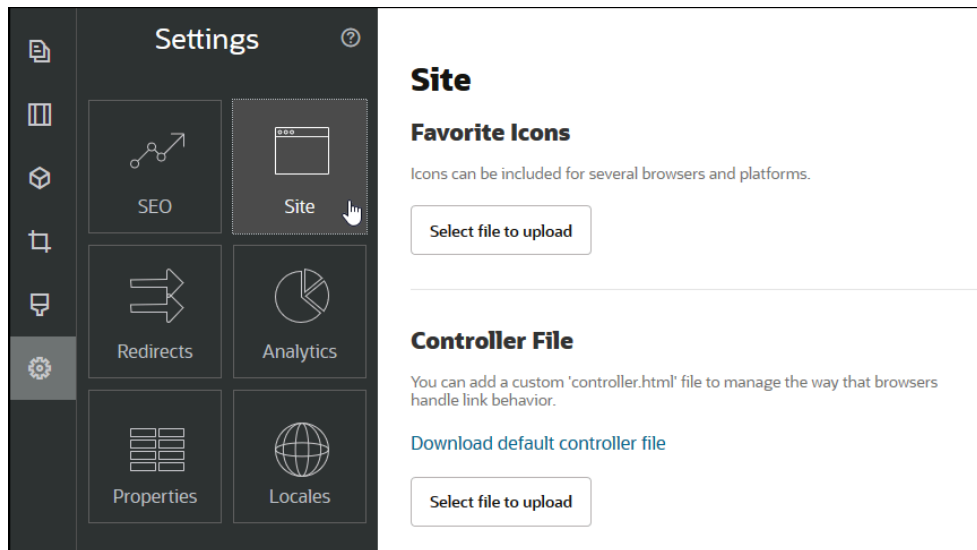
```
<!-- The following image tag displays an animated circle by default if
it takes too long for a page to display. The wait image must have the
id scsWaitImage. -->

</body></html>
```

## 기본 controller.html 파일 수정

사이트의 controller.html 파일을 다운로드하고 각 사이트 페이지의 동작에 영향을 주도록 수정합니다. 사이트의 controller.html 파일을 다운로드하려면:

1. 사이트 작성기에서 수정할 사이트를 열고 **편집**으로 설정합니다.
2. 기존 업데이트를 선택하거나 새 업데이트를 생성합니다.
3. 측면 탐색 메뉴에서 **설정**을 누른 후 **사이트**를 누릅니다.



4. [컨트롤러 파일] 섹션에서 **기본 컨트롤러 파일 다운로드**를 누릅니다.
5. 로컬 드라이브에 저장하고 편집 사항에 따라 controller.html 파일을 사용자정의합니다.
6. 편집이 완료되었으면 사이트 설정 페이지로 돌아가고 [컨트롤러 파일] 섹션에서 **업로드할 파일 선택**을 누릅니다.
7. 수정한 컨트롤러 파일로 이동하고 이를 업로드합니다.

**주:**

컨트롤러 파일의 사용자정의는 온라인 사이트에서만 사용됩니다. 사이트를 편집할 때나 사이트 미리보기에서는 사용되지 않습니다. 사이트에 대해 사용자정의 `controller.html` 파일을 업로드한 후 변경사항을 적용하려면 사이트 업데이트를 커밋하고 사이트를 게시해야 합니다.

## SCS 객체 정보

컨트롤러 JavaScript는 전역 SCS 객체에 정의된 변수 및 속성을 사용하여 웹 페이지를 렌더링합니다. 전역 JavaScript 객체는 `controller.js` 파일이 포함되기 전 컨트롤러 파일에 정의되어 있어야 합니다. SCS 객체에는 사용자정의에 사용할 수 있는 두 가지 기본 속성이 있습니다.

- [SCS.sitePrefix](#)
- [SCS.preInitRendering](#)
- [SCS.getDeviceInfo](#)

### SCS.sitePrefix

`SCS.sitePrefix` 변수는 온라인 사이트의 경로 접두어를 정의합니다. 일반적으로 기본값은 `site/<siteName>/`입니다.

컨트롤러 JavaScript는 이 값을 사용하여 표시할 웹 페이지를 결정합니다. 예를 들어, 브라우저가 `/site/SampleSite/products/index.html`에서 페이지를 요청하는 경우, `sitePrefix /site/SampleSite/`는 JavaScript가 사이트의 `products/index.html` 페이지를 표시해야 하는지 계산하도록 허용합니다. `sitePrefix`는 "/" 문자로 시작하고 끝나는 문자열 값이어야 합니다.

Akamai 및 URL 매핑 규칙과 같은 프록시 또는 다른 인프라가 사용되는 경우 이 변수를 수정하여 사이트 접두어를 사용자정의할 수 있습니다. 예를 들어, `sitePrefix`를 `/intranet/ExampleSite/`로 설정하면 기본 `/site/SampleSite/` 경로 접두어 대신 브라우저에서 해당 경로 이름으로 웹 사이트를 제공할 수 있습니다.

**주:**

브라우저의 URL이 사이트 접두어와 일치하지 않으면 내부적으로 기본 컨트롤러 JavaScript가 "/"를 `sitePrefix`로 사용합니다. 이렇게 하면 사용자정의 없이 `vanity` 도메인을 사용하여 사이트를 제공할 수 있습니다.

### SCS.preInitRendering

정의된 경우 계산 논리가 실행되기 전 컨트롤러 JavaScript 코드에서 `SCS.preInitRendering` 함수가 호출됩니다. 이것은 컨트롤러 JavaScript의 기본 작업을 무효화하도록 사용자정의 컨트롤러 파일에서 정의하는 데 유용한 함수입니다.

정의되었으면 `SCS.preInitRendering`이 함수여야 합니다. 이 함수는 인수 없이 호출되며, 반환 값이 요구되거나 처리되지 않습니다.

## SCS.getDeviceInfo

SCS.getDeviceInfo 함수는 컨트롤러 JavaScript 내의 장치 감지 논리의 사용자정의를 허용합니다. 이러한 감지는 페이지의 모바일 또는 반응형 버전을 제공해야 할지 여부를 결정하기 위해 사용됩니다.

SCS.getDeviceInfo 함수는 인수를 사용하지 않으며, 두 속성이 포함된 JavaScript 객체를 반환합니다.

- **isMobile**—현재 장치가 스마트 폰과 같은 모바일 클라이언트임을 나타내는 부울 속성입니다.
- **isIOS**—현재 장치가 iOS 기반 운영체제에서 실행되고 있음을 나타내는 부울 속성입니다.

사용자정의 컨트롤러가 SCS.getDeviceInfo를 무효화하지 않으면 내장된 기본 구현이 사용됩니다.

## 사용자정의하지 않아야 하는 컨트롤러 파일 섹션

컨트롤러 파일의 특정 섹션은 CDN을 사용하고 URL 캐싱 세그먼트를 정의하기 위해 Oracle Content Management 서버에서 처리됩니다. 이러한 내용은 아래 샘플에 인라인으로 표시되어 있습니다.

```
<!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>
```

## 사용자정의 컨트롤러 파일 이식성 허용을 위한 토큰 사용

사이트가 생성되면 사이트 이름이 컨트롤러 파일에 사용됩니다. 이렇게 해서 컨트롤러 파일이 이름이 지정된 사이트에 연결되고 해당 컨트롤러 파일을 사이트 간에 사용할 수 없게 됩니다.

컨트롤러 파일에 사용되는 것과 다른 이름으로 사이트에서 컨트롤러 파일을 사용하는 방식은 작동하지 않습니다. 이것은 또한 사용자정의 컨트롤러 파일이 포함된 사이트 템플릿에서 생성되는 사이트에도 적용됩니다. 다음과 같이 동적으로 평가되는 토큰은 컨트롤러 파일을 여러 사이트에 사용하도록 허용합니다.

토큰	설명
[!-\$SCS_SITE_PREFIX-]	현재 사이트의 사이트 접두어로 평가됩니다. 예를 들어 값에 다음이 포함될 수 있습니다.  /site/MySite/  또는  /site/authsite/MySecureSite/
[!-\$SCS_SITE_PATH-]	현재 제품 CDN 위치로 평가됩니다. 값에 후행 '/' 문자가 포함되지 않습니다. 예를 들어 값에 다음이 포함될 수 있습니다.  /site/MySite/_cache_0000  또는  /site/authsite/MySecureSite
[!-\$SCS_PRODUCT_PATH-]	캐시 키를 포함하여, 현재 제품 URL로 평가됩니다(해당되는 경우). 값에 후행 '/' 문자가 포함되지 않습니다. 예를 들어 값에 다음이 포함될 수 있습니다.  https://www.example.com/cdn/cec/v21.1.2.23  또는  https://www.example.com/cdn/cec/v21.1.3.18

## 사용자정의 컨트롤러 파일 샘플

샘플은 다음 사용자정의 제어 파일에 대해 제공됩니다.

- 사이트 접두어 변경
- 대기 그래픽 사용자정의
- 즐겨찾기 아이콘 사용자정의
- 비JavaScript Crawler에 대한 <noscript> 및 <meta> 태그 사용자정의
- JavaScript 파일 사전 인출
- 추가 마크업으로 사이트 소유권 확인
- 충분한 장치 감지
- controller.htm 이식성 향상을 위한 토큰 사용

### 사이트 접두어 변경

다음 샘플 컨트롤러 파일은 사이트를 여러 접두어로 제공할 수 있도록 preInitRendering 함수를 정의합니다.

#### 주:

이 샘플을 사용하려면 정의된 접두어에 응답하도록 CDN 또는 다른 프록시가 구성되어 있어야 합니다. 또한 함수에 정의된 추가 접두어가 브라우저 URL과 일치하지 않으면 기본 동작이 사용됩니다.

```
<!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>

```

## 대기 그래픽 사용자정의

다음 샘플 컨트롤러 파일은 네트워크 지연으로 인해 렌더링이 지연되었을 때 표시되는 대기 그래픽을 사용자정의합니다.



주:

이 샘플은 렌더링 중 개별 요청을 발생시키는 대신 이미지의 데이터 URL을 사용합니다.

```

<!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>

```

## 즐거찾기 아이콘 사용자정의

다음 샘플 컨트롤러 파일은 사이트의 사용자정의 즐거찾기 아이콘을 정의합니다. 이것은 브라우저가 렌더링된 페이지의 문서 객체 모델에서 즐거찾기 아이콘을 동적으로 로드하지 않는 경우에 유용합니다.

```

<!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>
<img id="scsWaitImage" style="display: none; margin-top: 5%; margin-

```



```
left: auto; margin-right: auto;" src="data:image/png;base64,..." />
</body></html>
```

## 비JavaScript Crawler에 대한 <noscript> 및 <meta> 태그 사용자정의

다음 샘플 컨트롤러 파일은 JavaScript를 처리하지 않는 Crawler에 대해 <noscript> 메시지 및 <meta> 태그를 사용자정의합니다.

### 주:

웹 사이트의 각 페이지에 대해 컨트롤러 파일이 제공되기 때문에 동일한 메타 태그가 모든 페이지에 표시됩니다.

```
<!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>
```

## JavaScript 파일 사전 인출

일부 브라우저에서는 현재 또는 이후 탐색에 필요한 리소스 선언이 허용됩니다. 컨트롤러 파일의 마크업은 사전 인출 또는 사전 로드되어야 하는 리소스를 선언할 수 있습니다.

### 주:

모든 OCE 웹 페이지에 require.js 및 renderer.js가 사용되기 때문에 이를 사전 로드 및 사전 인출하는 것이 좋습니다. 또한 href가 <link> 태그에 처음 나열된 경우에는 CDN에서 제공되도록 OCE 서버에서 해당 값이 수정됩니다.

```
<!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>
```

## 추가 마크업으로 사이트 소유권 확인

타사 Crawler 및 검색 엔진은 사이트 소유권 확인을 위해 추가적인 마크업이 필요할 수 있습니다. 이 마크업은 필요에 따라 컨트롤러 파일에 삽입될 수 있습니다.

```
<!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>
```

## 충분적인 장치 감지

컨트롤러에서 장치 감지 코드를 사용자정의하여 새 모바일 장치 또는 사용자정의된 장치의 사용자 에이전트 식별자를 인식할 수 있습니다.

 주:

- `getDeviceInfo` 함수는 모바일 장치가 사용되는지 여부를 확인하기 위해 컨트롤러에서 호출됩니다. 이 샘플은 기본 `getDeviceInfo` 호출을 무효화합니다.
- `isMobile` 속성은 페이지를 렌더링할 때 사용되는 레이아웃을 결정합니다.
- `isIOS` 속성은 맵 구성요소에 사용되는 제공자를 결정합니다.
- 내장된 기본 `isMobile` 테스트는 다음과 같습니다.

```
/Mobi|iPhone|iPod|BlackBerry|IEMobile|Opera Mini/  
i.test(userAgent) && !/iPad/i.test(userAgent)
```

- 내장된 기본 `isIOS` 테스트는 다음과 같습니다.

```
/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>
```

## controller.htm 이식성 향상을 위한 토큰 사용

기본적으로 사이트 이름이 `controller.html` 파일에 하드 코딩되어, 사이트 이름을 바꾸거나 여러 사이트 간에 동일한 사용자정의 컨트롤러를 재사용하는 것이 어렵습니다. 토큰을 사용하면 이러한 문제를 완화할 수 있습니다. 다음 샘플은 이식성 토큰 사용 방법을 보여줍니다.

 주:

샘플의 <link> 태그는 시연 목적으로만 사용됩니다. 기존 controller.html에는 이러한 항목이 포함되지 않습니다.

```
<!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>
```

# V부

## 기타 툴로 사이트용 개발

Oracle Content Management는 기타 타사 툴을 사용하여 Oracle Content Management 내부 또는 Oracle Content Management 외부에서 개발된 애플리케이션 및 환경과 상호작용할 수 있는 여러 가지 방법을 제공합니다.

### OCE Toolkit 사용

Oracle Content Management OCE Toolkit 및 SDK는 Oracle Content Management 저장소에서 관리되는 콘텐츠를 소비할 사용자정의 애플리케이션을 개발하도록 도와줍니다. 이 애플리케이션은 Oracle Content Management에서 또는 타사 툴을 사용하여 개발할 수 있습니다.

OCE Toolkit은 다음 작업을 도울 수 있습니다.

- 구성요소, 템플릿, 테마, 콘텐츠 레이아웃의 로컬 개발 및 테스트에 Oracle Content Management 인스턴스를 사용할 수 있도록 로컬 개발 환경을 설정합니다.
- 샘플에서 구성요소, 사이트 템플릿 및 콘텐츠 레이아웃을 생성하고 테스트 하네스에서 실행하고 탐색하며 Developer Cloud Service 환경에서 구성요소, 템플릿, 테마 및 콘텐츠 레이아웃을 개발합니다.
- 소스 관리 및 추가 개발을 위해 Oracle Content Management에서 생성된 구성요소와 사이트 템플릿을 Developer Cloud Service 프로젝트 및 환경으로 임포트합니다.
- Oracle Content Management에서 사용할 수 있도록 Developer Cloud Service 환경에서 구성요소, 템플릿 및 콘텐츠 레이아웃을 익스포트합니다.
- 기존 구성요소, 템플릿 또는 콘텐츠 레이아웃 복사
- 단위 테스트 쓰기
- 구성요소 최적화
- Oracle Content Management에 구성요소 및 템플릿 배치

추가 정보는 <https://github.com/oracle/content-and-experience-toolkit#readme>에서 찾아볼 수 있습니다.

### 환경 조정

Oracle Content Management 이외의 다른 툴을 사용하여 환경을 생성하는 경우 콘텐츠 생성자가 작업할 때 사이트 변경사항을 미리볼 수 있고 콘텐츠가 변경되거나 게시될 때 자동 구축이 트리거될 수 있도록 해당 환경에 Oracle Content Management 저장소를 연결할 수 있습니다. 이 환경 조정은 환경 관리 및 게시를 간소화하기 위해 콘텐츠 제공자와 사이트 개발자 간의 워크플로우 프로세스를 자동화합니다.

## OCE Toolkit으로 개발

OCE Toolkit은 Oracle Content Management용 사이트 템플릿, 테마, 사용자정의 구성요소 및 콘텐츠 레이아웃 개발을 돕습니다.

OCE Toolkit으로 고유의 개발 환경에서 작업하면서 Oracle Content Management의 자산 저장소, 파일 및 폴더를 사용할 수 있습니다. OCE Toolkit에는 테마와 콘텐츠 레이아웃을 포함한 사용자정의 구성요소 및 사이트 템플릿을 생성하고 개발하는 도구가 있습니다. 여기에는 빠르고 반복적인 개발을 위한 로컬 테스트 하네스와 시각용 샘플 단위 테스트가 포함됩니다.

다음 항목은 OCE Toolkit을 설정하고 로컬 시스템에서 또는 Developer Cloud Service 프로젝트로 개발하는 방법을 설명합니다.

- 로컬 시스템에서 OCE Toolkit 설정
- jQuery 3.5.x로 업그레이드
- Developer Cloud Service로 Oracle Content Management용 개발
- OCE Toolkit을 사용하여 테스트에서 운용으로 변경사항 전달
- 템플릿에서 사이트 생성 및 콘텐츠에 동일한 GUID 유지
- 분류법 임포트 및 익스포트
- 권장사항 임포트 및 익스포트
- OCE Toolkit을 사용하여 사용자정의 필드 편집기 개발
- 한 서버에서 다른 서버로 사이트 전송 또는 업데이트
- OCE Toolkit으로 사이트 페이지 인덱스화
- OCE Toolkit으로 다국어 사이트 인덱스화
- 쉬운 구성요소 개발을 위해 단순화된 구성요소 생성
- 사이트 컴파일로 사이트 페이지의 런타임 성능 향상
- Oracle Content Management 서버에서 새 사이트 또는 자산 번역 작업 생성
- 언어 서비스 제공자를 사용하여 사이트 번역

## 로컬 시스템에서 OCE Toolkit 설정

로컬 시스템의 GitHub에서 OCE Toolkit을 다운로드할 수 있습니다.

OCE Toolkit을 설정하려면 다음 지침을 따르십시오.

<https://github.com/oracle/content-and-experience-toolkit/blob/master/README.md>

설정을 완료하고 OCE Toolkit 사용을 준비합니다.

1. npm을 통해 종속성 설치
2. cec 명령행 유틸리티 사용

## 3. 로컬 테스트 하네스로 테스트

## npm을 통해 종속성 설치

npm(노드 패키지 관리자)을 사용하여 프로젝트에 대한 사이트 종속성을 설치합니다.

인터넷 액세스에 프록시를 사용하는 경우 `npm config` 명령으로 npm의 프록시를 설정합니다. <https://docs.npmjs.com/misc/config>를 참조하십시오. bower의 프록시를 설정하려면 <https://bower.io/docs/config/>를 참조하십시오.

 주:

Node.js 8.0.0 이상(<https://nodejs.org/>)이 로컬 컴퓨터에 설치되었는지 확인하십시오.

## cec 명령행 유틸리티 사용

cec 교차 플랫폼 명령행 유틸리티는 템플릿 및 구성요소를 생성하고 관리하기 위한 명령을 제공합니다.

cec 명령행 유틸리티를 사용하기 전에 Oracle Content Management 서버 접속이 필요한 명령을 위해 로컬 개발 환경에 소스를 생성하십시오.

통합 도움말은 명령 실행을 위한 정보를 예제와 함께 제공합니다. cec 통합 도움말을 보려면 명령행에 `cec` 명령을 입력할 수 있습니다.

- 터미널 창에서 `cec` 디렉토리로 이동합니다.
- `cec`를 입력하여 `cec` 명령을 나열합니다.

```
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]
  cec upload-folder <path>              Uploads folder to
OCM server.                               [alias: ulfd]
```

cec delete-folder <path> server.	Deletes folder on OCM [alias: ]
cec download-file <file> from OCM server.	Downloads file <file> [alias: dlf]
cec upload-file <file> OCM server.	Uploads file <file> to [alias: ulf]
cec delete-file <file> server.	Deletes file on OCM [alias: ]
<b>Components</b>	
cec create-component <name> <name>.	Creates the component [alias: cc]
cec copy-component <source> [<destination>] component named <source> to <destination>.	Copies an existing [alias: cpc]
cec import-component <zip> from <zip>.	Imports a component [alias: ic]
cec export-component <name> <name> as a zip file.	Exports the component [alias: ec]
cec download-component <names> components <names> from the OCM server.	Downloads the [alias: dlcp]
cec upload-component <names> <names> to the OCM server.	Uploads the components [alias: ulcp]
cec control-component <action> <action> on components on OCM server.	Performs action [alias: ctcp]
cec share-component <name> users and groups on OCM server.	Shares component with [alias: sc]
cec unshare-component <name> access to a component on OCM server.	Deletes user or group [alias: usc]
<b>Templates</b>	
cec create-template <name> <name>.	Creates the template [alias: ct]
cec create-template-from-site <name> <name> from site <site> on the OCM server.	Creates the template [alias: ctfs]
cec download-template <name> <name> from the OCM server.	Downloads the template [alias: dlt]
cec compile-template <source> within the template.	Compiles the site [alias: cmpt]
cec copy-template <source> [<destination>] template named <source> to <destination>.	Copies an existing [alias: cpt]
cec import-template <zip> <zip>.	Imports a template from [alias: it]
cec export-template <name> <name> as a zip file.	Exports the template [alias: et]
cec upload-template <name> <name> to the OCM server.	Uploads the template [alias: ult]
cec delete-template <name> <name> on the OCM server.	Deletes the template [alias: ]
cec share-template <name> users and groups on OCM server.	Shares template with [alias: stm]
cec unshare-template <name> access to a template on OCM server.	Deletes user or group [alias: ustm]
cec update-template <action> local template.	Performs action on a [alias: ut]
cec describe-template <name>	Describes the template



<name> package. cec create-template-report <name> usage report for the template <name> package.	[alias: dst] Generates an asset [alias: cctr]
 Themes	
cec add-component-to-theme <component> to a theme.	Adds a component [alias: actt]
cec remove-component-from-theme <component> component from a theme. rcft]	Removes a [alias:
cec control-theme <action> <action> on theme on OCM server.	Performs action [alias: ctt]
cec share-theme <name> users and groups on OCM server.	Shares theme with [alias: sth]
cec unshare-theme <name> group access to a theme on OCM server.	Deletes user or [alias: usth]
 Sites	
cec create-site <name> Site <name>.	Creates Enterprise [alias: cs]
cec copy-site <name> Site <name>.	Copies Enterprise [alias: cps]
cec update-site <name> Site <name>.	Update Enterprise [alias: us]
cec transfer-site <name> from one OCM server to another.	Transfers a site [alias: ts]
cec transfer-site-content <name> transfer site content from one OCM server to another.	Creates scripts to [alias: tsc]
cec validate-site <name> <name>.	Validates site [alias: vs]
cec control-site <action> <action> on site on OCM server.	Performs action [alias: cts]
cec share-site <name> users and groups on OCM server.	Shares site with [alias: ss]
cec unshare-site <name> group access to a site on OCM server.	Deletes user or [alias: uss]
cec get-site-security <name> on OCM server.	Gets site security [alias: gss]
cec set-site-security <name> on OCM server.	Sets site security [alias: sss]
cec index-site <site> content of site <site> on OCM server.	Index the page [alias: is]
cec create-site-map <site> for site <site> on OCM server.	Creates a site map [alias: csm]
cec create-rss-feed <site> for site <site> on OCM server.	Creates RSS feed [alias: crf]
cec create-asset-report <site> usage report for site <site> on OCM server.	Generates an asset [alias: car]
cec upload-static-site-files <path> render statically from a site on OCM server.	Uploads files to [alias: ulss]
cec download-static-site-files <site> static files from a site on OCM server. dlss]	Downloads the [alias:
cec delete-static-site-files <site> files from a site on OCM server.	Deletes the static [alias: ]

cec refresh-prerender-cache <site> cache for a site on OCM server.	Refreshes pre-render [alias: rpc]
cec migrate-site <site> OCI IC server to EC server.	Migrates a site from [alias: ms]
<b>Assets</b>	
cec download-content OCM server.	Downloads content from [alias: dlc]
cec upload-content <name> to a repository on OCM server.	Uploads local content [alias: ulc]
cec control-content <action> <action> on channel items on OCM server.	Performs action [alias: ctct]
cec transfer-content <repository> transfer content from one OCM server to another.	Creates scripts to [alias: tc]
cec list-assets server.	Lists assets on OCM [alias: la]
cec create-digital-asset asset	Creates digital [alias: cda]
cec update-digital-asset <id> asset	Updates digital [alias: uda]
cec copy-assets <repository> another repository on OCM server.	Copies assets to [alias: ca]
cec create-asset-usage-report <assets> usage report for assets on OCM server.	Generates an asset [alias: caur]
<b>Content</b>	
cec create-repository <name> OCM server.	Creates a repository on [alias: cr]
cec control-repository <action> <action> on repositories on OCM server.	Performs action [alias: ctr]
cec share-repository <name> users and groups on OCM server.	Shares repository with [alias: sr]
cec unshare-repository <name> access to a repository on OCM server.	Deletes user or group [alias: usr]
cec create-channel <name> OCM server.	Creates a channel on [alias: cch]
cec share-channel <name> users and groups on OCM server.	Shares channel with [alias: sch]
cec unshare-channel <name> access to a channel on OCM server.	Deletes user or group [alias: usch]
cec create-localization-policy <name> policy on OCM server.	Creates a localization [alias: clp]
cec list-server-content-types from server.	Lists all content types [alias: lsct]
cec share-type <name> and groups on OCM server.	Shares type with users [alias: st]
cec unshare-type <name> access to a type on OCM server.	Deletes user or group [alias: ust]
cec download-type <name> OCM server.	Downloads types from [alias: dltp]
cec upload-type <name> server.	Uploads types to OCM [alias: ultp]
cec update-type <action> <action> on a type	Performs action [alias: utp]
cec download-recommendation <name>	Downloads a

```

recommendation from the OCM server.                                [alias:
dlr]
    cec upload-recommendation <name>                               Uploads a
recommendation to the OCM server.
[alias:  ulr]
    cec create-contentlayout <name>                                Creates a content
layout based on a content type.                                    [alias:  ccl]
    cec add-contentlayout-mapping <contentlayout>                 Creates content
type and content layout mapping.                                  [alias:  aclm]
    cec remove-contentlayout-mapping <contentlayout>             Removes a content
layout mapping.                                                  [alias:  rclm]
    cec add-field-editor <name>                                    Adds a field
editor to a field in a content type.                              [alias:
afe]
    cec remove-field-editor <name>                                Removes a field
editor from a field in a content type.                            [alias:  rfe]
    cec migrate-content <name>                                    Migrates content
from OCI IC server to EC server.                                  [alias:  mc]
    cec compile-content                                          Compiles the
content items generating HTML renditions.                         [alias:
cmpc]
    cec upload-compiled-content <path>                            Uploads the
compiled content to OCM server.                                    [alias:
ulcc]

Taxonomies
    cec download-taxonomy <name>                                   Downloads a
taxonomy from OCM server.                                         [alias:
dltx]
    cec upload-taxonomy <taxonomy>                                Uploads a taxonomy
to OCM server.                                                    [alias:  ultx]
    cec control-taxonomy <action>                                 Performs action on
taxonomy on OCM server.                                           [alias:  cttx]

Translation
    cec list-translation-jobs                                     Lists translation
jobs.                                                              [alias:  ltj]
    cec create-translation-job <name>                             Creates a
translation job <name> for a site on OCM server.
[alias:  ctj]
    cec download-translation-job <name>                           Downloads
translation job <name> from OCM server.
[alias:  dtj]
    cec submit-translation-job <name>                             Submits
translation job <name> to translation connection <connection>.
[alias:  stj]
    cec refresh-translation-job <name>                            Refreshes
translation job <name> from translation connection.
[alias:  rtj]
    cec ingest-translation-job <name>                             Gets translated
job <name> from translation connection and ingest.                [alias:  itj]
    cec upload-translation-job <name>                             Uploads
translation job <name> to OCM server.
[alias:  utj]
    cec create-translation-connector <name>                       Creates

```

```
translation connector <name>.                                [alias:
ctc]
  cec start-translation-connector <name>                    Starts translation
connector <name>.                                          [alias: stc]
  cec register-translation-connector <name>                  Registers a translation
connector.                                                  [alias: rtc]
```

#### Groups

```
  cec create-group <name>                                    Creates an OCM group on
OCM server.                                                [alias: cg]
  cec delete-group <name>                                    Deletes an OCM group on
OCM server.                                                [alias: ]
  cec add-member-to-group <name>                             Adds users and groups
to an OCM group on OCM server.                             [alias: amtg]
  cec remove-member-from-group <name>                       Removes users and
groups from an OCM group on OCM server.                   [alias: rmfg]
```

#### Local Environment

```
  cec create-encryption-key <file>                          Create an encryption
key to encrypt/decrypt password for servers.              [alias: cek]
  cec register-server <name>                                 Registers a OCM
server.                                                    [alias: rs]
  cec set-oauth-token <token>                               Set OAuth token for a
registered server.                                        [alias: sot]
  cec list                                                   Lists local or server
resources.                                                [alias: l]
  cec execute-get <endpoint>                                Makes an HTTP GET
request to a REST API endpoint on OCM server              [alias: exeg]
  cec install                                               Creates source
tree.                                                      [alias: i]
  cec develop                                               Starts a test
server.                                                    [alias: d]
  cec sync-server                                           Starts a sync
server.                                                    [alias: scs]
  cec webhook-server                                        Starts a webhook
server.                                                    [alias: whs]
```

#### Options:

```
--version, -v Show version number [boolean]
--help, -h Show Help
```

```
-----
cec create-folder
-----
```

```
Usage: cec create-folder <name>
```

Create a folder or folder hierarchy on OCM server. Specify the server with -s <server> or use the one specified in cec.properties file.

#### Options:

```
--help, -h Show Help
--server, -s <server> The registered OCM server
```

#### Examples:

```
cec create-folder Projects                                Creates folder Projects under the Home
```

```

folder
  cec create-folder Projects/Blogs  Creates folder Projects under the
  Home folder and folder Blogs under Projects
  cec create-folder Projects -s UAT  Creates folder Projects under the
  Home folder on the registered server UAT

```

```

-----
cec share-folder
-----

```

Usage: cec share-folder <name>

Shares folder with users and groups on OCM server and assign a role. Specify the server with -s <server> or use the one specified in cec.properties file. The valid roles are

```

  manager
  contributor
  downloader
  viewer

```

Options:

```

  --help, -h      Show Help
  --users, -u     The comma separated list of user names
  --groups, -g    The comma separated list of group names
  --role, -r      The role [manager | contributor | downloader | viewer]
  to assign to the users or groups [required]
  --server, -s   <server> The registered OCM server

```

Examples:

```

  cec share-folder Projects/Blogs -u user1,user2 -r manager
  Share folder Projects/Blogs with user user1 and user2 and assign
  Manager role to them
  cec share-folder Projects/Blogs -u user1,user2 -g group1 -r manager
  Share folder Projects/Blogs with user user1, user2 and group group1 and
  assign Manager role to them
  cec share-folder Projects/Blogs -g group1,group2 -r manager
  Share folder Projects/Blogs with group group1 and group2 and assign
  Manager role to them
  cec share-folder Projects/Blogs -u user1,user2 -r manager -s UAT
  Share folder Projects/Blogs with user user1 and user2 and assign
  Manager role to them on the registered server UAT

```

```

-----
cec unshare-folder
-----

```

Usage: cec unshare-folder <name>

Deletes user or group access to a shared folder on OCM server. Specify the server with -s <server> or use the one specified in cec.properties file.

Options:

```

  --help, -h      Show Help
  --users, -u     The comma separated list of user names

```

```
--groups, -g The comma separated list of group names
--server, -s <server> The registered OCM server
```

Examples:

```
cec unshare-folder Projects/Blogs -u user1,user2
cec unshare-folder Projects/Blogs -g group1,group2
cec unshare-folder Projects/Blogs -u user1,user2 -g group1,group2
cec unshare-folder Projects/Blogs -u user1,user2 -s UAT
```

```
-----
cec list-folder
-----
```

Usage: cec list-folder <path>

Displays folder and all its content on OCM server. Specify the server with -s <server> or use the one specified in cec.properties file.

Options:

```
--help, -h Show Help
--server, -s The registered OCM server
```

Examples:

```
cec list-folder Releases/1
cec list-folder Releases/1 -s UAT
cec list-folder site:blog1
cec list-folder theme:blog1Theme
cec list-folder component:Comp1/assets
```

```
-----
cec download-folder
-----
```

Usage: cec download-folder <path>

Downloads folder and all its content from OCM server. Specify the server with -s <server> or use the one specified in cec.properties file. Optionally specify -f <folder> to save the folder on the local system.

Options:

```
--help, -h Show Help
--folder, -f <folder> Local folder to save the folder on OCM server
--server, -s <server> The registered OCM server
```

Examples:

```
cec download-folder Releases/1 Downloads folder
Releases/1 from OCM server and save to local folder src/documents/
cec download-folder / Downloads all
documents from OCM server and save to local folder src/documents/
cec download-folder Releases/1 -s UAT Downloads folder
Releases/1 from the registered server UAT and save to local folder src/
documents/
cec download-folder Releases/1 -f ~/Downloads Downloads folder
Releases/1 from OCM server and save to local folder ~/Download/
cec download-folder Releases/1 -f . Downloads folder
Releases/1 from OCM server and save to the current local folder
cec download-folder site:blog1 -f ~/Downloads/blog1Files Downloads all
```

```
files of site blog1 and save to local folder ~/Download/blog1Files
cec download-folder theme:blog1Theme Downloads
all files of theme blog1Theme and save to local folder src/documents/
blog1Theme/
cec download-folder component:Comp1/assets Downloads
all files in folder assets of component Comp1 and save to local folder
src/documents/Comp1/assets/
```

```
-----
cec upload-folder
-----
```

Usage: cec upload-folder <path>

Uploads folder and all its content to OCM server. Specify the server with -s <server> or use the one specified in cec.properties file. Optionally specify -f <folder> to set the parent folder on OCM server.

Options:

```
--help, -h Show Help
--folder, -f <folder> The parent folder on OCM server
--server, -s <server> The registered OCM server
```

Examples:

```
cec upload-folder ~/Downloads/docs
Uploads all content from ~/Downloads/docs to folder docs on the server
cec upload-folder ~/Downloads/docs/
Uploads all content from ~/Downloads/docs to the Home folder on the
server
cec upload-folder ~/Downloads/docs -f Mydoc
Uploads all content from ~/Downloads/docs to folder Mydoc/docs on the
server
cec upload-folder ~/Downloads/docs/ -f Mydoc
Uploads all content from ~/Downloads/docs to folder Mydoc on the server
cec upload-folder ~/Downloads/docs -s UAT
Uploads all content from ~/Downloads/docs to folder docs on the
registered server UAT
cec upload-folder ~/Downloads/docs/ -f site:blog1/settings/misc
Uploads all content from ~/Downloads/docs to folder settings/misc of
site blog1
cec upload-folder ~/Downloads/docs -f theme:blog1Theme
Uploads all content from ~/Downloads/docs to folder docs of theme
blog1Theme
cec upload-folder ~/Downloads/docs -f component:Comp1
Uploads all content from ~/Downloads/docs to folder docs of component
Comp1
```

```
-----
cec delete-folder
-----
```

Usage: cec delete-folder <path>

Deletes folder and all its content on OCM server. Specify the server with -s <server> or use the one specified in cec.properties file. Optionally specify -p to permanently delete the folder.

Options:  
 --help, -h Show Help  
 --server, -s <server> The registered OCM server  
 --permanent, -p Delete the folder permanently

Examples:  
 cec delete-folder Import/docs  
 cec delete-folder Import/docs -s UAT  
 cec delete-folder Import/docs -p  
 cec delete-folder site:blog1/docs  
 cec delete-folder theme:blog1Theme/docs  
 cec delete-folder component:Comp1/docs

-----  
 cec download-file  
 -----

Usage: cec download-file <file>

Downloads file <file> from OCM server. Specify the server with -s <server> or use the one specified in cec.properties file. Optionally specify -f <folder> to save the file on the local system.

Options:  
 --help, -h Show Help  
 --folder, -f <folder> Local folder to save the file  
 --server, -s <server> The registered OCM server

Examples:  
 cec download-file Releases/Projects.pdf Downloads  
 the file from OCM server and save to local folder src/documents/  
 cec download-file Releases/Projects.pdf -s UAT Downloads  
 the file from the registered server UAT and save to local folder src/  
 documents/  
 cec download-file Releases/Projects.pdf -f ~/Downloads Downloads  
 the file from OCM server and save to local folder ~/Download/  
 cec download-file Releases/Projects.pdf -f . Downloads  
 the file from OCM server and save to the current local folder  
 cec download-file site:blog1/siteinfo.json Downloads  
 the file from folder blog1 and save to local folder src/documents/blog1  
 cec download-file theme:blog1Theme/designs/default/design.css Downloads  
 the css file from folder designs/default of theme blog1Theme and save to  
 local folder src/documents/blog1Theme/designs/default/  
 cec download-file component:Comp1/assets/render.js Downloads  
 the js file from folder assets of component Comp1 and save to local folder  
 src/documents/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 Sitel -t SitelCopy           Copies a
standard site
cec copy-site Sitel -t SitelCopy -r Repository1   Copies an
enterprise site
cec copy-site Sitel -t SitelCopy -r Repository1 -x sitelc  Copies an
enterprise site and sets the site prefix to sitelc
```

```
-----
cec update-site
-----
```

Usage: cec update-site <name>

Update Enterprise Site on OCM server using the content from the template. Specify the server with -s <server> or use the one specified in cec.properties file.

Options:

```
--help, -h          Show Help
--template, -t      <template> Template [required]
--excludecontenttemplate, -x  Exclude content template
--server, -s        <server> The registered OCM server
```

Examples:

```
cec update-site Sitel -t Templatel       Updates site Sitel using the
content from template Templatel
cec update-site Sitel -t Templatel -x    Updates site Sitel using the
content from template Templatel excluding the "Content Template"
```

```
-----
cec transfer-site
-----
```

Usage: cec transfer-site <name>

Transfers a site from one OCM server to another. By default all assets are transferred, optionally specify -p to transfer only published assets. Specify the source server with -s <server> and the destination server with -d <destination>. If the site contains assets from other repositories, optionally provide the repository mapping otherwise those assets will not be transferred.

Options:

```
--help, -h          Show Help
--server, -s        The registered OCM server the site is from
[required]
--destination, -d   The registered OCM server to create or
update the site [required]
--repository, -r    Repository, required for creating
enterprise site
```

```

--localizationPolicy, -l Localization policy, required for creating
enterprise site
--sitePrefix, -f Site prefix
--publishedassets, -p The flag to indicate published assets only
--repositorymappings, -m The repositories for assets from other
repositories
--excludecontent, -x Exclude content
--excludecomponents, -e Exclude components
--excludetheme, -c Exclude theme
--excludetype, -t Exclude content types
--includestaticfiles, -i Include site static files
--suppressgovernance, -g Suppress site governance controls

```

Examples:

```

cec transfer-site Site1 -s DEV -d UAT -r Repository1 -l
L10NPolicy1
Creates site Site1 on server UAT based on site Site1 on server DEV
cec transfer-site Site1 -s DEV -d UAT -r Repository1 -l L10NPolicy1 -
p
Creates site
Site1 on server UAT based on site Site1 on server DEV with published assets
cec transfer-site Site1 -s DEV -d UAT -r Repository1 -l L10NPolicy1 -
x
Creates site
Site1 on server UAT based on site Site1 on server DEV without content
cec transfer-site Site1 -s DEV -d UAT -r Repository1 -l L10NPolicy1 -
e
Creates site
Site1 on server UAT based on site Site1 on server DEV without transferring
components to server UAT
cec transfer-site Site1 -s DEV -d UAT -r Repository1 -l L10NPolicy1 -e -
c
Creates site Site1
on server UAT based on site Site1 on server DEV without transferring
components and theme to server UAT
cec transfer-site Site1 -s DEV -d UAT -r Repository1 -l L10NPolicy1 -m
"Shared Images:Shared Images,Shared Video:Shared Video" Creates site Site1
on server UAT based on site Site1 on server DEV and transfer the assets
from repository Shared Images and Shared Video
cec transfer-site Site1 -s DEV -d UAT -r Repository1 -l L10NPolicy1 -
i
Creates site
Site1 on server UAT based on site Site1 on server DEV with static files from
DEV
cec transfer-site Site1 -s DEV -d
UAT
Updates site Site1 on server UAT based on site Site1 on server DEV
cec transfer-site StandardSite1 -s DEV -d
UAT
Creates standard site on server UAT based on site StandardSite1 on server DEV

-----
cec transfer-site-content
-----

Usage: cec transfer-site-content <name>

```

Creates scripts to transfer Enterprise Site content from one OCM server to another. This command is used to transfer large number of content items and the items are transferred in batches. By default the scripts will not be executed by this command. By default all assets are transferred, optionally



specify `-p` to transfer only published assets. Specify the source server with `-s <server>` and the destination server with `-d <destination>`. Optionally specify `-n` for the number of items in each batch, defaults to 500. If the site contains assets from other repositories, optionally provide the repository mapping otherwise those assets will not be transferred.

Options:

<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:

<code>--help, -h</code>	Show Help
<code>--server, -s</code>	<code>&lt;server&gt;</code> The registered OCM server

Examples:

```
cec validate-site Site1           Validate site Site1 on the server
specified in cec.properties file
cec validate-site Site1 -s UAT   Validate site Site1 on the registered
server UAT
```

```
-----
cec control-site
-----
```

Usage: cec control-site <action>

Perform <action> on site on OCM server. Specify the site with -s <site>. Specify the server with -r <server> or use the one specified in cec.properties file. The valid actions are

```
publish
unpublish
bring-online
take-offline
```

Options:

```
--help, -h           Show Help
--site, -s           <site> Site [required]
--usedcontentonly, -u Publish used content only
--compilesite, -c    Compile site after publish
--staticonly, -t     Only publish site static files
--fullpublish, -f    Do a full publish
--server, -r         <server> The registered OCM server
```

Examples:

```
cec control-site publish -s Site1           Publish site Site1 on the
server specified in cec.properties file
cec control-site publish -s Site1 -u       Publish the site and all
assets added to the site's pages
cec control-site publish -s Site1 -c       Compile and publish site
Site1
cec control-site publish -s Site1 -t       Only publish the static
files of site Site1
cec control-site publish -s Site1 -f       Do a full publish of Site1
cec control-site publish -s Site1 -r UAT   Publish site Site1 on the
registered server UAT
cec control-site unpublish -s Site1 -r UAT Unpublish site Site1 on the
registered server UAT
cec control-site bring-online -s Site1 -r UAT Bring site Site1 online on
the registered server UAT
cec control-site take-offline -s Site1 -r UAT Take site Site1 offline on
the registered server UAT
```

```
-----
cec share-site
-----
```

Usage: cec share-site <name>

Shares site with users and groups on OCM server and assign a role. Specify

the server with `-s <server>` or use the one specified in `cec.properties` file. The valid roles are

```
manager
contributor
downloader
viewer
```

Options:

```
--help, -h      Show Help
--users, -u     The comma separated list of user names
--groups, -g    The comma separated list of group names
--role, -r      The role [manager | contributor | downloader | viewer]
to assign to the users or groups [required]
--server, -s    <server> The registered OCM server
```

Examples:

```
cec share-site Site1 -u user1,user2 -r manager
Share site Site1 with user user1 and user2 and assign Manager role to
them
cec share-site Site1 -u user1,user2 -g group1,group2 -r manager
Share site Site1 with user user1 and user2 and group group1 and group2
and assign Manager role to them
cec share-site Site1 -u user1,user2 -r manager -s UAT
Share site Site1 with user user1 and user2 and assign Manager role to
them on the registered server UAT
```

```
-----
cec unshare-site
-----
```

Usage: `cec unshare-site <name>`

Deletes user or group access to a site on OCM server. Specify the server with `-s <server>` or use the one specified in `cec.properties` file.

Options:

```
--help, -h      Show Help
--users, -u     The comma separated list of user names
--groups, -g    The comma separated list of group names
--server, -s    <server> The registered OCM server
```

Examples:

```
cec unshare-site Site1 -u user1,user2
cec unshare-site Site1 -u user1,user2 -g group1,group2
cec unshare-site Site1 -u user1,user2 -s UAT
```

```
-----
cec get-site-security
-----
```

Usage: `cec get-site-security <name>`

Gets site security on OCM server. Specify the server with `-s <server>` or use the one specified in `cec.properties` file.

```
Options:
  --help, -h      Show Help
  --server, -s <server> The registered OCM server
```

```
Examples:
  cec get-site-security Site1
  cec get-site-security Site1 -s UAT
```

```
-----
cec set-site-security
-----
Usage: cec set-site-security <name>
```

Makes the site publicly available to anyone, restrict the site to registered users, or restrict the site to specific users. Specify the server with `-r <server>` or use the one specified in `cec.properties` file. Optionally specify `-a <access>` to set who can access the site. The valid group names are

```
Cloud users
Visitors
Service users
Specific users
```

```
Options:
  --help, -h      Show Help
  --signin, -s    If require sign in to access site: yes | no [required]
  --access, -a    The comma separated list of group names
  --addusers, -u  The comma separated list of users to access the site
  --deleteusers, -d The comma separated list of users to remove access from
the site
  --server, -r    <server> The registered OCM server
```

```
Examples:
  cec set-site-security Site1 -s no
make the site publicly available to anyone
  cec set-site-security Site1 -s no -r UAT
make the site publicly available to anyone on server UAT
  cec set-site-security Site1 -s yes
Require everyone to sign in to access this site and any authenticated user
can access
  cec set-site-security Site1 -s yes -a "Visitors,Service users"
Require everyone to sign in to access this site and all service visitors and
users can access
  cec set-site-security Site1 -s yes -a "Specific users" -u user1,user2
Require everyone to sign in to access this site and only user1 and user2 can
access
  cec set-site-security Site1 -s yes -d user1
Remove user1's access from the site
```

```
-----
cec index-site
-----
Usage: cec index-site <site>
```

Creates content item for each page with all text on the page. If the page index content item already exists for a page, updated it with latest text on the page. Specify `-c <contenttype>` to set the page index content type. Optionally specify `-p` to publish the page index items after creation or update. Specify the server with `-s <server>` or use the one specified in `cec.properties` file.

Options:

```
--help, -h          Show Help
--contenttype, -c  <contenttype> page index content type
--publish, -p      publish page index items
--server, -s       <server> The registered OCM server
```

Examples:

```
cec index-site Site1 -c PageIndex
cec index-site Site1 -c PageIndex -p
cec index-site Site1 -c PageIndex -s UAT
```

```
-----
cec create-site-map
-----
```

Usage: `cec create-site-map <site>`

Creates a site map for site on OCM server. Specify the server with `-s <server>` or use the one specified in `cec.properties` file. Optionally specify `-p` to upload the site map to OCM server after creation. Optionally specify `-c <changefreq>` to define how frequently the page is likely to change. Optionally specify `-t <toppagepriority>` as the priority for the top level pages. Also optionally specify `<file>` as the file name for the site map.

The valid values for `<changefreq>` are:

```
always
hourly
daily
weekly
monthly
yearly
never
auto
```

Options:

```
--help, -h          Show Help
--url, -u           <url> Site URL [required]
--changefreq, -c   How frequently the page is likely to
change.
--file, -f         Name of the generated site map file
--languages, -l    <languages> The comma separated list
of languages used to create the site map
--publish, -p      Upload the site map to OCM server
after creation
--toppagepriority, -t Priority for the top level pages, a
decimal number between 0 and 1
```

```
--server, -s                <server> The registered OCM server
--newlink, -n              Generate new 19.3.3 detail page link
--noDefaultDetailPageLink, -o Do not generate detail page link for items/
content lists that use the default detail page
```

Examples:

```
cec create-site-map Site1 -u http://www.example.com/site1
cec create-site-map Site1 -u http://www.example.com/site1 -s UAT
cec create-site-map Site1 -u http://www.example.com/site1 -t 0.9
cec create-site-map Site1 -u http://www.example.com/site1 -f sitemap.xml
cec create-site-map Site1 -u http://www.example.com/site1 -p
cec create-site-map Site1 -u http://www.example.com/site1 -c weekly -p
cec create-site-map Site1 -u http://www.example.com/site1 -l de-DE,it-IT
```

```
-----
cec create-rss-feed
-----
```

Usage: cec create-rss-feed <site>

Creates RSS feed for site <site> on OCM server. Specify the server with -s <server> or use the one specified in cec.properties file. Optionally specify -x <template> to specify the RSS template. Optionally specify -p to upload the RSS feed to OCM server after creation.

Options:

```
--help, -h                Show Help
--url, -u                 <url> Site URL [required]
--query, -q              Query for content items [required]
--limit, -l              The limit of the items returned from the query
[required]
--orderby, -o            The order by for the query [required]
--language, -i          The language for the query
--template, -x          The RSS xml template
--javascript, -j        Javascript file that contains functions to process
Mustache data
--title, -t             The RSS feed title
--description, -d       The RSS feed description
--ttl                  How long the data will last in number of minutes
--file, -f             Name of the generated RSS feed file
--publish, -p          Upload the RSS feed to OCM server after creation
--server, -s           <server> The registered OCM server
--newlink, -n         Generate new 19.3.3 detail page link
```

Examples:

```
cec create-rss-feed Site1 -u http://www.example.com/site1 -q 'type eq
"BlogType"' -l 10 -o name:asc -t "Blog RSS"
cec create-rss-feed Site1 -u http://www.example.com/site1 -q 'type eq
"BlogType"' -l 10 -o name:asc -t "Blog RSS" -x ~/Files/RSSTemplate.xml
cec create-rss-feed Site1 -u http://www.example.com/site1 -q 'type eq
"BlogType"' -l 10 -o name:asc -t "Blog RSS" -x ~/Files/RSSTemplate.xml -i fr-
FR -f rssfrFR.xml
```

```
-----
cec create-asset-report
-----
```

Usage: cec create-asset-report <site>

Generates an asset usage report for site <site> on OCM server. Specify the server with -s <server> or use the one specified in cec.properties file. Optionally specify -o to save the report to a json file.

Options:

--help, -h Show Help  
--output, -o Output the report to a JSON file  
--server, -s The registered OCM server

Examples:

```
cec create-asset-report Site1
cec create-asset-report Site1 -s UAT
cec create-asset-report Site1 -o                               The
report will be saved to Site1AssetUsage.json at the current local
location
cec create-asset-report Site1 -o ~/Documents                  The
report will be saved to ~/Documents/Site1AssetUsage.json
cec create-asset-report Site1 -o ~/Documents/Site1Report.json The
report will be saved to ~/Documents/Site1Report.json
```

-----  
cec upload-static-site-files  
-----

Usage: cec upload-static-site-files <path>

Uploads files to render statically from a site on OCM server. Specify the site <site> on the server. Specify the server with -r <server> or use the one specified in cec.properties file.

Options:

--help, -h Show Help  
--site, -s The site on OCM server [required]  
--server, -r The registered OCM server

Examples:

```
cec upload-static-site-files ~/Documents/localBlog -s BlogSite
cec upload-static-site-files ~/Documents/localBlog -s BlogSite -r UAT
```

-----  
cec download-static-site-files  
-----

Usage: cec download-static-site-files <site>

Downloads the static files from a site on OCM server. Specify the server with -s <server> or use the one specified in cec.properties file. Optionally specify -f <folder> to save the files on the local system.

Options:

--help, -h Show Help  
--folder, -f <folder> Local folder to save the static files  
--server, -s The registered OCM server

## Examples:

```
cec download-static-site-files BlogSite
Download the files and save to local folder src/documents/BlogSite/static
cec download-static-site-files BlogSite -f ~/Documents/BlogSite/static
Download the files and save to local folder ~/Documents/BlogSite/static
cec download-static-site-files BlogSite -s UAT
```

```
-----
cec delete-static-site-files
-----
```

Usage: cec delete-static-site-files <site>

Deletes the static files from a site on OCM server. Specify the server with -s <server> or use the one specified in cec.properties file.

## Options:

```
--help, -h    Show Help
--server, -s  The registered OCM server
```

## Examples:

```
cec delete-static-site-files BlogSite
cec delete-static-site-files BlogSite -s UAT
```

```
-----
cec refresh-prerender-cache
-----
```

Usage: cec refresh-prerender-cache <site>

Refreshes pre-render cache for a site on OCM server. Specify the server with -s <server> or use the one specified in cec.properties file.

## Options:

```
--help, -h    Show Help
--server, -s  The registered OCM server
```

## Examples:

```
cec refresh-prerender-cache BlogSite
cec refresh-prerender-cache BlogSite -s UAT
```

```
-----
cec migrate-site
-----
```

Usage: cec migrate-site <site>

Migrates a site from OCI IC server to EC server. Specify the IC server with -s <server> and the EC server with -d <destination>.

## Options:

```
--help, -h          Show Help
--server, -s        The registered IC server the site is from
--destination, -d  The registered EC server to create the site [required]
--repository, -r   Repository [required]
--template, -t     The site template
--name, -n         Site name
--description, -p  Site description
```



--sitePrefix, -x Site Prefix

Examples:

```
cec migrate-site Site1 -s ICServer -d ECServer -r
Repol Migrates site Site1 from ICServer to
ECServer
cec migrate-site Site1 -s ICServer -d ECServer -r Repol -n
newSite Migrates site Site1 from ICServer to ECServer and
rename to newSite
cec migrate-site Site1 -d ECServer -t ~/Documents/Site1Template.zip -
r Repol Migrates site Site1 to ECServer with template
Site1Template.zip from IC server
```

-----  
cec download-content  
-----

Usage: cec download-content

Downloads content from OCM server. By default all assets are downloaded, optionally specify -p to download only published assets. Specify the server with -s <server> or use the one specified in cec.properties file.

Options:

```
--help, -h Show Help
--publishedassets, -p The flag to indicate published assets only
--collection, -c Collection name
--repository, -r Repository name, required when <collection> is
specified
--query, -q Query to fetch the assets
--assets, -a The comma separated list of asset GUIDS
--assetsfile, -f The file with an array of asset GUIDS
--name, -n The name for this download, default to the
channel or repository name
--server, -s The registered OCM server
```

Examples:

```
cec download-content
Site1Channel
Download all assets in channel Site1Channel and save to local folder
src/content/Site1Channel
cec download-content Site1Channel -n
Site1Assets Download all
assets in channel Site1Channel and save to local folder src/content/
Site1Assets
cec download-content Site1Channel -
p Download
published assets in channel Site1Channel
cec download-content Site1Channel -s
UAT Download all
assets in channel Site1Channel on server UAT
cec download-content Site1Channel -q 'fields.category eq
"RECIPE"' Download assets from the channel
Site1Channel, matching the query, plus any dependencies
cec download-content Site1Channel -r Repol -c
```

```

Collection1                                Download assets from the
repository Repol, collection Collection1 and channel Site1Channel
cec download-content Site1Channel -r Repol -c Collection1 -q
'fields.category eq "RECIPE"' Download assets from repository Repol,
collection Collection1 and channel Site1Channel, matching the query, plus
any dependencies
cec download-content -a
GUID1,GUID2                                Download
asset GUID1 and GUID2 and all their dependencies
cec download-content -r
Repol                                        Download
assets from the repository Repol

```

Please specify the channel, repository, query or assets

```

-----
cec upload-content
-----

```

Usage: cec upload-content <name>

Uploads local content from channel <name>, template <name> or local file <name> to repository <repository> on OCM server. Specify -c <channel> to add the template content to channel. Optionally specify -l <collection> to add the content to collection. Specify the server with -s <server> or use the one specified in cec.properties file.

Options:

```

--help, -h          Show Help
--repository, -r <repository> The repository for the types and items
[required]
--template, -t      Flag to indicate the content is from template
--file, -f          Flag to indicate the content is from file
--channel, -c       <channel> The channel to add the content
--collection, -l   <collection> The collection to add the content
--server, -s        <server> The registered OCM server
--update, -u        Update any existing content instead of creating new items
--types, -p         Upload content types and taxonomies only

```

Examples:

```

cec upload-content Site1Channel -r Repol                                Upload
content to repository Repol, creating new items, and add to channel
Site1Channel
cec upload-content Site1Channel -r Repol -u                            Upload
content to repository Repol, updating existing content to create new
versions, and add to channel Site1Channel
cec upload-content Site1Channel -r Repol -l Site1Collection           Upload
content to repository Repol and add to collection Site1Collection and
channel Site1Channel
cec upload-content Site1Channel -r Repol -p                            Upload
content types from content SiteChannel to the server
cec upload-content Site1Channel -r Repol -s UAT                        Upload
content to repository Repol on server UAT and add to channel Site1Channel
cec upload-content Templatel -t -r Repol -c channell                 Upload
content from template Templatel to repository Repol and add to channel
channell

```

```
cec upload-content ~/Downloads/content.zip -f -r Repol -c channel1
Upload content from file ~/Downloads/content.zip to repository Repol
and add to channel channel1
```

```
-----
cec control-content
-----
```

```
Usage: cec control-content <action>
```

Performs action <action> on channel items on OCM server. Specify the channel with -c <channel>. Specify the server with -s <server> or use the one specified in cec.properties file. The valid actions are

```
publish
unpublish
add
remove
```

Options:

```
--help, -h          Show Help
--channel, -c       Channel
--repository, -r    Repository, required when <action> is add
--collection, -l    Collection
--query, -q         Query to fetch the assets
--assets, -a        The comma separated list of asset GUIDS
--server, -s        The registered OCM server
--date, -d          Date to publish items
--name, -n          Name of the scheduled publishing job to create
```

Examples:

```
cec control-content publish -c
Channel1
Publish all items in channel Channel1 on the server specified in
cec.properties file
cec control-content publish -c Channel1 -a
GUID1,GUID2
Publish asset GUID1 and GUID2 in channel Channel1
cec control-content publish -c Channel1 -s
UAT
Publish all items in channel Channel1 on the registered server UAT
cec control-content unpublish -c Channel1 -s
UAT
Unpublish all items in channel Channel1 on the registered server UAT
cec control-content add -c Channel1 -r Repol -s
UAT
Add all items in repository Repol to channel Channel1 on the registered
server UAT
cec control-content add -c Channel1 -r Repol -q 'type eq "BlogType"' -
s UAT
Add all items in
repository Repol, matching the query to channel Channel1 on the
registered server UAT
cec control-content add -c Channel1 -r Repol -q 'channels co
"CHANNELF43508F995FE582EC219EFEF03076128932B9A3F1DF6"' -s UAT
Add all
items in repository Repol and Channel2 to channel Channel1 on the
```

```

registered server UAT
  cec control-content add -c Channel1 -r Repol -a GUID1,GUID2 -s
UAT                                     Add asset GUID1
and GUID2 in repository Repol to channel Channel1
  cec control-content remove -c Channel1 -s
UAT
Remove all items in channel Channel1 on the registered server UAT
  cec control-content add -l Collection1 -r Repol -s
UAT                                     Add
all items in repository Repol to collection Collection1 on the registered
server UAT
  cec control-content remove -l Collection -s
UAT
Remove all items in collection Collection1 on the registered server UAT
  cec control-content publish -c C1 -r R1 -s UAT -d "2021/9/21 0:30:00 PST" -
n Name                                Create a publishing job
called Name to publish all items in channel C1 on the specified date.
Requires server version: 21.2.1

```

```

-----
cec transfer-content
-----
Usage: cec transfer-content <repository>

```

Creates scripts to transfer content from one OCM server to another. This command is used to transfer large number of content items and the items are transferred in batches. By default the scripts will not be executed by this command. By default all assets are transferred, optionally specify `-p` to transfer only published assets. Specify the source server with `-s <server>` and the destination server with `-d <destination>`. Optionally specify `-n` for the number of items in each batch, defaults to 200.

Options:

<code>--help, -h</code>	Show Help
<code>--server, -s</code>	The registered OCM server the content is from
[required]	
<code>--destination, -d</code>	The registered OCM server to transfer the content
[required]	
<code>--channel, -c</code>	The channel
<code>--publishedassets, -p</code>	The flag to indicate published assets only
<code>--number, -n</code>	The number of items in each batch, defaults to 200
<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 Channell List all assets  
from channel Channell  
cec list-assets -r Repol -l Collection1 List all assets  
from collection Collection1 and repository Repol  
cec list-assets -q 'fields.category eq "RECIPE"' List all assets  
matching the query
```

```
-----  
cec create-digital-asset  
-----
```

Usage: cec create-digital-asset

Creates digital asset on OCM server. Specify the server with `-s <server>` or use the one specified in `cec.properties` file. Specify the asset attributes in JSON file, e.g.

```
{  
  "imagetitle": "Logo",  
  "copyright": "Copyright © 1995, 2021, Company and/or its affiliates"  
}
```

Options:

```
--help, -h          Show Help  
--from, -f          The digital asset source file [required]  
--type, -t          The digital asset type [required]  
--repository, -r    The repository to add the asset [required]  
--slug, -l          The slug for the asset when create a single asset  
--attributes, -a    The JSON file of asset attributes  
--server, -s        The registered OCM server
```

## Examples:

```

cec create-digital-asset -f ~/Documents/logo.jpg -t Image -r
Repol                                     Create asset of type
Image
cec create-digital-asset -f ~/Documents/logo.jpg -t Image -r Repol -l
company-logo                             Create asset of type Image and
set slug to company-logo
cec create-digital-asset -f "~/Documents/demo.mp4,~/Documents/demo2.mp4" -
t Video -r Repol                         Create two assets of type Video
cec create-digital-asset -f ~/Documents/logo.jpg -t MyImage -r Repol -a ~/
Documents/logoattrs.json                Create asset of type MyImage with
attributes
cec create-digital-asset -f ~/Documents/logo.jpg -t MyImage -r Repol -l
company-logo -a ~/Documents/logoattrs.json Create asset of type MyImage
with slug and attributes
cec create-digital-asset -f ~/Documents/images -t Image -r
Repol                                     Create assets for
all images files from folder ~/Documents/images
Missing required arguments: from, type, repository

```

```

-----
cec update-digital-asset
-----

```

Usage: cec update-digital-asset <id>

Uploads a new version or updates attributes for a digital asset on OCM server. Specify the server with `-s <server>` or use the one specified in `cec.properties` file. Specify the asset attributes in JSON file, e.g.

```

{
  "imagetitle": "Logo2",
  "copyright": "Copyright © 1995, 2021, Company and/or its affiliates"
}

```

## Options:

```

--help, -h          Show Help
--from, -f          The digital asset source file for the new version
--slug, -l          The slug for the asset
--attributes, -a    The JSON file of asset attributes
--server, -s        The registered OCM server

```

## Examples:

```

cec update-digital-asset CORED129ACD36FCD42B1B38D22EEA5065F38 -l company-
logo                                     Update asset
slug
cec update-digital-asset CORED129ACD36FCD42B1B38D22EEA5065F38 -f ~/
Documents/logo2.jpg                    Upload a
new version
cec update-digital-asset CORED129ACD36FCD42B1B38D22EEA5065F38 -f ~/
Documents/logo2.jpg -l company-logo -a ~/Documents/logoattrs2.json Upload a
new version and update slug and attributes

```

```

-----
cec copy-assets

```

-----

Usage: cec copy-assets <repository>

Copies assets to another repository on OCM server. Specify the server with -s <server> or use the one specified in cec.properties file.

Options:

--help, -h            Show Help  
 --collection, -l    Collection name  
 --channel, -c        Channel name  
 --query, -q          Query to fetch the assets  
 --assets, -a        The comma separated list of asset GUIDS  
 --target, -t        The target repository [required]  
 --server, -s        The registered OCM server

Examples:

```

cec copy-assets Repo1 -t
Repo2                               Copy all assets in
repository Repo1 to Repo2
cec copy-assets Repo1 -t Repo2 -s
UAT                                  Copy all assets in
repository Repo1 to Repo2 on server UAT
cec copy-assets Repo1 -a GUID1,GUID2 -t
Repo2                               Copy asset GUID1 and GUID2 and all
their dependencies in Repo1 to Repo2
cec copy-assets Repo1 -q 'fields.category eq "RECIPE"' -t
Repo2                               Copy assets from repository Repo1, matching the
query, plus any dependencies to Repo2
cec copy-assets Repo1 -c Channel1 -t
Repo2                               Copy assets from the repository
Repo1 and channel Channel1 to Repo2
cec copy-assets Repo1 -l Collection1 -t
Repo2                               Copy assets from the repository
Repo1 and collection Collection1 to Repo2
cec copy-assets Repo1 -c Channel1 -q 'fields.category eq "RECIPE"' -t
Repo2                               Copy assets from repository Repo1, channel Channel1, matching
the query, plus any dependencies to Repo2

```

-----  
 cec create-asset-usage-report  
 -----

Usage: cec create-asset-usage-report <assets>

Generates an asset usage report for assets on OCM server. Optionally specify -o to save the report to a json file. Specify the server with -s <server> or use the one specified in cec.properties file.

Options:

--help, -h            Show Help  
 --output, -o        Output the report to a JSON file  
 --server, -s        The registered OCM server

Examples:

```

cec create-asset-usage-report GUID1
cec create-asset-usage-report GUID1 -s UAT

```

```

    cec create-asset-usage-report GUID1 -o                               The report
will be saved to GUID1AssetUsage.json
    cec create-asset-usage-report GUID1,GUID2 -o                       The report
will be saved to GUID1_GUID2AssetUsage.json
    cec create-asset-usage-report GUID1,GUID2 -o ItemReport.json     The report
will be saved to ItemReport.json

```

```

-----
cec create-repository
-----

```

Usage: cec create-repository <name>

Creates a repository on OCM server. Specify the server with `-s <server>` or use the one specified in `cec.properties` file. Optionally specify `-d <description>` to set the description. Optionally specify `-t <contenttypes>` to set the content types. Optionally specify `-c <channels>` to set the publishing channels. Optionally specify `-l <defaultlanguage>` to set the default language. Optionally specify `-p <type>` to set the repository type. The valid repository types are

```

    asset
    business

```

Options:

```

    --help, -h                Show Help
    --description, -d         The description for the repository
    --type, -p                The repository type [asset | business]. Defaults to
asset
    --contenttypes, -t       The comma separated list of content types for the
repository
    --channels, -c           The comma separated list of publishing channels to
use in this repository
    --defaultlanguage, -l    The default language
    --server, -s             The registered OCM server

```

Examples:

```

    cec create-repository Repol
    cec create-repository BusinessRepo -p business
    cec create-repository Repol -d "Blog Repository" -t BlogType,AuthorType -c
channel1,channel2 -l en-US -s UAT

```

```

-----
cec control-repository
-----

```

Usage: cec control-repository <action>

Performs action <action> on repositories on OCM server. Specify the server with `-s <server>` or use the one specified in `cec.properties` file. The valid actions are

```

    add-type
    remove-type
    add-channel
    remove-channel

```



```
add-taxonomy
remove-taxonomy
```

Options:

```
--help, -h          Show Help
--repository, -r    The comma separated list of content repositories
[required]
--contenttypes, -t  The comma separated list of content types
--channels, -c      The comma separated list of publishing channels
--taxonomies, -x    The comma separated list of promoted taxonomies
--server, -s        The registered OCM server
```

Examples:

```
cec control-repository add-type -r 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
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
Repo1                               Migrates content from collection
collection1 on ICServer to repository Repo1 on ECServer
cec migrate-content collection1 -s ICServer -d ECServer -r Repo1 -l
newCollection                       Migrates content from collection collection1
on ICServer to repository Repo1 and collection newCollection on ECServer
cec migrate-content collection1 -s ICServer -d ECServer -r Repo1 -l
newCollection -c channel1           Migrates content from collection collection1
on ICServer to repository Repo1, 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:

<code>--help, -h</code>	Show Help
<code>--name, -n</code>	Taxonomy name
<code>--id, -i</code>	Taxonomy Id
<code>--publishable, -p</code>	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 Site1Job           File will
  be uploaded to the Home folder.
  cec upload-translation-job Site1Job -s UAT   File will
  be uploaded to the Home folder on registered server UAT
  cec upload-translation-job Site1Job -f Import/TranslationJobs File will
  be uploaded to folder Import/TranslationJobs.
  cec upload-translation-job Site1Job -v      Validate
  the translation job without import.
-----

cec create-translation-connector
-----
Usage: cec create-translation-connector <name>

Creates the translation connector <name>. By default, it creates a
mockTranslationConnector. Optionally specify -f <source> to create from a
different source.

Valid values for <source> are:
  mockTranslationConnector

```



```
Options:
  --help, -h  Show Help
  --from, -f  <source> to create from
```

```
Examples:
  cec create-translation-connector connector1
```

```
-----
cec start-translation-connector
-----
```

```
Usage: cec start-translation-connector <name>
```

Starts translation connector <name>. Optionally specify -p <port> to set the port, default port is 8084.

```
Options:
  --help, -h  Show Help
  --port, -p  Set <port>. Defaults to 8084.
  --debug, -d Start the translation connector server with "--inspect"
option
```

```
Examples:
  cec start-translation-connector connector1
  cec start-translation-connector connector1 -p 7777
  cec start-translation-connector connector1 -d          Start the
translation connector server with "--inspect" option to allow debugger
to be attached.
```

```
-----
cec register-translation-connector
-----
```

```
Usage: cec register-translation-connector <name>
```

Registers a translation connector. Specify -c <connector> for the connector. Specify -s <server> for the connector server URL. Specify -u <user> and -p <password> for connecting to the server. Specify -f <fields> for custom fields.

```
Options:
  --help, -h          Show Help
  --connector, -c    <connector> Connector name [required]
  --server, -s       <server> Server URL [required]
  --user, -u         <user> User name [required]
  --password, -p    <password> password [required]
  --fields, -f      <fields> translation connector custom fields
```

```
Examples:
  cec register-translation-connector connector1-auto -c connector1 -s
http://localhost:8084/connector/rest/api -u admin -p SamplePass1 -f
"BearerToken:Bearer token1,WorkflowId:machine-workflow-
id,AdditionalData:{}"
```

```
-----
cec create-group
-----
```

Usage: cec create-group <name>

Creates an OCM group on OCM server. Specify the server with -s <server>. Set the group type with -t <type>. The valid group types are

```
PUBLIC_OPEN
PUBLIC_CLOSED
PRIVATE_CLOSED
```

Options:

```
--help, -h      Show Help
--type, -t      The group type [PUBLIC_OPEN | PUBLIC_CLOSED | PRIVATE_CLOSED]
--server, -s    <server> The registered OCM server
```

Examples:

```
cec create-group Group1          Create group Group1, people can
add themselves to the group and share content with the group
cec create-group Group1 -t PUBLIC_CLOSED  Create group Group1, only group
managers can add members but people can share content with the group
cec create-group Group1 -t PRIVATE_CLOSED Create group Group1, only group
managers can add members and only members can share content with the group
cec create-group Group1 -s DEV
```

```
-----
cec delete-group
-----
```

Usage: cec delete-group <name>

Deletes an OCM group on OCM server. Specify the server with -s <server>.

Options:

```
--help, -h      Show Help
--server, -s    <server> The registered OCM server
```

Examples:

```
cec delete-group Group1
cec delete-group Group1 -s DEV
```

```
-----
cec add-member-to-group
-----
```

Usage: cec add-member-to-group <name>

Adds users and groups to an OCM group and assign a role on OCM server. Specify the server with -s <server>. The valid roles are

```
MANAGER
MEMBER
```

Options:

```
--help, -h      Show Help
--users, -u     The comma separated list of user names
--groups, -g    The comma separated list of group names
```

```
--role, -r    The role [MANAGER | MEMBER] to assign to the users or
groups [required]
--server, -s  The registered OCM server
```

Examples:

```
cec add-member-to-group Group1 -u user1,user2 -g Group2,Group3 -r
MEMBER
cec add-member-to-group Group1 -u user1,user2 -g Group2,Group3 -r
MEMBER -s DEV
```

```
-----
cec remove-member-from-group
-----
```

Usage: cec remove-member-from-group <name>

Removes users and groups from an OCM group on OCM server. Specify the server with -s <server>.

Options:

```
--help, -h    Show Help
--members, -m The comma separated list of user and group names
[required]
--server, -s  The registered OCM server
```

Examples:

```
cec remove-member-from-group Group1 -m user1,user2,Group2,Group3
cec remove-member-from-group Group1 -m user1,user2,Group2,Group3 -s
DEV
```

```
-----
cec create-encryption-key
-----
```

Usage: cec create-encryption-key <file>

Create an encryption key to encrypt/decrypt password for servers and save to <file>. Use NodeJS 10.12.0 or later.

Options:

```
--help, -h  Show Help
```

Examples:

```
cec create-encryption-key ~/.ceckey  Create encryption key and save
to file ~/.ceckey
```

```
-----
cec register-server
-----
```

Usage: cec register-server <name>

Registers a OCM server. Specify -e <endpoint> for the server URL. Specify -u <user> and -p <password> for connecting to the server. Optionally specify -k <key> to encrypt the password. Optionally specify -t <type> to set the server type. The valid values for <type> are:

```
pod_ec
```

```

pod_ic
dev_ec
dev_pod
dev_osso

```

and the default value is pod\_ec.

For pod\_ec server, optionally specify <idcsurl>, <clientid>, <clientsecret> and <scope> for headless commands.

Options:

```

--help, -h          Show Help
--endpoint, -e      <endpoint> Server endpoint [required]
--user, -u          <user> User name [required]
--password, -p      <password> Password [required]
--key, -k           The key file used to encrypt the password
--type, -t          <type> Server type
--idcsurl, -i       <idcsurl> Oracle Identity Cloud Service Instance URL
--clientid, -c      <clientid> Client ID
--clientsecret, -s <clientsecret> Client secret
--scope, -o         <clientsecret> Scope
--timeout, -m       Timeout in millisecond when try to login to the
server. Defaults to 30000ms.

```

Examples:

```

cec register-server server1 -e http://server1.com -u user1 -p samplePass1 -
i http://idcs1.com -c clientid -s clientsecret -o https://primary-audience-
and-scope The server is a tenant on Oracle Public cloud

```

```

cec register-server server1 -e http://server1.com -u user1 -p
samplePass1

```

The server is a tenant on Oracle Public cloud

```

cec register-server server1 -e http://server1.com -u user1 -p samplePass1 -
m
60000

```

The server is a tenant on Oracle Public cloud

```

cec register-server server1 -e http://server1.git.oraclecorp.com.com -u
user1 -p samplePass1 -t

```

```

dev_ec The server
is a standalone development instance

```

```

cec register-server server1 -e http://server1.com -u user1 -p samplePass1 -
k

```

```

~/ceckey

```

The password will be encrypted

```

-----
cec set-oauth-token
-----

```

Usage: cec set-oauth-token <token>

Set OAuth token for a registered server.

Options:

```

--help, -h          Show Help
--server, -s        The registered OCM server [required]

```

Examples:

```
cec set-oauth-token token1 -s UAT Set OAuth token for server UAT,
all CLI commands using UAT will be headless
```

```
-----
cec list
-----
```

Usage: cec list

Lists local or server resources such components and templates. Specify the server with -s <server> or use the one specified in cec.properties file. Optionally specify -t <types> to list specific types of resources on the OCM server.

Valid values for <types> on the server are:

```
channels
components
localizationpolicies
recommendations
repositories
sites
templates
taxonomies
translationconnectors
```

Options:

```
--help, -h      Show Help
--types, -t     <types> The comma separated list of resource types
--server, -s    <server> The registered OCM server
```

Examples:

```
cec list                List all local resources
cec list -s             List resources on the server
specified in cec.properties file
cec list -t components,channels -s    List components and channels
on the server specified in cec.properties file
cec list -t components,channels -s UAT List components and channels
on the registered server UAT
```

```
-----
cec execute-get
-----
```

Usage: cec execute-get <endpoint>

Makes an HTTP GET request to a REST API endpoint on OCM server. Specify the server with -s <server>.

Options:

```
--help, -h      Show Help
--file, -f      The file to save the result [required]
--server, -s    The registered OCM server
```

Examples:

```
cec exeg "/sites/management/api/v1/sites?links=none" -f allsites.json -s
DEV
cec exeg "/content/management/api/v1.1/channels?links=none" -f
allchannels.json -s DEV
cec exeg "/documents/api/1.2/folders/self/items" -f homefolderitems.json -
s DEV
```

```
-----
cec install
-----
```

Usage: cec install

Creates an initial source tree in the current directory.

With cec install, your source can be in a separate directory to the cec command install files, and you no longer need your source to be within a sites-toolkit directory.

The cec.properties file can be used to specify server settings. It will be picked up from the source directory, or can be specified with environment variable CEC\_PROPERTIES

Use cec develop to start a dev/test server for your source. Different ports can be used for the server, to enable multiple source trees to exist.

Options:

--help, -h Show Help

Examples:

```
cec install
```

```
-----
cec develop
-----
```

Usage: cec develop

Starts a test server in the current folder. Specify the server with -s <server> or use the one specified in cec.properties file. Optionally specify -p <port> to set the port, default port is 8085.

Options:

--help, -h Show Help  
--port, -p Set <port>. Defaults to 8085.  
--server, -s The registered OCM server  
--debug, -d Start the server with "--inspect"

Examples:

```
cec develop
cec develop -p 7878
cec develop -p 7878 -s UAT
```

```
-----
cec sync-server
-----
```

Usage: cec sync-server

Starts a sync server in the current folder to sync changes notified by web hook from <server> to <destination> server. Specify the source server with -s <server> and the destination server with -d <destination>. Optionally specify -p <port> to set the port, default port is 8086. To run the sync server over HTTPS, specify the key file with -k <key> and the certificate file with -c <certificate>. Set authorization option with -a and the valid values are

```
none
basic
header
```

Options:

```
--help, -h          Show Help
--server, -s        The registered OCM server for sync source
[required]
--destination, -d   The registered OCM server for sync destination
[required]
--authorization, -a The authorization method [none | basic | header]
for the web hook event, defaults to basic
--username, -u      The username used to authenticate the web hook
event when <authorization> is basic
--password, -w      The password used to authenticate the web hook
event when <authorization> is basic
--values, -v        The comma separated list of name-value pairs
used to authenticate the web hook event when <authorization> is header
--port, -p          Set port. Defaults to 8086.
--key, -k           The key file for HTTPS
--certificate, -c   The certificate file for HTTPS
```

Examples:

```
cec sync-server -s DEV -d UAT -u admin -w
samplePass1          Use Basic authorization
cec sync-server -s DEV -d UAT -u admin -w samplePass1 -p
7878                Use Basic authorization and port set to 7878
cec sync-server -s DEV -d UAT
Use Basic authorization and the username and password will be prompted
to enter
cec sync-server -s DEV -d UAT -u admin
Use Basic authorization and the password will be prompted to enter
cec sync-server -s DEV -d UAT -a header -v key1:value1,key2:value2
Use Header authorization
cec sync-server -s DEV -d UAT -a none
No authorization
cec sync-server -s DEV -d UAT -k ~/keys/key.pem -c ~/keys/cert.pem
The sync server will start over HTTPS
Missing required arguments: server, destination
```

```
-----
cec webhook-server
-----
Usage: cec webhook-server
```

Starts a server in the current folder to handle events notified by web hook from <server>. Optionally specify -p <port> to set the port, default port is 8087. The supported event types are

```
seo - refresh Detailed page in the Prerender cache
```

Options:

```
--help, -h          Show Help
--type, -t          The webhook server type [seo] [required]
--contenttype, -c  The content type [required]
--detailpage, -d   The full url of the site detail page for this type
[required]
--server, -s       The registered OCM server [required]
--port, -p         Set port. Defaults to 8087.
```

Examples:

```
cec webhook-server -t seo -s DEV -c Blog -d "/site/blogsite/detailpage"
cec webhook-server -t seo -s DEV -c Blog,Author -d "/site/blogsite/
blogdetail,/site/blogsite/authordetail"
cec webhook-server -t seo -s DEV -c Blog -d "/site/blogsite/detailpage" -p
7878
Missing required arguments: type, contenttype, detailpage, server
```

## 로컬 테스트 하네스로 테스트

사용자정의 구성요소, 템플릿 및 콘텐츠 레이아웃을 Oracle Content Management로 импорт하기 전에 로컬 테스트 하네스에서 실행하십시오.

로컬 테스트 하네스를 시작하려면 다음과 같이 하십시오.

1. 터미널 창에 `cd cec`를 입력합니다.
2. `cec develop &` 또는 `cec develop --server <server-name> &`를 입력합니다.
3. `http://localhost:8085`에서 브라우저를 열어 로컬 테스트 하네스에서 실행 중인 구성요소, 템플릿 및 콘텐츠 레이아웃을 확인합니다.
4. 다음 디렉토리에서 구성요소, 템플릿, 테마 등을 찾을 수 있습니다.
  - `cec/src/main/components`
  - `cec/src/main/templates`
  - `cec/src/main/themes`

## jQuery 3.5.x로 업그레이드

개발자는 OCE Toolkit을 사용하여 부적절하게 닫힌 html 태그를 식별할 수 있습니다.

jQuery 3.5.x로 업그레이드하여 HTML 구문분석기의 보안 수정을 가져오십시오.

<https://blog.jquery.com/2020/04/10/jquery-3-5-0-released>



jQuery 3.5.x에서 특정 HTML 문자열을 구문분석하면 동일한 문자열을 3.4.x에서 구문분석할 때와 다른 결과가 나타납니다. 이러한 문자열에는 HTML 표준을 위반하는 자체 닫힘 태그가 관여합니다. 예: "<div />". 특히 커다란 태그 시퀀스의 일부일 때 해당 문자열은 3.4.x와 비교해 3.5.x에서 다르게 구문분석될 수 있습니다.

`cec create-asset-report <site>`를 실행하여 페이지 JSON 파일과 구성요소 HTML 및 JS 파일에서 부적절하게 닫힌 html 태그를 식별할 수 있습니다.

## Developer Cloud Service로 Oracle Content Management용 개발

OCE Toolkit(Developer Cloud Service 통합)은 Oracle Content Management용 사이트 템플릿, 테마, 사용자정의 구성요소 및 콘텐츠 레이아웃 개발을 돕습니다.

OCE Toolkit으로 Oracle Content Management의 자산 저장소, 파일 및 폴더를 사용할 수 있습니다. OCE Toolkit에는 테마와 콘텐츠 레이아웃을 포함한 사용자정의 구성요소 및 사이트 템플릿을 생성하고 개발하는 도구가 있습니다. 여기에는 Git 저장소는 물론, 빠르고 반복적인 개발을 위한 로컬 테스트 하네스와 시작용 샘플 단위 테스트가 포함됩니다.

OCE Toolkit은 다음 작업을 도울 수 있습니다.

- 구성요소, 템플릿, 테마, 콘텐츠 레이아웃의 로컬 개발 및 테스트에 Oracle Content Management 인스턴스를 사용할 수 있도록 로컬 개발 환경을 설정합니다.
- 샘플에서 구성요소, 사이트 템플릿 및 콘텐츠 레이아웃을 생성하고 테스트 하네스에서 실행하고 탐색하며 Developer Cloud Service 환경에서 구성요소, 템플릿, 테마 및 콘텐츠 레이아웃을 개발합니다.
- 소스 관리 및 추가 개발을 위해 Oracle Content Management에서 생성된 구성요소와 사이트 템플릿을 Developer Cloud Service 환경으로 임포트합니다.
- Oracle Content Management에서 사용할 수 있도록 Developer Cloud Service 환경에서 구성요소, 템플릿 및 콘텐츠 레이아웃을 익스포트합니다.
- 기존 구성요소, 템플릿 또는 콘텐츠 레이아웃 복사
- 단위 테스트 쓰기
- 구성요소 최적화
- Oracle Content Management에 구성요소 및 템플릿 배치

다음 항목은 사용자정의 구성요소, 사이트 템플릿, 테마 및 콘텐츠 레이아웃 개발을 위해 Developer Cloud Service를 설정하는 방법을 설명합니다.

1. [Developer Cloud Service 사용 정보](#)
2. [Oracle Content Management용 Developer Cloud Service 콘솔에 사인인](#)
3. [Developer Cloud Service에서 프로젝트 생성](#)
4. [새 Git 저장소에서 프로젝트 코드에 OCE Toolkit 추가](#)
5. [로컬 테스트 하네스에서 사용자정의 구성요소, 템플릿 및 콘텐츠 레이아웃 테스트](#)
6. [변경사항 병합](#)

다음 항목은 Oracle Content Management OCE Toolkit 사용에 대한 정보를 제공합니다.

- [cec 명령행 유틸리티 사용](#)
- [Developer Cloud Service로 사용자정의 구성요소 개발](#)

- Developer Cloud Service로 템플릿 개발
- 콘텐츠 레이아웃 개발

## Developer Cloud Service 사용 정보

Oracle Developer Cloud Service는 클라우드 기반의 소프트웨어 개발 PaaS(서비스형 플랫폼) 이자 호스트된 애플리케이션 개발 인프라 환경입니다. Oracle Cloud 내에서의 애플리케이션 개발, 공동 작업 및 배포를 위해 오픈 소스 표준 기반 통합이 제공됩니다.

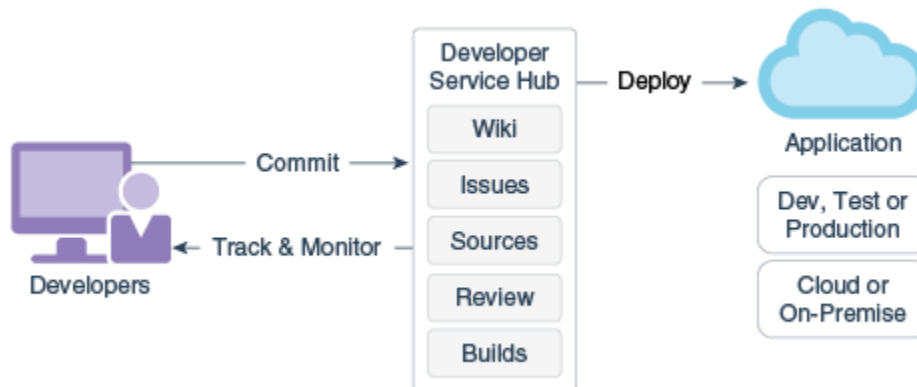
Developer Cloud Service는 Git, Maven, 이슈 및 위키와 통합을 통해 효과적으로 애플리케이션 개발 수명 주기를 관리할 수 있도록 Oracle Cloud에 호스트된 소프트웨어 및 서비스 모음입니다. Oracle Developer Cloud Service를 사용하면 애플리케이션 소스 코드를 Oracle Cloud의 Git 저장소로 커밋하고, 지정된 문제 및 결함을 온라인으로 추적하고, 위키 페이지를 사용하여 정보를 공유하고, 소스 코드 동료 평가를 수행하고, 프로젝트 빌드를 모니터링할 수 있습니다. 성공적인 테스트 후 Oracle Content Management에 프로젝트를 배치할 수 있습니다.

## Oracle Content Management용 Developer Cloud Service 콘솔에 사인인

Developer Cloud Service 콘솔에서 Oracle Content Management용 사용자정의 구성요소 개발을 시작합니다.

Oracle Cloud 서비스 관리자는 [내 서비스 관리]를 사용하여 클라우드 서비스를 생성하고 관리할 수 있습니다. Oracle Content Management의 서비스 인스턴스 관리자이자 Standard Developer Service의 서비스 관리자인 경우 서비스를 설정하고 사용을 시작할 수 있습니다.

1. 계정에 제공된 정보를 사용하여 Oracle Cloud에 사인인합니다.
2. [내 서비스 관리]에 사인인하여 Oracle Content Management 인스턴스 및 Standard Developer Service를 생성하고 관리합니다.



3. 요청에 따라 Oracle Developer Cloud Service 전자메일을 확인합니다.
4. 서비스 구독 세부정보를 사용하여 Oracle Content Management 인스턴스를 설정하고 인스턴스의 Oracle Content Management URL로 이동합니다.
5. Standard Developer Service의 URL로 이동합니다.
6. Oracle Developer Cloud Service 계정에 사인인합니다.

Developer Cloud Service URL에 액세스하여 콘솔에 사인인합니다.

## Developer Cloud Service에서 프로젝트 생성

"Content Experience Cloud" 프로젝트 템플릿을 사용하여 Developer Cloud Service에서 프로젝트를 생성하거나, 빈 Git 저장소로 프로젝트를 생성하고 Oracle Content Management 인스턴스에서 OCE Toolkit을 임포트할 수 있습니다.

- [Oracle Content Management 템플릿으로 Developer Cloud Service 프로젝트 생성](#)
- [Oracle Content Management에서 다운로드한 OCE Toolkit으로 Developer Cloud Service에서 프로젝트 생성](#)
- [새 Git 저장소에서 프로젝트 코드에 OCE Toolkit 추가](#)

## Oracle Content Management 템플릿으로 Developer Cloud Service 프로젝트 생성

Developer Cloud Service에서 사용자정의 구성요소, 템플릿, 테마 및 콘텐츠 레이아웃 개발을 위한 프로젝트를 생성합니다.

프로젝트를 생성하려면 다음과 같이 하십시오.

1. Developer Cloud Service 콘솔에 사인인한 후 **새 프로젝트**를 누릅니다.
2. 템플릿 목록에서 **Content Management**를 선택하고 **다음**을 누릅니다.
3. [프로젝트 속성] 아래의 속성에서 **위키 마크업 필드**에 **CONFLUENCE**를 선택합니다.

## Oracle Content Management에서 다운로드한 OCE Toolkit으로 Developer Cloud Service에서 프로젝트 생성

Developer Cloud Service에서 사용자정의 구성요소, 템플릿, 테마 및 콘텐츠 레이아웃 개발을 위한 프로젝트를 생성합니다.

프로젝트를 생성하려면 다음과 같이 하십시오.

1. Developer Cloud Service 콘솔에 사인인한 후 **새 프로젝트**를 누릅니다.
2. 프로젝트 이름을 지정하고, 원하는 다른 프로젝트 세부정보를 입력하거나 선택하고, **다음**을 누릅니다.
3. 템플릿 목록에서 **초기 저장소**를 선택하고 **다음**을 누릅니다.
4. [프로젝트 속성] 아래의 속성에서 초기 저장소에 대해 빈 저장소를 선택합니다. **완료**를 누릅니다.

## 새 Git 저장소에서 프로젝트 코드에 OCE Toolkit 추가

프로젝트를 위한 새로운 빈 Git 저장소에 OCE Toolkit을 추가할 수 있습니다.

1. 새 프로젝트에서 **REPOSITORIES** 아래에 프로젝트 Git 저장소의 HTTP URL을 복사합니다.
2. 터미널 창을 열고 `git clone <your-project >.git` 명령을 입력합니다.
  - a. 요청 시 Developer Cloud Service의 비밀번호를 입력합니다.
  - b. "git is not a command" 오류가 나타나면 <https://git-scm.com/downloads>에서 Git을 설치하고 `git clone` 명령을 다시 입력합니다.

3. `git clone git@github.com:oracle/content-and-experience-toolkit.git`  
또는 `https://github.com/oracle/content-and-experience-toolkit/archive/master.zip`에서 다운로드할 수 있습니다.
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`

## 로컬 테스트 하네스에서 사용자정의 구성요소, 템플릿 및 콘텐츠 레이아웃 테스트

사용자정의 구성요소, 템플릿 및 콘텐츠 레이아웃을 Oracle Content Management로 임포트하기 전에 로컬 테스트 하네스에서 실행하십시오.

로컬 테스트 하네스를 시작하려면 다음과 같이 하십시오.

1. 터미널 창에 `cd cec-components`를 입력합니다.  
`npm start &`를 입력합니다.
2. `http://localhost:8085`에서 브라우저를 열어 로컬 테스트 하네스에서 실행 중인 구성요소, 템플릿 및 콘텐츠 레이아웃을 확인합니다.

로컬 서버에서 구성요소를 테스트할 때 로컬 템플릿 또는 Oracle Content Management 서버의 콘텐츠를 사용하도록 선택할 수 있습니다.

## 변경사항 병합

로컬 시스템에서 구성요소, 템플릿 또는 콘텐츠 레이아웃을 생성하거나 소스 코드를 편집한 후에 신규/변경된 구성요소와 템플릿을 프로젝트 Git 저장소로 병합해야 합니다.

Git 저장소로 변경사항을 병합하려면 터미널 창에서 다음 명령을 순서대로 입력합니다.

```
cd cec-components git pull
git add .
git status
git commit -a -m "Your comments" git pull
git push
```

## OCE Toolkit을 사용하여 테스트에서 운용으로 변경사항 전달

사이트 템플릿을 개발한 후 OCE Toolkit의 CLI(명령행 인터페이스)를 사용하여 Oracle Content Management 서버의 개발에서 테스트, 운용 환경으로 템플릿을 전달할 수 있습니다.

변경사항을 전달하려면 Toolkit 명령을 사용하여 사이트를 생성하고 개발, 테스트 및 운용 서버에서 수명 주기를 관리할 수 있습니다. 개발 환경에서 사이트를 변경하고 이 변경사항을 테스트 및 운용 환경으로 전달할 수 있습니다. 이 명령행 유틸리티 집합을 스크립팅 환경으로 통합하여 배치를

관리할 수도 있습니다. CLI 유틸리티를 사용하여 자산 및 구성요소와 같은 새 항목뿐만 아니라 기존 콘텐츠의 업데이트를 돌아웃할 수 있습니다.

다음 단계는 OCE Toolkit CLI를 사용하여 개발에서 테스트, 운용 환경으로 변경사항을 전달하는 방법을 보여줍니다.

1. 동일한 저장소 및 지역화 정책으로 개발, 테스트 및 운용 서버를 설정합니다.

개발 서버에서 테스트 서버로, 그 다음 운용 서버로 변경사항을 전달하려면 각각의 3대 서버에서 동일한 이름과 지역화 정책으로 저장소를 설정해야 합니다. 자산 저장소의 기본 지역화 정책은 en-US이지만, 3대 서버에서 모두 같다면 다른 것을 사용할 수 있습니다.

자산 저장소 설정을 참조하십시오.

2. Oracle Content Management에 개발, 테스트 및 운용 서버를 등록합니다.

사이트로 변경사항을 전달하기 전에 각 서버를 등록해야 합니다. OCE Toolkit에서 제공한 `cec register-server` 명령으로 서버를 등록할 수 있습니다.

```
cec register-server <name>
```

다음 명령 옵션을 지정합니다.

- `-e <endpoint>`는 서버 URL용
- `-u <user>` 및 `-p <password>`는 서버 접속용
- `-t <type>`은 선택사항, 서버 유형 설정. 기본값은 `pod_ec`입니다.

Oracle Public Cloud의 Oracle Content Management 테넌트에 접속할 때 `pod_ec`만 사용하십시오.

예를 들어, 다음 명령은 Oracle Public Cloud의 테넌트인 서버를 등록합니다.

```
cec register-server DEV -e https://DEV.example.com -u user1 -p
<password>
```

다음 명령은 Oracle Content Management의 독립형 개발 인스턴스를 등록합니다.

```
cec register-server DEV -e https://DEV.git.oraclecorp.example.com -
u user1 -p <password>
```

Oracle Content Management 서버를 등록한 후 `cec list Toolkit` 명령으로 콘텐츠를 나열할 수 있습니다.

다음 명령은 개발 서버의 콘텐츠를 나열합니다.

```
cec-compontents> 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	Theme	Type
Online		
Name	StarterSiteTheme	Enterprise
Templates:		
Name	Theme	Type
StarterTemplate	StarterTheme	Standard

### 3. 사이트 템플릿을 개발 서버로 업로드하고 템플릿에서 사이트를 생성합니다.

cec create-template 명령으로 사이트 템플릿을 생성하고 개발 서버로 템플릿을 업로드합니다. 그런 다음 cec create-site 명령으로 템플릿에서 사이트를 생성할 수 있습니다. 다음 명령은 템플릿을 생성하고 템플릿을 업로드합니다.

```
- cec create-template blog -f BlogTemplate
- cec upload-template blog -s DEV
```

다음 명령은 업로드한 템플릿에서 blog라는 사이트를 생성합니다.

```
cec-components> cec create-site blog -t blog -r r -l "en-US" -d "en-US" --
server DEV
- Logged in to remote server: <https:<host:<port>
- establish user session
- get template
- get repository
- get localization policy
- creating enterprise site . . .
  name          blog
  template      blog
  site prefix   blog
  repository    r
  localization policy en-US
  default language en-US
- submit create site site
- create site in process: percentage 95
- create site in process: percentage 95
- create site in process: percentage 95
- create site in process: percentage 95
- create site in process: percentage 95
- site created
```

### 4. 개발 서버에 사이트를 게시하고 온라인으로 가져옵니다.

사이트를 생성한 후 `cec control-site` 명령을 사용하여 사이트를 게시하고 온라인으로 가져올 수 있습니다.

```
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. DEV에서 UAT로 사이트를 이동하려면 사이트를 패키징해야 합니다. 서버 간의 사이트 이동을 위한 패키징 모델은 템플릿입니다. 개발 서버에서 생성한 사이트에서 새 템플릿을 생성하고 템플릿을 다운로드합니다.

다음 예제의 `cec create-template-from-site` 명령은 `blog`에서 `blog2`라는 사이트 템플릿을 생성합니다.

```
cec create-template-from-site blog2 -s blog
```

cec download-template 명령을 사용하여 개발 사이트에서 생성한 템플릿을 다운로드합니다.

```
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. 템플릿을 업로드하여 콘텐츠 유형 및 콘텐츠 레이아웃 맵을 생성합니다.

```
cec upload-template blog2 --server UAT
```

7. 템플릿을 업로드하되 템플릿에서 콘텐츠 항목(콘텐츠 템플릿)을 제외합니다.

```
cec upload-template blog2 --server UAT -x
```

이렇게 하면 원래 사이트와 동일한 GUID를 가진 콘텐츠로 사이트를 생성할 수 있습니다. 콘텐츠가 포함된 템플릿에서 사이트를 생성할 때 새 사이트의 모든 콘텐츠는 새 GUID를 갖습니다. 따라서 새 콘텐츠를 생성하는 대신 콘텐츠 업데이트를 허용하려면 템플릿에서 콘텐츠를 제외해야 합니다.

8. 템플릿에서 사이트를 생성합니다.

```
cec create-site blog -t blog2 -r r -l "en-US" -d "end-US" --server UAT
```



9. 사이트의 채널 및 모음으로 콘텐츠 템플리트를 업로드합니다. 단계 7에서 템플리트에서 콘텐츠를 제외했기 때문에 이를 수행해야 합니다.

```
cec upload-content blog2 -t -r r -c blog -l "blog site" --server UAT
```

10. 테스트 서버에 사이트를 게시하고 온라인으로 가져옵니다.

cec control-site 명령을 사용하여 사이트를 게시하고 온라인으로 가져옵니다.

```
cec-components> cec control-site publish --site blog --server UAT
- Logged in to the remote server: https://<host>:<port>
- establish user session
- get site: runtimeStatus: offline publishStatus: unpublished
- submit publish site
- publish in process: percentage 20
- publish in process: percentage 40
- publish in process: percentage 49
- publish in process: percentage 49
- publish in process: percentage 50
- publish in process: percentage 50
- publish blob finished
```

11. DEV 서버의 blog 사이트를 변경한 경우 이미 UAT 서버에 생성한 사이트로 변경사항을 전달할 수 있습니다.

12. 변경사항을 포함할 다른 템플리트를 사이트에서 생성합니다.

```
cec create-template-from-site blog3 -s blog --server DEV
```

13. 템플리트를 다운로드합니다.

```
cec download-template blog3 -s DEV
```

14. 템플리트를 업로드하고 사이트를 생성하여 테스트 환경으로 변경사항을 전달합니다.

```
cec upload-template blog3 -s UAT
```

이 명령은 변경된 구성요소와 테마를 생성하거나 업데이트하되 콘텐츠는 제외합니다.

15. 이제 update-site 명령을 사용하여 콘텐츠를 가져오고 페이지를 업데이트합니다.

```
cec update-site blog -t blog3 - UAT
```

예를 들어, 다음과 같습니다.

```
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. 사이트에서 변경사항이 전달되었는지 확인합니다.
17. DEV 서버에서 UAT 서버로 사이트를 이동할 때와 동일한 단계를 수행하여 UAT에서 PROD로 이동합니다.
18. 운용 서버에 사이트를 생성하고 온라인으로 가져와서 변경사항을 확인합니다.

cec list 명령을 사용하여 운용 사이트의 콘텐츠를 보고 개발 환경에서 수행한 변경사항이 포함되었는지 확인할 수 있습니다. 또한 사이트에서 변경사항이 운용 환경으로 전달되었는지 확인할 수 있습니다.

## 비밀번호 암호화

OCE Toolkit에 서버를 등록할 때 로컬에서 서버를 사용할 수 있도록 비밀번호를 암호화해야 합니다.

1. cec register-server 명령을 사용하여 일반 텍스트 비밀번호가 포함된 Oracle Content Management 서버를 등록합니다.
2. cec create-encryption-key 명령으로 비밀번호를 암호화합니다.

```

cec create-encryption-key <file> [alias: cek]
Create an encryption key to encrypt/decrypt password for servers.

```

3. 암호화 키로 서버를 다시 등록하여 로컬 개발 및 테스트에 서버를 사용할 수 있도록 합니다. 암호화된 비밀번호는 서버 접속 파일에 저장됩니다. 등록된 서버에 접속할 때 비밀번호가 해독됩니다.

## 서버 등록

OCE Toolkit에 서버를 등록할 수 있습니다.

cec register-server 명령을 암호화 키와 함께 사용하여 로컬 개발 및 테스트용 Oracle Content Management 서버를 등록합니다.

서버를 등록할 때 비밀번호를 암호화하면 OCE Toolkit과 함께 서버를 사용할 수 있습니다. [비밀번호 암호화](#)를 참조하십시오.

## 사이트의 사용 및 권한 보고서 생성

사이트를 테스트에서 운용 환경으로 전달할 때 보고서를 생성하여 대상 서버 멤버의 권한을 검증/수정할 수 있습니다.

cec create-asset-report를 사용하여 다음과 같이 확인합니다.

1. 모든 사이트 아티팩트의 멤버십 및 채널 지정을 확인합니다.

- 테마
- 템플릿
- 구성요소
- 콘텐츠 유형

2. 해결할 수 있는 문제에 플래그를 표시합니다.

예를 들어, 다음과 같습니다.

```
cec create-asset-report blog1 -s <registered-server> -o
```

```
cec create-asset-report trbcent -s <registered-server> -o
```

보고서가 JSON 파일로 생성되고, 사용 및 권한에 문제가 있는지 확인할 수 있습니다. 다음 명령은 권한 수정에 사용할 수 있습니다.

- cec share-type: Oracle Content Management 서버에서 사용자와 유형을 공유합니다.
- cec unshare-type: Oracle Content Management 서버에서 주어진 사용자의 유형 액세스를 제거합니다.
- cec share-repository: Oracle Content Management 서버에서 저장소(및 저장소에서 사용된 유형)를 공유합니다.
- cec unshare-repository: Oracle Content Management 서버에서 사용자의 저장소 액세스를 제거합니다.
- 예를 들어, 다음과 같습니다.

```
cec share-repository Repol -u <user-name1>,<user-name2> -r manager -t -s <registered-server>
```

## 문서 및 폴더 다운로드 및 업로드

Oracle Content Management 서버 간에 문서 및 폴더를 다운로드 및 업로드할 수 있습니다.

다음 명령은 문서 및 폴더 다운로드 및 업로드에 사용할 수 있습니다.

```
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>

```

cec-share-folder <name>의 경우 Oracle Content Management 서버의 사용자와 폴더를 공유하고 사용자에게 롤을 지정할 수 있습니다. -s <server>로 서버를 지정하거나 cec.properties 파일에 지정된 서버를 사용합니다. 적합한 롤은 다음과 같습니다.

- manager
- contributor
- downloader
- viewer

다운로드를 위해 폴더 계층을 지정할 수 있습니다.

## 템플릿에서 사이트 생성 및 콘텐츠에 동일한 GUID 유지

개발자는 OCE Toolkit 명령을 사용하여 템플릿에서 Oracle Content Management 사이트를 생성하면서 콘텐츠에 동일한 GUID를 유지할 수 있습니다.

다음 OCE Toolkit 명령을 사용합니다.

```
update create-site-from-template --reuse-content
```

이 명령은 Oracle Content Management 서버에 사이트를 생성하고, 사이트를 생성할 때 콘텐츠 ID를 보존합니다. 여러 번의 테스트-운용 실행에서 대상 서버에 중복된 콘텐츠 항목이 생기지 않도록 하려면 콘텐츠 ID를 보존해야 합니다.

## 표준 사이트에서 엔터프라이즈 템플릿 생성

개발자는 OCE Toolkit CLI 명령을 사용하여 표준 사이트에서 엔터프라이즈 템플릿을 생성할 수 있습니다.

기본적으로 create-template 명령은 사이트가 표준 사이트이면 표준 템플릿을, 사이트가 엔터프라이즈 사이트이면 엔터프라이즈 템플릿을 생성합니다. 표준 사이트에서도 엔터프라이즈 템플릿을 생성할 수 있습니다.

cec create-template를 새로운 -enterprise 옵션과 함께 실행합니다.

```
cec create-template EnterpriseTemp1 -s StandardSite1 -e
```

## 분류법 импорт 및 익스포트

OCE Toolkit 명령을 사용하여 로컬 시스템에서 Oracle Content Management 서버로 분류법을 импорт하거나, 서버에서 로컬 시스템으로 분류법을 익스포트할 수 있습니다.

cec download-taxonomy <name> 명령은 Oracle Content Management에서 분류법을 익스포트합니다. Oracle Content Management 서버에서 분류법을 다운로드합니다.

이 명령에 다음 옵션을 사용할 수 있습니다.

- `--status, -t [promoted | published] [required]`: 분류법 상태를 지정합니다.
- `--id, -i`: 다른 분류법이 동일한 이름을 사용하면 분류법 ID를 지정합니다.
- `--server, -s`: 등록된 Oracle Content Management 서버를 지정하거나 `cec.properties` 파일에 지정된 서버를 사용합니다.
- `--help, -h`: 명령의 도움말을 표시합니다.

`download-taxonomy` 명령의 몇 가지 예제는 다음과 같습니다.

```
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>` 명령은 Oracle Content Management로 분류법을 임포트합니다. Oracle Content Management 서버로 분류법을 업로드합니다.

이 명령에 다음 옵션을 사용할 수 있습니다.

- `--createnew, -c`: 새 분류법을 생성합니다.
- `--name, -n`: 새 분류법의 이름을 지정합니다.
- `--abbreviation, -a`: 새 분류법의 약어를 지정합니다.
- `--description, -d`: 새 분류법의 설명을 지정합니다.
- `--file, -f`: 분류법이 파일에서 오는지 여부를 나타냅니다.
- `--server, -s`: 등록된 Oracle Content Management 서버를 지정하거나 `cec.properties` 파일에 지정된 서버를 사용합니다.
- `--T2P`: 대상 Oracle Content Management 인스턴스에 동일한 ID의 자산 또는 분류법이 존재하는지 확인합니다. true인 경우, 자산이 새 버전으로 추가되고 분류법이 초안으로 추가됩니다(기존 초안 무효화). 그렇지 않으면 새 자산 또는 분류법이 동일한 ID로 생성됩니다.
- `--New`: 항상 대상 Oracle Content Management 인스턴스에 새 자산 또는 분류법을 생성합니다.
- `--help, -h`: 명령의 도움말을 표시합니다.

`upload-taxonomy` 명령의 몇 가지 예제는 다음과 같습니다.

```
cec upload-taxonomy Taxonomy1
```

Create a new taxonomy or a draft of an existing taxonomy on upload

```
cec upload-taxonomy Taxonomy1 -s UAT
```

Create a new taxonomy or a draft of an existing taxonomy on upload on the registered server UAT

```
cec upload-taxonomy Taxonomy1 -c
```

Create a new taxonomy on upload

```
cec upload-taxonomy Taxonomy1 -c -n Taxonomy1_2 -a t12 -d
```

Create a new taxonomy on upload with the given name, abbreviation

**"Taxonomy1 copy" and description cec upload-taxonomy**

Create a new taxonomy or a draft of an existing taxonomy in <file-name>.json -f and upload the JSON file

OCE Toolkit 테스트-운용 CLI 유틸리티를 사용하여 소스 Oracle Content Management 서버에서 대상 Oracle Content Management 서버로, 콘텐츠 모델 및 해당 종속성과 함께 자산 импорт/ 익스포트를 자동화할 수 있습니다.

관리자나 콘텐츠 관리자는 테스트-운용 환경에서 OCE Toolkit 명령으로 분류법을 импорт/ 익스포트할 수 있습니다. 관리자 권한으로 분류법의 초안을 추가할 수 있습니다. 콘텐츠 관리자 권한으로 새 분류법을 생성할 수 있습니다.

분류법 승격, 저장소에 지정, 게시와 같은 분류법 수명 주기 작업을 테스트-운용 환경에 사용할 수 있습니다. 범주화 정보 및 분류법과 함께 자산을 импорт 파일로 импорт할 수 있으며, 익스포트 파일에서 범주화 정보 및 분류법과 함께 자산을 익스포트할 수 있습니다.

## 권장사항 импорт 및 익스포트

OCE Toolkit 명령을 사용하여 Oracle Content Management 서버에서 로컬 시스템으로 권장사항을 импорт하거나, 로컬 시스템에서 서버로 권장사항을 익스포트할 수 있습니다.

cec download-recommendation <name> 명령은 Oracle Content Management에서 권장사항을 익스포트합니다. Oracle Content Management 서버에서 권장사항을 다운로드합니다.

이 명령에 다음 옵션을 사용할 수 있습니다.

- --status, -t [promoted | published] [required]: 권장사항 상태를 지정합니다.
- --id, -i: 다른 권장사항이 동일한 이름을 사용하면 권장사항 ID를 지정합니다.
- --server, -s: 등록된 Oracle Content Management 서버를 지정하거나 cec.properties 파일에 지정된 서버를 사용합니다.
- --help, -h: 명령의 도움말을 표시합니다.

download-recommendation 명령의 몇 가지 예제는 다음과 같습니다.

```
cec download-recommendation Recommendation1 -t promoted
```

```
cec download-recommendation Recommendation1 -i
6A6DE836572C468B90F2A1C17B7CE5E4 -t promoted
```

```
cec download-recommendation recommendation -t published -s UAT
```

cec upload-recommendation <name> 명령은 Oracle Content Management로 권장사항을 импорт합니다. Oracle Content Management 서버로 권장사항을 업로드합니다.

이 명령에 다음 옵션을 사용할 수 있습니다.

- --createnew, -c: 새 권장사항을 생성합니다.
- --name, -n: 새 권장사항의 이름을 지정합니다.
- --abbreviation, -a: 새 권장사항의 약어를 지정합니다.
- --description, -d: 새 권장사항의 설명을 지정합니다.

- `--file, -f`: 권장사항이 파일에서 오는지 여부를 나타냅니다.
- `--server, -s`: 등록된 Oracle Content Management 서버를 지정하거나 `cec.properties` 파일에 지정된 서버를 사용합니다.
- `--help, -h`: 명령의 도움말을 표시합니다.

**upload-recommendation** 명령의 몇 가지 예제는 다음과 같습니다.

**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

## 모음 콘텐츠 추가 또는 제거

개발자는 OCE Toolkit 명령을 사용하여 모음에 콘텐츠를 추가하고 모음에서 콘텐츠를 제거할 수 있습니다.

다음 `cec-control-content` 명령은 **Repo1** 저장소의 모든 항목을 **UAT** 등록 서버의 **Collection1** 모음에 추가합니다.

```
cec control-content add -l Collection1 -r Repo1 -s UAT
```

다음 `cec-control-content` 명령은 **UAT** 등록 서버의 **Collection1** 모음에서 모든 항목을 제거합니다.

```
cec control-content remove -l Collection -s UAT
```

## OCE Toolkit을 사용하여 사용자정의 필드 편집기 개발

OCE Toolkit은 필드 모양 유형의 구성요소 개발을 지원합니다. 개발자는 사용자정의 필드 편집기를 생성하고 관리할 수 있습니다.

필드 모양 유형의 구성요소에 대해 다음 작업을 수행할 수 있습니다.

- 구성요소 열기, 복사 또는 삭제

- 구성요소 게시 또는 게시 취소
- 구성요소 익스포트 또는 임포트
- 구성요소의 멤버 추가 또는 제거
- 속성 보기
- 구성요소 로고 선택

필드 모양 유형으로 구성요소 목록을 필터링할 수 있습니다.

다음 OCE Toolkit 명령은 사용자정의 필드 편집기 개발에 사용할 수 있습니다.

```
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
```

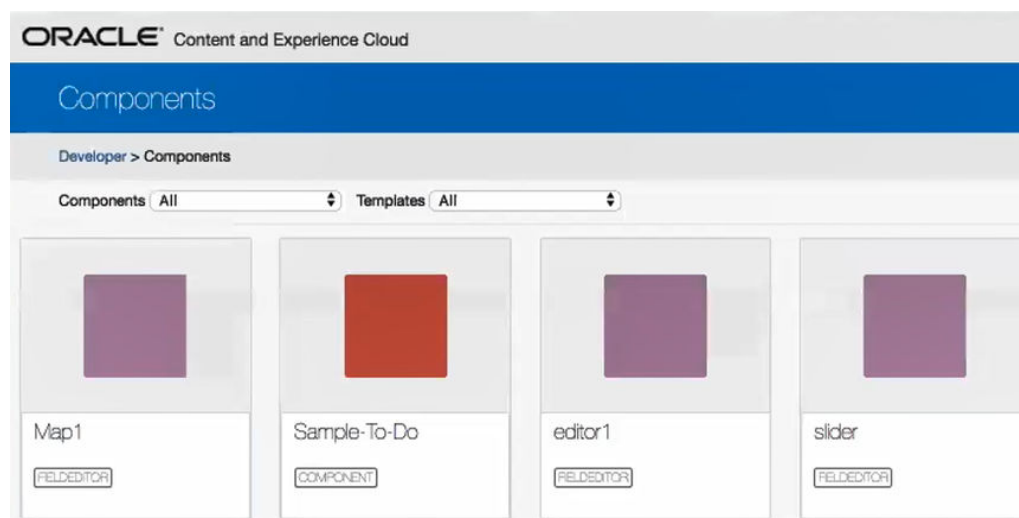
다음과 같은 필드 모양 구성요소 샘플이 OCE Toolkit에 포함됩니다.

- TextFieldEditor
- SliderFieldEditor
- MapFieldEditor

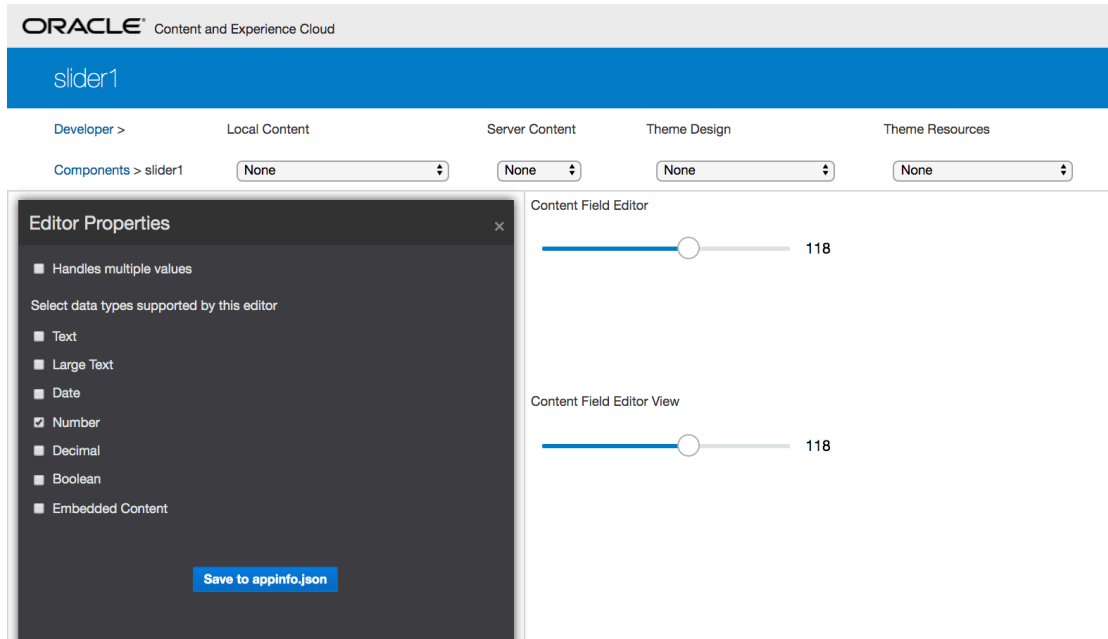
다음 이미지는 샘플 필드 모양 구성요소 개발에 사용할 수 있는 OCE Toolkit 명령을 보여줍니다.

```
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
```

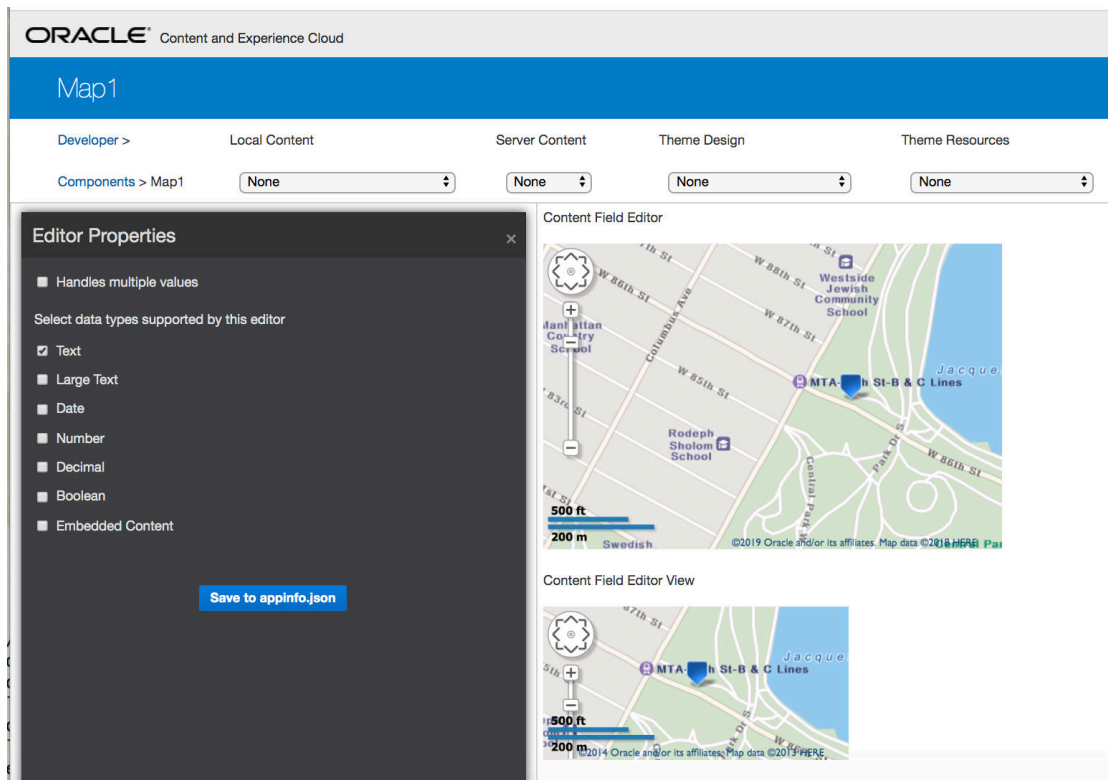
미리 정의된 필드 모양 구성요소를 로컬 서버에 생성하고 테스트하여 Oracle Content Management 인스턴스로 업로드할 수 있습니다. 다음 이미지는 localhost:8085에서 해당 구성요소를 보여줍니다.



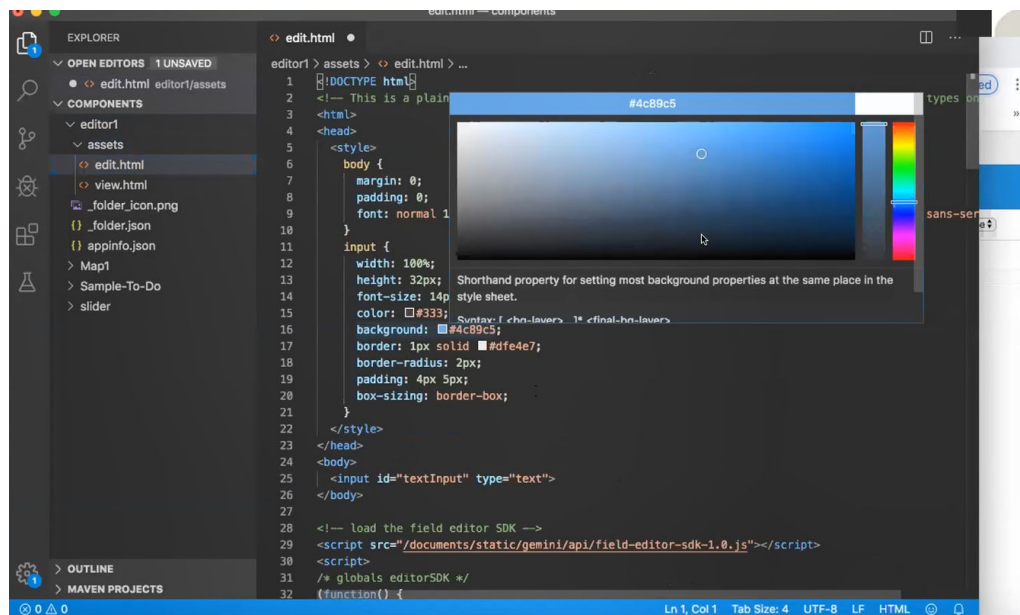
슬라이더와 같은 각 구성요소를 로컬 서버에서 테스트할 수 있습니다. 여기서 구성요소의 속성을 선택하고 저장할 수 있습니다.



맵 구성요소의 경우 맵 주위를 눌러 편집기 값으로 위치를 제공할 수 있습니다.



구성요소의 HTML 파일을 편집하여 배경색과 같은 설정을 변경할 수 있습니다.



OCE Toolkit 명령을 사용하여 로컬에서 필드 편집기를 콘텐츠 유형의 필드와 연관할 수 있습니다.

```
cec add-field-editor editor1 -t SimpleContent -n -c SimpleType -f title
```

```
cec add-field-editor slider -t SimpleContent -n -c SimpleType -f value
```

사용자정의 필드 편집기 구성 및 테스트를 마친 후에 OCE Toolkit 명령을 사용하여 Oracle Content Management 인스턴스로 업로드할 수 있습니다.

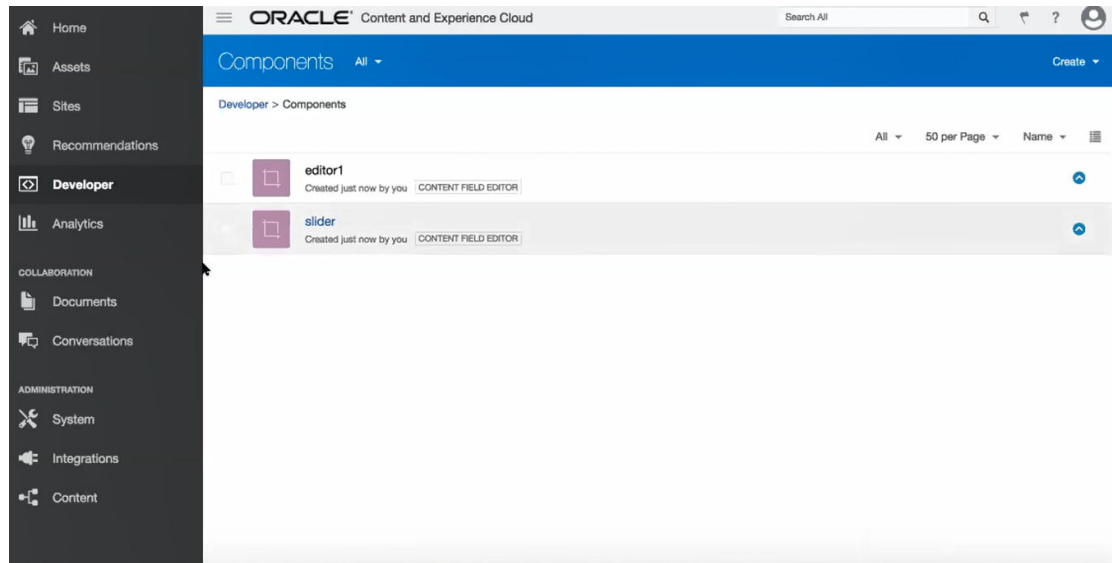
```
# upload editors
cec ulcp editor1,slider -p -s Latest
```

```
# upload content
```

```
cec cr Repo5 -s
```

```
cec upload-content SimpleContent -r Rpo5 -s
```

사용자정의 필드 편집기를 업로드하면 구성요소가 Oracle Content Management 인스턴스로 임포트됩니다.



## 한 서버에서 다른 서버로 사이트 전송 또는 업데이트

개발자는 OCE Toolkit 명령을 사용하여 서버 A에서 서버 B로 사이트와 콘텐츠를 생성하거나 업데이트할 수 있습니다.

기본적으로는 모든 자산이 전송됩니다. 선택적으로 게시된 자산만 전송하려면 `-p`를 지정합니다. `-s <server>`로 소스 서버를 지정하고 `-d <destination>`으로 대상 서버를 지정합니다. 사이트에 다른 저장소의 자산이 포함된 경우 선택적으로 저장소 매핑을 제공합니다. 그렇지 않으면 해당 자산이 전송되지 않습니다.

다음 명령을 사용하여 테스트에서 운용 환경으로 사이트를 업데이트하거나 전송할 수 있습니다.

```
cec transfer-site site --from server --to server --repository r
--localization-policy l
```

추가 옵션은 [cec 명령행 유틸리티 사용](#)을 참조하십시오.

## 콘텐츠 항목이 없는 사이트 전송

개발자는 OCE Toolkit을 사용하여 한 Oracle Content Management 서버에서 다른 서버로 콘텐츠 항목이 없는 사이트를 전송할 수 있습니다.

수많은 콘텐츠 자산이 있는 페이지를 전송하려면 사이트와 콘텐츠를 분리해야 합니다. `transfer-site` 명령에 `--excludecontent` 옵션(단축키 `-x`)을 추가할 수 있습니다.

이 옵션이 설정되면 사이트만 전송됩니다.

예를 들어, 다음과 같습니다.

```
cec transfer-site Site1 -s DEV -d UAT -r Repository1 -l LocalizationPolicy1 -x
```

## 그룹으로 사이트의 콘텐츠 항목 다운로드 또는 업로드

개발자는 한 Oracle Content Management 서버에서 다른 서버로 사이트를 전송할 때 사이트의 콘텐츠 항목을 그룹으로 다운로드/업로드할 수 있습니다.

수많은 콘텐츠 자산이 있는 페이지를 전송하려면 사이트와 콘텐츠를 분리해야 합니다. 콘텐츠 없이 사이트를 전송한 후(`cec transfer-site --excludecontent`) `cec transfer-site-content`를 사용하여 사이트의 콘텐츠를 전송합니다.

예를 들어, 다음과 같습니다.

```
cec transfer-site-content <name>
```

이 명령은 한 Oracle Content Management 서버에서 다른 서버로 엔터프라이즈 사이트 콘텐츠를 전송하는 스크립트를 생성합니다. 명령은 수많은 콘텐츠 항목을 전송하는 데 사용되며 항목은 일괄로 전송됩니다. 기본적으로 이 명령은 스크립트를 실행하지 않으며 모든 자산이 전송됩니다. `-p`를 지정하여 게시된 자산만 전송할 수 있습니다.

`-s <server>`로 소스 서버를 지정하고 `-d <destination>`으로 대상 서버를 지정합니다.

### 옵션

```
--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]
```

`--execute` 옵션이 설정되지 않은 경우 `cec transfer-site-content`가 완료된 후 생성된 스크립트 `<site name>_downloadcontent`를 실행하여 소스 서버에서 콘텐츠를 다운로드하고, `<site name>_uploadcontent`를 실행하여 다운로드한 콘텐츠를 대상 서버로 업로드합니다.

### 예제

```
cec transfer-site-content Sitel -s DEV -d UAT -r Repository1 Generate
script Sitel_downloadcontent and Sitel_uploadcontent
cec transfer-site-content Sitel -s DEV -d UAT -r Repository1 -e
Generate script Sitel_downloadcontent and Sitel_uploadcontent and
execute them
cec transfer-site-content Sitel -s DEV -d UAT -r Repository1 -n 200
cec transfer-site-content Sitel -s DEV -d UAT -r Repository1 -p
```

## OCE Toolkit으로 사이트 페이지 인덱스화

OCE Toolkit을 사용하여 사이트 페이지에서 텍스트에 대한 콘텐츠 항목을 생성하고 사이트에 대한 페이지 검색을 사용으로 설정할 수 있습니다.

다음 섹션은 OCE Toolkit으로 사이트 페이지를 인덱스화하는 방법을 설명합니다.

1. 사이트 페이지 텍스트에 대한 콘텐츠 유형 생성
2. OCE Toolkit으로 페이지 인덱스 콘텐츠 항목 생성
3. Oracle Content Management의 사이트에 콘텐츠 검색 추가

## 사이트 페이지 텍스트에 대한 콘텐츠 유형 생성

콘텐츠 유형에 대해 이름, 필수 필드 값과 유형의 기본 콘텐츠 레이아웃을 지정합니다.

- 유형 이름  
적합한 콘텐츠 유형 이름을 지정합니다.
- 필드  
다음 필드는 필수입니다.

필드 이름	필드 유형	데이터 필드 값 수	설명
site	텍스트	단일	사이트 이름
pageid	텍스트	단일	페이지 ID
pagename	텍스트	단일	페이지 이름
pageurl	텍스트	단일	페이지 URL
pagedescription	텍스트	단일	페이지 설명
keywords	텍스트	다중 (최대 없음)	OCE Toolkit index-site 명령으로 획득한, 페이지상 모든 텍스트와 페이지상 콘텐츠 항목의 모든 텍스트 필드 값

```

{{#fields}}
<div class="indextype"></div>
<div>
  <a href="{{pageFullURL}}" title="{{pagename}}">{{pagename}}</a>
</div>
{{/fields}}

```

```

content.fields.pageFullURL = SCSRenderAPI.getSitePrefix() +
content.fields.pageurl;

```

- 유형의 콘텐츠 레이아웃을 생성합니다.  
콘텐츠 레이아웃은 사이트 이름과 페이지로 이동하는 URL을 표시해야 합니다. 예를 들어 layout.html의 경우:

```

{{#fields}}
<div class="indextype"></div>
<div>
  <a href="{{pageFullURL}}"title="
{{pagename}}">{{pagename}}</a>
</div>
{{/fields}}

```

- `render.js`에서 페이지 전체 URL을 생성합니다.

```
content.fields.pageFullURL =
SCSRenderAPI.getSitePrefix() +
content.fields.pageurl;
```

- 콘텐츠 레이아웃을 유형의 기본 콘텐츠 레이아웃으로 설정합니다.

```
content.fields.pageFullURL =
SCSRenderAPI.getSitePrefix() + content.fields.pageurl;
```

## OCE Toolkit으로 페이지 인덱스 콘텐츠 항목 생성

OCE Toolkit 명령을 사용하여 페이지 인덱스 콘텐츠 항목을 생성할 수 있습니다.

필요 조건:

- OCE Toolkit이 로컬 시스템에 설치 및 설정되었습니다.
- Oracle Content Management의 사이트가 게시되었습니다.
- 사이트 페이지의 콘텐츠 항목이 사이트 채널에 게시되었습니다.

명령행 인터페이스에서 다음 OCE Toolkit 명령을 입력합니다.

```
cec index-site site name -c content type name -p
```

명령에서 *site name*은 사이트 이름이고, *content type name*은 페이지 텍스트에 대해 생성된 콘텐츠 유형이고, `-p` 옵션은 생성 후 페이지 인덱스 콘텐츠 항목을 게시할지 나타냅니다.

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
```

사용법을 보려면 `cec index-site -h`를 입력할 수 있습니다.

## Oracle Content Management의 사이트에 콘텐츠 검색 추가

검색 페이지 및 검색 필드를 사용하여 Oracle Content Management 사이트에 콘텐츠 검색을 추가할 수 있습니다.

사이트에 콘텐츠 검색을 추가하려면 다음과 같이 하십시오.

1. 사이트에 검색 페이지 추가
2. 테마에 검색 필드 추가

### 사이트에 검색 페이지 추가

사이트에 검색 페이지를 추가하고 검색 페이지에 콘텐츠 목록 구성요소를 추가할 수 있습니다.

검색 페이지 추가:

1. 사이트에 페이지를 추가하고 검색 페이지로 설정합니다.
2. 검색 페이지에 콘텐츠 목록 구성요소를 추가합니다.
3. 콘텐츠 유형을 이전에 생성한 페이지 인덱스 콘텐츠 유형으로 설정합니다.

### 테마에 검색 필드 추가

사이트의 모든 페이지에 검색 필드를 표시하려면 테마의 레이아웃 HTML 페이지에 검색 필드를 추가할 수 있습니다.

예를 들어, 다음과 같습니다.

```
<div align="center">  
<input id="searchonpage" type="text" size="30" placeholder="Search on  
page. . ."/>  
</div>
```

1. 입력 필드를 추가합니다.

```
<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. HTML 본문 끝에 JavaScript를 추가합니다.

## OCE Toolkit으로 다국어 사이트 인덱스화

OCE Toolkit을 사용하여 번역과 페이지 및 콘텐츠 항목 검색을 위해 다국어(MLS) 사이트를 인덱스화할 수 있습니다.

다국어 사이트 인덱스를 작성하고 사이트를 게시하기 전에 테스트할 수 있습니다. OCE Toolkit `cec index-site` 명령을 사용하여 다국어 사이트를 인덱스화합니다. `cec-components` 디렉토리로 이동하고 이 명령을 아무 옵션 없이 실행하여 명령에 대한 도움말 정보를 봅니다.

```

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

```

페이지 인덱스 항목은 페이지당 언어별로 존재합니다. 각 언어마다 생성된 페이지 인덱스 콘텐츠 항목은 기본 언어 페이지 인덱스 항목의 번역으로 생성됩니다. 실행 중인 사이트에서 질의를 수행하면 검색 및 콘텐츠 목록은 사이트 URL에서 언어를 가져옵니다. 그러면 자동으로 검색을 필터링합니다.

다국어 사이트를 게시하기 전에 인덱스화하고 번역해야 합니다(번역 작업 필요). [Oracle Content Management 서버에서 새 사이트 또는 자산 번역 작업 생성](#)을 참조하십시오.

기본 언어인 영어는 필수입니다. 각 지원 언어(필수 및 선택사항)에 대해 인덱스 생성을 실행하고 인덱스 항목의 번역을 생성합니다. 인덱스를 두 번 실행하면 바로 업데이트됩니다.

OCE Toolkit 명령을 사용하여 다국어 사이트를 인덱스화, 번역 및 게시하려면 다음과 같이 하십시오.

1. 사이트의 콘텐츠 유형을 생성하고 저장소에서 사용 가능하도록 합니다. [사이트 페이지 텍스트에 대한 콘텐츠 유형 생성](#)을 참조하십시오.
2. 검증 정책을 선택합니다.
  - a. 왼쪽 탐색 메뉴에서 **자산**을 누릅니다.
  - b. **자산** 메뉴에서 **지역화 정책**을 선택합니다.
  - c. 지역화 정책을 선택합니다.
  - d. 필요한 경우 지역화 정책을 수정하여 사이트 인덱스화 및 번역에 사용할 언어를 포함합니다. 예를 들어, 정책이 영어뿐이면 프랑스어와 스페인어를 추가할 수 있습니다.  
모든 번역은 영어를 기초로 합니다.
3. 번역 작업을 다운로드합니다. 사이트에서 사용된 자산만 번역할 수 있습니다.
4. 사이트를 **번역**합니다. 사이트 번역은 수동으로 또는 통합 커넥터를 통해 수행할 수 있습니다.
5. 번역이 완료되면 번역 작업을 업로드합니다.

6. `cec index-site` 명령을 사용하여 사이트를 인덱스화합니다. `-c <contenttype>`을 지정하여 페이지 인덱스 콘텐츠 유형을 설정합니다.

또한 `-p` 옵션을 지정하여 사이트를 게시할 수 있습니다. 그런 다음 라이브 사이트에 변경사항을 게시하기 전에 인덱스화 및 번역을 검증할 수 있습니다.

예를 들어, 다음 `cec index-site` 명령은 영어, 프랑스어, 스페인어를 사용하는 사이트의 사이트 인덱스를 작성합니다. 기본 언어를 비롯하여 사이트에서 지원되는 언어는 지정된 L10n 정책에서 옵니다.

```
cec index-site Demo2 -c search_content_type -p
- Logged in to remote server: server-URL
- establish user session
- get CSRF token
- site: Demo2, default language: en-US, channel token: channel-token
- site localization policy: search_localization_policy
- query site repository
- query content type search_content_type
- query site structure
- content types used in the site: search_blog
- query page data
- query content on the pages
- will create 11 page index items
- will update 0 page index items
- will remove 1 page index items
- create page index item for Blog
- create page index item for Privacy Policy
- create page index item for Search
- create page index item for Components
- create page index item for Navigtion
- create page index item for Detail Page
- create page index item for Pages
- create page index item for Page Content
- create page index item for Developing Templates
- create page index item for Themes
- add page index items to site channel
```

- remove page index items for page Search from site channel
- will create/update translate for fr-FR,es-ES
- query site structure with locale fr-FR
- query page data (fr-FR)
- query content on the pages (fr-FR)
- will create 11 page index items (fr-FR)
- will update 0 page index items (fr-FR)
- will remove 1 page index items (fr-FR)
- create page index item for Themes (fr-FR)
- create page index item for Navigation (fr-FR)
- create page index item for Pages (fr-FR)
- create page index item for Detail Page (fr-FR)
- create page index item for Search (fr-FR)
- create page index item for Page Content (fr-FR)
- create page index item for Components (fr-FR)
- create page index item for Developing Templates (fr-FR)
- create page index item for Blog (fr-FR)
- create page index item for Home (fr-FR)
- create page index item for Privacy Policy (fr-FR)
- add page index items to site channel
- set page index items in fr-FR as translated
- remove page index items for page Search from site channel
- query site structure with locale es-ES
- query page data (es-ES)
- query content on the pages (es-ES)
- will create 11 page index items (es-ES)
- will create 0 page index items (es-ES)
- create page index item for Pages (en-ES)
- create page index item for Home (en-ES)
- create page index item for Themes (en-ES)
- create page index item for Components (en-ES)
- create page index item for Privacy Policy (en-ES)
- create page index item for Detail Page (en-ES)
- create page index item for Page Content (en-ES)
- create page index item for Navigation (en-ES)
- create page index item for Developing Templates (en-ES)
- create page index item for Search (en-ES)
- create page index item for Blog (en-ES)
- add page index items to site channel
- set page index items in es-ED as translated
- publish job submitted
- publish in proogress
- publish in progress
- publish page index items finished

7. 번역을 포함하는 사이트를 게시합니다.

## 쉬운 구성요소 개발을 위해 단순화된 구성요소 생성

OCE Toolkit을 사용하여 쉬운 개발을 위해 단순화된 구성요소를 생성합니다.

SimpleHTML은 OCE Toolkit에 제공된 단순화된 구성요소로, 사용자정의 구성요소 개발을 위해 간편한 시작점을 제공합니다.

```
cec create-component -f SimpleHTML
```

JET 구성요소의 샘플을 사용하여 시작할 수도 있습니다.

```
cec create-component MyComp -f JET-CCA-Demo-Card
```

## 사이트 컴파일 서비스 설정

컴파일 서버에 대한 Docker 이미지를 생성하여 Oracle Content Management에서 사이트 컴파일 서비스를 설정할 수 있습니다. 이 서비스는 게시 전에 사용할 컴파일 서버를 정의하는 옵션을 제공합니다.

컴파일 서버에는 **관리>시스템>사이트 및 자산 페이지의 컴파일 끝점 URL** 필드에 URL 형태의 끝점이 있습니다. 여기서 서버에 등록할 전체 URL을 입력한 후 **테스트**를 눌러 끝점을 검증할 수 있습니다. 컴파일 끝점 URL 설정을 참조하십시오.

사이트 컴파일 서비스는 사이트 페이지를 컴파일하여 게시된 사이트에 액세스할 때 HTML 페이지가 반환되도록 합니다. Oracle Content Management 서버가 사이트를 게시할 때 사이트 컴파일 서비스가 구성된 경우 서비스를 호출하여 페이지를 컴파일합니다.

사이트의 끝점을 지정한 후 **사이트 속성 대화상자의 정적 사이트 전달** 탭에서 자동 컴파일을 사용으로 설정할 수 있습니다. 사이트를 게시하면 컴파일 서버를 통해 컴파일됩니다.

사이트 컴파일 서비스는 Toolkit 명령의 확장입니다. `cec compilation-server`를 자체 실행할 수 있지만, Docker 이미지를 통해 표준 모드로 컴파일 서버를 생성하고 서비스 구성을 조정할 수 있습니다.

Docker 이미지를 생성하고 Docker 파일을 게시하려면 다음과 같이 하십시오.

1. GitHub에서 Docker 이미지 정보를 다운로드합니다.  
GitHub에서 정보를 다운로드하면 컴파일 서버 아래에 추가 3개 파일이 있습니다. Docker 이미지 중 하나는 Docker 파일 및 readme 파일을 포함한 컴파일 서버입니다.
2. `run.sh` 명령은 Docker 이미지가 생성된 후 실행됩니다.  
이 명령은 OCE Toolkit을 다운로드하여 설치하고 소스 디렉토리를 생성합니다.
3. 그런 다음 환경을 사용자정의하고 `run.sh` 명령을 실행할 수 있습니다. 사용할 포트, 시간 초과 및 기타 변경할 사항을 지정하고 `.sh` 명령을 업데이트하고 사용할 수 있습니다.
4. Docker 이미지 정보를 다운로드하고 컴파일 서버 디렉토리를 변경한 후 Docker 파일을 빌드할 수 있습니다.  
이미 OCE Toolkit을 다운로드한 경우 다시 다운로드하지 않으려면 `no-cache` 옵션을 제거할 수 있습니다.
5. Docker 이미지를 나열합니다.
6. Docker 이미지가 사용 가능해지면 등록합니다.
7. 작동하는지 확인합니다.  
예외사항 파일 REST API로 이동하여 지원되는 버전을 확인할 수 있습니다. 이 테스트를 통해 실행되는지 검증할 수 있습니다.
8. 이제 서버를 등록해야 합니다. 서버 이름의 첫 부분이 필요하며 그 안의 모든 것을 처리합니다.

9. 시스템 디렉토리 및 사이트 자산으로 돌아가서 컴파일 끝점을 등록하고 **저장**을 누를 수 있습니다.
10. 사이트 속성으로 이동하여 게시할 항목과 게시할 시기를 지정합니다.

## 사이트 컴파일로 사이트 페이지의 런타임 성능 향상

Oracle Content Management의 사이트를 컴파일하면 사이트 페이지의 런타임 성능 및 동작이 향상될 수 있습니다. 이를 위해 사이트의 각 페이지에 대해 원본 페이지와 똑같이 동작하는 정적 HTML 파일을 생성하여 컴파일하면 됩니다.

### 사이트 컴파일 개요

사이트 "페이지"를 구성하는 메타데이터 파일은 컴파일 중에 결합되므로 일반적으로 런타임에 필요한 서버 요청을 피할 수 있습니다. 페이지 뷰당 렌더링 비용을 일회성 컴파일타임 비용으로 효과적으로 이동하는 것입니다.

런타임 요청 수를 줄여서 성능을 향상시킬 수 있습니다. 질의 결과를 페이지로 직접 컴파일할 수 있기 때문에 모든 콘텐츠 질의를 피할 수 있습니다.

페이지가 즉시 렌더링되므로 동작이 향상될 수 있습니다. 따라서 "FOUC(스타일이 지정되지 않은 콘텐츠의 깜빡임)" 또는 페이지에 구성요소가 불확실한 순서로 표시되는 문제(바닥글이 즉시 나타나고 페이지의 다른 항목이 렌더링되면서 페이지 아래로 이동)를 피할 수 있습니다.

Oracle Content Management 사이트 내의 페이지를 컴파일하려면 사이트를 템플릿으로 익스포트한 다음, OCE Toolkit을 사용하여 템플릿을 컴파일하고 생성된 정적 페이지를 원래 사이트로 업로드해야 합니다. 다음 단계는 템플릿 컴파일 프로세스를 안내하고 이 모델을 사용하여 사이트를 컴파일하는 방법을 알려줍니다.

### 사전 렌더링과 상호작용

사전 렌더링은 검색 엔진에 적합한 결과를 반환하는 데 중점을 둡니다. 사전 렌더링으로 생성된 페이지는 원본 페이지처럼 실행되지 않으며 JavaScript도 실행되지 않습니다. 단순히 브라우저에서 사전 렌더링된 HTML을 반환하여 SEO 텍스트 검색을 지원합니다.

사이트 컴파일을 통해 생성된 정적 페이지는 브라우저에서 실행되고 원래 동적 사이트 페이지와 똑같이 동작합니다. 여기서 비컴파일 구성요소를 동적으로 실행하고, 컴파일타임에 데이터를 수정하는 대신 결과에 완전한 최신 데이터를 원하는 사용자를 위해 런타임 질의를 실행합니다.

테넌트 관리자가 사전 렌더링 솔루션을 사용으로 설정한 경우 정적 페이지도 사전 렌더링 서버를 통해 전달할 수 있습니다. 인덱서와 크롤러가 정적 페이지 부분을 처리하며, 콘텐츠 목록처럼 여전히 동적으로 렌더링할 수도 있습니다.

콘텐츠 질의는 서식 있는 텍스트 필드 값을 포함하는 콘텐츠 항목을 반환하므로 콘텐츠 레이아웃이 콘텐츠 목록에 사용될 때 별도로 값을 인출하지 않아도 됩니다.

### 컨트롤러 사이트 페이지 렌더링

`cec compile-template` 명령을 사용하여 Oracle Content Management 사이트 페이지 컨트롤러를 거치지 않고 브라우저에서 직접 사이트 페이지를 렌더링할 수 있습니다.

정적 HTML 페이지 생성을 위해 컴파일러를 사용하지 않고 Oracle Content Management 사이트는 컨트롤러 모델을 사용하여 페이지를 렌더링합니다. 여기에는 실제로 브라우저에

페이지를 표시하기 전에 사이트 정보를 얻기 위한 수많은 요청, 사이트 내의 페이지와 페이지 렌더링에 사용되는 템플릿이 관여합니다.

### 주:

controller.js 파일을 가져오는 등의 더 많은 요청이 관여합니다.

## 컴파일된 사이트 페이지 렌더링

cec compile-template 명령을 사용하여 모든 단계를 정적 HTML 파일로 컴파일하고 해당 서버 요청을 줄이거나 없애서 런타임 사이트의 성능을 향상시킬 수 있습니다. 또한 런타임에 변경할 필요 없는 리소스에 대한 요청을 더 줄여서 정적 페이지로 컴파일할 수 있습니다.

기본 컴파일 프로세스는 기존 런타임 컨트롤러의 작업을 맡아서 사이트의 각 page.json 파일에 대한 물리적 HTML 페이지를 생성합니다. 그런 다음 이 페이지를 사이트에 배치할 수 있고 사이트와 함께 게시되어 해당 페이지 URL에 도달할 때 사용됩니다.

## 템플릿 컴파일

사이트를 컴파일하려면 사이트 내의 테마 및 구성요소에 액세스할 수 있어야 합니다. 사이트 템플릿은 사이트의 패키징 모델입니다. 템플릿에는 사이트 이전 및 실행에 필요한 모든 리소스가 포함됩니다.

사이트 템플릿을 사용하여 OCE Toolkit 환경으로 사이트를 익스포트하면 템플릿 내의 사이트를 컴파일할 수 있습니다. 다음 섹션은 OCE Toolkit에 시드된 **BlogTemplate** 내의 사이트 컴파일을 안내합니다.

## 설정

cec compile-template 명령은 OCE Toolkit을 통해 제공되며 툴킷을 설치할 때 설치됩니다. OCE Toolkit 설치를 위한 표준 지침을 따르십시오.

OCE Toolkit이 설치된 후에 cec 명령을 실행하여 템플릿을 생성할 수 있습니다.

1. OCE Toolkit 개발 디렉토리를 설치합니다.

```
mkdir cec-src
cd ./cec-src
cec install

# Start up the CEC Toolkit Development server for testing
cec develop &
```

2. 템플릿을 생성합니다.
 

시드된 BlogTemplate가 사용자정의 컴파일러로 업데이트되었습니다. 이 소스에서 템플릿을 생성합니다.

```
# create a new template (must be in the cec-src folder if not already
there)
cd ./cec-src
cec create-template BlogTemplate -f BlogTemplate
```

시드된 **BlogTemplate**는 사용자정의 컴파일러로 이전 템플릿을 확장합니다.

- 콘텐츠 레이아웃 컴파일러

```
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
```

- 페이지 레이아웃 컴파일러

```
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
```

## 템플릿 컴파일

템플릿을 컴파일하면 템플릿 내의 사이트에 대한 `structure.json` 파일에 각 페이지의 정적 HTML 페이지가 생성됩니다.

컴파일 단계에서 페이지 메타데이터(예: `100.json`)를 페이지 레이아웃(예: `about.html`)과 결합하므로 런타임에 컨트롤러 없이 즉시 HTML 페이지가 표시됩니다. 사용자정의 컴파일러를 통해 이전의 동적 항목(예: 탐색 및 구성요소)을 페이지로 컴파일하면 런타임 JavaScript 실행을 더 줄일 수 있습니다.

`cec compile-template` 명령은 사이트의 각 페이지에 대해 다음 동작을 수행합니다.

- 사이트 및 페이지 메타데이터 읽기
- 페이지 레이아웃을 읽고 페이지 레이아웃 컴파일러 적용
- 페이지상 각 슬롯에 대해:
  - `page.json` 파일에 정의된 그리드로 슬롯 확장
  - 슬롯 내의 각 구성요소에 대해:
    - \* 구성요소 컴파일러 적용
    - \* 생성된 구성요소 마크업을 슬롯 내의 해당 위치에 삽입
- 컴파일된 마크업에서 매크로를 확장하고 런타임에 Oracle Content Management 렌더러가 사용할 SCS JavaScript 객체 삽입
- 컴파일된 페이지 마크업을 `src/templates/<template>/static` 폴더 아래에 저장

사이트를 컴파일하려면 다음을 실행합니다.

```
> 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.

```

## 컴파일 오류

컴파일 동안 정보, 경고, 오류라는 세 가지 유형의 메시지를 받습니다.

1. 정보 메시지는 사용자가 인지할 문제에 관한 것이지만 다음 메시지는 컴파일의 일부로 예상됩니다.
  - 런타임에 렌더링할 위치 표시자 콘텐츠 항목.
  - 누락된 페이지 레이아웃 컴파일러. 페이지 레이아웃에 동적 요소가 없을 수 있으므로 누락되어도 문제로 간주되지 않습니다.
  - "render on access"로 표시된 항목. 즉, 사이트 개발자는 페이지가 컴파일되더라도 이 구성요소가 런타임에 동적으로 렌더링되기를 원합니다.



2. 경고 메시지는 실행 중인 사이트의 성능에 영향을 미치는 문제에 관한 것이며 가능한 수정해야 합니다. 이 범주의 항목은 다음과 같습니다.
  - 누락된 콘텐츠 레이아웃 또는 사용자정의 구성요소 컴파일러. 이것이 없으면 구성요소가 예전과 같이 동적으로 페이지에 렌더링됩니다.
  - 누락된 콘텐츠 레이아웃 맵. 콘텐츠를 항목을 렌더링하기 위해 시스템 기본 콘텐츠 레이아웃을 사용하고 싶지는 않습니다.
3. 오류 메시지는 컴파일 실패를 나타내는 문제에 관한 것입니다. 페이지는 가능한 계속 컴파일되지만 전체 컴파일은 오류와 함께 종료됩니다.
  - 대부분 사용자정의 컴파일러의 JavaScript 오류로 인해 발생하며 모든 오류를 해결해야 합니다.



#### 주:

반복 메시지를 줄이기 위해 여러 페이지에 발생하든 관계없이 동일 메시지가 컴파일당 한번만 나타납니다.

앞의 예제에는 다음과 같은 컴파일 종료 상태가 있습니다.

```
Compilation completed with 0 errors and 3 warnings.
to display warnings, run with --verbose (-v) option.
```

"-v" 옵션과 함께 컴파일 단계를 재실행하면 다음 출력이 표시됩니다.

```
> 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-e606ca18alf4
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
```

## 세부정보 페이지 컴파일

세부정보 페이지 컴파일은 컴파일 중에 발생한 모든 콘텐츠 항목을 조합합니다. 그런 다음 새 세부정보 페이지의 URL을 정의하는 슬러그 값을 사용하여 발견한 각 콘텐츠 항목의 세부정보 페이지를 재컴파일합니다.

앞의 컴파일 출력은 다음 두 섹션에 나타납니다.

1. 페이지 컴파일
2. 세부정보 페이지 컴파일

앞의 예제에서 **Starter-Blog-Author** 페이지에 대한 콘텐츠 레이아웃 맵 항목이 없다는 경고가 표시됩니다. 그러나 **Starter-Blog-Author**에 대한 세부정보 페이지는 원하지 않습니다. 세부정보 페이지는 **Starter-Blog-Post** 콘텐츠 항목에만 필요합니다. 이 오류를 제거하기 위해 해당 설정에서 참조된 명시적 세부정보 페이지가 없는 경우, 다음 옵션으로 세부정보 페이지 컴파일에서 콘텐츠 항목을 제외할 수 있습니다.

```
--noDefaultDetailPageLink, -o Do not generate compiled detail page for
items/content lists
    that use the default detail page.
```

따라서 기본 세부정보 페이지를 사용하는 세부정보 페이지 생성을 제외하고 컴파일을 재실행하면 다음 출력이 생성됩니다.

```
> 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

```

## 특정 페이지 컴파일

템플릿의 모든 페이지를 컴파일할 필요는 없으며 컴파일할 페이지를 선택할 수 있습니다. 이는 전체 사이트를 계속해서 재컴파일하지 않고 특정 페이지를 디버그하거나 업데이트하려는 경우 유용합니다.

특정 페이지를 컴파일하려면 `--pages (-p)` 옵션 뒤에 컴파일할 페이지 목록을 사용합니다.



주:

참고: 목록 참조 세부정보 페이지에 콘텐츠 항목이 있는 경우 명시적으로 포함하지 않더라도 세부정보 페이지도 컴파일됩니다.

```

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

```

## 사이트 컴파일

앞의 단계는 로컬 템플릿을 생성하고 컴파일하는 방법의 개요를 보여줍니다. 일반적으로 실제 Oracle Content Management 사이트를 컴파일합니다.

Oracle Content Management 사이트를 컴파일하려면 사이트를 템플릿으로 익스포트한 다음 템플릿을 컴파일해야 합니다. 템플릿 패키지는 사이트뿐만 아니라, 템플릿과 함께 익스포트되는 테마 및 구성요소에서 사이트 컴파일에 필요한 사용자정의 코드가 있습니다.

## 사이트 컴파일을 위한 필요 조건

다음 단계에서는 Oracle Content Management 서버에 **BlogSite**라는 사이트를 생성했다고 가정합니다. 앞의 **BlogTemplate**를 업로드하고 이 사이트를 템플릿에서 생성할 수 있습니다.

예를 들어, 다음과 같습니다.

```
> cec upload-template BlogTemplate --server UAT

> cec create-site BlogSite --template BlogTemplate --repository
<yourRepository> --
localizationPolicy <yourLocalizationPolicy> --defaultLanguage en-US --
server UAT
```

사이트를 선택하거나 생성한 후 다음 섹션의 단계를 수행하여 사이트를 컴파일할 수 있습니다.

## 사이트 컴파일

OCE Toolkit의 `cec create-template-from-site` 명령은 사이트 템플릿을 생성하여 사이트를 컴파일할 수 있도록 합니다.

```
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]
```

사이트를 컴파일하려면 다음과 같이 하십시오.

1. 사이트에서 템플릿을 생성합니다.

```
> cec create-template-from-site BlogTemplate --site BlogSite --
includeunpublishedassets -
server UAT
```

2. 템플릿을 다운로드합니다.

```
> cec download-template BlogTemplate --server UAT
```

3. 템플릿을 컴파일합니다.

```
> cec compile-template BlogTemplate --noDefaultDetailPageLink --
verbose --server UAT
--channelToken e1bb88cdc1e025c8dd278f6b676877a3
```

 주:

사이트의 게시 채널에 대한 서버에서 사이트의 채널 토큰(--channelToken (-c) 옵션)을 가져와야 합니다. 그러면 템플릿 사이트 내의 모든 질의에 사용됩니다.

4. 컴파일된 사이트 페이지를 업로드합니다.  
컴파일된 정적 파일을 사이트의 정적 폴더로 복사합니다.

```
> cec upload-static-site-files src/templates/BlogTemplate/static --site BlogSite --server UAT
```

**비컴파일 동작으로 되돌리기**

비컴파일 동작으로 되돌리려면 사이트에 업로드한 정적 파일을 제거해야 합니다.

```
> cec delete-static-site-files BlogSite --server UAT
```

그리고 사이트가 현재 게시되어 있으면 사이트를 재게시합니다. 그러면 사이트에서 "정적" 폴더가 제거되지만 사이트는 제거하지 않습니다.

## 사용자정의 컴파일러

컴파일 프로세스는 사용자정의 컴파일러로 더욱 향상될 수 있습니다. 사용자정의 컴파일러를 호출하여 페이지 레이아웃, 섹션 레이아웃, 사용자정의 구성요소 또는 콘텐츠 레이아웃을 페이지로 컴파일하면 런타임에 구성요소를 동적으로 추가할 필요가 없습니다.

구성요소에 대한 사용자정의 컴파일러가 없거나 구성요소 컴파일러가 마크업을 반환하지 않는 경우 구성요소는 한번도 컴파일되지 않은 것처럼 런타임에 렌더링됩니다.

다음 구성요소에 대해 사용자정의 컴파일러가 지원됩니다.

- 페이지 레이아웃
- 섹션 레이아웃
- 사용자정의 구성요소
- 콘텐츠 레이아웃

다음 샘플은 페이지 레이아웃 및 콘텐츠 레이아웃 컴파일러를 보여줍니다. 섹션 레이아웃 및 사용자정의 구성요소 컴파일러는 콘텐츠 레이아웃 컴파일러와 동일한 모델을 따릅니다.

사용자정의 컴파일러가 없으면 구성요소는 `render.js` 파일을 통해 렌더링됩니다.

사용자정의 컴파일러가 존재하면 호출 후 결과 HTML이 페이지에 삽입됩니다. 사용자정의 컴파일러는 런타임 구성요소에 JavaScript 하이드레이션이 필요함을 나타낼 수도 있습니다. 이 경우 `render.js` 파일 내에서 구성요소의 `hydrate()` 함수가 호출됩니다. 구성요소에 하이드레이션이 필요하지 않으면 구성요소의 `render.js` 파일이 로드되지 않습니다.

 주:

컴파일은 런타임 (게시된 사이트) 전용 기능입니다. 사이트가 편집, 탐색 또는 미리보기 모드로 표시될 때 페이지는 평소처럼 작동하며 모든 구성요소는 항상 동적으로 페이지에 추가됩니다.

## 제약 조건

cec compile-template 작업은 NodeJS 애플리케이션이며 브라우저 밖에서 실행됩니다. 페이지가 브라우저에 렌더링되지 않기 때문에 DOM 또는 창 객체가 없으며 JQuery, VueJS 또는 KnockoutJS와 같은 클라이언트측 JavaScript 라이브러리가 작동하지 않습니다.

JSDOM과 같은 라이브러리를 사용하여 DOM 객체를 생성하고 이러한 클라이언트측 라이브러리를 실행할 수 있지만 여기에는 거의 이점이 없습니다. 간단한 HTML 구문분석에는 cheerio와 같은 NodeJS HTML 구문분석기를 사용할 수 있습니다.

마크업에서 반환된 HTML은 적합한 HTML이어야 합니다. 구문분석기를 통과하여 구문분석된 HTML만 페이지에 추가됩니다. 이는 컴파일된 HTML에 슬롯을 손상시킬 수 있는 불일치 태그가 없는지 확인하는 것입니다.

## 사용자정의 컴파일러 디버그

사용자정의 컴파일러를 개발할 때 코드를 디버그해야 합니다.

cec compile-template 명령에는 --debug (-d) 옵션이 제공되며 디버거를 프로세스에 연결할 수 있도록 설정된 --inspect-brk 플래그와 함께 컴파일러가 시작됩니다. 그런 다음 표준 노드 디버깅에 따라 코드를 검사할 수 있습니다.

```
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
```

또한 --pages (-p) 옵션을 사용하여 컴파일할 페이지를 특정 페이지로 제한할 수 있습니다.

## 페이지 레이아웃 컴파일러

페이지 레이아웃 컴파일러는 해당 페이지 레이아웃을 컴파일하는 NodeJS (CommonJS) JavaScript 모듈입니다.

특정 페이지 레이아웃에 대한 페이지 레이아웃 컴파일러는 -compile.js 확장과 이름을 연관시켜 정의됩니다.

- src
  - themes
    - \* <yourTheme>
      - \* layouts
        - \* <yourPageLayout>.html
        - \* <yourPageLayout>-compile.js

페이지 레이아웃에 대한 `-compile.js`가 없으면 사용자정의 컴파일이 적용되지 않습니다.

페이지 레이아웃 컴파일러는 `Promise`를 반환하는 `compile()` 인터페이스(예: `about-compile.js`)를 구현해야 합니다.

```
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();
```



## 구성요소 컴파일러

사용자정의 구성요소 컴파일러는 모두 페이지 컴파일러와 동일한 모델을 따르며 섹션 레이아웃, 사용자정의 구성요소 및 콘텐츠 레이아웃을 위해 생성할 수 있습니다.

컴파일 동안 `cec compile-template` 명령은 구성요소의 `render.js` 파일과 동일한 위치에서 `compile.js` 파일을 찾습니다.

- `src`
  - `components`
    - \* `<yourComponent>`
      - \* `assets`
        - \* `render.js`
        - \* **`compile.js`**

이 파일이 없으면 구성요소가 컴파일되지 않고 런타임에 렌더링됩니다.

파일이 존재하면 `Promise`를 반환하는 `compile()` 인터페이스를 구현해야 합니다. 예를 들어, 다음 `Starter-Blog-Author-Summary`는 사용자정의 콘텐츠 레이아웃 컴파일러입니다.

```
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;

```

## 콘텐츠 레이아웃 컴파일러

콘텐츠 레이아웃 컴파일러를 사용하여 지정된 콘텐츠 유형의 콘텐츠 항목을 게시할 때 콘텐츠 레이아웃용 HTML을 출력할 수 있습니다.

컴파일 동안 `cec compile-content` 명령은 구성요소의 `render.js` 파일과 동일한 위치에서 `compile.js` 파일을 찾습니다.

- src
  - components
    - \* <yourComponent>
      - \* assets
        - \* render.js
        - \* **compile.js**

이 파일이 없으면 레이아웃이 컴파일되지 않고 런타임에 렌더링됩니다.

파일이 존재하면 `Promise`를 반환하는 `compile()` 인터페이스를 구현해야 합니다. 예를 들어, 다음 콘텐츠 레이아웃 컴파일러를 실행하면 결과 `layout.html`이 출력됩니다.

```

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;

```

결과 `layout.html`은 웹 애플리케이션 또는 그 밖에 콘텐츠 레이아웃의 정적 HTML 출력이 필요한 곳에 삽입할 수 있습니다. `layout.html`은 `compile.js` 파일과 동일한 디렉토리에 출력됩니다.

```

{{{style}}}
{#{fields}}
<div class="author-container">
  <span class="author-name"
  onclick='{{scsData.contentTriggerFunction}}'("starter-blog-post_author

```

```
eg \"{{author_id}}\"')>{{starter-blog-author_name}}</span>
{{/fields}}
```

## SCSCompileAPI

SCSRenderAPI와 유사하게 각 컴파일 함수로 전달되는 SCSCompileAPI가 있습니다.

다음 속성 및 함수를 포함합니다.

- 속성:
  - **navigationRoot**: 사이트의 루트인 노드의 ID입니다.
  - **navigationCurr**: 현재 페이지 노드의 ID입니다.
  - **structureMap**: 사이트 계층의 모든 노드로, ID를 통해 액세스됩니다.
  - **siteInfo**: 모든 사이트 속성입니다.
- 함수:
  - **getContentClient**: Content API 호출에 사용하도록 contentClient 인스턴스를 가져옵니다.
  - **getCustomSiteProperty**: 사이트 컴파일 중에 사용자정의 속성을 가져옵니다. 사이트 레벨의 사용자정의 데이터 및 속성을 siteinfo.json 파일에 저장하여 데이터 및 속성에 준하여 사이트를 사용자정의할 수 있습니다. 예를 들어 페이지 배경 질감, 바닥글 링크 또는 웹 사이트 이름을 변경할 수 있습니다.

## 구성요소 하이드레이션

하이드레이션은, 브라우저에서 HTML을 렌더링할 때 페이지의 컴파일된 HTML에 JavaScript 동작을 다시 추가하는 프로세스를 말합니다.

예를 들어, 페이지에 마스터/디테일로 렌더링할 두 구성요소가 있는 경우 마스터의 항목을 누르면 디테일 구성요소를 업데이트해야 합니다. 이것은 모두 페이지에서 실행되는 JavaScript에 의해 처리됩니다. 이 동작이 작동하려면 두 구성요소가 페이지에 렌더링된 후 HTML을 *하이드레이션*해야 합니다. on click 이벤트를 처리기를 마스터 구성요소의 요소와 디테일 구성요소 컨테이너의 리스너에 추가하여 이벤트에 전달된 페이로드에 따라 on click 이벤트가 발생할 때 다시 렌더링되도록 합니다.

구성요소 컴파일러는 페이지에 HTML을 삽입합니다. 구성요소가 런타임에 이벤트 처리기 따위를 추가하는 별도의 JavaScript를 실행해야 하는 경우 인라인 JavaScript 또는 Hydrate 함수의 두 가지 옵션이 있습니다. 어떤 솔루션을 선택할지는 요구사항에 따라 다릅니다.

## 인라인 JavaScript

반환되는 컴파일된 마크업에 직접 <script> 태그를 삽입할 수 있습니다. 페이지를 실행할 때 스크립트가 실행됩니다.

예를 들어, 다음과 같습니다.

```
<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 함수

JavaScript 인라인 대신, `render.js` 파일에 `hydrate` 함수를 포함할 수 있습니다. 이때 컴파일된 마크업을 반환할 때 런타임에 구성요소 *하이드레이션*이 필요합니다. 이 하이드레이션은 반복되는 `<script>` 태그를 피하고 기존 JavaScript 코드를 활용하여 이벤트를 관리할 수 있습니다.

`render.js` 파일이 로드되더라도 하이드레이션 중에 `render()` 함수는 호출되지 *않습니다*. `hydrate()` 함수만 호출됩니다.

### 주:

컴파일된 구성요소에 하이드레이션이 필요하지 않으면 구성요소의 `render.js` 파일이 로드되지 않습니다.

예를 들어, 사용자정의 콘텐츠 레이아웃 컴파일러는 - { hydrate: true }를 반환합니다.

```
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
});
```

또한 필요한 경우 사용자정의 컴파일러는 런타임에 발견될 *hydrate* 속성을 추가할 수 있습니다. 예를 들어, 다음과 같습니다.

```
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}}">
```

마지막으로, 구성요소에 하이드레이션이 필요하다면 런타임에 구성요소의 `render.js` 파일이 로드되고 `hydrate()` 함수가 호출되어 컴파일된 마크업을 포함하는 컨테이너 `<div>`를 전달합니다.

예를 들어, `render.js - hydrate()` 함수를 참조하십시오.

```
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);
  }
}
}

```

## 게시

컴파일된 정적 페이지가 생성되고 사이트의 정적 폴더에 업로드된 후에 페이지가 활성화되도록 사이트를 게시하거나 재게시해야 합니다. 마찬가지로, 비컴파일 사이트 전달 동작으로 되돌리려면 사이트에서 정적 파일을 제거한 후 게시하거나 재게시해야 합니다.

게시 중에 업로드된 정적 페이지를 전달할 수 있습니다. 이 파일은 게시 프로세스 중에 복사되므로 파일 수에 비례해서 게시 작업의 성능이 저하될 수 있습니다.

게시 작업은 현재 정적 파일 집합을 가져와서 전달할 수 있도록 합니다. 이 파일은 동적 사이트에서 발생한 변경사항과 동기화되거나 동기화되지 않을 수 있으며, 동적 사이트를 미러링하거나 미러링하지 않을 수 있습니다. 적절한 시간에 정적 파일 수집을 업데이트하는 것은 사이트 개발자의 역할입니다.

## 정적 사이트 전달 우선순위

사이트에 연관된 정적 파일이 있는 경우 이 파일은 서버로 들어오는 일치 URL에 전달됩니다. 수신 URL이 정적 파일과 일치하지 않으면 사이트의 controller.html 파일이 요청에 반환됩니다. 이는 사이트 전달의 기존 동적 모델을 따릅니다.

Oracle Content Management 사이트는 연관된 JSON 파일을 통해 301 및 302 재지정을 정의할 수도 있습니다. 재지정이 구성된 경우 재지정이 정적 파일보다 우선합니다. URL이 재지정 규칙 및 정적 파일과 둘 다 일치하면 서버에서 재지정이 전달됩니다.

사이트 전달을 위한 URL 평가는 다음 플로우를 따릅니다.

1. URL이 구성된 재지정과 일치합니까?  
그렇다면 재지정 응답을 발행합니다.
2. URL이 정적 파일에 해당합니까?  
모바일 정적 사용자 에이전트 목록이 사이트에 구성되었고 목록과 일치하는 브라우저에서 요청이 오는 경우 모바일 정적 파일을 전달합니다.
3. 그렇지 않으면 동적 사이트 controller.html 파일을 전달합니다.

 주:

모바일 정적 파일이 사이트와 연관되었고 고객이 CDN을 전달에 사용하는 경우 표준 데스크톱 요청과 별도로 모바일 브라우저 요청을 캐싱하도록 CDN(대개 Akamai)을 구성해야 합니다.

CDN이 별도의 모바일/표준 캐싱으로 구성되지 않은 경우 모바일 브라우저가 표준 응답을 수신할 수 있고 데스크톱 브라우저가 모바일 브라우저용 응답을 받을 수 있습니다.

## 캐싱 헤더

웹 서버 응답의 HTTP 헤더는 브라우저의 페이지 캐싱 방법을 결정하는 데 도움을 줍니다. 또한 정적 페이지는 캐싱 헤더와 함께 전달되어 브라우저 캐싱을 용이하게 해줍니다.

보안 사이트의 경우 다음 헤더가 응답과 함께 전송됩니다.

- Cache-Control: no-store
- Pragma: no-cache

표준 비보안 사이트의 경우 다음 헤더가 전송됩니다.

- Cache-Control: max-age=300
  - Edge-Control: !no-store,max-age=2592000,downstream-ttl=1800
- Edge-Control 헤더는 CDN 캐싱 동작을 용이하게 해줍니다.

이러한 두 영역 중 하나에서 헤더를 사용자정의한 경우 여기에 나열된 표준 헤더 대신 사용자정의 헤더가 응답에 포함됩니다.

테넌트 레벨 또는 사이트 레벨에서 해당 응답을 제어할 수 있습니다.

## 세부정보 페이지

Oracle Content Management 사이트의 세부정보 페이지는 수많은 콘텐츠 항목의 정보를 보여주는 단일 페이지입니다.

예를 들어, 동일한 세부정보 페이지를 사용하여 수많은 URL을 처리할 수 있습니다. 각 URL은 동일한 페이지 구조를 표시하지만 슬러그 값이 각각 item1.html, item2.html, item3.html인 콘텐츠 항목과 관련된 콘텐츠를 보여줍니다. 이 상황에서 cec 템플릿 컴파일러는 다음 4개 파일을 생성할 수 있습니다.

- /detail/item1.html
- /detail/item2.html
- /detail/item3.html
- /detail.html

최종 파일은 사이트를 재컴파일하고 재게시할 필요 없이 새로 게시된 자료를 웹 사이트에 표시할 수 있습니다. 이 예제에서 슬러그 값이 item4.html인 콘텐츠 항목은 사이트가 온라인이 된 후에 게시됩니다. 정적 /detail.html 페이지는 새 항목을 사이트에 동적으로 표시할 수 있습니다. URL /detail/item4.html은 detail.html 페이지를 전달하지만 item4.html 콘텐츠 항목과 관련된 콘텐츠를 보여줍니다.



cec 컴파일러는 detail.html 페이지를 생성하여 콘텐츠 항목을 표시합니다. 이러한 이유로 컴파일된 detail.html 페이지 안의 상대 URL에는 추가 상위 세그먼트(../)가 있습니다. 따라서 직접 참조하면 detail.html 페이지 자체는 제대로 표시되지 않습니다. 이러한 이유로 detail.html 페이지 자체를 참조하거나 페이지 탐색에 추가해서는 안 됩니다.

## 콘텐츠 항목을 채널에 추가

OCE Toolkit control-content 명령을 사용하여 Oracle Content Management 서버의 채널에 콘텐츠 항목을 추가할 수 있습니다.

control-content <action> 명령에는 Oracle Content Management 채널에 콘텐츠 항목을 추가하기 위한 add 동작이 있습니다.

```
cec control-content add -c Channel1 -r Repol -s UAT
```

이 명령은 Repol 저장소의 모든 항목을 등록 서버 UAT의 Channel1 채널에 추가합니다.

-s <server>로 서버를 지정하거나 cec.properties 파일에 지정된 서버를 사용할 수 있습니다.

content-usage 명령에 적합한 동작은 다음과 같습니다.

- publish
- unpublish
- add
- remove

content-usage 명령의 옵션은 다음과 같습니다.

- --channel, -c Channel [필수]
- --repository, -r Repository [<action>이 add일 때 필수]
- --server, -s 등록된 Oracle Content Management 서버
- --help, -h 도움말 표시 [부울]

control-content 명령의 예제는 다음과 같습니다.

```
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

## 모바일 장치에 대한 사이트 컴파일

OCE Toolkit을 사용하여 사이트 웹 페이지의 모바일 레이아웃을 컴파일할 수 있습니다. 동일한 콘텐츠에 대해 모바일 레이아웃은 데스크톱 페이지 레이아웃과 다를 수 있습니다. 또는 모바일과 데스크톱 레이아웃이 같을 수 있습니다.

사이트 편집기에서 모바일 장치에 대해 데스크톱 레이아웃과 동일한 페이지 레이아웃을 선택하거나, 다른 페이지 레이아웃을 지정할 수 있습니다. OCE Toolkit으로 모바일 장치용 정적 레이아웃을 별도로 컴파일할 수 있습니다.

모바일 장치에서 사이트 페이지가 다르게 보일 수 있습니다. 모바일 장치에 렌더링된 페이지에는 데스크톱 레이아웃에 있던 배너가 없을 수도 있습니다.

OCE Toolkit에서 `cec compile-template`의 도움말 페이지는 사이트 템플릿을 컴파일할 때 특정 장치를 타겟팅하기 위한 `targetDevice` 옵션을 보여줍니다.

```
C:\git\webclient\developer\test\sites-compiler\cec-install>cec compile-template --help
Usage: cec compile-template <source>

Compiles all the pages within the site of the template and places the compiled pages under the sites assets folder.
Optionally specify -s <server> to make content queries against this server (requires channelToken).
Optionally specify -c <channelToken> to use this channelToken when generating any content URLs.
Optionally specify -t <contentType> [draft | published] content to retrieve from the server type, defaults to published.
Optionally specify -p <pages> the set of pages to compile.
Optionally specify -d <debug> to start the compilation with --inspect-brk flag.
Optionally specify -r <recurse> recurse through all child pages of specified pages.
Optionally specify -l <includeLocale> include default locale when creating pages.
Optionally specify -a <targetDevice> [desktop | mobile] target device type when using adaptive layouts.
Optionally specify -v <verbose> to display all warning messages during compilation.

Options:
--server, -s          The registered CEC server
--channelToken, -c   The channel access token to use for content URLs
--type, -t           The type of content to retrieve from the server [published | draft]
--pages, -p         The list of pages to compile
--recurse, -r       Compile all child pages of those specified in the page list
--debug, -d         Start the compiler with "--inspect-brk" option to debug compilation
--noDetailPages, -e Do not generate compiled detail pages
--noDefaultDetailPageLink, -o Do not generate compiled detail page for items/content lists that use the default detail page
--targetDevice, -a  The target device type when using adaptive layouts [desktop | mobile]
--includeLocale, -l Include default locale when creating pages
--verbose, -v       Run in verbose mode to display all warning messages during compilation.
--help, -h         Show help

Examples:
cec compile-template 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>
```

사이트를 컴파일할 때 `desktop` 또는 `mobile` 용도로 컴파일할지 지정할 수 있습니다. 데스크톱 파일은 `static/_files` 아래에 놓입니다. 모바일 파일은 `static/_mobilefiles` 아래에 놓입니다.

```

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/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/themes.html
createPage: Processing pageId 110. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/developing-templates/pages.html
createPage: Processing pageId 120. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/developing-templates/navigation.html
createPage: Processing pageId 130. Preview URL: http://localhost:8085/templates/Corporate-Site-Template/developing-templates/page-content.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

```

모바일 장치용 템플릿을 컴파일한 후 OCE Toolkit 명령 `upload-static-site-files`는 모바일 파일을 지원합니다.

## 사이트 수명 주기 및 컴파일된 페이지

사이트에서 템플릿을 생성할 때 컴파일된 페이지는 템플릿에 포함되지 않습니다. 이는 사이트 개발자가 동적 페이지를 전달해야 하는데 정적 페이지가 전달되는 문제를 피하기 위한 것입니다.

그 뒤에 템플릿에서 사이트를 생성하는 경우 사이트 페이지를 컴파일하고 새 사이트로 업로드해야 합니다.

## Oracle Content Management 서버에서 새 사이트 또는 자산 번역 작업 생성

OCE Toolkit을 사용하여 Oracle Content Management에서 사이트 또는 자산의 번역 작업을 생성할 수 있습니다.

다국어 사이트를 인덱스화하기 전에 번역 작업이 필요합니다. 번역 작업을 생성하려면 다음과 같이 하십시오.

1. 사이트 페이지의 상단 메뉴에서 **번역**을 누릅니다.
2. **번역 작업 생성** 대화상자에 작업 이름을 입력하고 기본 소스 언어, 대상 언어 및 번역 작업 콘텐츠를 선택합니다.

번역 패키지에 모든 사이트 콘텐츠 및 대상 자산, 사이트 콘텐츠만, 또는 사이트의 게시 채널 대상으로 지정된 자산만 포함할지 선택할 수 있습니다.

**번역하지 않음** 텍스트 설정으로 구성된 콘텐츠 항목은 번역에서 제외합니다. 예를 들어, 제품 이름은 일반적으로 번역하지 않습니다.

3. **생성**을 눌러 번역 작업을 생성합니다.

4. OCE Toolkit 명령을 사용하여 사용 가능한 작업을 나열합니다.

```
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
US                                fr-FR,es-ES  READY      en-
                                fr-FR,es-ES

searchdemo1
US                                fr-FR,es-ES  TRANSLATED en-
```

5. 번역 작업을 다운로드합니다.

```
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. 번역 번들을 열고 번역하려는 언어의 리소스 폴더를 빌드합니다.

```
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

```

#### 7. 번역 작업을 임포트합니다.

```

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

```

## 언어 서비스 제공자를 사용하여 사이트 번역

OCE Toolkit 명령행 인터페이스 및 LSP(언어 서비스 제공자)를 사용하여 여러 언어로 사이트 번역을 관리할 수 있습니다.

사이트의 지역화 정책은 미국 영어(en-US)와 같은 기본 언어와, 독일어 및 프랑스어와 같은 하나 이상의 사이트 대체 언어를 지정합니다. 사이트의 텍스트 문자열을 지정된 대체 언어로 번역할 수 있습니다. 번역 전에 사이트의 언어를 변경해도 텍스트 문자열은 여전히 기본 언어로 나타납니다.

OCE Toolkit은 명령행 인터페이스에서 다음 번역 옵션을 제공합니다.

Translation	
cec list-translation-jobs	Lists translation
jobs.	[alias: ltj]
cec create-translation-job <name>	Creates a translation
job <name> for a site on CEC server.	[alias: ctj]
cec download-translation-job <name>	Downloads translation
job <name> from CEC server.	[alias: dtj]
cec submit-translation-job <name>	Submits translation job
<name> to translation connection <connection>.	[alias: stj]
cec ingest-translation-job <name>	Gets translated job
<name> from translation connection and ingest.	[alias: itj]
cec upload-translation-job <name>	Uploads translation job
<name> to CEC server.	[alias: utj]
cec create-translation-connector <name>	Creates translation
connector <name>.	[alias: ctc]
cec start-translation-connector <name>	Starts translation
connector <name>.	[alias: stc]
cec register-translation-connector <name>	Registers a translation
connector.	[alias: rtc]

cec list-translation-jobs 명령을 사용하여 이미 서버에 있는 번역 작업을 나열할 수 있습니다. 예를 들어, 다음과 같습니다.

```
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
```

cec 명령을 매개변수 없이 입력하거나 -h와 함께 입력하면 명령 도움말이 제공됩니다. [cec 명령행 유틸리티 사용](#)을 참조하십시오.

다음 섹션은 LSP를 사용한 사이트 번역에 대한 정보를 제공합니다.

1. [OCE Toolkit으로 번역 작업 생성](#)
2. [번역 작업 나열](#)
3. [번역 커넥터 생성](#)
4. [다국어 사이트의 사이트 맵 생성](#)
5. [언어 서비스 제공자에게 번역 작업 제출](#)

## 6. 서버에 번역 작업 업로드

# OCE Toolkit으로 번역 작업 생성

OCE Toolkit 명령을 사용하여 로컬 시스템에 사이트 번역 작업을 생성할 수 있습니다.

사이트의 새 번역 작업을 생성하려면 `cec create-translation-job` 명령을 사용합니다. 이 명령은 사이트의 모든 자산을 찾아서 해당 사이트에서 번역해야 하는 모든 항목의 zip 파일을 생성합니다.

```
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
```

번역 옵션은 [Oracle Content Management 서버에서 새 사이트 또는 자산 번역 작업 생성](#)을 참조하십시오.

## 번역 작업 나열

서버의 번역 작업을 나열하여 내 작업이 생성되었고 작업할 준비가 되었는지 확인할 수 있습니다.

```
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
```

FridayDemo 작업이 READY 상태임을 알 수 있습니다.

## 번역 커넥터 생성

LSP(언어 서비스 제공자)가 사이트 번역을 도울 수 있습니다. LSP용 번역 커넥터를 사용하여 번역 작업을 제출하고 입수할 수 있습니다.

번역 작업을 제출하기 전에 번역 커넥터를 생성해야 합니다. LSP 없이 사이트를 번역하려면 모의 실행할 번역 커넥터를 생성할 수 있습니다. `cec create-translation-`

connector 명령을 사용하여 번역 커넥터를 생성하고 cec start-translation-connector 명령을 사용하여 시작합니다.

```
cec create-translation-connector connector1
- translation connector connector1 created at <sites-toolkit folder>/cec-
components/src/main/connectors/connector1
- install connector
. . .
Start the connector: cec start-translation-connector connector1 [-p <port>]
cec start-translation-connector connector1 -p 7777
NodeJS running. . .:
Site page: http://localhost:7777
```

OCE Toolkit을 사용하여 예상 API를 통해 번역 커넥터를 실행하여 테스트합니다.

1. OCE Toolkit에 커넥터를 등록합니다.

```
>cec register-translation-connector
```

2. 툴킷을 열고 "Translation Connections" 페이지로 이동합니다.

```
>http://localhost:8085/public/translationconnections.html
```

3. 번역 커넥터 검증 페이지의 단계를 실행합니다. 이 단계는 커넥터 환경에서 /data 폴더의 translationBundle.zip 파일을 사용하여 커넥터를 검증합니다.

번역 커넥터 SDK를 사용하여 Oracle Content Management용 번역 커넥터를 개발할 수 있습니다. 이 SDK는 번역 커넥터 API의 샘플 NodeJS를 구현합니다. 샘플은 Oracle Content Management 번역 작업 zip 파일을 수락하고, 파일의 모든 리소스를 번역하고, 모든 번역이 포함된 새 zip 파일을 반환합니다.

SDK에서 사용자가 실제 문자열을 번역하려면 LSP에 액세스할 수 있어야 합니다. SDK에 포함된 모의 LSP 서버는 간단히 대상 로케일을 문자열 앞에 추가하여 LSP의 응답을 모방합니다.

번역 커넥터 SDK는 세 가지 주요 모듈로 구성됩니다.

- **커넥터:** 필요한 Oracle Content Management 번역 커넥터 API를 구현하는 번역 커넥터입니다.
- **작업 관리자:** 언어 서비스 제공자가 작업을 번역하는 동안 커넥터 작업 상태를 유지 관리하는 파일 시스템 기반의 샘플 작업 관리자입니다.
- **제공자:** LSP가 번역할 문서를 제출하고 번역된 문서를 읽어들이기 위해 필요한 특정 API 집합을 구현합니다.

모의 번역 제공자 JS를 복사하여 그 안의 모든 메소드를 구현할 수 있습니다.

## 다국어 사이트의 사이트 맵 생성

OCE Toolkit을 사용하여 다국어 사이트의 사이트 맵을 생성하고 사이트에 맵을 게시할 수 있습니다.

cec create-site map <site> 명령을 사용하여 Oracle Content Management 서버에 다국어 사이트의 사이트 맵을 생성할 수 있습니다. 예를 들어, 다음과 같습니다.

```
cec create-site-map Site1 -u http://www.example.com/site1
```



이 명령은 사이트 구조를 순회하고, 사이트 페이지 계층과 일치하는 사이트 맵 계층을 생성하고, Oracle Content Management 서버에 지정된 사이트 URL에 사이트 맵을 생성합니다.

명령 옵션은 다음과 같습니다.

```
--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> 옵션에 적합한 값은 다음과 같습니다.

- always
- hourly
- daily
- weekly
- monthly
- yearly
- never
- auto

cec create-site-map 명령의 예제는 다음과 같습니다.

```
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
```

사이트 맵을 게시하기 위해 사이트 업데이트가 생성되고 사이트 맵이 업데이트된 다음 업데이트가 커밋됩니다.

## 언어 서비스 제공자에게 번역 작업 제출

OCE Toolkit은 번역 작업을 시작하도록 언어 서비스 제공자에게 보낼 수 있는 zip 파일을 제공합니다.

번역 커넥터를 통해 LSP에 번역 작업을 제출할 수 있습니다. 커넥터가 zip 파일의 압축을 풀고 개별 파일을 모두 LSP에 제출해야 하기 때문에 다소 시간이 걸립니다. 그러면 LSP가 번역 작업의 프로젝트를 생성할 수 있습니다. 파일이 프로젝트로 импорт된 후에 번역할 파일 선택을 시작할 수 있습니다. 그러면 LSP가 번역 상태의 모니터링을 시작합니다.

상태를 확인하려면 cec list-translation-jobs 명령을 옵션 없이 사용하여 로컬에 번역 작업을 나열합니다. 작업 상태가 READY TO INGEST이면 LSP에서 zip 파일을 다운로드하여 번역 작업을 입수할 수 있습니다. 번역 커넥터가 zip 파일을 LSP에 제출했고,

LSP가 파일 목록을 번역했으며, 커넥터는 다운로드 및 입수가 가능한 zip 파일로 LSP로부터 다시 파일을 읽어들이었습니다.

```
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
```

zip 파일을 입수하면 번역 작업이 커넥터에서 OCE Toolkit으로 되돌아옵니다.

```
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
```

zip 파일을 입수한 후 로컬에 번역 작업을 나열하면 번역 작업의 상태는 TRANSLATED입니다.

```
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
```

번역된 작업을 Oracle Content Management 서버로 업로드할 수 있습니다. 일반적으로 작업은 초기 빠른 번역을 거치고 돌려보내서 검토합니다. LSP가 반환한 번역 작업을 입수하고 번역을 교정하고 번역 작업을 다시 제출하는 과정을 거쳐 사이트 번역을 완료하려면 몇 주가 걸릴 수 있습니다.

## 서버에 번역 작업 업로드

번역 작업을 입수한 후 Oracle Content Management 서버로 업로드한 다음 사이트에서 번역을 확인할 수 있습니다.

cec upload-translation-job 명령을 사용하여 번역 zip 파일을 서버로 업로드합니다.

```
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
```

번역 작업을 업로드한 후 서버의 작업 상태는 INPROGRESS입니다.

```
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
```

번역을 검증하려면 번역 중인 사이트의 자산에서 텍스트 문자열을 확인할 수 있습니다.

# 기부 부록

다음 부록을 사용할 수 있습니다.

- [자습서: Knockout으로 구성요소 개발](#)
- [Sites SDK 참조](#)
- [문제 해결](#)

# 26

## 자습서: Knockout으로 구성요소 개발

이 자습서는 표준 Knockout ViewModel 및 템플릿 기능을 활용하는 일련의 JavaScript 객체를 사용하여 Oracle Content Management 구성요소 카탈로그에 저장된 구성요소를 생성하는 과정을 안내합니다.

- Knockout으로 구성요소 개발 소개 및 필요 조건
- 단계 1: 구성요소 생성
- 단계 2: 로컬 구성요소 렌더링의 구조 검토
- 단계 3: 로컬 구성요소 설정의 구조 검토
- 단계 4: 구성요소에 새 속성 표시
- 단계 5: 트리거 등록
- 단계 6: 트리거 발생
- 단계 7: 작업 등록
- 단계 8: 작업 실행
- 단계 9: 각 구성요소 인스턴스에 대한 고유 제목 생성
- 단계 10: 인라인 편집과 함께 중첩 구성요소 사용
- 단계 11: 다양한 레이아웃 지원
- 단계 12: 사용자정의 스타일 정의
- 단계 13: 인라인 프레임으로 구성요소 렌더링
- 단계 14: 구성요소가 인라인 프레임으로 렌더링될 때 사용자정의 스타일 사용
- 단계 15: 페이지 실행 취소 및 재실행 동작과 통합
- 단계 16: 자산 관리
- 자습서 검토

### Knockout으로 구성요소 개발 소개 및 필요 조건

이 자습서는 표준 Knockout JS ViewModel 및 템플릿 기능을 활용하는 JavaScript 객체를 사용하여 샘플 구성요소를 생성하는 단계 및 확인 절차를 제공합니다.

이 단계에서 참조된 코드(구성요소를 생성할 때 시드된 파일에 제공)를 가져와 .html 템플릿과 JavaScript viewModel만 자체 코드로 업데이트할 수 있어야 합니다.

#### 주:

Oracle Content Management는 구성요소 생성에 사용할 JavaScript 기술을 지시하지 않지만, 일반적으로 어떤 JavaScript 프레임워크를 선택하든 각 구성요소 구현에 대해 팩토리 JavaScript 함수는 동일합니다.

## 필요 조건

이 자습서는 구성요소 구현에만 초점을 맞춥니다. 구성요소에 대한 더 일반적인 정보는 구성요소 개발을 참조하십시오.

이 자습서의 단계를 완료하려면 다음 요구사항을 충족해야 합니다.

- 사이트 및 구성요소를 생성할 수 있는 권한으로 Oracle Content Management 인스턴스에 액세스할 수 있어야 합니다.
- Oracle Content Management 인스턴스 서버가 Oracle Content Management 데스크톱을 사용하거나 사용자정의 구성요소를 사용하는 로컬 컴퓨터와 동기화되었습니다. Developer Cloud Service로 사용자정의 구성요소 개발을 참조하십시오.

또한 다음 JavaScript 개념 및 프레임워크에 익숙해야 합니다.

- JavaScript 브라우저 디버깅
- JavaScript 클로저
- JavaScript 비동기 모듈 정의(AMD) 개발
- RequireJS 및 KnockoutJS 프레임워크

[단계 1: 구성요소 생성](#)을 계속합니다.

# 단계 1: 구성요소 생성

이 단계는 Oracle Content Management에서 사용자정의 구성요소를 생성하는 방법을 설명합니다.

사용자정의 구성요소를 생성할 때 Oracle Content Management에서 사용 가능하도록 등록해야 합니다. Oracle Content Management에 구성요소를 알려려면 사이트 작성기의 [구성요소] 페이지에서 구성요소를 등록합니다.

등록할 구성요소는 두 가지 유형이 있습니다.



- **로컬 구성요소:**
  - Oracle Content Management 인스턴스 서버에 파일이 저장된 구성요소입니다.
  - 주 장점은 파일이 사이트에 있기 때문에 교차 도메인 또는 교차 프로토콜 문제를 걱정할 필요가 없다는 것입니다.
  - 단점은 Oracle Content Management 서버에서 중간 계층 논리를 실행할 수 없기 때문에 CORS를 지원하는 원격 서버에 REST API를 사용해야 한다는 것입니다.
  - 이 유형의 구성요소는 페이지에 직접 내장될 수 있습니다. 또는 인라인 프레임으로 페이지에 구성요소를 렌더링하도록 선택할 수 있습니다.
- **원격 구성요소:**
  - 원격 서버에 파일이 저장되고 구성요소의 렌더링 및 설정 패널에만 URL을 등록하는 구성요소입니다.
  - 원격 구성요소는 구성요소의 콘텐츠를 생성할 때 서버측 논리를 실행해야 하는 경우 장점을 제공합니다.
  - 단점은 이 URL에 액세스하려면 교차 도메인 및 보안 문제를 해결해야 한다는 것입니다.
  - 원격 구성요소는 항상 인라인 프레임으로 페이지에 렌더링됩니다.

로컬 구성요소를 생성 및 등록하려면 다음과 같이 하십시오.

1. Oracle Content Management 홈 페이지에서 **개발자**를 누릅니다.  
개발자 페이지가 표시됩니다.
2. **모든 구성요소 보기**를 누릅니다.
3. 메뉴에서 **로컬 구성요소 생성**을 선택합니다.
4. 구성요소의 이름(예: **A\_Local\_Component**)을 입력합니다.
5. 선택적 설명을 입력합니다.
6. **생성**을 누릅니다.  
이 작업을 마치면 구성요소 목록에 A\_Local\_Component라는 구성요소가 표시됩니다.

### 단계 1의 결과 확인

구성요소를 성공적으로 생성했으므로 이제 생성한 사이트에 대한 구성요소 팔레트에서 구성요소를 볼 수 있습니다. 다음 단계에 따라 구성요소 생성을 검증하십시오.

1. localComponentTest라는 사이트를 생성합니다.
2. 사이트를 선택하고 **열기**를 누릅니다.
3. **편집**을 누릅니다.
4. 사이트의 업데이트를 생성하고 이름과 선택적 설명을 제공합니다.
5. 사이트의 페이지를 선택합니다.
6. 측면 팔레트에서 을 누르고 **사용자정의**를 선택하여 사용자정의 구성요소 목록을 표시합니다.
7. 사용자정의 구성요소 목록에서 A\_Local\_Component를 선택하고 페이지 위에 끌어 놓습니다.  
이제 생성한 로컬 구성요소에 대한 기본 렌더링이 표시되어야 합니다.
8. 방금 페이지 위에 놓인 구성요소의 배너에서 을 선택합니다.
9. **설정**을 선택합니다.
10. 구성요소의 맞춤을 변경하고 스타일을 설정합니다.
11. 설정 패널을 닫습니다.

다음 단계는 사용자정의 구성요소의 빌드 방법을 안내하고 목적에 맞게 수정하는 방법을 설명합니다. [단계 2: 로컬 구성요소의 구조 검토](#)를 계속합니다.

## 단계 2: 로컬 구성요소 렌더링의 구조 검토

이 단계에서는 로컬 구성요소에 대해 생성된 기본 파일의 구조를 검토합니다.

간단한 Hello World 예제의 경우 4개의 JavaScript 객체와 코드 행 수가 너무 많아 보일 수 있지만, 이는 더 복잡한 구성요소를 빌드하고 Oracle Cloud Sites Service 페이지 수명 주기와의 상호작용을 다루기 위한 기초를 제공하는 것입니다.

로컬 구성요소의 구조를 검토하려면 다음과 같이 하십시오.

1. Oracle Content Management 홈 페이지에서 **개발자**를 누릅니다.  
개발자 페이지가 표시됩니다.

2. 모든 구성요소 보기를 누릅니다.
3. 메뉴에서 로컬 구성요소 생성을 선택합니다.
4. 구성요소의 이름(예: **A\_Local\_Component**)을 입력합니다.
5. 선택적 설명을 입력합니다.
6. 생성을 누릅니다.

이 작업을 마치면 구성요소 목록에 `A_Local_Component`라는 구성요소가 표시됩니다.

1. Oracle Content Management 데스크톱 동기화 클라이언트를 사용하여 구성요소를 찾아 파일 시스템과 동기화합니다.

데스크톱 클라이언트가 없는 경우 Oracle Content Management 인터페이스의 [구성요소] 페이지에서 모든 구성요소를 보고 구성요소를 선택하고 드릴다운하여 파일을 볼 수 있습니다.

2. 구성요소 아래의 파일을 나열하면 다음 파일이 표시됩니다.

```
assets
  render.js
  settings.html
  appinfo.json
  _folder_icon.jpg
```

3. `/assets` 디렉토리 아래의 `render.js` 파일을 엽니다.

`render.js` 파일의 요점은 다음과 같습니다.

- 페이지에 “필수”가 되도록 JavaScript AMD 모듈로 구조화되어 있습니다.
- 이미 Oracle Content Management 페이지의 일부로 로드된 KnockoutJS 및 JQuery에 대한 참조도 포함합니다.

`render.js` 파일의 구조를 고려하십시오.

`render.js` 파일의 내용에는 필수 Oracle Content Management 구성요소 API를 구현하는 두 개의 JavaScript 객체인 `sampleComponentFactory` 및 `SampleComponentImpl`이 있습니다. 이러한 객체는 KnockoutJS 기반 구성요소를 생성하기 위한 구현의 예입니다. 이러한 객체의 구현은 사용할 기술에 따라 변경됩니다.

- `sampleComponentFactory`
  - 이 객체는 `render.js` AMD 모듈에 의해 반환됩니다.
  - 매우 간단한 팩토리 객체이며 단일 `createComponent()` 인터페이스를 구현합니다.
  - 더 복잡한 구현에는 전달된 `args` 값을 사용하여 `viewModel` 매개변수에 따라 다른 구성요소 구현을 반환할 수 있습니다. 이렇게 하면 사이트 작성기와 비교해 런타임에 구성요소를 훨씬 경량으로 구현할 수 있습니다.
- `SampleComponentImpl`
  - 이 객체 내의 주 함수는 페이지에 구성요소를 렌더링하는 데 사용되는 `render` 함수입니다.
 

Knockout 구성요소를 페이지에 렌더링하기 위해 `render` 함수는 동적으로 페이지에 템플릿을 추가한 다음, `viewModel` 바인딩을 템플릿에 적용합니다.
  - 나머지 구현에서는 `viewModel` 매개변수 및 템플릿의 초기화와, 페이지와 구성요소 간의 메시징 처리를 다룹니다.



render.js 파일의 마지막 두 객체인 sampleComponentTemplate 및 SampleComponentViewModel은 구성요소에 대한 사용자정의 구현을 제공합니다. 이 구현은 요구사항에 따라 달라집니다.

- sampleComponentTemplate
  - 이 객체는 KnockoutJS 템플릿 생성을 제공합니다. 항목을 표시하려고 시도하기 전에 구성요소는 모든 데이터가 초기화될 때까지 기다립니다.
- SampleComponentViewModel
  - viewModel은 구성요소 대신 Oracle Content Management에 저장된 정보를 읽어들이고, 해당 데이터에 따라 구성요소를 적절히 레이아웃하는 방법을 선택합니다.
  - 구성요소 대신 저장된 메타데이터에 대한 액세스를 처리하기 위해 템플릿에서 사용되는 일반적인 Knockout observable:

```
self.imageWidth = ko.observable('200px');
self.alignImage = ko.observable();
self.layout = ko.observable();
self.showTopLayout = ko.observable();
self.showStoryLayout = ko.observable();
```

- 트리거 및 작업 통합:

**트리거:** 구성요소에서 Oracle Content Management 트리거를 발생시키는 함수로, 페이지상 다른 구성요소의 작업에 바인드할 수 있습니다.

```
self.imageClicked = function (data, event) {
  self.raiseTrigger("imageClicked"); // matches appinfo.json
};
```

**작업:** 주어진 페이로드로 작업을 실행하라고 구성요소에 지시할 때 콜백을 처리하는 함수입니다.

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

요청 시 등록된 작업을 실행하는 콜백입니다.

```
SitesSDK.subscribe(SitesSDK.MESSAGE_TYPES.EXECUTE_ACTION,
$.proxy(self.executeActionsListener, self));
```

- 구성요소 수명 주기 구독:

- \* 구성요소 초기화: 모든 데이터가 인출될 때까지 구성요소가 렌더링되지 않도록 합니다. 이는 Knockout observable을 통해 처리됩니다.

```
self.componentLayoutInitialized = ko.observable(false);
self.customSettingsDataInitialized = ko.observable(false);
```

필수 속성에 대한 초기 값을 가져옵니다. 이는 데이터를 읽어들이는 콜백에 의해 처리됩니다.

```
SitesSDK.getProperty('componentLayout',
self.updateComponentLayout);
SitesSDK.getProperty('customSettingsData',
self.updateCustomSettingsData);
```

- \* 메타데이터 업데이트: 구성요소 대신 정의된 구성요소 메타데이터가 변경될 때마다(예: 사용자가 설정 패널을 호출하고 데이터를 업데이트할 때) 콜백합니다.

```
SitesSDK.subscribe(SitesSDK.MESSAGE_TYPES.SETTINGS_UPDATED,
$.proxy(self.updateSettings, self));
```

### 주:

Oracle Content Management 서버는 항상 .html 파일에 대해 mime 유형을 설정하기 때문에 .html 파일을 업로드하고 필요한 "text!" 플러그인을 사용하여 로드할 수 없습니다. 따라서 템플릿의 경우 "text!" 플러그인을 사용하여 로드하려면 다른 확장을 사용하거나, 시드된 데이터에 표시된 대로 JavaScript에 직접 인라인으로 로드해야 합니다.

## 단계 2의 결과 확인

이제 사용자정의 구성요소 렌더러의 구조가 생성되는 방법의 개요를 수립해야 합니다. 작동하는지 검증하려면 다음과 같이 하십시오.

1. 다음 행을 변경하도록 render.js 파일의 sampleComponentTemplate 객체를 업데이트합니다. 이 코드를 변경합니다.

```
'<!-- ko if: initialized -->'+
```

이 코드를 대신 사용합니다.

```
'<!-- ko if: initialized -->'+
'<div data-bind="text:\'image width is: \' + imageWidth()"></div>' +
```

2. Oracle Content Management 인스턴스 서버와 구성요소를 동기화하거나 업로드합니다.
3. 사이트 내의 페이지를 편집하고 A\_Local\_Component 사용자 구성요소를 페이지 위에 놓습니다.

이 시점에서 구성요소에 image width is: 260px가 표시됩니다.

4. 설정 패널을 불러와서 **사용자정의 설정** 단추를 누릅니다.
5. **이미지 너비** 필드를 300px로 변경합니다.
6. 이 시점에서 구성요소에 두 가지 일이 발생합니다.
  - a. 기본 이미지 크기가 260px에서 300px로 확장됩니다.
  - b. 추가한 텍스트가 `image width is 300px`로 업데이트됩니다.

단계 3: 로컬 구성요소 설정의 구조 검토를 계속합니다.

## 단계 3: 로컬 구성요소 설정의 구조 검토

이 단계에서는 로컬 구성요소에 대해 지정된 설정의 구조를 검토합니다.

`/assets` 디렉토리의 `render.js` 파일과 유사하게, 동일한 디렉토리에 미리 생성된 `settings.html` 파일이 있습니다. `settings.html` 파일은 구성요소에 대한 사용자정의 설정 데이터를 렌더링합니다. 기본 구현에서는 사용자정의 설정 데이터에 단일 속성 `imageWidth`가 있습니다.

로컬 구성요소의 구조를 검토하려면 다음과 같이 하십시오.

1. Oracle Content Management 데스크톱 동기화 클라이언트를 사용하여 구성요소를 찾아 파일 시스템과 동기화합니다.

데스크톱 동기화 클라이언트가 없는 경우 Oracle Content Management 웹 인터페이스의 **구성요소** 탭에서 구성요소를 선택하고 드릴다운하여 파일을 볼 수 있습니다.

2. 구성요소 아래의 파일을 나열하면 다음 파일이 표시됩니다.

```
assets
  render.js
  settings.html
  appinfo.json
  _folder_icon.jpg
```

`/assets` 디렉토리 아래의 `settings.html` 파일을 열고 내용을 검토합니다. `render.js` 파일과 달리 `settings.html` 파일은 사이트 작성기의 설정 패널에서 인라인 프레임을 사용하는데, 이 때문에 인라인 프레임 안에 올바르게 렌더링하려면 지원 파일에 액세스해야 합니다. 사이트 작성기는 사이트 관리에 필요하므로 JavaScript 코드의 오류가 사이트 작성기에서 격리될 수 있기 때문에 `settings.html` 파일은 인라인 프레임을 사용합니다.

`settings.html` 파일의 주요 영역은 다음과 같습니다.

- 설정 패널을 렌더링하는 Knockout 템플릿.

```
<!-- 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 -->
```

- 설정 패널이 렌더링된 후 인라인 프레임의 높이를 조정하는 사용자정의 바인딩 처리기.

```
ko.bindingHandlers.scsCompComponentImpl
```

- Knockout 템플릿에 적용할 Knockout ViewModel.

```
SettingsViewModel
```

SettingsViewModel의 주요 요소는 다음과 같습니다.

- 구성요소 수명 주기 구독.
- 구성요소 초기화:
  - 모든 데이터가 인출될 때까지 구성요소가 렌더링되지 않도록 합니다. 이는 Knockout observable을 통해 처리됩니다.

```
self.initialized = ko.observable(false);
```

- 준비될 때까지 데이터 업데이트를 시도하지 않습니다.

```
self.saveData = false;
```

- 필수 속성에 대한 초기 값을 가져옵니다. 이는 데이터를 읽어들이는 콜백에 의해 처리됩니다.

```
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;
});
```

- 사용자정의 설정 데이터에 대한 속성 변경사항을 저장합니다.

```
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);
```

캡처할 다른 속성을 추가하려면 여러 단계가 필요합니다.

1. 새 값을 표시하도록 사용자 인터페이스를 업데이트합니다.

2. 구성요소에 대해 저장된 현재 값으로 값을 초기화합니다.
3. 값 변경사항을 구성요소에 다시 저장합니다.

사용자정의 구성요소에 다른 속성을 추가하려면 `settings.html` 파일을 다음과 같이 변경합니다.

1. 새 속성을 처리하도록 다른 `observable`을 추가합니다. 이 코드를 변경합니다.

```
self.width = ko.observable();
```

이 코드를 대신 사용합니다.

```
self.width = ko.observable();
self.imageBannerText = ko.observable();
```

2. 처음 설정 패널이 표시될 때 새 속성의 현재 값을 가져옵니다. 이 코드를 변경합니다.

```
self.width(data.width);
```

이 코드를 대신 사용합니다.

```
self.width(data.width);
self.imageBannerText(data.imageBannerText);
```

3. 새 속성의 변경사항을 저장합니다. 이 코드를 변경합니다.

```
'width': isNaN(self.width()) ? self.width() : self.width() + 'px'
```

이 코드를 대신 사용합니다.

```
'width': isNaN(self.width()) ? self.width() : self.width() + 'px',
'imageBannerText': self.imageBannerText()
```

4. 새 필드를 표시하도록 사용자 인터페이스를 추가합니다. 이 코드를 변경합니다.

```
<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">
```

이 코드를 대신 사용합니다.

```
<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. `settings.html` 파일을 동기화하거나 업로드합니다.

지금 실행했으면 필드가 표시됩니다. 그러나 설정 패널의 크기는 자동으로 변경되지 않습니다. 패널의 크기를 늘렸기 때문에 `components.json` 등록 항목도 새 크기로 업데이트해야 합니다.

1. 구성요소의 `assets/` 디렉토리와 동일 레벨에 있는 `appinfo.json` 파일을 다운로드하고 설정 패널의 크기를 업데이트합니다. 이 코드를 변경합니다.

```
"settingsHeight": 90,
```

이 코드를 대신 사용합니다.

```
"settingsHeight": 160,
```

2. `appinfo.json` 파일을 동기화하거나 업로드합니다.

### 단계 3의 결과 확인

이제 설정 패널에 추가한 새 속성을 보고 입력할 수 있어야 합니다.

1. 사이트 작성기가 구성요소의 변경사항을 가져올 수 있도록 사이트의 페이지를 새로고침합니다.
2. 페이지를 편집 모드로 전환합니다.
3. 구성요소를 페이지 위에 끌어 놓습니다.
4. 구성요소에 대해 설정 패널을 불러옵니다.
5. **사용자정의 설정** 단추를 누릅니다.

`settings.html` 파일에 있는 각 속성에 대해 두 개의 필드가 표시됩니다.

단계 4: 구성요소에 새 속성 표시를 계속합니다.

## 단계 4: 구성요소에 새 속성 표시

이 섹션이 끝나면 설정 패널에 새 속성 값을 입력하고 새 값이 반영되도록 사용자정의 구성요소가 변경되는 것을 확인할 수 있습니다. 속성 업데이트도 페이지와 함께 자동으로 저장됩니다.

`render.js` 파일에서 구성요소의 두 JavaScript 객체를 업데이트해야 합니다.

- `SampleComponentViewModel`
- `sampleComponentTemplate`

`render.js`를 편집하고 새 속성을 포함하도록 `SampleComponentViewModel` 구성요소를 업데이트합니다. 이 속성을 변경합니다.

```
self.showStoryLayout = ko.observable();
```

다음을 대신 사용합니다.

```
self.showStoryLayout = ko.observable();
self.imageBannerText = ko.observable();
```

값 변경사항을 가져오도록 `SampleComponentViewModel`을 업데이트합니다. 이 속성을 변경합니다.

```
self.imageWidth(customData && customData.width);
```

다음을 대신 사용합니다.

```
self.imageWidth(customData && customData.width);
self.imageBannerText(customData && customData.imageBannerText);
```

새 속성을 표시하도록 `sampleComponentTemplate`를 변경합니다. 이 속성을 변경합니다.

```
'<div data-bind="text: \'image width is: \' + imageWidth()"'></div>' +
```

다음을 대신 사용합니다.

```
'<div data-bind="text: imageBannerText"'></div>' +
```

Oracle Content Management 서버와 구성요소를 동기화하거나 업로드합니다.

이제 새 속성을 표시하도록 구성요소를 변경했습니다. 페이지에 인라인 프레임으로 내장된 설정 패널과 달리, 구성요소가 페이지에 직접 삽입되므로 크기가 늘어남에 따라 사용 가능한 영역이 자동으로 증가합니다.

#### 단계 4의 결과 확인

새 속성이 표시되는지 확인하려면 다음과 같이 하십시오.

1. 사이트 작성기가 구성요소의 변경사항을 가져올 수 있도록 사이트의 페이지를 새로고침합니다.
2. 페이지를 편집 모드로 전환합니다.
3. 구성요소를 페이지 위에 끌어 놓습니다.
4. 구성요소에 대해 설정 패널을 불러옵니다.
5. **사용자정의 설정** 단추를 누릅니다.
6. Image Banner를 Workspace로 변경합니다.

이미지 위에 workspace가 나타나도록 페이지의 구성요소가 업데이트됩니다.

[단계 5: 트리거 등록](#)을 계속합니다.

## 단계 5: 트리거 등록

이 단계에서는 Oracle Content Management 트리거를 등록할 수 있는 방법을 검토합니다. 그러면 구성요소의 설정 패널에서 [링크] 탭 아래의 [트리거 작업] 옵션을 사용하여 선택할 수 있습니다.

트리거는 Oracle Content Management 구성요소 상호 통신의 일부입니다. 모든 구성요소는 원하는 수의 트리거를 발생시킬 수 있습니다. 구성요소는 트리거에 대한 페이로드를 제공할 수 있으며, 이는 트리거가 발생할 때 실행되는 작업으로 전달됩니다. 사용자는 각 트리거에 대해 실행할 작업을 선택할 수 있습니다. 마지막으로, 함께 작동하도록 제작된 구성요소들은 사용자가 구성요소 간에 상호작용을 정의할 필요 없이 자동으로 트리거를 발생시켜 다른 구성요소에 작업을 실행할 수 있습니다.

직접 추가한 구성요소의 경우 트리거는 구성요소 등록 데이터의 일부로 등록됩니다. 트리거를 추가하려면 구성요소가 지원하는 각 트리거로 "triggers" 속성 배열을 업데이트합니다. 또한 트리거가 지원하는 페이로드를 지정해야 사용자 인터페이스를 생성하여 사용자가 페이로드 내의 값을 작업에서 지원하는 속성에 매핑할 수 있습니다.

appinfo.json 파일을 열고 "triggers":[], 항목을 검토합니다.

```
"triggers": [{
  "triggerName": "imageClicked",
  "triggerDescription": "Image clicked",
  "triggerPayload": [{
    "name": "payloadData",
    "displayName": "Trigger Payload Data"
  }]
}],
```

이 항목에서 다음을 확인합니다.

- triggerName인 "imageClicked"는 고유 값이어야 하며 일반적으로 사용자정의 구성요소 ID에 따라 네임스페이스가 지정됩니다.
- triggerDescription인 "Image clicked"는 사용자 인터페이스 대화상자에서 트리거를 표시할 때 사용됩니다.
- 트리거에 대한 단일 값 triggerPayload인 "payloadData". 사용자는 이 페이로드의 항목을 선택하여 작업 필드에 매핑할 수 있습니다.

### 단계 5의 결과 확인

구성요소의 설정 패널에서 링크 탭으로 이동할 때 트리거를 보고 선택할 수 있습니다.

1. 사이트 작성기가 구성요소의 변경사항을 가져올 수 있도록 사이트의 페이지를 새로고침합니다.
2. 페이지를 편집 모드로 전환합니다.
3. 구성요소를 페이지 위에 끌어 놓습니다.
4. 구성요소에 대해 설정 패널을 불러옵니다.
5. 설정 패널의 상단에서 링크 탭을 선택합니다.
6. 링크 유형으로 트리거 작업을 누릅니다.
7. 등록된 Image clicked 트리거를 누릅니다.
8. 대화상자의 [페이지 작업] 섹션 내에서 경고 표시 작업을 끌어옵니다. (페이지 작업은 Oracle Content Management에서 제공한 내장 작업입니다.)
9. 메시지 필드에서 Trigger Payload Data 값을 선택합니다. 이것은 트리거를 등록할 때 보았던 페이로드의 항목 이름입니다. 원하는 이름으로 변경할 수 있습니다.

이제 트리거를 등록하고 값을 통과하여 트리거를 내장 작업에 매핑할 수 있습니다. 다음 단계에서는 작업을 실행하도록 트리거를 발생시키는 방법을 검토합니다.

단계 6: 트리거 발생을 계속합니다.

## 단계 6: 트리거 발생

이 단계에서는 등록했던 트리거를 발생시키는 방법을 보여줍니다.



트리거는 언제든지 구성요소에 의해 발생할 수 있습니다. 일반적으로 단추 누르기 또는 테이블의 행 선택과 같은 사용자 상호작용에 의해 발생합니다. 그러나 구성요소는 REST 호출로 인해 데이터가 변경될 때와 같은 조건에 준하여 트리거를 발생시킬 수 있습니다.

이 샘플의 경우 이미지를 누를 때 whoAreYou 속성의 현재 값을 통과하여 트리거가 발생합니다.

render.js 파일을 검토하고 SampleComponentViewModel 객체를 살펴봅니다.

트리거를 발생시키려면 다음과 같이 하십시오.

1. SampleComponentViewModel 객체에서 Sites SDK를 호출하여 트리거를 발생시키는 함수를 검토합니다.

```
self.raiseTrigger = function (triggerName) {
  SitesSDK.publish(SitesSDK.MESSAGE_TYPES.TRIGGER_ACTIONS, {
    'triggerName': triggerName,
    'triggerPayload': {
      'payloadData': 'some data here'
    }
  });
};
```

2. 이제 사용자 인터페이스에서 트리거를 발생시키는 함수를 호출할 무언가가 필요합니다. render.js 파일을 검토하고 이 항목을 포함하도록 sampleComponentTemplate 객체를 업데이트합니다.

```
'<div data-bind="attr: {style: imageStyle, \'data-layout\':
alignImage()}, click: imageClicked">' +
```

SampleComponentViewModel 객체에서 이미지를 누를 때 호출되는 JavaScript 함수를 볼 수 있습니다. 이 함수는 Sites SDK를 호출하여 "imageClicked" 트리거(단계 2의 click 바인딩에서 전달된 값)에 대해 정의된 모든 작업을 트리거합니다. 또한 단일 필드 payloadData가 있는 triggerPayload를 통과하고 정적 값 'some data here'를 통과합니다. imageClicked 및 whoAreYou 값은 이전 단계에서 트리거가 등록된 appinfo.json 파일의 값과 일치합니다.

샘플 코드에서 트리거는 click 바인딩의 데이터 바인드에 의해 발생하고 트리거 이름 imageClicked를 전달합니다. 사용자가 선택한 레이아웃에 준하여 현재 세 가지 <scs-image> 구성요소 렌더링이 있습니다. 각 레이아웃에 대해 트리거가 발생하는지 확인하려면 render.js 파일을 편집하여 다음과 같이 변경합니다.

- 다른 레이아웃에서 트리거를 발생시킵니다. 이 코드의 두 항목을 찾습니다.

```
'<div data-bind="attr: {style: imageStyle, \'data-layout\':
alignImage()}">' +
```

다음으로 코드를 변경합니다.

```
'<div data-bind="attr: {style: imageStyle, \'data-layout\':
alignImage()}, click: imageClicked">' +
```

- 트리거에 전달할 페이로드를 지정합니다. 이 코드를 변경합니다.

```
self.raiseTrigger = function (triggerName) {
  SitesSDK.publish(SitesSDK.MESSAGE_TYPES.TRIGGER_ACTIONS, {
    'triggerName': triggerName,
```

```

        'triggerPayload': {
          'payloadData': 'some data here'
        }
      });
    };
  };
};

```

이 코드를 대신 사용합니다.

```

self.raiseTrigger = function (triggerName) {
  SitesSDK.publish(SitesSDK.MESSAGE_TYPES.TRIGGER_ACTIONS, {
    'triggerName': triggerName,
    'triggerPayload': {
      'payloadData': self.imageBannerText() // pass banner text
    }
  });
};

```

- `render.js` 파일을 Oracle Content Management 인스턴스 서버와 동기화하거나 업로드합니다.

이제 필요한 코드를 검토했으므로 단추를 누를 때 사용자정의 구성요소에 트리거가 발생하도록 트리거를 후크할 수 있습니다.

### 단계 6의 결과 확인

이제 트리거에 대해 실행할 작업을 등록하고 트리거가 발생할 때 작업이 실행되도록 해야 합니다.

1. 사이트 작성기가 구성요소의 변경사항을 가져올 수 있도록 사이트의 페이지를 새로고침합니다.
2. 페이지를 편집 모드로 전환합니다.
3. 구성요소를 페이지 위에 끌어 놓습니다.
4. 구성요소에 대해 설정 패널을 불러옵니다.
5. 설정 패널의 상단에서 링크 탭을 선택합니다.
6. 링크 유형으로 **트리거 작업**을 선택합니다.
7. 등록했던 **imageClicked** 트리거를 누릅니다.
8. 대화상자의 **페이지 작업** 섹션에서 **경보 표시** 작업을 끌어옵니다.
9. **메시지 필드**에서 **payloadData** 값을 선택합니다. 이것은 트리거를 등록할 때 입력한 페이로드입니다.
10. 설정 패널을 닫고 사이트 작성기를 미리보기 모드로 전환합니다.
11. 구성요소에서 이미지를 누릅니다.  
 imageBannerText 값을 지정하지 않았기 때문에 정의된 메시지 없음을 표시하는 경보가 나타납니다.
12. 페이지를 편집 모드로 전환하고 구성요소의 설정 패널을 다시 불러옵니다.
13. **사용자정의 설정**을 누르고 **Workplace**를 입력합니다.
14. 설정 패널을 닫고 페이지를 미리보기 모드로 전환합니다.
15. 구성요소에서 이미지를 누릅니다.

이제 click 바인딩의 변경 결과로 호출되는 업데이트된 페이로드 Workplace가 표시되어야 합니다.

트리거가 발생할 때 원하는 수의 작업을 실행할 수 있습니다.

#### 주:

작업을 실행할 때 미리 정의된 순서는 없습니다. 각 작업은 나열된 순서대로 호출되지만, 다음 작업을 호출하기 전에 현재 작업이 완료되기를 기다리지는 않습니다. 작업이 비동기로 호출되는 경우 다음 작업을 실행하기 전에 현재 작업이 완료되지 않을 수 있습니다.

단계 7: 작업 등록을 계속합니다.

## 단계 7: 작업 등록

Oracle Content Management 작업은 트리거가 발생할 때 구성요소에서 호출됩니다.

구성요소는 원하는 수의 작업을 등록하고 작업이 지원하는 페이로드를 정의할 수도 있습니다. 사용자가 작업을 선택하면 페이로드를 채워서 작업으로 전달할 수 있습니다.

트리거 등록과 마찬가지로, appinfo.json 파일 등록 데이터에 구성요소가 지원하는 작업을 등록할 수 있습니다. 구성요소의 샘플 작업 등록을 검토하려면 appinfo.json 파일을 열고 "actions" 코드를 찾습니다.

```
"actions": [{
  "actionName": "setImageWidth",
  "actionDescription": "Update the image width",
  "actionPayload": [{
    "name": "imageWidth",
    "description": "Image Width in pixels",
    "type": {
      "ojComponent": {
        "component": "ojInputText"
      }
    },
    "value": ""
  }]
}]
```

이 등록된 작업은 구성요소의 설정 패널에 있는 링크 탭에서 트리거를 누를 때 호출되는 작업 대화상자에서 볼 수 있습니다.

### 단계 7의 결과 확인

1. 사이트 작성기가 구성요소의 변경사항을 가져올 수 있도록 사이트의 페이지를 새로고침합니다.
2. 페이지를 편집 모드로 전환합니다.
3. 구성요소를 페이지 위에 끌어 놓습니다.
4. 단추 구성요소를 페이지 위에 놓습니다.
5. 단추 구성요소에 대해 설정 패널을 불러옵니다.

6. [일반 사항] 탭에서 단추 레이블을 **Click me!**로 변경합니다..
7. 설정 패널에서 링크 탭을 선택합니다.
8. 링크 유형으로 트리거 작업을 선택합니다.
9. 단추 구성요소에 대해 단추 누르기 트리거를 누릅니다.
10. 대화상자의 왼쪽 팔레트에서 A\_Local\_Component 구성요소를 확장합니다.
11. A\_Local\_Component 구성요소의 **Update the image width** 작업을 페이지 위에 끌어 놓습니다.
12. **Image width in pixels** 필드에 **300px**를 입력합니다.

이제 작업을 등록할 수 있는 방법과 해당 작업이 사용자 인터페이스에 어떻게 표시되는지 살펴보았습니다. 다음 단계에서는 호출 시 구성요소 내의 작업을 처리하는 방법을 배웁니다.

단계 8: 작업 실행을 계속합니다.

## 단계 8: 작업 실행

이 항목이 끝나면 구성요소 내의 작업을 실행할 페이지에 구성요소를 놓을 수 있습니다. 이전 단계에서 생성한 작업 등록을 활용합니다.

구성요소가 작업을 실행하려면 EXECUTE\_ACTION 메시지를 수신해야 합니다. 이 메시지에는 작업으로 전달된 페이로드도 포함되며, 여기서 예상 값을 추출합니다.

EXECUTE\_ACTION 메시지를 수신하려면 render.js 파일을 편집하고 다음 항목으로 SampleComponentViewModel 객체를 업데이트합니다.

```
SitesSDK.subscribe('EXECUTE_ACTION', $.proxy(self.executeActionsListener, self));
```

EXECUTE\_ACTION 메시지가 수신되면 연관된 콜백 함수가 실행됩니다.

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

작업을 실행하는 JavaScript 함수가 생성되고 EXECUTE\_ACTION 메시지가 발생할 때마다 Sites SDK를 사용하여 함수를 호출합니다.

EXECUTE\_ACTION 메시지가 발생할 때마다 구성요소가 호출되며, 구성요소는 처리하도록 설계된 작업만 처리할 수 있습니다. 이를 위해 작업 이름을 검사하여 구성요소가 처리할 수 있는 작업인지 확인해야 합니다.

작업 페이로드는 값의 배열입니다. 일반적으로 배열에서 원하는 페이로드 값을 찾아야 합니다.

#### 주:

작업 리스너는 콜백이기 때문에 함수를 실행할 때 `viewModel`에 액세스할 수 있도록 JavaScript 클로저를 사용하거나 함수를 적절히 바인딩해야 합니다.

#### 단계 8의 결과 확인

1. 사이트 작성기가 구성요소의 변경사항을 가져올 수 있도록 사이트의 페이지를 새로고침합니다.
2. 페이지를 편집 모드로 전환합니다.
3. 구성요소를 페이지 위에 끌어 놓습니다.
4. 단추 구성요소를 페이지 위에 끌어 놓습니다.
5. 단추 구성요소에 대해 설정 패널을 불러옵니다.
6. [일반 사항] 탭에서 단추 레이블을 `Click me!`로 변경합니다.
7. 설정 패널의 상단에서 **링크** 탭을 선택합니다.
8. 링크 유형으로 **트리거 작업**을 선택합니다.
9. 단추 구성요소에 대해 **단추 누르기** 트리거를 누릅니다.
10. 대화상자의 왼쪽에서 `A_Local_Component` 구성요소를 확장합니다.
11. `A_Local_Component` 구성요소의 **Update the image width** 작업을 오른쪽에 끌어 놓습니다.
12. **Image Width in pixels** 필드에 `300px`를 입력합니다.
13. 페이지를 미리보기 모드로 전환합니다.
14. **Click me!** 단추를 누릅니다.

이 시점에서 이미지 크기가 `300px`로 증가합니다.

#### 주:

트리거 및 작업은 구성요소 상호 통신을 지원하도록 설계되었습니다. 상태를 생성하거나 관리하도록 설계되지 않았습니다. 페이지를 새로고침하면 트리거가 발생하거나 작업이 실행되지 않은 것처럼 페이지가 원래 상태로 되돌아갑니다.

단계 9: 각 구성요소 인스턴스에 대한 고유 제목 생성을 계속합니다.

## 단계 9: 각 구성요소 인스턴스에 대한 고유 제목 생성

이 단계는 다양한 구성요소 인스턴스에 대한 고유 제목을 생성하는 방법을 설명합니다.

구성요소를 페이지 위에 놓으면 구성요소의 배너가 `A_Local_Component`라는 것을 알 수 있습니다. 사용자가 구성요소들 중 하나만 페이지에 놓는 경우에는 괜찮지만, 사용자가 다양한 구성요소 인스턴스를 구별할 수 있도록 고유 제목을 생성할 수 있습니다.

Sites SDK를 사용하여 구성요소의 제목을 업데이트할 수 있습니다. 이 단계에서는 `"imageBannerText"` 속성에 준하여 업데이트합니다.

제목을 업데이트하려면 `render.js` 파일을 편집하고 이 코드를 `SampleComponentViewModel` 객체에 추가합니다.

```
self.updateDescription = ko.computed(function () {
    SitesSDK.setProperty('description', self.imageBannerText());
});
```

이 Knockout 컴퓨팅은 `imageBannerText observable`이 변경될 때마다 구성요소의 설명을 업데이트합니다.

### 단계 9의 결과 확인

1. 사이트 작성기가 구성요소의 변경사항을 가져올 수 있도록 사이트의 페이지를 새로고침합니다.
2. 페이지를 편집 모드로 전환합니다.
3. 구성요소를 페이지 위에 놓습니다.
4. 구성요소에 대해 설정 패널을 불러옵니다.
5. **사용자정의 설정** 단추를 누릅니다.
6. **Image Banner**를 **Workplace**로 변경합니다.
7. 설정 패널을 닫고 구성요소 위로 커서를 가져가서 배너를 표시합니다.  
이제 `A_Local_Component Workplace`가 표시되어야 합니다.

단계 10: 인라인 편집과 함께 중첩 구성요소 사용을 계속합니다.

## 단계 10: 인라인 편집과 함께 중첩 구성요소 사용

Oracle Content Management 구성요소는 KnockoutJS 구성요소 구조를 사용하여 구현됩니다. 즉, KnockoutJS를 사용하여 구성요소를 구현하는 경우 템플릿에 직접 Oracle Content Management 내장 구성요소를 포함할 수 있습니다.

### 주:

Oracle Content Management 내장 구성요소는 Oracle Content Management 페이지에서만 실행할 수 있으므로 구성요소가 인라인 프레임으로 렌더링된 경우 중첩 구성요소를 사용할 수 없습니다.

중첩 구성요소를 활용하려면 다음과 같이 하십시오.

1. KnockoutJS를 사용하여 구성요소를 구현합니다.
2. RequireJS를 사용하여 구성요소를 포함하고 Oracle Content Management에서 생성된 동일한 Knockout "ko" 인스턴스 변수를 사용합니다.  
Oracle Content Management가 구성요소로 Knockout을 확장하고 고유의 KnockoutJS 인스턴스를 사용할 경우 이 구성요소가 제공되지 않기 때문에 필요합니다.  
이 단계에서는 Oracle Content Management 이미지, 단락 및 제목 구성요소가 사용자정의 구성요소에 렌더링되는 방법을 검토합니다. 사용자는 페이지 내에서 직접 편집하고 중첩 구성요소의 설정 패널에 액세스할 수 있습니다.

이 구성요소가 템플릿에 어떻게 포함되는지 보려면 `render.js` 파일을 편집하고 `sampleComponentTemplate` 객체를 살펴봅니다. 렌더링되는 기본 섹션은 다음과 같습니다.

```
'<!-- 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 -->' +
```

`<scs-image>` 중첩 구성요소를 살펴보면 다음 항목이 표시됩니다.

```
'<scs-image params="{ scsComponent: { \'renderMode\': mode, \'parentId\':
id, \'id\': \'imageId\', \'data\': imageData } }"></scs-image>' +
```

`params` 템플릿 바인딩에 전달된 `scsComponent` 데이터에는 다음이 포함됩니다.

- `renderMode`: 사이트 작성기가 있는 모드를 나타냅니다. 이를 통해 기능을 사용 및 사용 안함으로 설정할 수 있습니다. 예를 들어, `<scs-title>` 구성요소에서 사용될 경우 `edit` 모드로 실행할 때 서식 있는 텍스트 편집기를 추가합니다.
- `parentId`: Oracle Content Management 구성요소가 중첩 구성요소로 렌더링되고 있음을 인지하려면 이 속성이 필요합니다. 중첩 구성요소의 모든 변경사항은 사용자정의 구성요소의 데이터에 저장됩니다.
- `id`: 중첩 구성요소의 고유 ID입니다. 사용자정의 구성요소의 ID에 따라 추가 네임스페이스가 지정됩니다.
- `data`: 중첩 구성요소의 초기 데이터입니다. 구성요소가 더 수정되지 않으면 이 초기 데이터로 렌더링됩니다.

참조된 `id` 및 `mode` 값은 `SampleComponentViewModel` 객체의 사용자정의 구성요소에 전달되므로 이 값을 가져오도록 객체를 수정할 필요가 없습니다.

```
// Store the args
self.mode = args.viewMode;
self.id = args.id;
```

그 밖에 모든 지원되는 중첩 구성요소의 구문은 `<scs-paragraph>`와 동일한 패턴을 따릅니다. 예: `<scs-image>`, `<scs-title>`, `<scs-button>`.

### 단계 10의 결과 확인

1. 사이트 작성기가 구성요소의 변경사항을 가져올 수 있도록 사이트의 페이지를 새로고침합니다.
2. 페이지를 편집 모드로 전환합니다.

3. 구성요소를 페이지 위에 끌어 놓습니다.
4. 구성요소의 *As a page author, you can edit. . .* 텍스트를 누르고 서식 있는 텍스트 편집기를 사용하여 설명을 업데이트합니다.
5. 미리보기 모드로 전환하여 업데이트를 확인합니다.
6. 편집 모드로 다시 전환합니다.
7. 구성요소에 대해 설정 패널을 불러옵니다.
8. 중첩 구성요소를 찾았으므로 지금 나타나는 **구성요소** 링크를 누릅니다.
9. 발견한 중첩 구성요소인 **단락**을 누릅니다.

이제 구성요소 내의 단락 구성요소에 대해 속성을 업데이트할 수 있습니다.

#### 주:

구성요소가 인스턴스화될 때까지 Oracle Content Management는 어떤 중첩 구성요소가 템플릿에 존재하는지 모릅니다. Oracle Content Management에 숨겨진 중첩 구성요소를 알려려면 `SitesSDK.setProperty('visibleNestedComponents', [])`; API를 사용할 수 있습니다. 숨겨진 중첩 구성요소를 기본적으로 표시하려면 구성요소 등록에서 `"nestedComponents": []` 배열을 업데이트해야 합니다.

단계 11: 다양한 레이아웃 지원을 계속합니다.

## 단계 11: 다양한 레이아웃 지원

이 단계에서는 구성요소 표시 방식을 사용자가 변경할 수 있는 레이아웃을 검토합니다.

사용자정의 구성요소는 사용자가 선택할 수 있는 레이아웃을 원하는 만큼 지원할 수 있습니다. 각 레이아웃은 사용자정의 구성요소의 표시 방식을 변경합니다. 레이아웃은 등록 데이터의 또 다른 확장입니다.

샘플 코드에 지원되는 세 가지 레이아웃을 검토하려면 `appinfo.json` 파일의 `"componentLayouts"` 항목을 검토합니다.

```
"componentLayouts": [
  {
    "name": "default",
    "displayName": "IMAGE_LEFT_LAYOUT"
  },
  {
    "name": "right",
    "displayName": "IMAGE_RIGHT_LAYOUT"
  },
  {
    "name": "top",
    "displayName": "IMAGE_TOP_LAYOUT"
  }
],
```



사용자정의 구성요소에 대해 설정 패널을 불러오면 레이아웃 간을 전환하는 옵션이 표시됩니다. 구성요소가 선택 변경에 반응할 수 있도록 `render.js` 파일에는 현재 선택된 값을 가져와서 이 값의 변경사항을 수신하는 코드가 있습니다.

`render.js` 파일을 편집하고 `SampleComponentViewModel` 객체를 살펴봅니다.

- 템플릿에서 참조된 layout `observable`이 있습니다.

```
self.layout = ko.observable();
```

- 이 값이 변경될 때마다 처리할 업데이트 함수가 있습니다.

```
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);
```

- 초기화 코드는 원래 레이아웃 값을 가져와서 업데이트 함수를 호출합니다.

```
SitesSDK.getProperty('componentLayout', self.updateComponentLayout);
```

속성 변경 리스너가 이 속성의 변경사항을 확인하고 업데이트 함수를 호출합니다.

```
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,
```

마지막으로 `sampleComponentTemplate` 템플릿 객체에는 이 값의 변경사항을 반영하는 코드가 있습니다.

```
'<!-- ko if: alignImage() === \'right\' -->' +
```

이러한 변경사항을 모아 설정 패널에서 레이아웃을 선택하고 구성요소를 업데이트할 수 있습니다.

### 단계 11의 결과 확인

1. 사이트 작성기가 구성요소의 변경사항을 가져올 수 있도록 사이트의 페이지를 새로고침합니다.
2. 페이지를 편집 모드로 전환합니다.
3. 구성요소를 페이지 위에 끌어 놓습니다.
4. 구성요소에 대해 설정 패널을 불러옵니다.
5. 레이아웃 속성에서 **이미지 오른쪽**을 선택합니다.

이 시점에서 "<scs-image>" 구성요소를 표시하도록 구성요소가 업데이트됩니다.

단계 12: 사용자정의 스타일 정의를 계속합니다.

## 단계 12: 사용자정의 스타일 정의

직접 생성한 구성요소는 사이트에 사용된 테마에서 design.json 및 design.css 파일의 다른 구성요소처럼 취급됩니다.

사용자정의 구성요소에 고유의 스타일을 추가하려면 구성요소를 등록할 때 사용한 id 값을 확인하십시오. appinfo.json 파일에서 "id": "hello-world"였습니다.

이 값을 사용하여 테마의 design.json 파일을 편집하고 해당 id에 대해 지원할 새 스타일을 추가합니다. 예를 들어, 테마의 /designs/default/design.json 파일을 편집하고 이 코드를 추가합니다.

```
"hello-world": {
  "styles": [{
    "name": "Plain",
    "class": "hello-world-default-style"
  },
  {
    "name": "Gothic",
    "class": "hello-world-gothic-style"
  }
  ]
},
```

구성요소에 대해 설정 패널을 불러오면 이제 스타일 탭에 나열된 두 옵션으로 **Plain(기본값)** 및 **Gothic**이 표시되어야 합니다. 그러나 이 옵션 간을 전환해도 실제로 design.css 파일에 나열된 스타일 클래스를 정의할 때까지 아무 것도 수행되지 않습니다.

테마의 design.css 파일을 편집하고 스타일의 CSS(중속형 스타일시트) 클래스를 추가합니다. 예를 들어, 테마의 /designs/default/design.css 파일을 편집하고 이 코드를 추가합니다.

```
.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; }
```

Oracle Content Management 인스턴스 서버에 파일을 저장하고 동기화합니다.

### 단계 12의 결과 확인

1. 사이트 작성기가 구성요소의 변경사항을 가져올 수 있도록 사이트의 페이지를 새로고침합니다.
2. 페이지를 편집 모드로 전환합니다.

3. 구성요소를 페이지 위에 끌어 놓습니다.
4. 구성요소에 대해 설정 패널을 불러옵니다.
5. [스타일] 탭으로 이동합니다.
6. design.json 파일에 정의된 **Gothic** 및 **Plain** 스타일 간을 전환합니다.

각 선택에 적용된 CSS 클래스 간을 전환할 때 변경사항이 반영되도록 구성요소 내의 글꼴 크기가 조정되는 것을 알 수 있습니다.

단계 13: 인라인 프레임으로 구성요소 렌더링을 계속합니다.

## 단계 13: 인라인 프레임으로 구성요소 렌더링

지금까지 샘플은 페이지에 인라인으로 렌더링된 로컬 구성요소를 보여주었습니다. 인라인 프레임으로 구성요소를 렌더링하도록 선택할 수도 있습니다.

예를 들어, 구성요소가 페이지에 전능하게 업데이트되지 않아서 속성이 변경될 때마다 페이지를 재생성해야 하는 경우 인라인 프레임으로 구성요소를 렌더링하도록 선택할 수 있습니다. 또한 원격 구성요소는 항상 인라인 프레임으로 렌더링됩니다.

이 섹션의 샘플은 로컬 구성요소를 생성할 때 **iframe에 렌더링하는 구성요소 생성** 옵션을 선택하면 자동 생성되는 파일에서 가져온 것입니다. 그러나 이 파일 집합을 가져와서 원격 서버에 호스팅하면 원격 구성요소에 똑같이 적용할 수 있습니다.

### 인라인 프레임과 비인라인 프레임 구성요소 간의 유사점

#### 설정 패널

설정 패널은 항상 인라인 프레임으로 페이지에 놓이기 때문에 구성요소가 인라인 프레임을 사용하는지 여부에 관계없이 설정 패널의 코드는 변경되지 않습니다. 두 사용 사례에 동일한 설정 패널 코드를 생성합니다.

#### Sites SDK API

SDK API는 두 사용 사례에 동일합니다. 동일한 코드를 사용하여 트리거를 발생시키고 작업을 수신하며 속성값을 가져오고 설정합니다. 특정 속성은 두 사례에 적용할 수 없지만(예: 인라인 프레임을 사용하지 않는 구성요소에 "height" 속성을 설정할 수 없음) API는 그대로 유지됩니다. 따라서 두 유형의 구성요소 간에 코드를 복사할 수 있으며 이 자습서에 설명된 예제 코드는 두 사례에 모두 작동합니다.

### 인라인 프레임과 비인라인 프레임 구성요소 간의 차이점

#### 파일 구조 및 종속성

로컬 구성요소를 생성할 때 **iframe에 렌더링하는 구성요소 생성**을 선택하면 다음 파일이 자동 생성됩니다.

```
<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
```

이 파일이 생성되면 페이지에 인라인 프레임으로 구성요소를 즉시 실행할 수 있습니다. 이 구조와 표준 로컬 구성요소 구조 간의 주요 차이점은 다음과 같습니다.

- JavaScript 종속성:
  - 구성요소가 실행되도록 이 파일의 전체 복사본을 가져옵니다. 이 파일은 샘플 인라인 프레임 구성요소 실행에 필요합니다. 요구사항에 따라 이 디렉토리의 내용을 추가하고 제거할 수 있습니다.
  - 구성요소가 게시될 때 구성요소의 `assets` 디렉토리 아래의 모든 항목이 공용 사이트로 푸시되기 때문에 `js` 디렉토리의 모든 항목은 사이트 작성기와 런타임에 모두 사용할 수 있습니다.
  - 참고: 이 파일은 간편한 사용을 위해 생성됩니다. 각 인라인 프레임 구성요소마다 이 파일의 개별 버전을 생성하는 대신, 이 파일을 테마나 다른 공용 위치에 통합해야 합니다.
- `render.html`:
  - AMD 모듈인 표준 구성요소의 `render.js` 파일과 달리 완전한 HTML 문서입니다.

### 구성요소 "높이" 관리

인라인 프레임 사용의 문제 중 하나는 인라인 프레임 자체의 높이 관리입니다. 이 문제가 발생하면 원하던 원하지 않은 페이지의 구성요소에 스크롤 막대가 나타납니다.

인라인 프레임의 높이를 관리하려면 구성요소가 원하는 인라인 프레임의 높이를 페이지에 알려주어야 합니다. 원격 구성요소로 교차 도메인 문제를 처리할 수 있으므로, Sites SDK 메시징을 사용하여 구성요소가 페이지에 렌더링된 후 필요한 높이로 인라인 프레임을 설정하도록 페이지에 요청해야 합니다. 이를 위해 `SitesSDK.setProperty('height', {value})` API를 사용할 수 있습니다. (Oracle Content and Experience SDK 참조.)

예를 들어, `setHeight` 함수와 사용자정의 바인딩 처리기를 생성하고 구성요소가 페이지에 렌더링될 때 호출합니다.

- `height` 함수 업데이트:

```
// 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');
};
```

- 구성요소가 페이지에 렌더링되거나 속성이 변경될 때마다 `setHeight`를 호출하는 Knockout 사용자정의 바인딩 처리기:

```
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();
    }
};

```

- 바인딩 처리기를 호출하도록 템플릿 업데이트:

```
<div data-bind="sampleAppSetAppHeight: true"></div>
```

### 트리거 및 작업 등록

인라인 프레임이 아닌 구성요소의 트리거/작업 등록은 appinfo.json 파일에 있지만, 인라인 프레임 구성요소의 경우 구성요소 자체가 이 정보를 제공해야 합니다. 다음 두 API를 사용하여 수행됩니다.

```

SitesSDK.subscribe('GET_ACTIONS', self.getAppActions);
SitesSDK.subscribe('GET_TRIGGERS', self.getAppTriggers);

```

다음은 이 API 사용의 예입니다.

```

// 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;
    };

```

### 테마 스타일에 액세스

구성요소가 인라인 프레임으로 렌더링되기 때문에 테마에서 제공되는 스타일에 액세스할 수 없습니다. Sites SDK는 이 스타일을 검색하는 API를 제공하므로 인라인 프레임 내의 요소에 적용할 수 있습니다.

이 항목은 [단계 14: 구성요소가 인라인 프레임으로 렌더링될 때 사용자정의 스타일 사용](#)에서 자세히 다룹니다.

### 혼합 HTTPS 대 HTTP 프로토콜

Oracle Content Management는 HTTPS 프로토콜을 사용하므로 페이지 내에 참조된 모든 리소스도 HTTPS를 사용해야 합니다. 리소스에는 인라인 프레임으로 렌더링될 기본 .html 파일과 함께 모든 참조 파일이 포함됩니다.

이 리소스 요구사항은 대부분 원격 구성요소에 적용되지만 이 제약 조건을 인지해야 합니다. 인라인 프레임을 사용하는 로컬 구성요소의 리소스는 Oracle Content Management 서버에서 제공하므로 이 구성요소는 이미 일치 프로토콜을 사용합니다.

[단계 14: 구성요소가 인라인 프레임으로 렌더링될 때 사용자정의 스타일 사용](#)을 계속합니다.

## 단계 14: 구성요소가 인라인 프레임으로 렌더링될 때 사용자정의 스타일 사용

인라인 프레임으로 렌더링된 구성요소는 design.css 파일에 직접 액세스할 수 없습니다. 대신 구성요소에 design.css의 URL을 가져와서 페이지에 추가하는 별도의 단계가 있습니다. 그런 다음 사용자가 선택한 스타일이 반영되도록 구성요소를 업데이트해야 합니다.

구성요소에 design.css 파일을 포함하고 사용하려면 render.html 파일을 변경해야 합니다.

1. design.css 파일의 URL을 찾아 포함합니다.
2. 변경될 때마다 선택한 스타일 클래스의 값을 가져옵니다.
3. 선택한 styleClass가 반영되도록 템플릿을 업데이트합니다.
4. 선택한 스타일 클래스의 변경사항을 구성요소에 반영합니다.
5. 스타일이 변경될 때 인라인 프레임 크기가 조정되는지 확인합니다.

다음은 render.html 파일 편집을 위한 자세한 지침입니다.

1. design.css 파일의 URL을 찾아 포함합니다.

페이지의 <head> 섹션에 design.css 파일을 동적으로 추가합니다. 로드된 후에 스타일 적용으로 변경되었을 수 있으므로 인라인 프레임의 높이를 설정합니다.

다음 코드를 viewModel 객체에 추가합니다.

```

// Dynamically add any theme design URL to the <head> of the page
self.loadStyleSheet = function (url) {
    var $style,

```

```

    styleSheetDeferred = new $.Deferred(),
    attempts = 100,
    numAttempts = 0,
    interval = 50,
    pollFunction = function () {
        // try to locate the style sheet
        for (var i = 0; i < document.styleSheets.length; i++) {
            try {
                // locate the @import sheet that has an href based on
our expected URL
                var sheet = document.styleSheets[i],
                    rules = sheet && sheet.cssRules,
                    rule = rules && rules[0];
                // check whether style sheet has been loaded
                if (rule && (rule.href === url)) {
                    styleSheetDeferred.resolve();
                    return;
                }
            } catch (e) {}
        }
        if (numAttempts < attempts) {
            numAttempts++;
            setTimeout(pollFunction, interval);
        } else {
            // didn't find style sheet so complete anyway
            styleSheetDeferred.resolve();
        }
    };

    // add the themeDesign stylesheet to <head>
    // use @import to avoid cross domain security issues when determining
when the stylesheet is loaded
    $style = $('<style type="text/css">@import url("'" + url + "'</
style>');
    $style.appendTo('head');

    // kickoff the polling
    pollFunction();

    // return the promise
    return styleSheetDeferred.promise();
};

// update with the design.css from the Sites Page
SitesSDK.getSiteProperty('themeDesign', function (data) {
    if (data && data.themeDesign && typeof data.themeDesign === 'string')
    {
        // load the style sheet and then set the height
        self.loadStyleSheet(data.themeDesign).done(self.setHeight);
    }
});

```

2. 변경될 때마다 선택한 스타일 클래스의 값을 가져옵니다.

styleClass 속성 값이 변경될 때 추적할 observable을 생성합니다.

```
self.selectedStyleClass = ko.observable();
```

스타일 클래스가 있을 때까지 렌더링할 수 없습니다. 이 코드를 변경합니다.

```
self.customSettingsDataInitialized = ko.observable(false);
self.initialized = ko.computed(function () {
    return self.customSettingsDataInitialized();
}, self);
```

이 코드를 대신 사용합니다.

```
self.customSettingsDataInitialized = ko.observable(false);
self.styleClassInitialized = ko.observable(false);
self.initialized = ko.computed(function () {
    return self.customSettingsDataInitialized() &&
self.styleClassInitialized();
}, self);
```

다음을 추가하여 선택한 스타일 클래스의 초기 값을 가져옵니다.

```
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. styleClass가 반영되도록 템플릿을 업데이트합니다. 이 코드를 변경합니다.

```
<p data-bind="attr: {id: 'titleId'}, text: titleText"></p>
```

이 코드를 대신 사용합니다.

```
<p data-bind="attr: {id: 'titleId'}, text: titleText, css:
selectedStyleClass"></p>
```

4. 선택한 스타일 클래스의 변경사항을 구성요소에 반영합니다. 이 코드를 변경합니다.

```
if (settings.property === 'customSettingsData') {
    self.updateCustomSettingsData(settings.value);
}
```

이 코드를 대신 사용합니다.

```
if (settings.property === 'customSettingsData') {
    self.updateCustomSettingsData(settings.value);
}
if (settings.property === 'styleClass') {
```



```

        self.updateStyleClass(settings.value);
    }

```

5. 스타일이 변경될 때 인라인 프레임 크기가 조정되는지 확인합니다. 이 코드를 변경합니다.

```

// 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();

```

이 코드를 대신 사용합니다.

```

// 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. Oracle Content Management 인스턴스 서버에 파일을 저장하고 동기화합니다.

#### 단계 14의 결과 확인

1. 사이트 작성기가 구성요소의 변경사항을 가져올 수 있도록 사이트의 페이지를 새로고침합니다.
2. 페이지를 편집 모드로 전환합니다.
3. 구성요소를 페이지 위에 끌어 놓습니다.
4. 구성요소에 대해 설정 패널을 불러옵니다.
5. [스타일] 탭으로 이동합니다.
6. design.json 파일에 정의된 **Gothic** 및 **Plain** 스타일 간을 전환합니다.

각 선택에 적용된 CSS 클래스 간을 전환할 때 변경사항이 반영되도록 구성요소 내의 글꼴 크기가 조정되는 것을 알 수 있습니다.

단계 15: 페이지 실행 취소 및 재실행 동작과 통합을 계속합니다.

## 단계 15: 페이지 실행 취소 및 재실행 동작과 통합

Oracle Content Management는 사용자정의 구성요소 대신 속성을 저장하므로 이 속성의 변경사항은 자동으로 페이지의 **실행 취소** 및 **재실행** 동작의 일부입니다.

사용자가 **실행 취소** 또는 **재실행**을 누를 때 어떤 일이 발생하는지 명확히 하려면 사용자가 실제로 페이지에 어떤 작업을 수행했을 때만 "실행 취소 이벤트"가 발생해야 합니다. 예를 들어, 사용자정의 구성요소 설정 패널을 불러와도 사용자가 실제로 속성을 변경할 때까지 페이지 내의 속성이 업데이트되지 않아야 합니다. 단순히 설정 패널의 속성을 초기화해도 업데이트 이벤트가 발생하지 않아야 합니다.

이 동작에 주의하지 않으면 예상치 않은 동작이 발생할 수 있습니다. 페이지는 계속 실행되지만 결국 사용자 경험을 해치게 됩니다. 예를 들어, 다음 동작이 발생할 수 있습니다.

- 단순히 설정 패널을 불러와도 **저장** 단추가 활성화됩니다.

- 효과가 보이기 전에 사용자가 **실행 취소**를 여러 번 눌러야 합니다.
- 구성요소가 예상치 않은 변경사항을 쓰기 저장하고 재실행 스택을 새 값으로 업데이트했기 때문에 재실행 스택이 제거됩니다.

이 자습서에 제공된 설정 패널의 샘플 코드는 초기화가 아닌 실제로 `saveData`를 호출할 준비가 되었을 때만 쓰기 저장하는 방법의 예를 제공합니다. 사용자 상호작용이 관여하지 않는 한 마찬가지로 구성요소 자체 내에서 `customSettingsData`를 업데이트하지 않도록 주의해야 하지만, 일반적으로 이는 문제가 되지 않습니다.

단계 16: 자산 관리를 계속합니다.

## 단계 16: 자산 관리

이 단계는 구성요소에서 사용된 자산을 관리하는 방법을 설명합니다.

자산에는 Oracle Content Management가 자산의 수명 주기를 관리하기 위해 알아야 하는 구성요소와 사용자정의 구성요소가 포함됩니다.

### Oracle Content Management content 폴더

Oracle Content Management에서 생성한 각 사이트에는 자체 `content` 폴더가 제공됩니다. 보통은 보이지 않는 숨겨진 폴더입니다. 사이트가 게시될 때 `content` 폴더의 모든 파일도 파일 시스템에 게시됩니다.

예를 들어, 이미지 구성요소를 사용하여 이미지를 선택하면 Oracle Content Management는 선택한 이미지의 복사본을 만들어 `content` 폴더에 놓습니다. URL은 항상 이 복사된 버전의 이미지를 가리키므로 원본 이미지를 삭제해도 사이트가 중단되지 않습니다. 이는 Oracle Content Management에서 제공한 다른 구성요소(갤러리, 갤러리 그리드, 문서, 소셜 표시줄, 파일 다운로드)는 물론 `slots` 및 `componentGroups`의 배경 이미지에도 적용됩니다.

사용자정의 구성요소가 이 자산 수명 주기에 참여하려면 사용자정의 구성요소 대신 서비스가 관리할 자산을 Oracle Content Management에 알려야 합니다. 여기에는 자산 복사본 만들기가 포함되므로 사용자정의 구성요소도 Oracle Content Management API를 사용하여 자산을 선택해야 관리 방법을 알 수 있습니다.

### URL 관리

자산 URL은 수많은 기준에 따라 변경됩니다.

- 구성요소의 런타임 URL은 구성요소의 사이트 작성기 URL과 다릅니다.
- 페이지를 복사할 경우 Oracle Content Management는 `content` 폴더에서 참조된 모든 자산의 복사본도 만들기 때문에 두 구성요소가 `content` 폴더의 동일한 자산을 가리키는 일은 없습니다.
- `componentGroup`을 페이지 위에 놓으면 `componentGroup`의 구성요소에서 참조된 모든 자산의 새 복사본을 만듭니다.

또한 상대 URL은 로컬 구성요소에 적합할 수 있지만, 원격 구성요소에는 Oracle Content Management가 대신 관리할 자산의 전체 URL이 필요합니다. 그래야 전체 URL로 인라인 프레임 콘텐츠를 렌더링할 수 있습니다.

남은 정적 URL에 의존할 수 없으므로 코드에는 자산 ID에 대한 참조만 넣고 자산을 렌더링할 때 자산의 URL을 검색해야 합니다.

## 자산 관리

다음 Sites SDK API는 자산 관리에 사용할 수 있습니다.

```
SitesSDK.getProperty('componentAssets', callback);
```

- 현재 자산의 배열을 가져옵니다.
- 각 자산 항목은 다음으로 구성됩니다.
  - **id**: 자산의 고유 ID입니다.
  - **제목**: Oracle Content Management 제목 메타데이터입니다.
  - **설명**: Oracle Content Management 설명 메타데이터입니다.
  - **fileName**: 선택한 파일의 원래 이름입니다. 사용자가 선택한 파일을 알 수 있도록 사용자정의 구성요소의 설정 패널에 표시하기에 유용합니다. content 폴더에 복사된 파일의 이름이 아닙니다.
  - **source**: 매크로가 사용된 자산 URL입니다. 이 값은 시간에 따라 변경되므로 구성요소에서 참조하지 않아야 하지만 자산의 일부로 저장해야 합니다.
  - **url**: getProperty()가 호출된 컨텍스트에 따른 자산의 전체 URL입니다.

```
SitesSDK.setProperty('componentAssets', [assets]);
```

- Oracle Content Management가 대신 관리할 모든 자산을 저장하려면 호출합니다.
- 호출하지 않으면 자산이 저장되지 않습니다.
- 이 배열에 없는 자산은 사이트를 게시할 때 삭제됩니다.
- assets 매개변수는 getProperty에서 반환된 것과 동일한 형식의 자산 배열이며 filePicker에서도 반환됩니다.

### 주:

url 값은 저장되지 않습니다. 이 값은 자산을 요청할 때 동적으로 생성됩니다.

```
SitesSDK.filePicker(options, callback);
```

- 자산 목록을 선택할 파일 선택기를 불러오는 API입니다.
- 성공적인 자산 선택 시 콜백을 호출하여 선택한 자산의 배열을 전달합니다.
- 이 시점에는 아무것도 저장되지 않으며 setProperty('componentAssets', [assets]);를 호출하는 구성요소만이 이 선택사항과 다른 저장될 자산을 결합한 항목을 저장할 수 있습니다.

## 자산 선택의 예

이 섹션은 자산을 선택하여 해당 ID를 저장하고 저장된 자산에서 실제 값을 다시 인출하는 방법을 보여줍니다.

1. settings.html 파일을 편집합니다.
2. Image selection을 포함하도록 템플릿 객체를 변경합니다.

```
<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. 선택한 자산의 ID를 저장할 observable을 추가하도록 viewModel을 변경합니다.

```
self.imageID = ko.observable();
```

4. 파일 선택기를 불러와서 선택한 자산의 이름을 표시하여 자산 선택을 관리하도록 viewModel을 변경합니다.

```
//
// 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. `customSettingsData`를 가져오기 전에 먼저 자산을 검색하도록 `viewModel`을 업데이트합니다. 또한 이 코드는 `self.imageID()` `observable`이 변경될 때 `self.imageName`을 호출합니다.

```
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. 마지막으로, `save` 함수를 업데이트하여 `imageID`를 저장하고 참조된 자산 목록으로 `componentAssets`를 업데이트해야 합니다.

```
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);
```

### 자산 선택의 결과 확인

1. 사이트 작성기가 구성요소의 변경사항을 가져올 수 있도록 사이트의 페이지를 새로고침합니다.
2. 페이지를 편집 모드로 전환합니다.
3. 구성요소를 페이지에 끌어 놓습니다.
4. 설정 패널을 불러옵니다.
5. **이미지 선택** 단추를 누릅니다.
6. 이미지를 찾아보고(또는 업로드) 선택합니다.  
선택한 이미지를 보여주며 이미지 이름이 저장됩니다.
7. 설정 패널을 닫습니다.
8. 설정 패널을 다시 불러옵니다.  
이미지 이름이 다시 반영됩니다.

## 자산 렌더링의 예

이 섹션은 자산을 검색하여 구성요소에 렌더링하는 방법을 보여주며, 설정 패널에서 값이 변경될 때마다 구성요소가 동적으로 업데이트되는지 확인합니다.

### 주:

여기서는 페이지의 인라인 프레임에 있는 로컬 구성요소의 예를 보여주지만, 비슷한 코드가 페이지에 인라인으로 렌더링된 구성요소에도 작동합니다.

1. render.html 파일을 편집합니다.
2. 자산을 포함하도록 템플릿을 업데이트합니다.

```
<!-- 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. viewModel에서 두 개의 observable을 생성하여 customSettingsData에서 imageID를 가져오고 저장된 자산 목록에서 검색된 imageURL을 저장합니다.

```
self.imageID = ko.observable();
self.imageURL = ko.observable();
```

4. imageID가 변경될 때마다 해당 이미지 자산 URL을 가져오도록 viewModel을 업데이트합니다.

```
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. customSettingsData에서 ID를 검색하도록 viewModel을 업데이트합니다.

## 자산 렌더링의 결과 확인

1. 사이트 작성기가 구성요소의 변경사항을 가져올 수 있도록 사이트의 페이지를 새로고침합니다.
2. 페이지를 편집 모드로 전환합니다.
3. 구성요소를 페이지에 끌어 놓습니다.
4. 설정 패널을 불러옵니다.

5. **이미지 선택** 단추를 누릅니다.
6. 이미지를 찾아보고(또는 업로드) 선택합니다.  
선택한 이미지를 보여주며 이미지 이름이 저장됩니다.
7. 설정 패널을 닫습니다.  
이 시점에서 선택한 이미지가 구성요소에 렌더링되어야 합니다.

자습서 검토를 계속합니다.

## 자습서 검토

이 자습서는 Knockout 구성요소 팩토리를 사용하여 사용자정의된 구성요소를 생성하는 방법의 개요를 제공합니다.

이 자습서의 주 목적은 이 패턴을 사용하여 `SampleComponentViewModel` 및 `sampleComponentTemplate` JavaScript 객체의 업데이트만으로 사용자정의 구성요소를 생성하는 것입니다. `sampleComponentFactory` 및 `SampleComponentImpl` 객체는 자습서 진행 중에 변경되지 않았습니다. 페이지와의 통신을 다루지 않아도 이 변경사항을 구현할 수 있었고 다음 작업을 수행할 수 있었습니다.

- 설정 패널의 변경사항을 구성요소로 전달하고 이 변경사항이 지속되도록 합니다.
- 트리거 및 작업을 실행하고 페이지의 다른 구성요소와 상호작용합니다.
- 레이아웃을 생성하고 중첩 구성요소를 활용합니다.
- 구성요소별 스타일을 정의합니다.

이 예제는 사용자정의 구성요소를 여러 파일로 분할했지만 이는 자습서의 명확성을 위한 것입니다. 최적화를 위해 여러 번의 다운로드를 피하려면 적절히 파일을 패키징해야 합니다.

마지막으로, 이 자습서는 Knockout 기반 구성요소에 적합하지만 다른 JavaScript 기술 스택(예: AngularJS)을 사용하여 사용자정의 구성요소를 생성하려면 `SampleComponentImpl` 객체를 다시 구현하여 실제 구성요소의 기술적 구현과 함께 해당 프레임워크와의 통신을 생성해야 합니다. 이 작업은 이 자습서의 범위를 벗어납니다.

## Sites SDK 참조

Oracle Content Management JavaScript SDK(Sites SDK)에는 구성요소가 Oracle Content Management와 더 통합된 경험을 제공할 수 있도록 지원하는 함수 집합이 있습니다.

Sites SDK는 Oracle Content Management 서버에서 다운로드할 수 있습니다.

`http://{server}/_sitesclouddelivery/renderer/app/sdk/js/sites.min.js`

Sites API(애플리케이션 프로그래밍 인터페이스) 정의는 전역 객체 `SitesSDK`의 모든 정의된 네임스페이스를 다룹니다.

- `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 SDK의 주 목적은 사용자가 애플리케이션을 빌드하고 관리하도록 하는 것입니다. 여러 기능을 제공합니다.

- 인라인 프레임 및 도메인 경계를 넘은 통신
  - 구성요소 호스팅 위치에 관계없이 구성요소가 페이지와 통신할 수 있음
  - 페이지 모델 수명 주기에 참여:
    - \* 구성요소 생성
    - \* 저장된 속성으로 구성요소 렌더링
    - \* 구성요소 업데이트 및 변경사항 저장
    - \* 페이지에서 구성요소 삭제
  - 페이지에 사용할 구성요소 구성에 사용된 사용자정의 설정 저장 및 검색
  - 구성요소간 통신을 사용하여 트리거 및 작업 참여
- 페이지에 형상 힌트(사용자정의 구성요소에는 적용되지 않음)
  - 구성요소가 지시할 때 페이지의 인라인 프레임 크기를 동적으로 조정할 수 있음



- 결과적으로 구성요소가 페이지의 일부인 것처럼 보이고 구성요소 크기 조정 이벤트에 응답하여 페이지가 이동함(스크롤 막대 없음)
- Oracle Content Management에서 호출될 때 구성요소 인지
  - 구성요소와 페이지 간의 통신에는 전용 키로 서명한 토큰이 관여합니다.
  - Sites SDK는 구성요소가 토큰을 디코딩하고 호출자 확인을 용이하게 합니다.

알려진 스크립트 경로를 제공하여 Sites SDK 함수를 포함합니다. 예를 들어, 다음과 같습니다.

```
<script type="text/javascript"
src="<sdk_install_dir_path_prefix>/sites.min.js">
</script>
```

Sites SDK에는 다음 전역 객체가 있습니다.

- SitesSDK
- SitesSDK.Utils

### SitesSDK

SitesSDK 전역 객체는 렌더링 및 설정 끝점에서 사용할 수 있습니다.

명령	작업
<code>SitesSDK.getProperty(propertyName, callback)</code>	호스트 사이트에서 명명된 속성 값을 가져옵니다.
<code>SitesSDK.setProperty(propertyName, propertyValue)</code>	명명된 속성 값을 <i>propertyValue</i> 로 설정합니다.
<code>SitesSDK.getSiteProperty(propertyName, callback)</code>	호스트 사이트에서 명명된 사이트 속성 값을 가져옵니다. 예를 들어, 호스트 사이트에서 사용 중인 현재 테마 디자인을 인출할 수 있습니다.
<code>SitesSDK.publish(messageType, payload)</code>	서버에 메시지를 전송하고 <i>messageType</i> 및 JSON 객체를 페이로드로 수락합니다.
<code>SitesSDK.subscribe(messageType, callback)</code>	호스트 사이트에서 작업 할당된 메시지의 메시지 리스너입니다. 비동기 호출입니다. 이것은 JavaScript 콜백이므로 함수에 JavaScript 클로저를 사용하거나 함수를 적절히 바인딩하여 적합한 컨텍스트에 액세스할 수 있도록 해야 합니다.

### SitesSDK.Utils

`SitesSDK.Utils` 네임스페이스에는 Oracle Content Management 구성요소의 모든 끝점에서 호출할 수 있는 유틸리티 함수가 있습니다. 이 네임스페이스는 원격 구성요소의 설정 패널에서 사용할 수 있지만 인라인 로컬 구성요소에는 제공되지 않습니다.

**▲ 주의:**

이 함수는 오라클이 공식적으로 지원하지 않으며 샘플 구현을 위한 것입니다. 사용자 책임하에 사용하십시오.

명령	작업
<code>SitesSDK.Utils.Logger</code>	Sites SDK에서 사용 중인 로거 객체를 반환합니다. 다음 중 하나로 로그 레벨을 설정할 수 있습니다. <ul style="list-style-type: none"> <li>• debug</li> <li>• log</li> <li>• info</li> <li>• error</li> <li>• warn</li> </ul>
<code>SitesSDK.Utils.addSiteThemeDesign(cssUrl)</code>	현재 사이트 테마 디자인을 구성요소에 추가합니다.
<code>Utils.getStyle(elem,styleProp)</code>	인라인 프레임의 높이를 계산하는 유틸리티 메소드입니다.

## SitesSDK.getProperty('componentAssets', callback)

이 함수는 사이트 내의 구성요소 대신 현재 저장 중인 자산 목록을 반환합니다.

### 매개변수

이름	유형	설명
<code>componentAssets</code>	문자열	목록 생성자를 호출합니다.
<code>callback</code>	함수	콜백 함수

### 사용법

```
// 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)

이 함수는 호스트 사이트에서 명명된 속성 값을 가져옵니다.

### 매개변수

이름	유형	설명
propertyName	문자열	속성의 이름입니다.
callback	함수	콜백 함수.

### 관련 속성

속성 이름	설명	설정	렌더링
height	프레임의 높이	해당 사항 없음	예
width	프레임의 너비	해당 사항 없음	예
customSettingsData	설정 데이터	예	예
styleClass	선택된 스타일 클래스	예	예
borderColor, borderRadius, alignment, borderStyle, margin 등의 기타 스타일 속성			

### 사용법

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

**렌더링 URL에서 인출된 추가 관련 속성**

속성 값을 인출하는 `getProperty()` 사용 외에도, 렌더링 URL에서 인출할 수 있는 다른 관련 속성은 다음과 같습니다.

속성 이름	설명
id	구성요소 ID
instance	암호화된 토큰
width	iframe의 너비
height	iframe의 높이
viewMode	렌더링 모드 값
locale	사이트 또는 브라우저에 설정된 언어
settingsId	현재 설정 ID

**설정 URL에서 인출된 추가 속성**

속성 값을 인출하는 `getProperty()` 사용 외에도, 설정 URL에서 인출할 수 있는 다른 속성은 다음과 같습니다.

속성 이름	설명
currCompId	편집 중인 구성요소의 ID
instance	암호화된 토큰
width	설정 iframe의 너비
locale	사이트 또는 브라우저에 설정된 언어
OrigSettingsId	구성요소의 원래 설정 ID
settingsId	새 설정 ID

## SitesSDK.getSiteProperty(propertyName, callback)

이 함수는 호스트 사이트에서 명명된 사이트 속성 값을 가져옵니다. 예제는 호스트 사이트에서 사용 중인 현재 테마 디자인을 인출하는 것입니다.

**매개변수**

이름	유형	설명
propertyName	문자열	사이트 속성의 이름
callback	함수	속성 값을 읽는 콜백 함수

**사이트 속성**

속성 이름	설명	설정	렌더링
theme	현재 테마 디자인의 URL	해당 사항 없음	예

## 사용법

```

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])

이 함수는 사용자정의 구성요소 대신 저장된 사이트 자산 목록을 업데이트합니다.

### 매개변수

이름	유형	설명
componentAssets	문자열	구성요소 자산 목록을 생성합니다.
assets	JSON 객체	선택된 자산 목록을 반환합니다.

### 인수

속성 이름	설명
filetypes	지원되는 파일 유형 목록입니다.
multiSelect	단일 또는 다중 파일을 선택합니다.

## 사용법

```

// get/set list of assets
SitesSDK.getProperty('componentAssets', callback);
SitesSDK.setProperty('componentAssets', [assets]);

// invoking list creator
// after calling this, the developer needs to call setProperty against
// 'componentAssets' to persist all currently used assets
//
// args:
// fileTypes - list of supported file types
// multiSelect - single or multiple file select
//
// returns a list of selected assets. If user cancels out of the
// picker,
// the callback is never called
SitesSDK.filePicker({options}, callback);

```

```
//
// example 'componentAssets' JSON returned:
//
'componentAssets': [{
  'name': <nameOfAssetInContentFolder>, // this is used to uniquely and
  persistently identify the asset. It is typically prefixed by the
  <originalName> followed by a 16 digit string
  'originalName': <originalName>, // name of the asset as selected
  from DOCS
  'description': <description>, // description/other attributes
  that are available from DOCS
  'url': <fully qualified url to the asset>
}]
```

## SitesSDK.setProperty(propertyName, propertyValue)

이 함수는 명명된 속성 값을 *propertyValue*로 설정합니다.

### 매개변수

이름	유형	설명
propertyName	문자열	속성의 이름입니다.
propertyValue	JSON 객체	속성의 값입니다.

### 관련 속성

이 표에는 setProperty()를 사용하여 인출할 수 있는 관련 속성 집합이 나와 있습니다.

속성 이름	설명	설정	렌더링
height	프레임의 높이	해당 사항 없음	예
width	프레임의 너비	해당 사항 없음	예
customSettingsData	설정 데이터	예	예
styleClass	선택된 스타일 클래스	예	예
description	페이지의 로컬 구성요소 인스턴스에 배너로 사용됩니다.		
layout	로컬 구성요소에만 현재 선택된 레이아웃을 반환합니다. appinfo.json 파일의 값을 기반으로 설정 패널에서 업데이트할 수 있습니다.		
renderStatus	사용자정의 구성요소가 렌더링 완료를 보고할 수 있습니다.	true	
borderColor, borderRadius, alignment, borderStyle, margin 등의 기타 스타일 속성			

## 사용법

```
// 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)

이 함수는 선택된 파일 목록을 반환합니다.

## 매개변수

이름	유형	설명
options	문자열	선택된 파일을 반환하도록 지정하는 옵션입니다.
callback	함수	콜백 함수. 사용자가 선택기를 취소하면 콜백이 호출되지 않습니다.

## 사용법

```
// 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)

이 함수는 선택된 파일 목록을 반환합니다.

SitesSDK.openDocumentPicker() 함수는 단일 인수를 취하여 선택한 문서로 분석되는 Promise를 반환합니다. 사용 가능한 옵션은 *Oracle Content Management 통합 및 확장*에 있는 Documents View Embed UI (documentsView Component) API V2의 documentsView 구성요소와 일치합니다.

### 사용법

```

SitesSDK.openDocumentPicker({
  selectable: "any",
  layout: "grid"
}).then(function (selection) {
  console.log(selection);
});

```

문서가 검색된 경우 [SitesSDK.getProperty\('componentAssets', callback\)](#) 및 [SitesSDK.setProperty\('componentAssets', \[assets\]\)](#) 속성을 사용하여 사이트에 해당 문서의 ID를 저장하도록 Oracle Content Management에 알려야 합니다. 그렇지 않으면 사이트에 게시되지 않으며 삭제될 수 있습니다.

## SitesSDK.openAssetPicker(options)

이 함수는 선택된 파일 목록을 반환합니다.

SitesSDK.openAssetPicker() 함수는 단일 인수를 사용하며 선택된 자산으로 분석되는 Promise를 반환합니다. 사용 가능한 옵션은 *Oracle Content Management 통합 및 확장*에 있는 Assets View Embed UI(assetsView 구성요소) API V2의 assetsView 구성요소와 일치합니다.

### 사용법

```

SitesSDK.openAssetPicker({
  select: "single"
}).then(function (selection) {
  console.log(selection);
});

```

자산이 검색된 경우 [SitesSDK.getProperty\('componentAssets', callback\)](#) 및 [SitesSDK.setProperty\('componentAssets', \[assets\]\)](#) 속성을 사용하여 사이트에 해당 자산의 ID를 저장하도록 Oracle Content Management에 알려야 합니다. 그렇지 않으면 사이트에 게시되지 않으며 삭제될 수 있습니다.



## SitesSDK.publish(messageType, payload)

이 함수는 서버에 메시지를 전송합니다. messageType 및 JSON 객체를 페이로드로 수락합니다. 메시지 유형이 인식되고 페이지에서 처리됩니다. 처리기가 없으면 전달된 메시지 유형이 무시됩니다.

### 매개변수

이름	유형	설명
messageType	문자열	메시지 유형
payload	JSON 객체	메시지 페이로드

### 메시지 유형

이 표에는 SitesSDK.publish() 함수를 사용하여 전송할 수 있는 로컬(인라인 프레임 사용) 또는 원격 구성요소에 알려진 메시지 유형이 나와 있습니다.

메시지 유형	설명
SETTINGS_UPDATED	설정 패널에서 사용되어 설정 업데이트 메시지를 렌더링 끝점으로 보냅니다.
TRIGGER_ACTIONS	구성요소 렌더링 끝점에서 사용되어 호스트 사이트에 작업을 트리거합니다.

### 사용법

```
// raise trigger
SitesSDK.publish(SitesSDK.MESSAGE_TYPES.TRIGGER_ACTIONS, {
    'triggerName': 'scsChangeSettings',
    'triggerPayload': { 'url':
this.configuration()['url'],'feedcount': this.configuration()['limit'] }
});
```

## SitesSDK.subscribe(messageType, callback)

이 함수는 사실상 호스트 사이트에서 작업 할당된 메시지의 메시지 리스너입니다. 비동기 호출입니다.

등록된 콜백은 SDK에서 특정 유형의 메시지를 수신할 때 호출됩니다. 콜백이 값을 반환하면 해당 값이 페이지로 다시 반환됩니다. 이 함수로 전달된 콜백이 없으면 특정 메시지 유형에 대해 등록된 리스너가 삭제됩니다.

이것은 JavaScript 콜백이므로 함수에 JavaScript 클로저를 사용하거나 함수를 적절히 바인딩하여 적합한 컨텍스트에 액세스할 수 있도록 해야 합니다.

## 매개변수

이름	유형	설명
messageType	문자열	메시지 유형: <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	함수	호스트 사이트에서 메시지가 수신될 때 함수입니다.

## 메시지 유형

메시지 유형	설명
SETTINGS_UPDATED	이 메시지를 구독하여 설정 패널에서 값이 변경될 때 통지를 받습니다. 일반적으로 구성요소는 사용자 인터페이스의 표현을 업데이트하여 응답합니다.
TRIGGER_ACTIONS	이 메시지를 게시하면 트리거가 발생하고 페이로드가 전달됩니다.
EXECUTE_ACTION	이 메시지를 구독하여 사용자정의 작업을 처리합니다. 리스너는 일반적으로 주어진 작업을 수행하여 이 메시지를 처리합니다.
GET_ACTIONS	이 메시지를 구독하여 구성요소가 실행할 수 있는 작업의 배열을 선언합니다. 작업의 배열을 반환합니다.
GET_TRIGGERS	이 메시지를 구독하여 구성요소가 발생시킬 수 있는 트리거의 배열을 선언합니다. 트리거의 배열을 반환합니다.
COPY_CUSTOM_DATA	이 메시지를 구독하여 클립보드로 사용자정의 설정 데이터 복사를 처리합니다. 클립보드에 넣을 데이터를 나타내는 객체를 반환합니다.
PASTE_CUSTOM_DATA	이 메시지를 구독하여 클립보드에서 사용자정의 설정 데이터 붙여넣기를 처리합니다.

## 사용법

일부 리스너는 데이터가 전달될 것으로 예상하고, 일부는 데이터를 반환할 것으로 예상됩니다.

### 예제 1:

메시지 유형 EXECUTE\_ACTION에 대한 메시지 리스너를 등록합니다.

```
SitesSDK.subscribe(SitesSDK.MESSAGE_TYPES.EXECUTE_ACTION,
self.executeAction,self);
```

등록된 callback 함수는 다음과 같이 인수를 수락해야 합니다.

```
// 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

}

```

**예제 2:**

COPY\_CUSTOM\_DATA 리스너는 일반적으로 다음과 비슷한 코드로 구현됩니다.

```

// listen for COPY_CUSTOM_DATA request
SitesSDK.subscribe(SitesSDK.MESSAGE_TYPES.COPY_CUSTOM_DATA,

```

copyCustomDataListener() 메소드는 다음과 같이 클립보드로 복사할 데이터를 나타내는 객체를 반환합니다.

```

// Handle Copy Style (save customSettingsData to the clipboard)
self.copyCustomDataListener = function() {
    return {
        prop1: this.prop1(),
        prop2: this.prop2()
    };
};

```

**예제 3:**

PASTE\_CUSTOM\_DATA 요청은 데이터를 수락하지만 아무것도 반환하지 않아도 됩니다.

```

// listen for PASTE_CUSTOM_DATA request
SitesSDK.subscribe(SitesSDK.MESSAGE_TYPES.PASTE_CUSTOM_DATA,
pasteCustomDataListener);

```

동반 항목:

```

// Handle Paste Style (apply customSettingsData from the clipboard)
self.pasteCustomDataListener = function(data) {
    ...
};

```

## SitesSDK.Utils.addSiteThemeDesign(cssUrl)

이 함수는 현재 페이지 HTML <head> 태그에 link 요소를 생성합니다. 소스는 cssUrl 경로로 설정됩니다.

**▲ 주의:**

Utils 네임스페이스의 함수는 오라클이 공식적으로 지원하지 않습니다. 샘플 구현으로 활용하기 위한 것입니다. 사용자 책임하에 사용하십시오.

## 매개변수

이름	유형	설명
cssUrl	문자열	현재 테마 디자인의 URL 경로. 원격 구성요소에만 사용됩니다. 로컬 구성요소(iframe으로 실현)는 URL 대신 페이지에서 속성을 인출해야 합니다.

## 사용법

이 함수는 다음 샘플 코드와 같이 호스트 사이트에서 현재 테마 디자인을 인출하는 것과 함께 사용됩니다.

```
// 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.Utills.addSiteThemeDesign(data.url);
        }
    }
});
```

## SitesSDK.Utills.Logger

이 함수는 Sites SDK에서 사용 중인 로거 객체를 반환합니다.

### ▲ 주의:

utils 네임스페이스의 함수는 오라클이 공식적으로 지원하지 않습니다. 샘플 구현으로 활용하기 위한 것입니다. Utills 함수는 사용자 책임하에 사용하십시오.

로거 레벨은 debug, log, info, error 또는 warn으로 설정할 수 있습니다.

# 28

## 문제 해결

사이트를 생성할 때 몇 가지 질문에 직면할 수 있습니다. 여기에 몇 가지 답변이 있습니다.

- 사이트를 생성하려고 시도하는 중이지만 템플릿이 없음
- 사이트를 삭제할 수 없는 경우
- 사이트 트리를 열거나 페이지를 편집할 수 없는 경우
- 구성요소를 추가했지만 페이지에 노출되지 않는 경우
- 내 폴더, 파일 및 대화 구성요소가 작동하지 않음
- 업로드한 새 버전의 이미지가 페이지에 표시되지 않는 경우
- 페이지 레이아웃 변경 후 내 콘텐츠 일부가 사라지는 경우
- 다른 서비스를 기반으로 추가한 구성요소가 작동하지 않음
- 내 엔터프라이즈 사이트에 경고가 표시됨

### 사이트를 생성하려고 시도하는 중이지만 템플릿이 없음

템플릿이 설치되고 공유되어야만 사이트를 생성할 수 있습니다.

관리자가 Oracle Content Management와 함께 제공되는 템플릿을 설치하고 공유해야만 사용할 수 있습니다. 관리자에게 문의하십시오.

마찬가지로 내가 생성한 템플릿은 명시적으로 공유하기 전까지 다른 사람이 사이트에서 импорт 또는 복사하거나 생성 기반으로 활용하는 등의 방식으로 사용할 수 없습니다.

*Administering Oracle Content Management*의 사이트 설정 구성을 참조하십시오.

### 사이트를 삭제할 수 없는 경우

사이트를 삭제할 수 없는 몇 가지 이유가 있습니다.

- 사이트가 온라인입니다.

먼저, 사이트를 오프라인으로 가져온 후 삭제하십시오. 사이트를 오프라인으로 가져오려면 사이트의 관리자 롤을 보유해야 합니다.

- 필요한 권한이 없습니다.

사이트를 직접 생성했거나(사이트 소유자) 누군가 나와 사이트를 공유하고 내게 제공자 또는 관리자 롤을 부여한 경우 사이트를 삭제할 수 있습니다.

자세한 내용은 [사이트 및 사이트 설정 관리](#)를 참조하십시오.

## 사이트 트리를 열거나 페이지를 편집할 수 없는 경우

사이트 작성기는 기본적으로 미리보기 모드로 열려서 사이트를 볼 수 있지만 편집할 수는 없습니다. 편집기가 편집 모드인 경우에만 사이트를 편집할 수 있습니다.



가 편집으로 설정되어 있는지 확인합니다.

## 구성요소를 추가했지만 페이지에 노출되지 않는 경우

여기에는 몇 가지 가능한 이유가 있습니다.

- 원격 구성요소(앱)와 연관된 URL이 적합한지 확인합니다. [원격 구성요소 등록](#)을 참조하십시오.
- 편집기에서 원격 구성요소는 인라인 프레임(iframe 태그)이라는 HTML 요소로 묶입니다. 모든 원격 구성요소가 인라인 프레임으로 묶이는 것은 아닙니다. 구성요소 제공자에게 문의하여 인라인 프레임으로 묶을 수 있는지 알아보십시오.

## 내 폴더, 파일 및 대화 구성요소가 작동하지 않음

Oracle Content Management와 통신하는 구성요소에는 특정 리소스와 설정이 필요합니다.

다음 구성요소를 사용하려면 Oracle Content Management의 리소스에 대한 액세스 권한이 필요합니다.

- 폴더 목록
- 파일 목록
- 문서 관리자
- 대화

폴더 및 파일 구성요소를 사용하려면 REST API(애플리케이션 프로그래밍 인터페이스)에 대한 액세스 권한이 필요합니다.

다음 사항을 확인하십시오.

- 폴더 및 파일 REST 작업에 액세스하려면 구성요소에서 API 끝점에 액세스할 수 있어야 합니다.

기본적으로 Oracle Content Management REST API 끝점은 사이트에 제공된 표준 URL을 사용하는 경우 제공됩니다. 사용자정의 URL을 사용할 경우 /documents REST API 끝점에 대한 액세스를 명시적으로 제공해야 할 수도 있습니다.

사이트 URL을 매핑하는 방법(를) 참조하십시오.

## 페이지 레이아웃 변경 후 내 콘텐츠 일부가 사라지는 경우

명명된 슬롯이 더 적거나 다른 레이아웃을 선택하면 다른 슬롯의 기존 콘텐츠는 새 레이아웃에 표시되지 않습니다.

콘텐츠는 삭제되지 않으며, 단지 선택한 레이아웃에 동일한 이름의 슬롯이 없는 한 콘텐츠를 표시할 수 없습니다.

슬롯 수가 동일한 레이아웃을 선택해야 합니다. 선택하는 레이아웃의 슬롯 수는 동일하지만 일부 콘텐츠가 표시되지 않을 경우 슬롯의 이름이 다른 것일 수 있습니다. 불일치를 해결하려면 테마 디자이너에게 문의하십시오.

## 업로드한 새 버전의 이미지가 페이지에 표시되지 않는 경우

구성요소에 사용할 이미지를 선택하는 경우 고유 복사본이 생성되어 사이트와 함께 저장됩니다.

파일을 업로드하면 사이트 저장소에 해당 파일이 저장됩니다. 기존 파일과 이름이 동일한 파일을 업로드하면 새 버전의 파일이 생성됩니다.

페이지에 표시되는 이미지 파일은 저장소에서 선택된 파일의 복사본이며 자동으로 업데이트되지 않습니다. 이는 설계 시 지정된 것으로 저장소에서 이미지 파일이 업데이트될 때 기존 페이지에서 의도치 않은 변경사항이 발생하지 않도록 방지합니다.

구성요소에 사용된 이미지를 업데이트하려면 구성요소 속성 페이지에서 명시적으로 이미지를 선택해야 합니다. 자세한 내용은 [이미지](#)를 참조하십시오.

## 다른 서비스를 기반으로 추가한 구성요소가 작동하지 않음

사이트를 다양한 프로세스 및 서비스와 통합할 수 있습니다.

Process Cloud Service와의 통합을 기준으로 구성요소를 생성한 경우 Oracle Intelligent Advisor(이전의 Oracle Policy Automation) 및 기타 서비스가 동일한 ID 도메인을 사용해야 합니다. 통합 설정자에게 두 서비스가 동일한 ID 도메인을 사용하는지 확인하십시오.

## 내 엔터프라이즈 사이트에 경고가 표시됨

엔터프라이즈 사이트에 v1.0 콘텐츠 레이아웃이 포함된 경우 사이트 작성기에 경고가 표시되며 사이트를 볼 때도 경고가 표시됩니다. 레이아웃을 렌더링할 수 없기 때문입니다. 따라서 v1.1 콘텐츠 레이아웃으로 업데이트해야 합니다.

이 문제는 다음 두 가지 방식 중 하나로 발생할 수 있습니다.

- 기존 사이트를 변환 가능 사이트로 설정하면 해당 사이트에 기본 언어 및 지역화 정책이 지정되어 사이트가 자동으로 엔터프라이즈 사이트로 변환됩니다. 사이트 템플릿에 v1.0 콘텐츠 레이아웃이 포함되면 콘텐츠 레이아웃을 렌더링할 수 없습니다.
- v1.0 콘텐츠 레이아웃을 사용하는 이전 템플릿을 기반으로 엔터프라이즈 사이트를 생성하면 콘텐츠 레이아웃을 렌더링할 수 없습니다.

이 문제를 해결하려면 콘텐츠 레이아웃을 v1.1 콘텐츠 레이아웃으로 업데이트하십시오. 차이가 구분되도록 구성요소 카탈로그에서 새 콘텐츠 레이아웃을 생성할 수 있습니다. 행을 추가하여 콘텐츠 레이아웃 버전을 등록하고 `data.fieldname`을 `fields.fieldname`으로 변경해야 합니다.