

# Getting started - Demo setup

The goal of this section is to show you how to set up your environment and copy the integration code in order to try out the Contentful and Composable Storefront Powertools Store demo. Each step will:

- Describe the required actions
  - Provide links to the official documentation from Contentful or SAP
  - Show an example of the expected output so you can verify that the step is completed.
- 

The assumption is that you will be starting with none of the required tools and applications set up, however, if you already have an SAP Commerce or a Contentful instance set up, feel free to skip some steps as long as you can verify that what you have matches the expected output for that particular section.

## Table of contents:

SAP Commerce Cloud Setup
Installing a local copy of Commerce Cloud 2211
Using an existing instance of Commerce Cloud 2211 in CCV2
Setting up additional HTTP response headers in order to use Live Preview
Verifying the installation of SAP Commerce Cloud 2211
SAP Composable Storefront Setup
Using the example code provided in the repository
Contentful Setup
Setting up the Contentful Access Tokens
Integration Demo Data Import
Marketplace App - Contentful Connector for SAP Commerce Cloud
Marketplace App - Flex Fields
Check content localization
Content preview platform for the Live Preview
Contentful and Composable Storefront Integration code
Verifying Demo Setup outcome
Updating Content Model
Using Live Preview

Before we begin the specifics of the integration between Contentful and Composable storefront, we would like to get to a state where we have both of those products installed, configured and working independently.

## SAP Commerce Cloud Setup

Since we will be using SAP Composable Storefront and one of the SAP's default storefronts for product information (Powertools Store), we need to have an SAP Commerce setup and initialize it with proper data.

We can do this in two ways:

1. Download and install a local copy of SAP Commerce Cloud 2211
2. Use an existing instance of SAP Commerce Cloud 2211 deployed on Commerce Cloud V2

## Installing a local copy of Commerce Cloud 2211

In order to correctly install Commerce Cloud 2211 on your local machine and initialize it with the Powertools Store data, please follow the official documentation found here:

## Using an existing instance of Commerce Cloud 2211 in CCV2

If you already have an instance of SAP Commerce Cloud deployed in SAP CCV2 cloud solution, you will have to set up additional HTTP response headers in order to let Contentful host the store website in an iframe when using the Live Preview functionality.

### Setting up additional HTTP response headers in order to use Live Preview

You can find the required headers in the official Contentful Live Preview setup guide:

#### [Contentful - Live Preview Setup](#)

Refer to the official SAP documentation on how to add HTTP Response header sets:

#### [SAP Commerce Cloud in the Public Cloud - Creating an HTTP Response Header Set](#)

At the time of writing, headers that need to be set up under **Security > HTTP Response Header Sets** in order for Live Preview to work are the following:

- **Header Name:** Content-Security-Policy
- **Header Value:** frame-ancestors 'self' https://app.contentful.com
- **Apply Condition:** ALWAYS
- **Apply Action:** SET

and:

- **Header Name:** X-Frame-Options
- **Header Value:**
- **Apply Condition:** ALWAYS
- **Apply Action:** UNSET

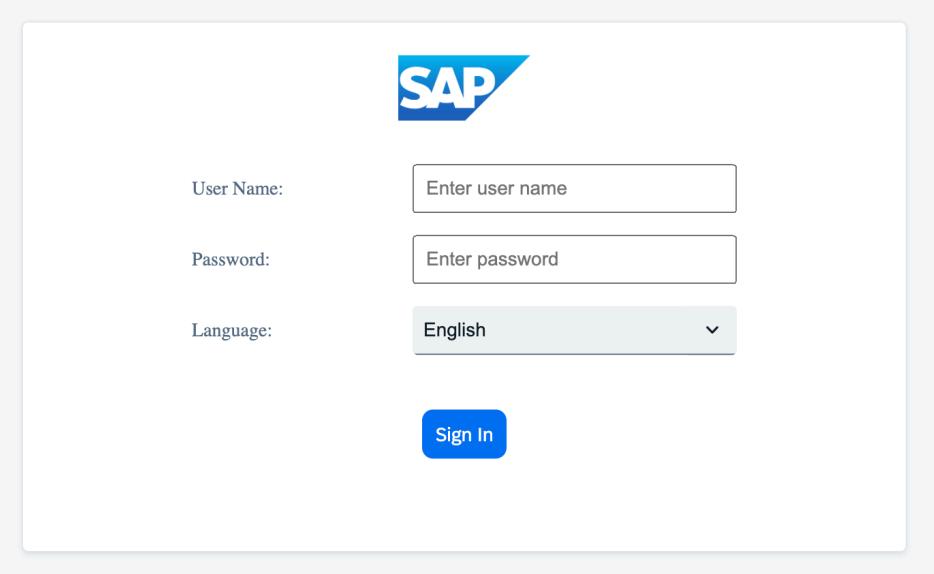
After you create a set containing these headers, you will have to assign the newly created set to the **JS Storefront** endpoint, please follow the official SAP documentation in order to achieve this:

#### [SAP Commerce Cloud in the Public Cloud - Assigning an HTTP Response Header Set to an Endpoint](#)

## Verifying the installation of SAP Commerce Cloud 2211

After you install your SAP Commerce Cloud 2211 instance, make sure to have access to the following resources:

1. SAP Commerce Backoffice
  - a. For local instances, you can access the Backoffice at <https://localhost:9002/backoffice/>
  - b. For CCV2 instances, you can find the Backoffice endpoint at **Environments > [environment] > Public Endpoints > Backoffice**



## 2. powertools-spa website within the **WCMS > Website** section of the Backoffice:

ID	Name	Active	Content Catalogs	Data Isolation Enabled
electronics-spa	Spartacus Electronics Site	true	SPA Electronics Content Catalog	false
electronics-standalone	Electronics Site Standalone	true	Electronics Content Catalog	true
electronics	Electronics Site	true	Electronics Content Catalog	false
apparel-uk-spa	Spartacus Apparel UK Site	true	SPA Apparel UK Content Catalog	false
apparel-de-standalone	Apparel Site DE Standalone	true	Apparel DE Content Catalog	true
apparel-de	Apparel Site DE	true	Apparel DE Content Catalog	false
apparel-uk-standalone	Apparel Site UK Standalone	true	Apparel UK Content Catalog	true
apparel-uk	Apparel Site UK	true	Apparel UK Content Catalog	false
<input checked="" type="checkbox"/> powertools-spa	Spartacus powertools Site	true	SPA Powertools Content Catalog	false
powertools-standalone	Powertools Site Standalone	true	Powertools Content Catalog	true
powertools	Powertools Site	true	Powertools Content Catalog	false

## SAP Composable Storefront Setup ↗

**Note:** For the purposes of this demo, we are using the Powertools SPA store which is a B2B store and in order to enable additional functionality such as adding products to cart and creating orders, additional configuration is needed, but for now the scope is narrowed down to a homepage and faq page experience and in order to keep the flow simple we will keep the default configuration from the SAP Composable Storefront installation documentation. Later, expanded versions of this guide will tackle the rest of the functionalities of this B2B store.

Installing a Composable Storefront Angular application should be done by following the official SAP documentation:

### [Building the Composable Storefront From Libraries](#)

After installing the libraries and running the application you should be greeted with the ability to access two SAP B2C stores: `electronics-spa` and `apparel-uk-spa`.

Language: English ▾ Currency: \$ USD ▾ Theme: Default ▾

[Quick Order](#) [Find a Store](#) [Contact Us](#) [Help](#)



Search

Enter product name or SKU



[Sign In / Register](#)



BRANDS ▾

DIGITAL CAMERAS ▾

FILM CAMERAS

CAMCORDERS

WEBCAMS

ACCESSORIES ▾

**SAVE  
BIG**

On select SLR & DSLR Cameras

[SHOP NOW](#)

SAVE BIG ON SELECT CAMERA ACCESSORIES & SUPPLIES [SHOP NOW](#)

**COMPACT CAMERAS**  
Large assortment of cool Compact Cameras

**CAMERA LENSES**  
Latest pro and consumer Camera Lenses in stock

**CAMCORDERS**  
Latest pro and consumer DV Camcorders in stock

**POWER SUPPLIES**  
Large assortment of Batteries & Power Supplies

Theme: Default ▾

[Quick Order](#) [Find a Store](#) [Contact Us](#) [Help](#)



Search

Enter product name or SKU



[Sign In / Register](#)



BRANDS ▾

STREETWEAR ▾

SNOW ▾

ACCESSORIES ▾

YOUTH ▾

**START  
YOUR  
SEASON**

SHOP WINTER NOW >

SAVE BIG ON SELECT STREETWEAR [SHOP NOW](#)

[SHOP WOMEN](#)

[SHOP MEN](#)

[SHOP TEENS](#)

**OAKLEY**

[SHOP BRANDS](#)

In order to gain access to the Powertools SPA store, that we will be using for the integration demo, you have to edit the `src/app/spartacus/spartacus-configuration.module.ts` file and change the existing context configuration:

```
1 provideConfig(<SiteContextConfig>{
2   context: {
3     urlParameters: ['baseSite', 'language', 'currency'],
4     baseSite: ['electronics-spa','apparel-uk-spa'],
5     currency: ['USD', 'GBP'],],
6   })
7 })
```

to the following:

```
1 provideConfig(<SiteContextConfig>{
2   context: {
3     urlParameters: ['baseSite', 'language', 'currency'],
4     baseSite: ['powertools-spa'],
5     currency: ['USD', 'EUR'],
6     language: ['en', 'de'],},
7 })
```

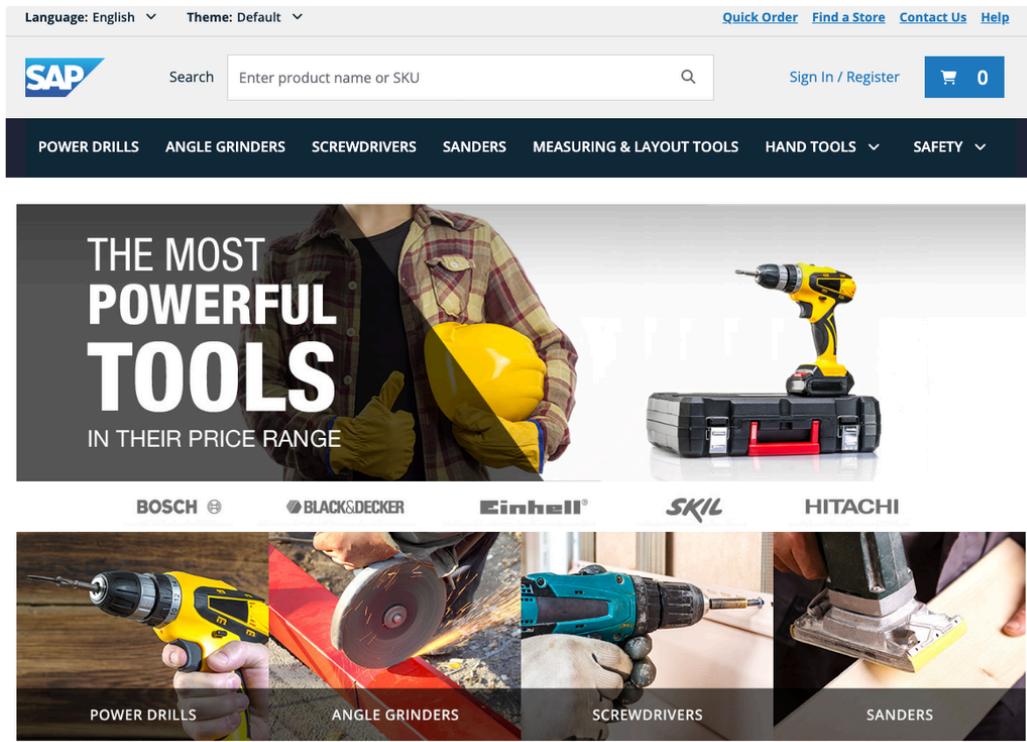
In the current version of Composable Storefront, the "my-account" translations are not automatically provided in the default setup. This may be a temporary issue, but you can manually fix it by modifying the i18n configuration.

```
1 provideDefaultConfig(<I18nConfig>{
2   i18n: {
3     resources: translations,
4     chunks: translationChunksConfig,
5     fallbackLang: 'en',
6   },
7 }),
```

to the following:

```
1 provideDefaultConfig(<I18nConfig>{
2   i18n: {
3     resources: translations,
4     chunks: {
5       ...translationChunksConfig,
6       myAccount: ['closeAccount', 'updatePasswordForm', 'updateProfileForm', 'consentManagementForm',
7       'myCoupons', 'notificationPreference', 'myInterests'],
8     },
9     fallbackLang: 'en',
10   },
11 }),
```

And after the application reload, you will be able to access the Powertools SPA store.



Additionally, you should edit `package.json` file's `start` script to make sure that the app starts in ssl mode:

```
"start": "ng serve --ssl"
```

Now, your Powertools SPA store should be available at:

```
https://localhost:4200/powertools-spa/en/USD/
```

### Using the example code provided in the repository

If you wish to get to the demo functionality as quickly as possible, we have prepared a demo repository with the sample code and the integration already in place.

1. Checkout the code in `contentful-spartacus-demo-app` folder
2. Place your [SAP RBSC key](#) in the `.npmrc` file
3. Edit the `src/app/contentful/root/config/default-contentful-config.ts` and add the required configuration
  - a. Space Id  
(described in [Getting started - Demo setup | Setting up the Contentful Access Tokens](#) )
  - b. Access Token  
(described in [Getting started - Demo setup | Setting up the Contentful Access Tokens](#) )
  - c. Preview Token  
(described in [Getting started - Demo setup | Setting up the Contentful Access Tokens](#) )
4. Make sure that `src/app/spartacus/spartacus-configuration.module.ts` has `baseUrl` property that points to the API of your instance of Commerce Cloud.
5. Run `npm install`
6. Run `npm start`

If you've gone through the Contentful setup as described in the rest of this guide, the Powertools store that reflects the Contentful Content Model and Content should be available at

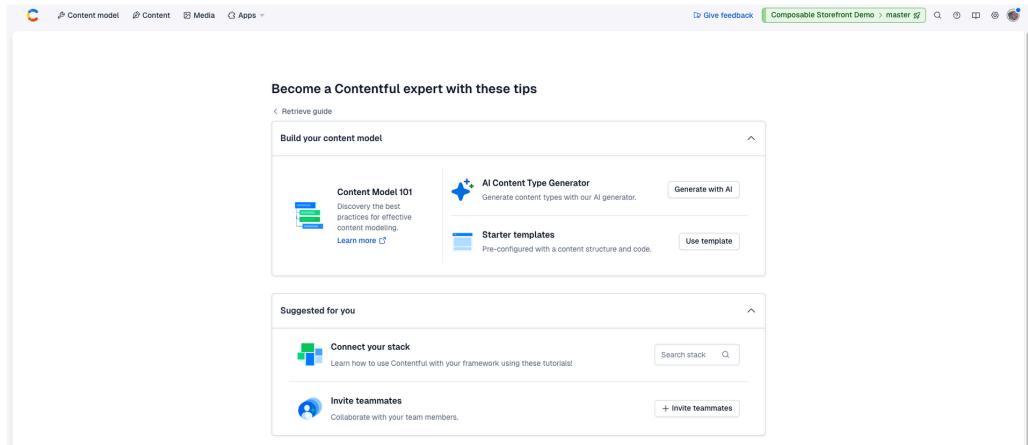
```
https://localhost:4200/powertools-spa/en/USD/
```

## Contentful Setup ↗

You will also need a Contentful Space for importing the content data we prepared for this demo. Simply by signing up with Contentful you will receive a single Space and access to everything that you need in order to get started.

For additional info we recommend the official docs:

[C Beginner's guide to Contentful | Contentful](#)



In order to set up Contentful for use with this demo, you will have to set up a couple of access tokens, import content model and content that we have prepared, install a couple of [Contentful Marketplace Apps](#), set up the localization options and the content preview platform for the Live Preview.

## Setting up the Contentful Access Tokens ↗

In order to successfully import the data that we have prepared and later connect to the Composable Storefront application, you will have to create the following tokens as described in the official documentation [C Authentication](#):

- [Content Management API](#) key. This key will be used during the data import.
- [Content Delivery API](#) and [Content Preview API](#) tokens. These tokens will be used in the Composable Storefront application in order to fetch the content and during the Live Preview to fetch the edited content.

## Integration Demo Data Import ↗

We have provided a set of data that has been exported from an existing Integration Demo space.

In order to import data:

1. Set up the Space localization to `en`

Default localization for a new Space is usually `en-US`, but for the data that we are importing we specifically need it to be `en`

Check the official docs on how to change the localization of the space: [C Manage locales | Contentful](#)

2. Extract the provided import files on your local machine and navigate to the

`powertools-demo-data-import` folder. The folder should have the following structure:

```
1 powertools-demo-data-import/
2   |- import-data/
3   |   |- images.ctfassets.net/
4   |   |   |- ...
5   |   |   |- ...
6   |   |   |- ...
7   |   |- import-data.json
8   |- import-config.json
9   |- README.md
```

Configure the `import-config.json` file by adding your:

- a. Space ID
- b. Environment ID ('master' will be used if left empty)
- c. Content Management API key (as described in

[Getting started - Demo setup | Setting up the Contentful Access Tokens](#))

3. Open the `powertools-demo-data-import` folder in your terminal

4. Follow the instructions from the [C Importing and exporting content with the Contentful CLI](#) in order to install contentful library and authenticate

5. Execute the following import command:

```
contentful space import --config import-config.json
```

You should expect the following output:

```
1 | The following entities are going to be imported:
2 |
3 | Content Types | 25
4 |
5 | Tags | 7
6 |
7 | Editor Interfaces | 25
8 |
9 | Entries | 332
10 |
11 | Assets | 50
12 |
13 | Locales | 2
14 |
15 | Webhooks | 0
16 |
17 |
18 ✓ Validating content-file
19 ✓ Initialize client (1s)
20 ✓ Checking if destination space already has any content and retrieving it (2s)
21 ✓ Apply transformations to source data (1s)
22 ✓ Push content to destination space
23   ✓ Connecting to space (1s)
24   ✓ Importing Locales (1s)
25   ✓ Importing Content Types (4s)
26   ✓ Publishing Content Types (12s)
27   ✓ Importing Tags (2s)
28   ✓ Importing Editor Interfaces (8s)
29   ↓ Uploading Assets [skipped]
30   ✓ Importing Assets (264s)
31   ✓ Publishing Assets (16s)
32   ✓ Archiving Assets (1s)
33   ✓ Importing Content Entries (48s)
34   ✓ Publishing Content Entries (129s)
35   ✓ Archiving Entries (0s)
36   ✓ Creating Web Hooks (1s)
37 Finished importing all data
38 |
39 | Imported entities |
40 |
41 | Locales | 2
42 |
43 | Content Types | 25
44 |
```

```

45 | Tags | 7 |
46 |
47 | Editor Interfaces | 25 |
48 |
49 | Assets | 50 |
50 |
51 | Published Assets | 50 |
52 |
53 | Archived Assets | 0 |
54 |
55 | Entries | 332 |
56 |
57 | Published Entries | 332 |
58 |
59 | Archived Entries | 0 |
60 |
61 | Webhooks | 0 |
62 |
63 The import took 8 minutes (480s)
64
65 The import was successful.

```

After refreshing your Contentful page, you should see the imported Content Model and Content:

The screenshot shows the Contentful interface with the 'Content' tab selected. On the left, there's a sidebar with filters for 'All content', 'Recent', 'Scheduled', 'Private views', 'Status' (with 'Published' selected), and 'Content Type' (with 15 items listed). The main area is titled 'All' and shows a table of entries. The columns are 'Name', 'Content Type', 'Updated', 'Last updated by', and 'Status'. There are 15 entries listed, all of which are 'Published' and were updated 'an hour ago' by 'Vladimir Baumgartner'. The entry types include 'Category Navigation Component', 'Footer Navigation Component', 'FooterNavLink', 'SAPCustomerExperienceNavLink', 'FollowUsNavLink', 'PowerToolsCategoryNavLink', 'HandToolsNavLink', 'SafetyNavLink', 'HandToolLinksNavLink', 'FootwearLinksNavLink', 'Homepage', 'FAQTextParagraph', 'FAQMenuParagraph', and 'Breadcrumb CMS Component'.

## Marketplace App - Contentful Connector for SAP Commerce Cloud

In order to choose the products that we want displayed in the carousel component on the homepage, we need an official SAP Commerce Cloud connector which will make it easier to browse and select the products from our Commerce Cloud 2211 instance.

### Our Bestselling Products



Without installing the app, in the Best Selling Products carousel component, you should be seeing something similar to this:

Products

Type the value and hit enter

- ⋮ https://localhost:9002/occ/v2/powertools-spa/products/2116282 X
- ⋮ https://localhost:9002/occ/v2/powertools-spa/products/3887477 X
- ⋮ https://localhost:9002/occ/v2/powertools-spa/products/2221933 X
- ⋮ https://localhost:9002/occ/v2/powertools-spa/products/3793682 X
- ⋮ https://localhost:9002/occ/v2/powertools-spa/products/3756505 X
- ⋮ https://localhost:9002/occ/v2/powertools-spa/products/3774127 X
- ⋮ https://localhost:9002/occ/v2/powertools-spa/products/3755205 X

On your Contentful web page go to the **Apps > Marketplace**, then find and install the Contentful Connector for SAP Commerce Cloud app.

**App details**

### Contentful Connector for SAP Commerce Cloud

App • Developed by [Contentful](#)

[Install](#)

---

**About the SAP Commerce Cloud Connector app**

This SAP Commerce Cloud Connector app brings your products from SAP Commerce Cloud into Contentful. The app allows your content creators to select one or more products from Commerce Cloud and embed them into Contentful entries. These products can then be used like other Contentful content, empowering more scalable and effective ecommerce.

---

**Features**

- Connect Contentful to your SAP Commerce Cloud product catalogues
- Sort, search, and select one or more products and embed them directly into a field on Contentful entries
- Supports one or more base sites

---

**Requirements**

To use this app, you will need:

- SAP Commerce Cloud Version 1905 or higher

---

**Screenshots**

**Developer**

 Contentful

**Links**

[Store Listing](#) [Documentation](#) [SAP Commerce Cloud Connector Source Code](#)

**Support**

[Get support](#)

**Shows in**

Web app

Compose

**Categories**

Ecommerce

**Legal**

[End User License Agreement](#) [Privacy Policy](#)

**App ID**

GdOFzqcDrEbmN2teDQE2i

In order to install, you will have to:

- Provide API endpoint
  - For CCV2 Commerce instances, it can be found in the CCV2 panel under **Environments > [environment] > Public Endpoints > API**
  - For locally installed Commerce instance, you can use the following value:  
`https://localhost:9002`
- Select the proper fields to connect:  
**Product Carousel > Products**

After installing the app, you should be able to add new products to the Product Carousel component.

Products

	KD990KA 2116282	X
	Hammer Drill 6270 3887477	X
	Laboratory Bottle 2221933	X
	BT-HS 12 3793682	X
	PSR 14.4 LI-2 3756505	X
	PSR 10.8 LI 3774127	X
	PSR 960 3755205	X

SAP Select products

You should be seeing Select Products button and after clicking it, you should see all the products for the selected store available for selection.

Select products

Select	Code	Name	Image
<input type="checkbox"/>	3879435	Rotary Hammer BT-RH 920 E	
<input type="checkbox"/>	3879436	PBH 3000 RE	
<input type="checkbox"/>	3879437	Rotary Hammer RT-RH 24	
<input type="checkbox"/>	3879438	Rotary Hammer RT-RH 32	
<input type="checkbox"/>	3879440	Impact Drill BT-ID 550 E	

**Note:** The app is using the URLs to find the data in the Commerce Cloud instance. We have provided data that relies on the local installation of Cloud Commerce but if you're providing a CCV2 instance data, or it runs on a different port in your local system, the imported url links might not match your environment, thus you would have to reimport the products to get the app to show product name and picture in the Contentful UI. The imported links should still work in the component itself because the product id data is what is being parsed and that should match the powertools-spa initialization data.

## Marketplace App - Flex Fields ↗

Another marketplace app that will be useful to us is Flex Fields.

This app will enable us to conditionally show what page sections are available during page creation. You can read more about this in the Content Architecture documentation.

**App details**

 **FlexFields**  
App • Developed by [Thrillworks](#)

**About the FlexFields app**

FlexFields™ by Thrillworks makes content authoring easier, faster, and more streamlined by creating flexible content models that hide fields for content authors based on the criteria set out by content architects. FlexFields provides the flexibility to adjust or modify content models, eliminating the need to create entirely new models when variations arise. The app works by adding a new entry editor tab for FlexFields which displays the fields conditionally for the respective entry.

**Features**

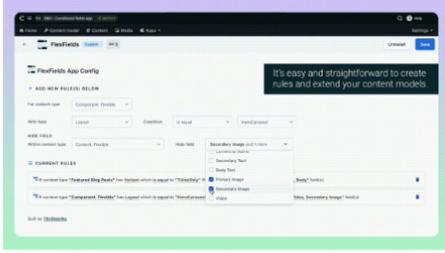
- Customizable content entry forms
- Flexibility to show and hide fields in the same content entry as well as referenced entries
- More efficient content authoring through a streamlined environment

**Requirements**

To use this app you will need:

- A Contentful account

**Screenshots**



The screenshot shows the 'FlexFields App Config' interface. It displays a configuration panel for 'CURRENT RULES'. One rule is visible: 'A content type "Thickend Big Post" has status which is equal to "Published"'. Another rule is partially visible: 'A content type "Component Post" has status which is equal to "Unpublished"'. The interface includes dropdown menus for 'Field Type' (Content, Text), 'Condition' (is equal), and 'Value' (Body Text, Secondary Text, Body Image, Secondary Image, Video). A note at the top right says 'It's easy and straightforward to create rules and extend your content models.'

**Developer**  
 [Thrillworks](#)

**Links**  
[FlexFields Source Code](#)

**Support**  
[Get support](#)

**Shows in**  
 Web app

**Categories**  
Developer Productivity, Editor Productivity

**Legal**  
[End User License Agreement](#)  
[Privacy Policy](#)

**App ID**  
7GbS2x3SdVh7D2hb1FFsS6 [Copy](#)

Before the installation (and, more importantly, configuration) of the Flex Fields, when you are creating or viewing a `CMSPage` entry, you will have all page sections available regardless of the template the page is based upon. This can get quite chaotic in time with more templates and different section naming schemes, this is why we will import FlexFields rules in order to show only the needed page sections.

**Note:** During the installation of the FlexFields app you might have to add a random rule in order for the app to let you proceed with the installation, you can delete the rule as soon as the installation is confirmed.

The screenshot shows the Contentful Content Editor interface. At the top, there is a dropdown menu labeled "Template (required)" containing the option "ContentPage1Template". Below this, the main area displays a hierarchical structure of sections:

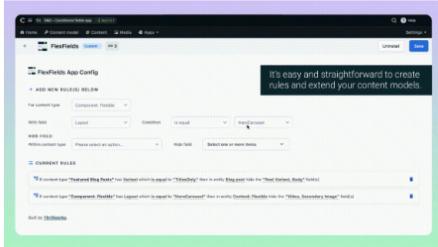
- Section1 | English**: Contains a dashed box for adding content with a "+ Add content" button.
- Section2A | English**: Contains a dashed box for adding content with a "+ Add content" button.
- Section2B | English**: Contains a dashed box for adding content with a "+ Add content" button.
- Section2C | English**: Contains a dashed box for adding content with a "+ Add content" button.
- Section3 | English**: Contains a dashed box for adding content with a "+ Add content" button.
- Section4 | English**: Contains a dashed box for adding content with a "+ Add content" button.
- Section5 | English**: Contains a dashed box for adding content with a "+ Add content" button.

In order to import the prepared rules for FlexFields, you should execute the script that we have provided with the Demo Sample Data, in the folder `flexfields-demo-data-import`.

Within the script itself you will have to configure the required properties:

1. Space ID (SPACEID)
2. Content Management API Token (CMATOKEN) (as described in  
[Getting started - Demo setup | Setting up the Contentful Access Tokens](#))
3. App Installation ID (APPINSTALLATION) should remain as provided, since every Marketplace App provides their own unique ID on the app installation info page

## Screenshots



Flex Fields App ID available at  
**Apps > Installed Apps > FlexFields > About**

After executing the script you will be seeing three rules configured for the FlexFields app.

A screenshot of the Contentful web app's 'FlexFields App Config' section after importing rules from a CSV file. It shows three rules listed under 'CURRENT RULES'. Rule 1: If content type "CMS Page" has 'Template' which is empty, then in entity 'CMS Page (Same Entry)' hide the "Section1, Section2A, Section2B, Section3, Section4, Section5" fields. Rule 2: If content type "CMS Page" has 'Template' which is equal to "LandingPage2Template", then in entity 'CMS Page (Same Entry)' hide the "Section1, Section2A, Section2B, Section3, Section4, Section5" fields. Rule 3: If content type "CMS Page" has 'Template' which is equal to "ContentPage1Template", then in entity 'CMS Page (Same Entry)' hide the "Section1, Section2A, Section2B, Section3, Section4, Section5" fields.

**Note:** After import of the FlexFields rules you might have to go into FlexFields Configuration and click Save and reload the Contentful web app in order to update the UI components properly and activate the rules.

After you import the rules, creating a CMSPage with template ContentPage1Template will now hide all the page sections that are not used by that specific template.

A screenshot of the Contentful web app's CMSPage editor. The 'Template (required)' dropdown is set to 'ContentPage1Template'. Below it, there are two sections: 'Section2A' and 'Section2B'. Each section is represented by a dashed rectangular area with a central 'Add content' button.

## Check content localization ☀

Demo data that we have provided has two locales: `en` for English and `de` for German language. After the import, you should be able to set up localized versions of content, but in order to see these options you will have to check the additional locale in the Web UI:

1. Open the `Homepage` `CMSPage` content
2. On the right side you should be seeing `Locales list` and the button `Edit locales list`

## Community

### Legal

[End User License Agreement](#)

[Privacy Policy](#)

### App ID

7GbS2x3SdVh7D2hbIFFsS6

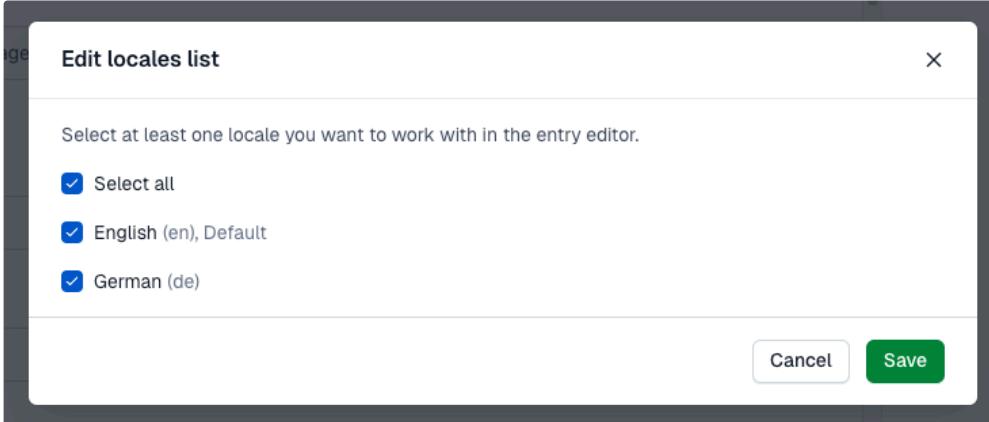


Locales

English (en) | Default

Edit locales list

3. After clicking the button you should be able to add the German language locale to the list



4. You should now see the input fields for all the fields that have a German localization

Title | English (required)

Homepage

8 characters Maximum 256 characters

Title | German (required)

Homepage

8 characters Maximum 256 characters

## Content preview platform for the Live Preview ↗

Once we set up the integration between Composable Storefront and Contentful you will be able to access Live Preview functionality directly from the Contentfull app, but you will have to set up the Content Preview settings.

Follow the official documentation [Set up content preview](#) and set up the live preview URL as follows:

- Content types: CMS Page
- Preview URL: `https://localhost:4200/powertools-spa/{locale}/USD/{entry.fields.previewUrl}?preview=true`

And then you should be seeing an `Open Live Preview` button for the platform you've created. This will become useful after the integration code is copied and installed in the next chapter.

Preview

Open Live Preview

Platform: Local Change

## Contentful and Composable Storefront Integration code ↗

In the repository we provided you will find the Composable Storefront Angular code that will be used to connect the Composable Storefront + Commerce Cloud to our Contentful instance.

The code that we provided should have the following structure:

```
1 contentful/
2   └── assets
3     └── translations
4       ├── de
5       │   └── index.ts
6       │   └── order.json
7       └── product.json
8     └── en
9       └── index.ts
10      └── order.json
11      └── product.json
12   └── translations.ts
13
14 └── cms
15   └── adapters
16     └── contentful-cms-component.adapter.ts
17     └── contentful-cms-component-adpater.spec.ts
18     └── contentful-cms-page-adapter.spec.ts
19     └── contentful-cms-page.adapter.ts
20   └── converters
21     └── components
22       └── contentful-cms-banner-component-normalizer.spec.ts
23       └── contentful-cms-banner-component-normalizer.ts
24       └── contentful-cms-navigation-component-normalizer.spec.ts
25       └── contentful-cms-navigation-component-normalizer.ts
26       └── contentful-cms-product-carousel-component-normalizer.spec.ts
27       └── contentful-cms-product-carousel-component-normalizer.ts
28       └── contentful-cms-component-normalizer.spec.ts
29       └── contentful-cms-component-normalizer.ts
30       └── contentful-cms-page-normalizer.spec.ts
31       └── contentful-cms-page-normalizer.ts
32   └── contentful.module.ts
33
34 └── core
35   └── contentful-core.module.ts
36   └── content-types.ts
37   └── decorators
38     └── contentful-component-decorator.spec.ts
39     └── contentful-component-decorator.ts
40     └── index.ts
41   └── helpers.ts
42   └── public_api.ts
43   └── services
44     └── contentful-angular.service.spec.ts
45     └── contentful-angular.service.ts
46     └── contentful-live-preview.service.spec.ts
47     └── contentful-live-preview.service.ts
48     └── contentful-restrictions.service.spec.ts
49     └── contentful-restrictions.service.ts
50     └── contentful.service.ts
51   └── type-guards.spec.ts
52   └── type-guards.ts
53
54 └── root
```

```

53   └── config
54     |   └── contentful-config.ts
55     |       └── default-contentful-config.ts
56   └── contentful-root.module.ts
57   └── feature-name.ts
58   └── index.ts
59   └── public_api.ts

```

In order to integrate Composable Storefront and Contentful you will need to:

1. Install the contentful libraries within your Composable Storefront Angular app

```
npm install contentful @contentful/live-preview
```

2. Copy the `contentful` folder from the provided codebase to the `src/app` folder within your Composable Storefront Angular app

3. Copy the `spartacus/features/contentful/contentful-cms-feature.module.ts` module file to the `src/app/spartacus/features/contentful/` folder within your Composable Storefront Angular app

4. Add the `ContentfulCMSFeatureModule` to the imports array of the `src/app/spartacus/features/spartacus-features.module.ts` :

```

1 import { NgModule } from '@angular/core';
2 // ... other imports
3 import { ContentfulCMSFeatureModule } from './features/contentful/contentful-cms-feature.module';
4
5 @NgModule({
6   declarations: [],
7   imports: [
8     AuthModule.forRoot(),
9     // ... other imports
10    ProductVariantsFeatureModule,
11    ProductImageZoomFeatureModule,
12    ContentfulCMSFeatureModule,
13  ],
14   providers: [provideFeatureToggles({
15     "showDeliveryOptionsTranslation": true,
16     //... other feature toggles
17   })]
18 })
19 export class SpartacusFeaturesModule {} 
20

```

5. Provide the correct configuration in the

`src/app/spartacus/spartacus-configuration.module.ts` file.

- Fill in `YOUR_SPACE_ID`, `YOUR_ACCESS_TOKEN`, `YOUR_PREVIEW_ACCESS_ID` and `YOUR_ENVIRONMENT`

**Note:** You should've created `accessToken` and `previewToken` (Content Delivery and Content Preview) during the Contentful setup, as described in

[Getting started - Demo setup | Setting up the Contentful Access Tokens](#)

- Add the slug map

In Composable Storefront, pages with dynamic paths (e.g., Order Details pages) include a page ID in the URL. To fetch the correct page from Contentful, a slug map is used to match the dynamic page paths to their corresponding slugs. You can extend the slug mapping configuration by adding custom mappings for your own dynamic pages. This allows you to correctly fetch pages from Contentful that follow a unique URL pattern.

```

1 provideConfig(<ContentfulConfig>{
2   contentful: {
3     spaceId: {{YOUR_SPACE_ID}},
4     accessToken: {{YOUR_ACCESS_TOKEN}},

```

```

5   previewAccessToken: {{YOUR_PREVIEW_ACCESS_ID}},
6   environment: {{YOUR_ENVIRONMENT}},
7   deliveryApiUrl: 'cdn.contentful.com',
8   previewApiUrl: 'preview.contentful.com',
9   slugMapping: {
10     '^organization/units/[a-zA-Z0-9]+$': 'organization/units',
11     '^organization/account-summary/details/[a-zA-Z0-9]+$': 'organization/account-summary/details',
12     '^my-account/saved-cart/[a-zA-Z0-9]+$': 'my-account/saved-cart',
13     '^my-account/order/[a-zA-Z0-9]+$': 'my-account/order',
14     '^my-account/order/cancel/confirmation/[a-zA-Z0-9]+$': 'my-account/order/cancel/confirmation',
15     '^my-account/order/cancel/[a-zA-Z0-9]+$': 'my-account/order/cancel',
16     '^my-account/support-ticket/[a-zA-Z0-9]+$': 'my-account/support-ticket',
17     '^my-account/unitLevelOrderDetails/[a-zA-Z0-9]+$': 'my-account/unitLevelOrderDetails',
18     '^my-account/order/return/confirmation/[a-zA-Z0-9]+$': 'my-account/order/return/confirmation',
19     '^my-account/order/return/[a-zA-Z0-9]+$': 'my-account/order/return',
20     '^my-account/return-request/[a-zA-Z0-9]+$': 'my-account/return-request',
21     '^my-account/quote/[a-zA-Z0-9]+$': 'my-account/quote',
22   },
23 },
24 }),

```

## 6. Run the app using

```
npm start
```

## 7. Verify that you still see the Powertools Homepage at:

```
https://localhost:4200/powertools-spa/en/USD/
```

## 8. Only the homepage and faq/help page should be accessible with this demo, other pages and shop functionality should follow soon as the demo gets expanded

## Verifying Demo Setup outcome ↗

In this section we will go through a couple of basic usecases to confirm that the demo setup has been completed successfully.

### Updating Content Model ↗

We will be updating the footer paragraph that contains copyright information by updating the year:



Copyright © 2024 SAP SE or an SAP affiliate company. All rights reserved.

Navigate to your Contentful space, browse the content and find the Notice Text Paragraph component.

Edit the component's Content field, change the year from 2024 to 2025 and hit Publish changes.

Return to the store page and hit refresh and the new copyright text should be there.

Copyright © 2025 SAP SE or an SAP affiliate company. All rights reserved.

## Using Live Preview ⚡

Navigate to your Contentful space, browse the content and find the Homepage CMSPage.

All

Content type Any Homepage

Created by me Status is Tags is one of Taxonomy

Sort by Name Content Type Updated Last updated by Status

Homepage CMS Page 7 minutes ago Vladimir Baumgartner Published

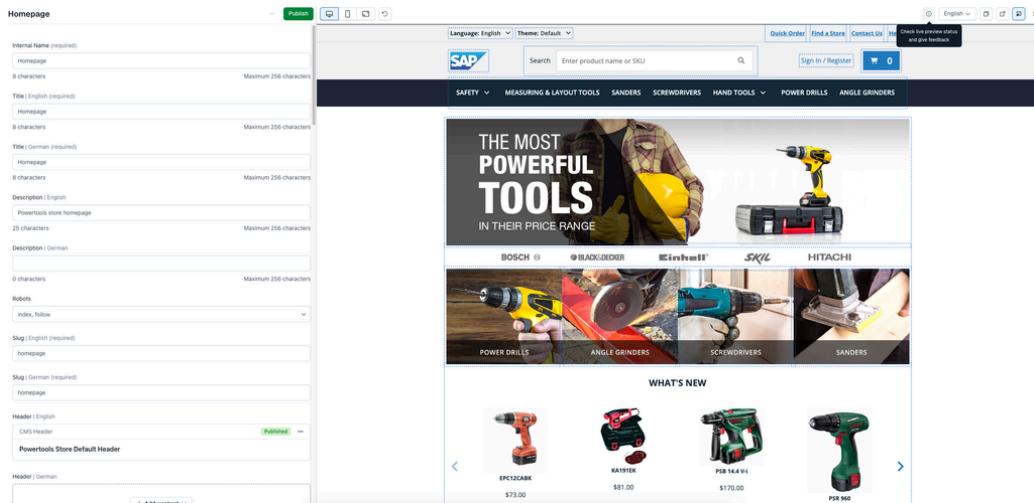
View Add entry Filter

Sort by Name Content Type Updated Last updated by Status

Homepage CMS Page 7 minutes ago Vladimir Baumgartner Published

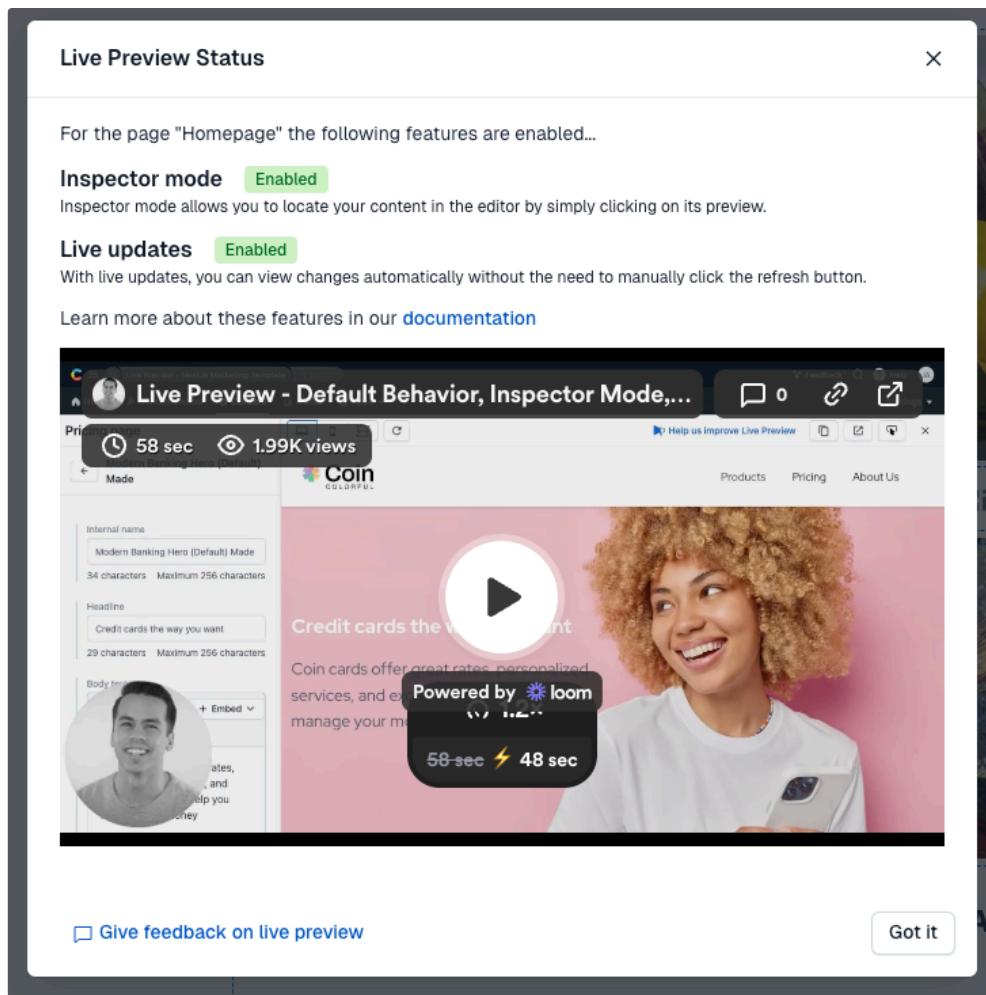
Previous Next

Open the page and click on the Live preview button. A new overlay structure will appear with the page data on the left and the preview window on the right.

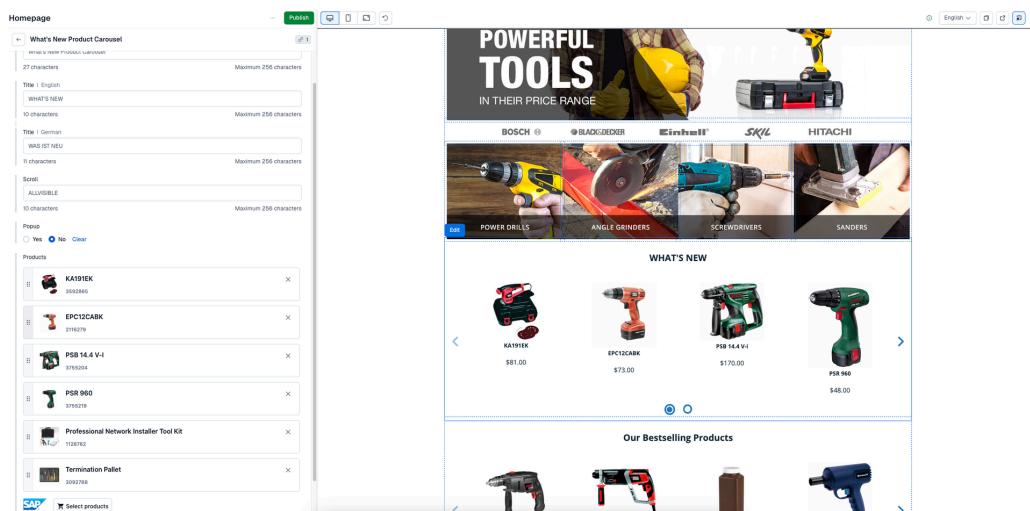


Click on the green i on the top right corner of the page and confirm that:

- Inspector Mode is on - this means that all of the components on the page will have an outline around them and you can click on the corresponding Edit button to quickly jump to the properties of that particular component
- Live Updated is on - this means that the changes will be visible as soon as you make them on the component



Hover one of the product carousels and click the Edit button that shows above them, the component properties will be shown on the left side of the Live Preview window. This is the Inspector Mode feature of the Live Preview.



In the products field, rearrange the positions of the products, swap the first and the second products by dragging one in front of the other and observe the instant change that occurs on the preview page. This is the Live updates functionality of the Live Preview.

Popup  
 Yes  No

Products

 EPC12CABK 2716279	X
 KA191EK 3592865	X
 PSB 14.4 V-I 3751204	X
 PSR 960 3755219	X

POWER DRILLS ANGLE GRINDERS SCREWDRIVERS SANDERS

WHAT'S NEW



EPC12CABK  
\$73.00



KA191EK  
\$81.00



PSB 14.4 V-I  
\$170.00



PSR 960  
\$48.00

< >

If you wish, you can click Publish and publish the changes that you've created so that they are visible when visiting the page directly at <https://localhost:4200/powertools-spa/en/USD/>