IMPORTING CUSTOM CONTENT TO SALESFORCE CMS

A guide to importing third-party content using JSON files, creating custom content types, and sharing them in the Community Cloud

Authors: Aleksander (Olek) Wojdyga

Categories: Salesforce

Keywords: SFDX, DevOps, CMS, Community Cloud, Content

Intended Audience: Salesforce architects, consultants, and senior developers that want to design and implement the custom CMS content types to fulfill clients' needs.

Author Bio: Aleksander (Olek) Wojdyga is an experienced software engineer who has worked on the Salesforce platform for the last 7 years. He holds several developer certificates and has worked with companies across the world to provide them with custom solutions. Olek has built applications in Sales Cloud, Service Cloud, and Community Cloud with Lightning Web Components, Lightning Aura Components, and VisualForce.

softserve

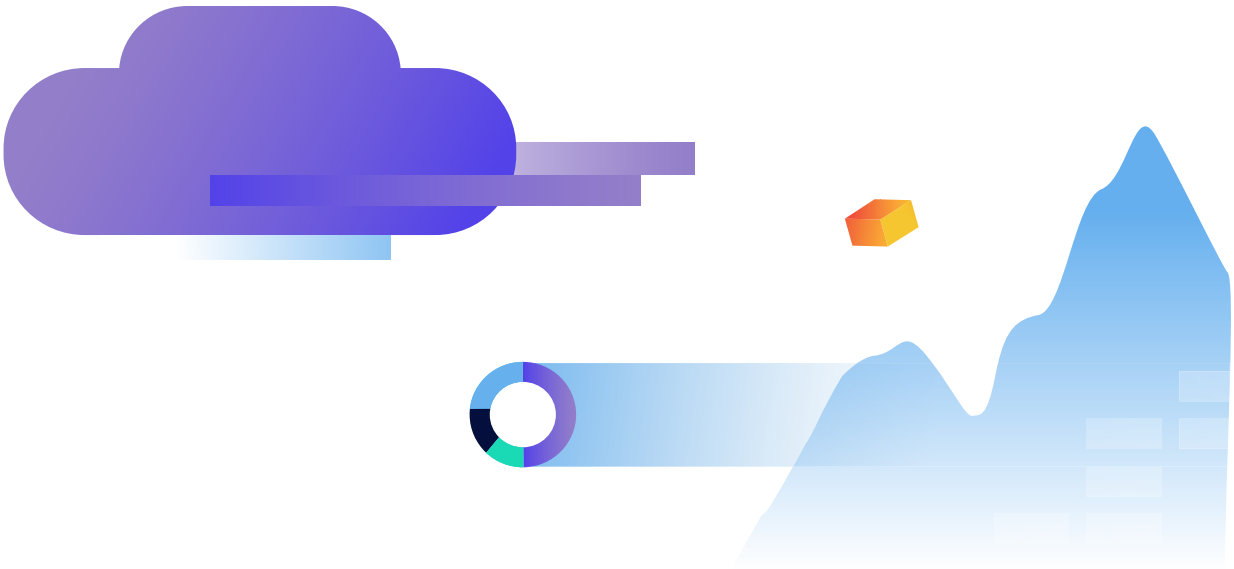
INTRODUCTION

Nearly every business creates content and needs a management system to organize it. Salesforce has built a content management system (CMS) that allows you to create, manage, and share content across multiple channels—regardless of whether you're a novice or an expert. It also connects your content delivery to your customer data through other Salesforce projects.

However, if your business has been around for years, you likely have content you've already created. The issue then becomes how to move your current content into Salesforce CMS.

Whether you're using a different CMS currently, you're looking to migrate your content to Salesforce CMS to phase out a legacy system, or you already have customers in Salesforce Community Cloud that will be consuming your content, you need to know how to import custom content types to Salesforce CMS.

Read on as we walk you through how to handle this type of import and manage the content in the Community Cloud.



IMPORTING CONTENT WITH SALESFORCE CMS

To import CMS data, you must have a Salesforce org to store it.

You will use the API names of the SObjects to accomplish this. All Salesforce records are natively represented as an sObject in Apex, meaning you can access Salesforce records and their fields directly from Apex. The API names of the SObjects for your content are ManagedVersion (key prefix 20Y) and ManagedContentVersion (key prefix is 5OU).

However, you can't query them directly using SOQL, and you can't use Apex to create the entity—only to access it. In order to create, edit, and publish CMS records, you can simply use the standard Salesforce UI in their CMS app.

This works fine if you're simply beginning with content management.

Chances are you already have content currently stored in a third-party system, which means you'll want to batch import it.

BATCH IMPORTING

The Salesforce CMS has a built-in import functionality, requiring a zip file for uploading content. While this may sound simple, you will need to prepare this zip file manually.

First, you will need to extract all the required data from your current CMS storage. Then, you will need to process it using whichever tools you prefer to create a JSON file before zipping it.

To process the JSON file format for mapping to Salesforce entities, you must first pick the CMS type where it belongs.

- Image = a standalone image like an infographic or an image embedded within another CMS item.
- Document = a link to a binary file stored in Salesforce's CMS.
- News = Can be a blog post or similar format and often contains rich text with embedded images.

EXAMPLE 1:
JSON file with CMS item

```
{
"content": [
{
"type":"cms _ image",
"urlName":"helpscreenshot",
"body":{
"title":"Salesforce Help CMS Screenshot",
"altText":"Salesforce Help CMS Screenshot",
"source":{
"ref":"images/help.png"
}
}
}]
}
```

Here is how we begin importing images into Salesforce CMS.

First, you can upload binary data for images from either the zip package or using a URL. If your image is publicly available—such as being stored on your CDN system—you'll find the URL option useful.

If you downloaded the binary image files to the local machine, you need to place the entire folder structure under _media folder.

This example places the image file in _media/images/help.png. The URL name is known as the "content slug," and it's a URL friendly way to link your content. It cannot contain spaces, but it may contain dashes, so you could use a title such as "help-screen-shot." Your title field is also used as the record's name field.

Here are the steps in this example:

1. Create the JSON file

2. Name it import.json

3. Save it in the same folder as _media.

4. Go to your CMS Workspace and use the Import Content button to upload the file.

EXAMPLE 2:
JSON response when uploading image


```
{
"dateTime": "2021-03-26 09:42:07",
"jobId": "0gU0H000000000B",
"details": [{
"contentKey": "MCTQQSQXQ4CJGRJKBECHTYF34XE4",
"managedContentVersionName": "Salesforce Help CMS
Screenshot",
"managedContentVersionId": "5OU0H000000GmkdWAC",
"managedContentId": "20Y0H000000GmeB",
"content": "{altText=Salesforce Help CMS Screenshot,
source={nodeType=MEDIA, mediaType=IMAGE,
isExternal=false, filename=help.png, mimeType=image/png,
url=/cms/media/5OU0H000000Gmkd/source?cb=05T0H0000
0n7o0x, ref=05T0H00000n7o0x}, title=Salesforce Help CMS
Screenshot}"
}]
}
```


Once the import process finishes, you will get an email with a link to the JSON response (see Example 2).

That is where you will have the contentKey field, which you can use to reference the image in other content items such as news.

CREATING CUSTOM CMS CONTENT TYPES

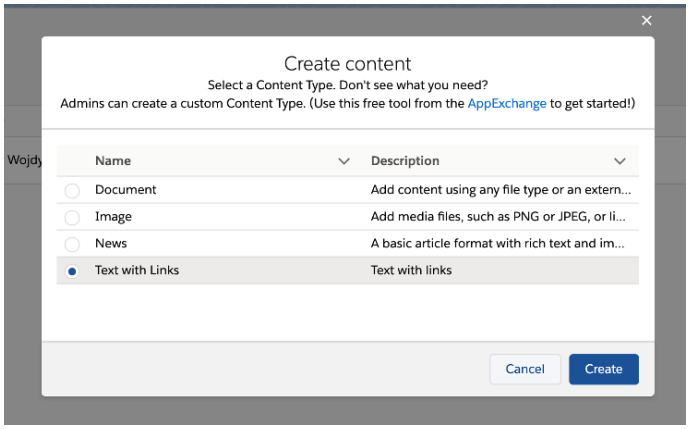
Now, what happens if your content items have a different structure? For example, maybe you want to have several images, but you don't want to display them as inline images within the text paragraphs. Or

perhaps you have a section with external links that you want to stand out from the other parts of the text.

This scenario is why you would need to use custom CMS content types.

You can create them manually using the metadata API. With this API, you can deploy the custom content types using Visual Studio Code or your favorite deployment tool.

Once you deploy your custom content, you can create latest content items using the Add Content button in your workspace. This newly added function is in the Create content dialog window, as seen in this image.



As you create a new ManagedContentType, you will use specific nodeTypes to differentiate them. For reference, here are the valid values for nodeType:

- TEXT - Simple text node (255 characters)
- MTEXT - Multi-line, non-formatted text node (2000 characters)
- RTE - Rich text node (65536 characters)
- IMG - Image node
- NAMEFIELD (80 characters)

EXAMPLE 3: Managed Content Type XML file

```
<?xml version="1.0" encoding="UTF-8"?>
<ManagedContentType
xmlns="http://soap.sforce.com/2006/04/metadata">
<description>Text with links</description>
<developerName>textWithLinks</developerName>
<managedContentNodeTypes>
<helpText>Title</helpText>
<isLocalizable>true</isLocalizable>
<isRequired>true</isRequired> <nodeLabel>Title</nodeLabel>
<nodeName>title</nodeName> <nodeType>NAMEFIELD</nodeType>
<placeholderText>Enter an SEO friendly title...</placeholderText> </
managedContentNodeTypes> <managedContentNodeTypes> <helpText>Main
image</helpText> <isLocalizable>false</isLocalizable>
<isRequired>false</isRequired> <nodeLabel>Primay Image</
nodeLabel> <nodeName>primaryImage</nodeName> <nodeType>IMG</
nodeType> </managedContentNodeTypes> <managedContentNodeTypes>
<helpText>Main content</helpText> <isLocalizable>true</
isLocalizable> <isRequired>true</isRequired> <nodeLabel>Main
Content</nodeLabel> <nodeName>body</nodeName> <nodeType>RTE</
nodeType> </managedContentNodeTypes> <managedContentNodeTypes>
<helpText>Section with links</helpText> <isLocalizable>true</
isLocalizable> <isRequired>true</isRequired> <nodeLabel>Links</
nodeLabel> <nodeName>links</nodeName> <nodeType>RTE</nodeType>
</managedContentNodeTypes> <masterLabel>Text with Links</
masterLabel> </ManagedContentType>
friendly title...</placeholderText> </managedContentNodeTypes>
<managedContentNodeTypes> <helpText>Main image</helpText>
<isLocalizable>false</isLocalizable> <isRequired>false</isRequired>
<nodeLabel>Primay Image</nodeLabel> <nodeName>primaryImage</
nodeName> <nodeType>IMG</nodeType> </managedContentNodeTypes>
<managedContentNodeTypes> <helpText>Main content</helpText>
<isLocalizable>true</isLocalizable> <isRequired>true</
isRequired> <nodeLabel>Main Content</nodeLabel> <nodeName>body</
nodeName> <nodeType>RTE</nodeType> </managedContentNodeTypes>
<managedContentNodeTypes> <helpText>Section with links</
helpText> <isLocalizable>true</isLocalizable> <isRequired>true</
isRequired> <nodeLabel>Links</nodeLabel> <nodeName>links</
nodeName> <nodeType>RTE</nodeType> </managedContentNodeTypes>
<masterLabel>Text with Links</masterLabel> </ManagedContentType>
```

In this example, you manually create the file for storing the content within the managed-ContentTypes folder. After that, you add the required extensions, so the file name reads as follows: textWithLinks.managedContentType-meta.xml.

The example XML file above has four nodes. The maximum number of nodes for a given node type is 15, so 15 RTE nodes at most.

You can create up to 20 custom content types in your org and can ask Salesforce to increase that number if needed.

Now you know how to create custom content types. But how do you upload the content items?

You can use the same import process as described above, using the JSON example below.

EXAMPLE 4:
JSON contents for custom type.

```
{
"content":[
{
"type":"textWithLinks",
"body":{
"title":"My Text with Links Item",
"body":"<p>This is the body of the item. We are
using the image already uploaded.</p>",
"primaryImage": {
"ref":"MCTQQSQXQ4CJGRJKBECHTYF34XE4"
},
"links":"<ul><li><a href=\"https://www.salesforce.
com\">Salesforce</a></li><li><a
href=\"https://www.softserveinc.com\">SoftServe</a></li></ul>"
}
}]
}
```

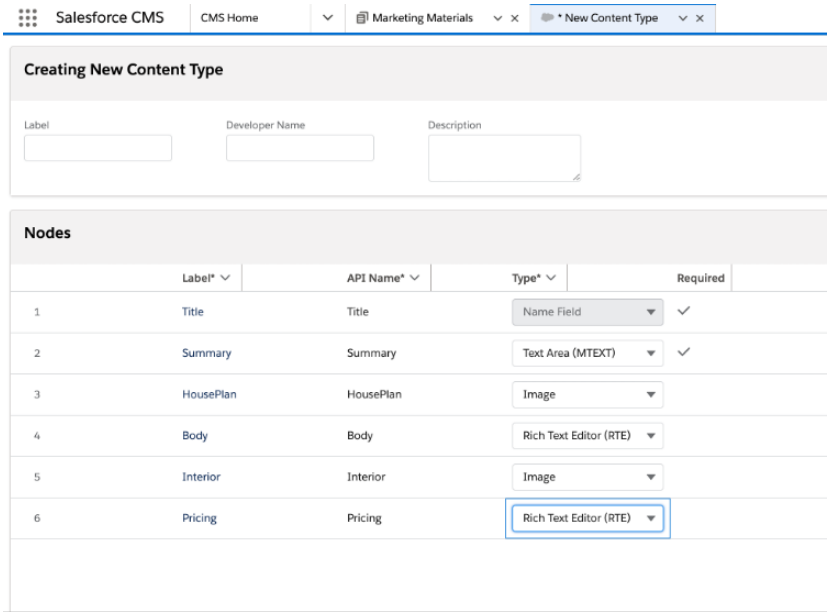
Note that the type field in JSON matches the developerName from the XML file.

Inside the body structure, you will use the nodeName as field names. The published image that we previously uploaded above can be used by referencing it with contentKey (see the JSON response from Example 2).

While manually creating XML files is easy, you can simplify the process even further with the CMS Content Type Creator app from Salesforce Labs.

Once you download the app from the Salesforce AppExchange and install it, you can immediately open it and create new content types.

To do this, use the Add Node button to add nodes. When selecting the node type, the values are the same as in the XML file. Below is an example of an added content type that includes two images.



CREATING THE CONTENT

You can now create the content using the added content types.

Navigate to the CMS Workspace and click on **Add Content** button.

Once the form opens, use the same fields and labels as you used above.

After you have created the custom content, you will be able to share the content in Community Cloud.

SHARING CUSTOM CONTENT TYPES IN COMMUNITY CLOUD

As stated at the beginning, there is always an option to use CMS Connect to share content from other content management systems.

This option implies that you're storing your content outside of Salesforce.

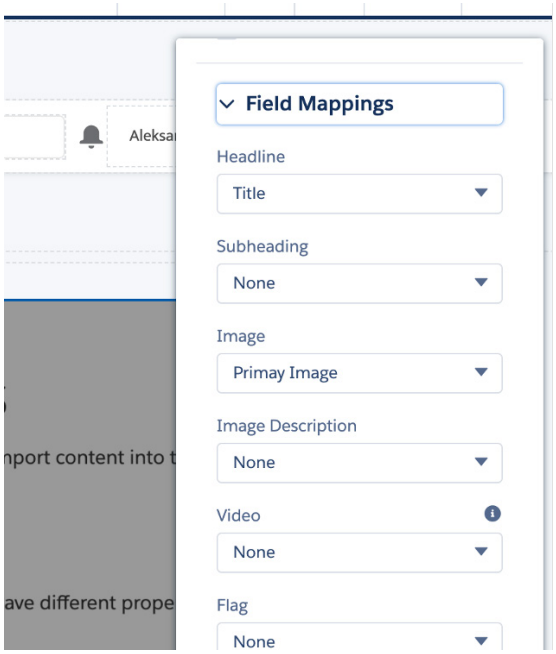
We have detailed how to create custom CMS content types because we want to store our CMS content within the Salesforce org. Having it stored locally means we can now share it in the Community Cloud.

As content items are records of some type—ManagedContent object—there are community pages and components that can natively display them.

Let's first create a collection for all the TextWithLinks published items and then create a new community page.

Once you open the Experience Builder, you'll need to navigate to the Content Management app and then to Collections. Here, you can create a new collection. It's important to note that you can only select one content type per collection. This will become obvious when you add a CMS Collection component to your community home page.

In the properties of the component, there is a section titled **Field Mappings.**

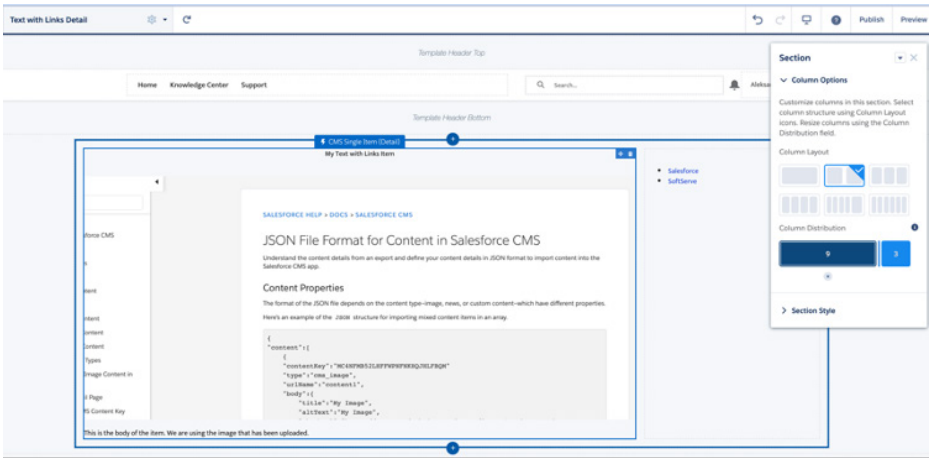


Here you can select the content item fields to display as a title or image. You're also able to have multiple images in an item, as the image above shows.

Now you should have the collection of CMS items displayed on the homepage.

The next step is to create a detail page for the TextWithLinks custom type. The goal is to display the links in a separate column, not below the main text of the blog post.

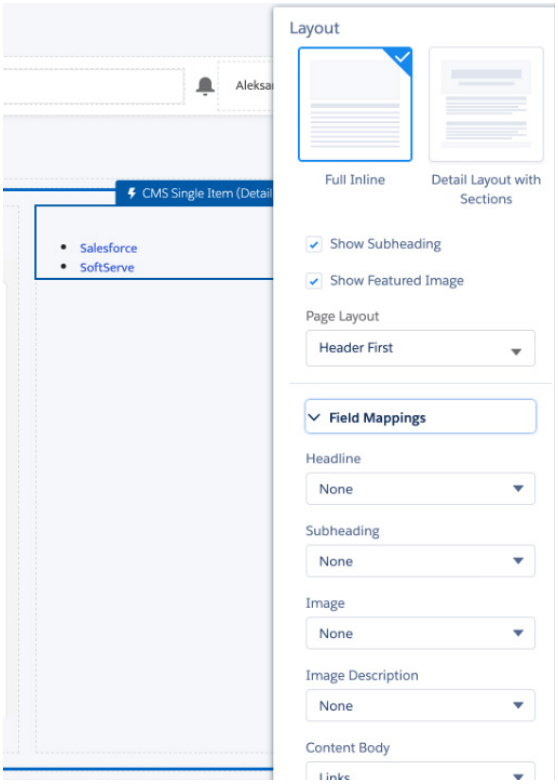
Open the **Pages** list and click on the **New Page** button. You will need to select CMS Content Page and then select your custom content type. Taking the full power of a 12-column layout, you can split the main section into two columns.



After this, you then add the CMS Single Item Detail components to both columns.

The wider column on the left will display images and text, while the one on the right will display links.

Use the **Field Mappings** section to select which nodes to display in the community from the content item.



CONCLUSION

This guide has covered importing the content item from a non-Salesforce CMS storage using JSON files, creating custom content types, and then displaying them in the Community Cloud. Salesforce makes this easy with their no-code and low-code solutions mentioned above—only the JSON files must be created manually. The examples throughout contain working source code so you can test and deploy them to your developer org. By enabling these Salesforce features, you can increase the reach of your content without extensive extra work. Sharing your content in the Community Cloud means reaching even more of your audience and increasing your overall business.

ABOUT US

SoftServe is a digital authority that advises and provides at the cutting-edge of technology. We reveal, transform, accelerate, and optimize the way enterprises and software companies do business. With expertise across healthcare, retail, energy, financial services, and more, we implement end-to-end solutions to deliver the innovation, quality, and speed that our clients' users expect.

SoftServe delivers open innovation, from generating compelling new ideas, to developing and implementing transformational products and services.

Our work and client experience is built on a foundation of empathetic, human-focused experience design that ensures continuity from concept to release.

We empower enterprises and software companies to (re)identify differentiation, accelerate solution development, and vigorously compete in today's digital economy-no matter where you are in your journey.

Visit our **website**, **blog**, **LinkedIn**, **Facebook**, and **Twitter** pages.

NORTH AMERICAN HQ

201 W 5th Street, Suite 1550
Austin, TX 78701 USA
+1 866 687 3588 (USA)
+1 647 948 7638 (Canada)

EUROPEAN HQ

14 New Street
London EC2M 4HE
United Kingdom

APAC HQ

6 Raffles Quay
#14-07
Singapore 048580
+65 31 656 887

info@softserveinc.com
www.softserveinc.com

softserve