About SAP Commerce Cloud

Generated on: 2024-12-03 11:12:39 GMT+0000

SAP Commerce Cloud in the Public Cloud | 2205

Public

Original content: https://help.sap.com/docs/SAP_COMMERCE_CLOUD_PUBLIC_CLOUD/20125f0eca6340dba918bda360e3cdfa?locale=en-US&state=PRODUCTION&version=v2205

Warning

This document has been generated from the SAP Help Portal and is an incomplete version of the official SAP product documentation. The information included in custom documentation may not reflect the arrangement of topics in the SAP Help Portal, and may be missing important aspects and/or correlations to other topics. For this reason, it is not for productive use.

For more information, please visit the https://help.sap.com/docs/disclaimer.

AddOn Concept

AddOns extend the functionality of the SAP Commerce Cloud Accelerator. They are a type of extension that allow you to add front-end files such JSP, HTML, CSS, and JavaScript files, and images without modifying the storefront front-end files directly.

⚠ **Caution**

This page refers to software that has been deprecated as part of the Accelerator UI and older OCC template extensions deprecation. For more information, see Deprecation of Accelerator UIs and Older OCC Template Extensions.

The overall extensibility of SAP Commerce Cloud Accelerator is improved in the following ways:

- You can generate your facade Data Transfer Objects (DTOs) and extend them, even if they are not defined in your own AddOn.
- You can plug populators into existing converters without having to redefine them.
- Some facades and services have been refactored so that they can be easily plugged in and their behavior can be easily modified.

For more information on extensions, see Extension Concept.

Overview

SAP Commerce Cloud Accelerator is built on an extensive framework that includes a template for storefront implementation. AddOns and extensions provide a well-defined infrastructure for third party developers to plug in their own functionality.

Using AddOns, you can extend the functionality of SAP Commerce Cloud Accelerator without editing the core code base. The core code base in this context means the Platform, and all additional extensions delivered with SAP Commerce Cloud Accelerator. An AddOn is a regular extension that may, or may not, provide additional front-end components to SAP Commerce Cloud Accelerator.

Creating Extensions

When creating extensions (including AddOns), it is a good idea to keep the logic and storefront-related features in separate extensions. That is why extensions are organized into the following groups:

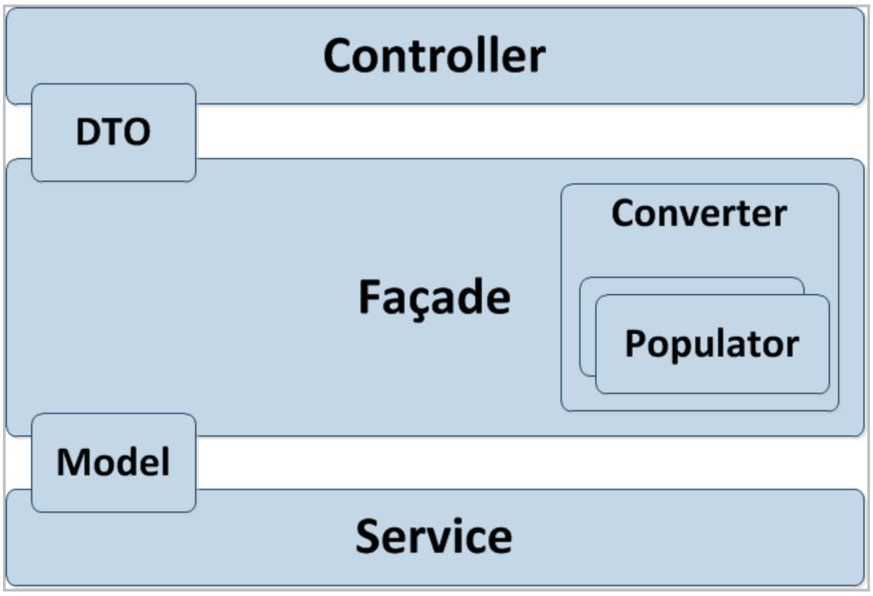
- Accelerator project extensions
- Accelerator cockpit extensions
- Accelerator sample and test data
- SAP Commerce Cloud storefront API extensions

AddOn Architecture

An AddOn is a regular SAP Commerce Cloud extension that is configured using the `extensioninfo.xml` file, located in the root folder of the AddOn. Additionally, the storefront extension that is extended by an AddOn has to reference this AddOn in the `extensioninfo.xml` file. The structure of this file is exactly the same as any regular extension. For more information about this file, see extensioninfo.xml.

Extensibility of the Framework

The diagram below illustrates the three-layer model of the extensible framework of SAP Commerce Cloud Accelerator:



At the bottom is the **Service** layer, which includes fine-grained business methods, such as the ones responsible for adding promotions to a cart, or for calculating the total value of the cart. These services expose the data model, which persists in the database.

On top of the Service layer, there are facades, which implement specific business use-cases, such as adding a product to a cart, placing an order, or searching for a product. The facades expose the **Data Transfer Objects (DTOs)**, which are completely independent from the underlying storage technology. There may be a one-to-one mapping of the model (such as store products), but there may also be a subset of the model, or aggregated models. The DTOs are not always stored in the database. An example of this is the Solr objects, which are stored in the Solr index.

The converters delegate to populators to convert the DTOs back and forth to models. For example, a product that has basic attributes, such as name, title, and description, can also have classification attributes. Therefore, you might have two populators, one for the basic attributes, and one for the classification attributes.

The facade layer, including the DTOs, represents the SAP Commerce Cloud OmniCommerce Connect. This is a business API, and the foundation for the web services.

On the top layer, the **Controllers** take the DTOs and expose them to the view. This is done using the Spring Model View Controller (MVC), which replaces all the facades, services, and controllers.

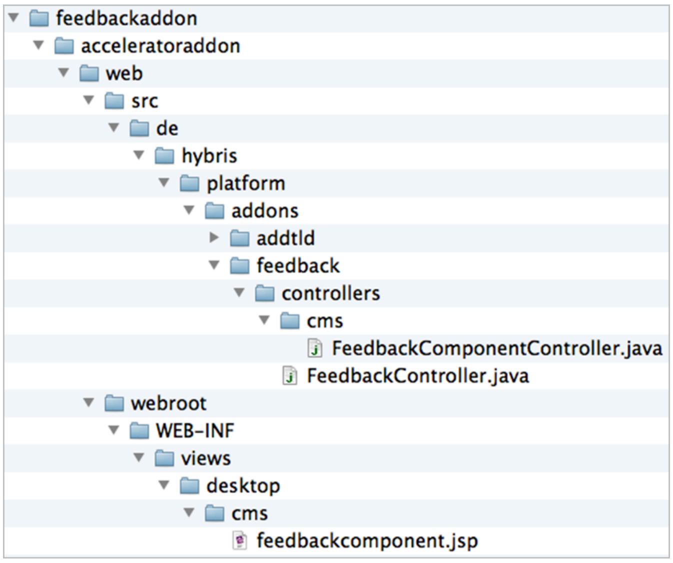
Extensibility of the Front-End

One example of extending the front-end is to enable customer feedback on your storefront. In this scenario, when a customer provides feedback, or rates a certain product, the system would automatically create a Customer Service Cockpit ticket for a customer service representative to review.

The AddOn Approach

The AddOn concept aims at extending the Accelerator storefront without touching its core code base. For example, you could implement customer feedback functionality on your storefront without editing the source code. Instead, you would plug in all the functionality from within your AddOn.

Continuing with this example, the AddOn would have an additional folder called `acceleratoraddon`, and the structure of the folders within it mirrors the structure of the folders containing the front-end components of a regular extension, as follows:

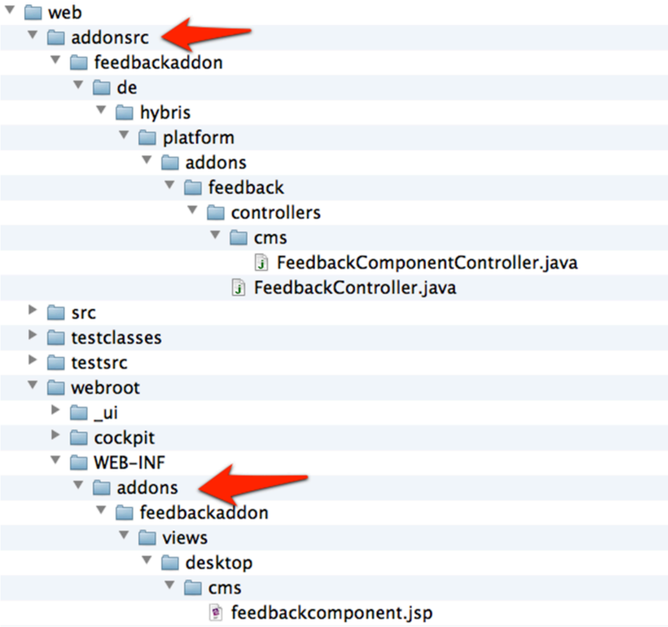


```
▼ feedbackaddon
  ▼ acceleratoraddon
    ▼ web
      ▼ src
        ▼ de
          ▼ hybris
            ▼ platform
              ▼ addons
                ▶ addtld
                ▼ feedback
                  ▼ controllers
                    ▼ cms
                        FeedbackComponentController.java
                      FeedbackController.java
      ▼ webroot
        ▼ WEB-INF
          ▼ views
            ▼ desktop
              ▼ cms
                  feedbackcomponent.jsp
```

Figure: `acceleratoraddon` folder structure

In this example, the `web` folder contains both the `src` and the `webroot` subdirectories. After putting all the components you want to add into the proper folders, you need to run the following command on your system: `ant build<Enter>`.

The system automatically copies the files to the target storefront extension during the build phase, and creates two additional folders containing the imported content:



```
▼ web
  ▼ addonsrc
    ▼ feedbackaddon
      ▼ de
        ▼ hybris
          ▼ platform
            ▼ addons
              ▼ feedback
                ▼ controllers
                  ▼ cms
                      FeedbackComponentController.java
                    FeedbackController.java
  ▶ src
  ▶ testclasses
  ▶ testsrc
  ▼ webroot
    ▶ _ui
    ▶ cockpit
    ▼ WEB-INF
      ▼ addons
        ▼ feedbackaddon
          ▼ views
            ▼ desktop
              ▼ cms
                  feedbackcomponent.jsp
```

After the build phase finishes, the target extension contains the following new folders:

- `web/addonsrc`, which contains the source code for each installed AddOn. This gets compiled automatically.
- `web/webroot/WEB-INF/addons`, which contains all the front-end components, such as images, JSP files, HTML files, and TAG files.

All the contents of these folders are processed automatically during the development phase. By default, this mechanism is disabled on the production systems so as not to impact the performance of the system. This is because the process continuously checks if the contents of the folders have changed.

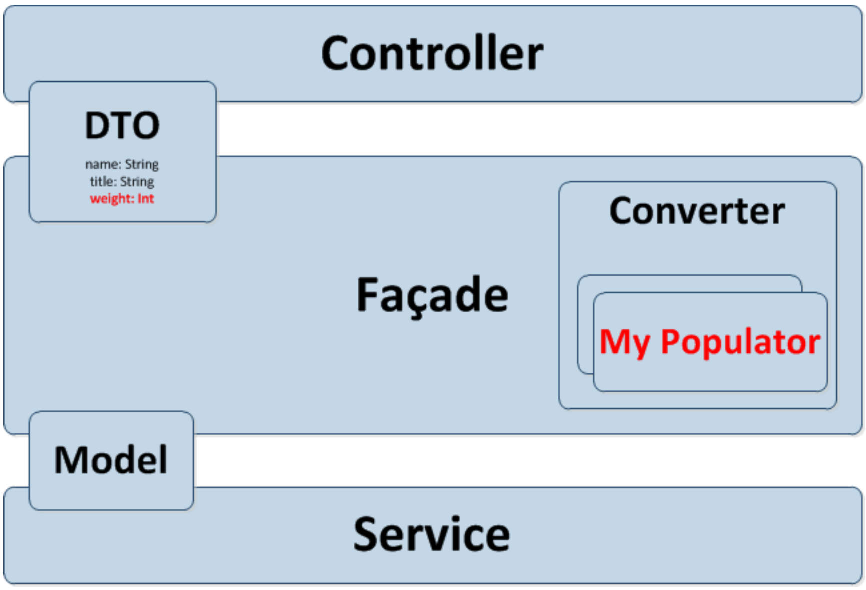
Benefits of the AddOn Approach

The benefits of the AddOn Concept approach are as follows:

- AddOn files are kept separate from the rest of the front-end files.
- When you upgrade the Accelerator, it will not overwrite your files.
- You can easily remove your AddOns without refactoring the code of your whole extension.

Improved Overall Extensibility

The AddOn approach makes it easier to modify the exposed data model. For example, if you want to add a new attribute to a product , you do not have to subclass the DTO. Instead, you can plug in a new populator, which is responsible for adding the new attribute:



Additionally, you can modify multiple extensions at once. Previously, due to Java restrictions, you could only modify the subclass once. Now, you can plug in new data from every extension using a new XML descriptor to generate the DTOs. Each extension can provide such a descriptor. All definitions are merged, so you can add attributes to existing DTOs and fill your attributes with pluggable populators. As a result, you can extend an existing API without needing to replace anything.

The following is an example of an XML descriptor:

```
<beans xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:noNamespaceSchemaLocation="beans.xsd">

        <bean class="de.hybris.platform.commercefacades.product.data.ProductData">
                <property name="weight" type="int" />
                <property name="instruction" type="String" />
                <property name="additionalImage"
                                                 type="de.hybris.platform.commercefacades.product.data.ImageData" />
        </bean>

        <enum class="de.hybris.platform.commercefacades.product.ProductOption">
                <value>DIMENSIONS</value>
                <value>MINIMAL</value>
        </enum>
</beans>
```

Similarly to the `item.xml` file, the definitions in the `beans.xml` file are merged automatically. In the example above, the `ProductData` file is not part of our example extension, but it is extended from the `commerceservices` extension. That means that during this process, you extended the API because you modified the data model. You do not need to replace the facade in this instance. All you need to do is plug in the populator from your extension, and it can be done multiple times.

AddOns Best Practices

We recommend that you put as much logic into your AddOns as possible. The advantage of this is that whenever you need to upgrade your Accelerator code base, you do not need to merge the changes manually. By having the logic put into the AddOns, you can build a library of self-contained, reusable components that you can use for your future projects.

Related Information

[About Extensions](#)
[Generating Beans and Enums](#)
[Customizing the B2C Accelerator](#)
[Storefront Web Application Deconstructed](#)
[commercefacades Extension](#)

Creating AddOns

Learn how to create an empty AddOn, which is the first step in creating your own custom AddOns.

AddOns allow you to extend or override storefront extensions without having to modify the extensions directly. In addition to simply using the preconfigured AddOns provided by SAP, you can create your own custom AddOns. To do so you create a blank AddOn, install it, and then modify its logic.

To create a blank AddOn.

1. In the command prompt window or Terminal, navigate to the *<HYBRIS_HOME>*/hybris/bin/platform folder.
2. Run the appropriate command:
   - setantenv.bat on Microsoft Windows systems.
   - . ./setantenv.sh on Unix-related systems (such as Linux or Mac OS).
3. Run the following command to create a new extension named myextension. The new extension is created under *<HYBRIS_HOME>*/hybris/bin/custom.

   ```
   ant extgen -Dinput.template=yaddon -Dinput.name=myextension -Dinput.package=com.myapp.myextension
   ```
4. Open the *<HYBRIS_HOME>*/hybris/config/localextensions.xml file and add your new extension:

   ```
   <extension name="myextension" />
   ```

Copying Files Between an AddOn and a Target Storefront

During the build process, the system copies files from the acceleratoraddon folder to the web folder of the target extension.

AddOns allow you to extend storefront extensions without having to modify the extensions directly. AddOn extensions are similar to regular extensions, but they allow you to add front-end related items (such as JSP files, CSS files, and images) without requiring you to build the platform. You can also add new Java code to the target extension, but that requires building the platform. The front-end items are stored in the acceleratoraddon folder of an AddOn.

The structure within the acceleratoraddon folder mirrors the folder structure within the web folder, which contains the front-end components of a regular extension.

Folder Structure

During the build process, the contents of the following folders are copied to the target extension:

Folder	Description	Target Folder
<addon_name>/acceleratoraddon/web/webroot/_ui	This folder contains static resources, such as CSS css files, JS files, and images.	*<yacceleratorstorefront>*/web/webroot/_ui/addons/*<addon_name>*
<addon_name>/acceleratoraddon/web/webroot/WEB-INF/*<folder/subfolder/resource>*	These folders contain TAG files, JSP files, TXT files, and so on.	*<yacceleratorstorefront>*/web/webroot/WEB-INF/*<folder/addons/addon_name/subfolder/resource>*
<addon_name>/acceleratoraddon/web/src	The subfolders of the src folder contains the web application Java source code.	*<yacceleratorstorefront>*/web/addonsrc/*<addon_name>*

Copying Files

During the build process, there is a build callback for copying the files after the yacceleratorstorefront is compiled. This occurs as follows:

1. The system scans for every extension that has a specific acceleratoraddon folder. Marker folders are defined as follows:

   ```
   ${extension-path}/acceleratoraddon/web/webroot/_ui
   ${extension-path}/acceleratoraddon/web/webroot/WEB-INF
   ```
2. For each AddOn that is found, the system identifies the target extension.
3. For each resource that is found in the acceleratoraddon folder, the system copies the contents to the target extension.

i Note

You can also copy certain AddOn files during runtime if the addonfilter is enabled. For more information about the addonfilter, see Storefront Web Application Deconstructed.

Deleting Copied Files

The system uses the ant sync task to keep the contents of the acceleratoraddon up to date with the contents of counterpart storefront extensions.

The AddOn directories in all storefront extensions that are defined in the localextensions.xml file are cleaned up with the ant clean command when building the Platform.

i Note

An AddOn can override any resource in the storefront extension. Every change done on behalf of the storefront extension from inside the acceleratoraddon folder is set up after the build.

Related Information

AddOn Concept

Installing an AddOn for a Specific Storefront

The `addoninstall` tool allows you to configure an AddOn for a storefront. It also adds the AddOn to the `extensioninfo.xml` file of the storefront, and generates the relevant `project.properties` file for the AddOn.

The addoninstall Tool

The following table lists the parameters of the `addoninstall` tool:

Parameter	Notes
`addonnames`	This parameter is mandatory. If more than one AddOn name is specified, the names must be separated by commas. If you do not include the `addonnames` parameter when you run the `addoninstall` command, you are asked to provide it.

`addonStorefront.`
`<storefrontTemplateName>`
	This parameter is used to indicate which storefronts you are installing the AddOn for. If more than one storefront is specified, the storefront names must be separated by commas. The *<storefrontTemplateName>* variable indicates the storefront template you wish to use for storefront generation. By default, the value is `yacceleratorstorefront`.

The `ant addoninstall` command has the following format:

```
ant addoninstall -Daddonnames="AddOnName1,AddOnName2" -DaddonStorefront.<storefrontTemplateName>="Storefront1,Storefront2"
```

The following is an example of the `ant addoninstall` command, where two AddOns (`NewAddOn1` and `NewAddOn2`) are installed for `B2CStorefront1` and `B2CStorefront2`, as well as for `B2BStorefront1` and `B2BStorefront2`:

```
ant addoninstall -Daddonnames="NewAddOn1,NewAddOn2" -DaddonStorefront.yacceleratorstorefront="B2CStorefront1,B2CStorefront2,B2BStorefront1,B2BStoref
```

The project.properties.template File

When you run the `ant addoninstall` command, the `project.properties` file for the AddOn is generated automatically. This is done using the `project.properties.template` file. Note that when you run the `ant addoninstall` command, the `project.properties.template` file overwrites the contents of the `project.properties` file of the AddOn.

When generating the `project.properties` file, the `addoninstall` tool executes the following steps:

- Deletes the old `project.properties` file.
- Generates the new `project.properties` file from the `project.properties.template` file.
- Adds a relative import file reference in the `yacceleratorstorefront/web/webroot/WEB-INF/_ui-src/responsive/lib/ybase-x.x.x/less/addons.less` file if such reference does not already exist and the AddOn's `_ui-src/reponsive/less` folder contains an *`<AddOn Name>`*`.less` file. For more information on LESS support for AddOns, see Responsive Website Build Process
- Replaces the storefront name with the following line in the `project.properties` file:

  ```
  <storefrontTemplateName>.additionalWebSpringConfigs.<AddOnName>=classpath:/<AddOnName>/web/spring/<AddOnName>-web-spring.xml
  ```

Configuring the AddOn Storefront Properties

When generating a new `project.properties` file, you can replace the `addonStorefront.`*`<storefrontTemplateName>`* property, or you can delete it, as described in the following sections.

Replacing the AddOn Storefront Properties

You can replace the default storefront properties by using the following parameter when you run the `ant addoninstall` command: `addonStorefront.` *`<storefrontTemplateName>`*`=Storefront1,Storefront2`.

For example, the default storefront properties for the B2C Accelerator appear in the `project.properties` file as follows:

```
yacceleratorstorefront.additionalWebSpringConfigs.<AddOnName>=classpath:/<AddOnName>/web/spring/<AddOnName>-web-spring.xml
```

If you specify the storefront parameters as `addonStorefront.yacceleratorstorefront=B2CStorefront1,B2CStorefront2`, the default line above is replaced in the `project.properties` file with the following:

```
B2CStorefront1.additionalWebSpringConfigs.<AddOnName>=classpath:/<AddOnName>/web/spring/<AddOnName>-web-spring.xml
B2CStorefront2.additionalWebSpringConfigs.<AddOnName>=classpath:/<AddOnName>/web/spring/<AddOnName>-web-spring.xml
```

Deleting the AddOn Storefront Properties

If you set the storefront parameter `addonStorefront.yacceleratorstorefront=blank`, the following property is deleted from the `project.properties` file:

```
yacceleratorstorefront.additionalWebSpringConfigs.<AddOnName>=classpath:/<AddOnName>/web/spring/<AddOnName>-web-spring.xml
```

Running the ant addoninstall Command

The following procedure describes how to run the `ant addoninstall` command.

1. Before you run the `ant addoninstall` command:
   - Ensure that the `addonsupport` extension is listed in `localextensions.xml` file, as follows:

```
...
<extension name="addonsupport"/>
...
```

- Also, ensure that the AddOn and storefront extension that you want to install are listed in `localextensions.xml` file.
- If you have all your AddOns prepared as a single ZIP file, perform the following steps:

  a. Unzip all the AddOns that you wish to install into a subfolder of the *<HYBRIS_BIN_DIR>* folder. In the following example, the subfolder is called `addoninstalldir`.

  b. Add the following line to your `localextensions.xml` file:

  ```
  ...
  <path autoload="true" dir="$\{HYBRIS_BIN_DIR\}/addoninstalldir"/>
  ...
  ```

- Finally, make sure you have properly defined the `project.properties.template` file for each AddOn that you wish to install.

2. Open the command prompt and navigate to the `hybris/bin/platform` directory.
3. Run the `addoninstall` command. The command has the following format:

```
ant addoninstall -Daddonnames="AddOnName1,AddOnName2" -DaddonStorefront.<storefrontTemplateName>="Storefront1,Storefront2"
```

4. When `addoninstall` has finished running successfully, rebuild the SAP system by running `ant clean all` from the `hybris/bin/platform` directory.

Uninstalling an AddOn for a Specific Storefront

The `addonuninstall` tool can be used to remove an AddOn from the storefront.

The addonuninstall Tool

The following table lists the parameters of the `addonuninstall` tool:

Parameter	Notes
`addonnames`	This parameter is mandatory. If more than one AddOn name is specified, the names must be separated by commas. If you do not include the `addonnames` parameter when you run the `addonuninstall` command, you are asked to provide it.
`addonStorefront.`*`<storefrontTemplateName>`*	This parameter is used to indicate which storefronts you are uninstalling the AddOn from. If more than one storefront is specified, the storefront names must be separated by commas. The *<storefrontTemplateName>* variable indicates the storefront template that was used for storefront generation. By default, the two possible values are `yacceleratorstorefront` and `yb2bacceleratorstorefront`.

The `ant addonuninstall` command has the following format:

```
ant addonuninstall -Daddonnames="AddOnName1,AddOnName2" -DaddonStorefront.<storefrontTemplateName>="Storefront1,Storefront2"
```

The following is an example of the `ant addonuninstall` command, where two AddOns (`NewAddOn1` and `NewAddOn2`) are uninstalled for `B2CStorefront1` and `B2CStorefront2`, as well as for `B2BStorefront1` and `B2BStorefront2`:

```
ant addonuninstall -Daddonnames="NewAddOn1,NewAddOn2" -DaddonStorefront.yacceleratorstorefront="B2CStorefront1,B2CStorefront2" -DaddonStorefront.yb2b
```

Running the ant addonuninstall Command

The following procedure describes how to run the `ant addonuninstall` command.

1. Before you run the `ant addonuninstall` command, make sure the `addonsupport` extension is listed in `localextensions.xml` file, as follows:

```
...
<extension name="addonsupport"/>
...
```

2. Open the command prompt and navigate to the `hybris/bin/platform` directory.
3. Run the `ant addonuninstall` command. The command has the following format:

```
ant addonuninstall -Daddonnames="AddOnName1,AddOnName2" -DaddonStorefront.<storefrontTemplateName>="Storefront1,Storefront2"
```

4. When `addonuninstall` has finished running successfully, rebuild SAP Commerce Cloud by running `ant clean all` from the `hybris/bin/platform` directory.

Customizing the Storefront Using an AddOn

You can customize a sample storefront and tailor it to your organization's requirements using AddOns.

Context

AddOns override the default configuration of the storefront so that it appears or behaves as desired, without the need to customize extensions.

Procedure

1. Create an AddOn using the procedure described in Creating AddOns.
2. Copy the storefront files that you wish to customize. The folder should have the same contents as the target storefront folder.

   For more information, see Copying Files Between an AddOn and a Target Storefront.
3. Change any of the following files, if applicable.
   - JSP Files
   - Impex Files
   - Tag Files

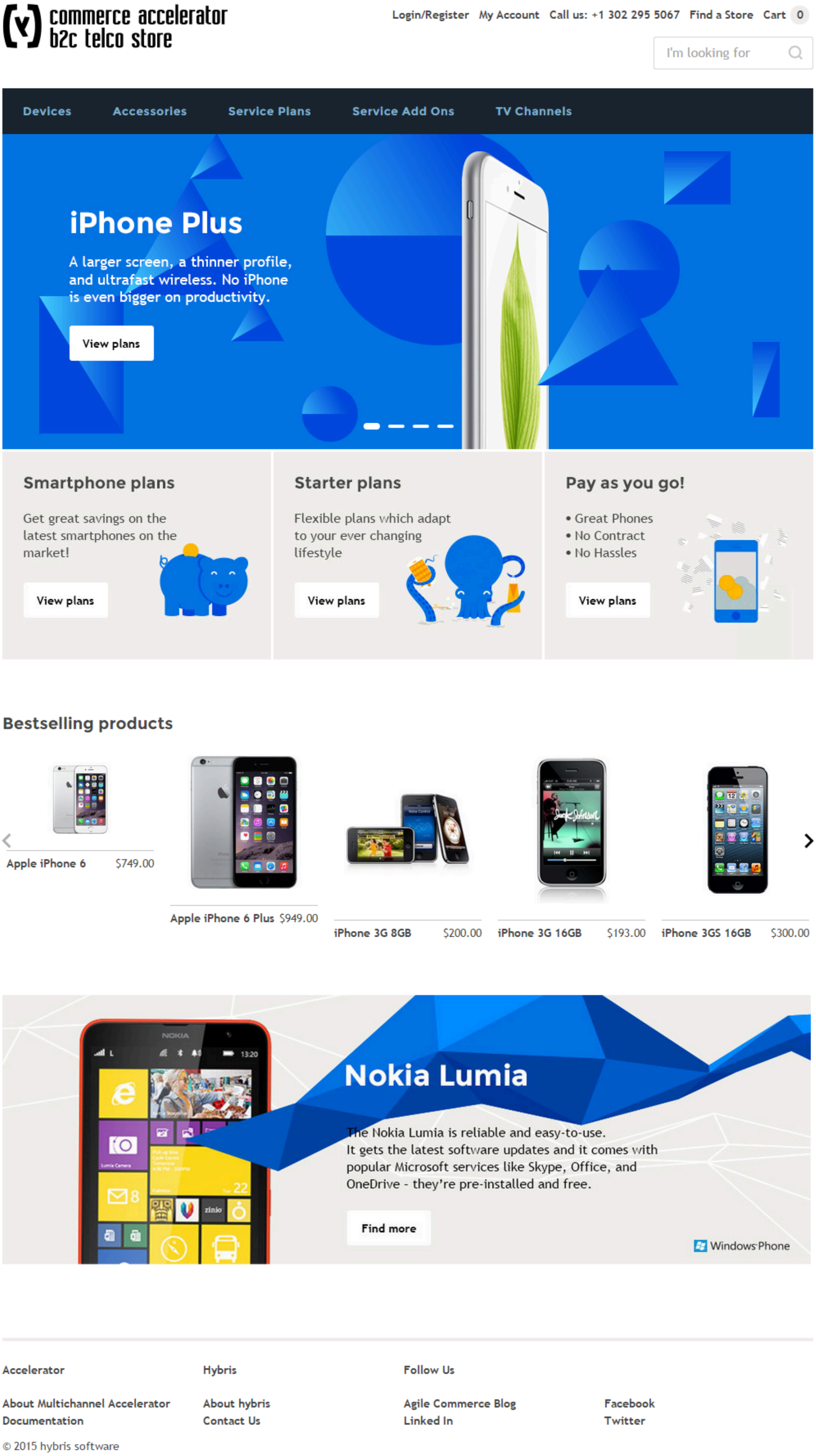
   You can copy an existing template and customize it accordingly.
4. Configure any of the following files, if applicable:
   - Spring
   - Controller
5. Install the AddOn as described in Installing an AddOn for a Specific Storefront.

Customizing the Home Page

The Home Page is completely configured using the CMS functions inherited from the B2C Accelerator. Modification of the Home Page is done via configuration of links, slots, and content via ImpEx, as for any other CMS managed page.

CMS Configuration

ContentSlotForTemplate	ContentSlot	Sample Data Slot Content CMS Components
SiteLogo-LandingPage2	SiteLogoSlot	SiteLogoComponent
HomepageLink-LandingPage2	HomepageNavLinkSlot	HomepageNavLink
NavigationBar-LandingPage2	NavigationBarSlot	NavBarComponent,breadcrumbComponent
MiniCart-LandingPage2	MiniCartSlot	MiniCart
Footer-LandingPage2	FooterSlot	FooterNavigationComponent
HeaderLinks-LandingPage2	HeaderLinksSlot	ContactInfo
SearchBox-LandingPage2	SearchBoxSlot	SearchBox,LangCurrencyComponent
TopHeaderSlot-LandingPage2	TopHeaderSlot	(No Components)
BottomHeaderSlot-LandingPage2	BottomHeaderSlot	(No Components)
PlaceholderContentSlot-LandingPage2	PlaceholderContentSlot	(No Components)



commerce accelerator b2c telco store

Login/Register My Account Call us: +1 302 295 5067 Find a Store Cart 0

I'm looking for

Devices Accessories Service Plans Service Add Ons TV Channels

iPhone Plus

A larger screen, a thinner profile, and ultrafast wireless. No iPhone is even bigger on productivity.

View plans

Smartphone plans

Get great savings on the latest smartphones on the market!

View plans

Starter plans

Flexible plans which adapt to your ever changing lifestyle

View plans

Pay as you go!

- Great Phones
- No Contract
- No Hassles

View plans

Bestselling products

Apple iPhone 6	$749.00
Apple iPhone 6 Plus	$949.00
iPhone 3G 8GB	$200.00
iPhone 3G 16GB	$193.00
iPhone 3GS 16GB	$300.00

Nokia Lumia

The Nokia Lumia is reliable and easy-to-use. It gets the latest software updates and it comes with popular Microsoft services like Skype, Office, and OneDrive - they're pre-installed and free.

Find more

Accelerator

About Multichannel Accelerator
Documentation

Hybris

About hybris
Contact Us

Follow Us

Agile Commerce Blog
Linked In

Facebook
Twitter

© 2015 hybris software

Customizing the Header and Footer

Customize the default B2C Accelerator header and footers supplied with the accelerator template to suit your needs.

Context

Variables:

Variable Name	Description	Value
<${addOnName}>	The name of the AddOn	`b2ctelcostorefront`
<${addOnPageRoot}>	The location where custom page fragments must go	`b2ctelcostorefront/acceleratoraddon/web/webroot/WEB-INF`

Available template files:

File Name	Source Path	Target Path
Layout	`/yacceleratorstorefront/web/webroot/WEB-INF/views/desktop/pages/layout/contentLayout1Page.jsp`	*<${addOnPageRoot}>*`/views/desktop/pages/layout/myNewLayoutPage.jsp`
Page Template	`/yacceleratorstorefront/web/webroot/WEB-INF/tags/desktop/template/page.tag`	*<${addOnPageRoot}>*`/tags/desktop/template/page.tag`
Header	`/yacceleratorstorefront/web/webroot/WEB-INF/tags/desktop/common/header/header.tag`	*<${addOnPageRoot}>*`/tags/desktop/common/header/header.tag`
Footer	`/yacceleratorstorefront/web/webroot/WEB-INF/tags/desktop/common/footer/footer.tag`	*<${addOnPageRoot}>*`/tags/desktop/common/footer/footer.tag`

Procedure

Create Custom JSP Files

1. Create a new page template based on the template file of the page type you need to customize.

   Copy an existing template. For example, if you want to customize the layout, copy `/yacceleratorstorefront/web/webroot/WEB-INF/views/desktop/pages/layout/contentLayout1Page.jsp` to `$`*<{addOnPageRoot}>*`/views/desktop/pages/layout/myNewLayoutPage.jsp`.

2. At the top of the layout page, change the reference for the template taglib.

   For example, find the following:

   ```
   <%@ taglib prefix="template"
                                                  tagdir="/WEB-INF/tags/desktop/template" %>
   ```

   Then change it to:

   ```
   <%@ taglib prefix="template"
                                                  tagdir="/WEB-INF/tags/addons/${AddOnName}/desktop/template"
                                                  %>
   ```

Change the Header

3. Change the reference for the **header** taglib.

   For example, find the following:

   ```
   <%@ taglib prefix="header" tagdir="/WEB-INF/tags/desktop/common/header"%>
   ```

   And change it to:

   ```
   <%@ taglib prefix="header" tagdir="/WEB-INF/tags/addons/${AddOnName}/desktop/common/header" %>
   ```

Change the Footer

4. Change the reference for the **footer** taglib.

   For example, find the following:

   ```
   <%@ taglib prefix="footer" tagdir="/WEB-INF/tags/desktop/common/footer" %>
   ```

   And change it to:

   ```
   <%@ taglib prefix="footer" tagdir="/WEB-INF/tags/addons/${AddOnName}/desktop/common/footer" %>
   ```

   You can now directly customize the footer and header in your custom AddOn.

Adding a New Page to the My Account Area

This document provides an example of how to add a new page to the My Account area. It shows you how to create a new page for listing subscriptions, integrate it into the navigation area, add custom controller logic, and define all the necessary CMS components in the ImpEx files.

Pages, Tags, and Templates

Variables

Variable Name	Description	Example Value
<${addOnName}>	The name of the AddOn	`subscriptionstorefront`
<${addOnPageRoot}>	The location where custom page fragments go	`subscriptionstorefront/acceleratoraddon/web/webroot/WEB-INF`
<${addOnWebSource}>	The location of Java code source	`subscriptionstorefront/acceleratoraddon/web/src/`
<${catalogName}>	Desired catalog name	`b2ctelcoContentCatalog`
<${myStorefrontProject}>	Your storefront project, derived from `yacceleratorstorefront`	`mystorefront`

Create the JSP Page Component

Create a new JSP page, in this case the `accountSubscriptionsPage.jsp` that will be used as a JSP include component on the actual page. The JSP page is located at *`<${addOnPageRoot}>`*`/views/desktop/pages/account/accountSubscriptionsPage.jsp` and will be copied (when the AddOn has been installed) to the *`<${myStorefrontProject}>`*`/web/webroot/WEB-INF/views/addons/${addOnName}/desktop/pages/account/accountSubscriptionsPage.jsp` location.

```
<%@ page trimDirectiveWhitespaces="true" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@ taglib prefix="template" tagdir="/WEB-INF/tags/desktop/template" %>
<%@ taglib prefix="theme" tagdir="/WEB-INF/tags/shared/theme" %>
<%@ taglib prefix="nav" tagdir="/WEB-INF/tags/desktop/nav" %>
<%@ taglib prefix="spring" uri="http://www.springframework.org/tags" %>
<%@ taglib prefix="cms" uri="http://hybris.com/tld/cmstags"%>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>
<%@ taglib prefix="fn" uri="http://java.sun.com/jsp/jstl/functions" %>
<%@ taglib prefix="common" tagdir="/WEB-INF/tags/desktop/common" %>
<%@ taglib prefix="ycommerce" uri="http://hybris.com/tld/ycommercetags" %>
<%@ taglib prefix="breadcrumb" tagdir="/WEB-INF/tags/desktop/nav/breadcrumb" %>
<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form" %>

        <div id="globalMessages">
                <common:globalMessages/>
        </div>
    <div class="headline"><spring:theme code="text.account.subscriptions" text="Subscriptions"/></div>
        <c:if test="${not empty subscriptions}">
        <div class="description"><spring:theme code="text.account.subscriptions.manageSubscriptions" text="Manage your subscriptions"/></div>
        <table class="subscriptionsTable">
            <thead>
                <tr>
                    <th><spring:theme code="text.account.subscriptions.productName" text="Product Name"/></th>
                    <th><spring:theme code="text.account.subscriptions.startDate" text="Start Date"/></th>
                    <th><spring:theme code="text.account.subscriptions.endDate" text="End Date"/></th>
                    <th class="subscriptionsTableState"><spring:theme code="text.account.subscriptions.status" text="Status"/></th>
                    <th class="subscriptionsTablePayment"><spring:theme code="text.account.subscriptions.paymentDetails" text="Payment Details"/></t
                    <th class="subscriptionsTableAction"><spring:theme code="text.account.orderHistory.actions" text="Actions"/></th>
                </tr>
            </thead>
            <tbody>
                                <c:forEach items="${subscriptions}" var="subscription">
                    <c:url value="/my-account/subscription/${subscription.id}" var="myAccountSubscriptionDetailsUrl"/>
                    <c:url value="${subscription.productUrl}" var="productDetailsUrl"/>
                    <tr>
                        <td>
                            <ycommerce:testId code="subscriptions_productName_link">
                                <a href="${productDetailsUrl}">${subscription.name}</a>
                            </ycommerce:testId>
                        </td>
                        <td>
                            <ycommerce:testId code="subscriptions_startDate_label">
                                <fmt:formatDate value="${subscription.startDate}" dateStyle="long" timeStyle="short" type="date"/>
                            </ycommerce:testId>
                        </td>
                        <td>
                            <ycommerce:testId code="subscriptions_endDate_label">
                                <fmt:formatDate value="${subscription.endDate}" dateStyle="long" timeStyle="short" type="date"/>
                            </ycommerce:testId>
                        </td>
                        <td>
                            <ycommerce:testId code="subscriptions_status_label">
                                ${subscription.subscriptionStatus}
                            </ycommerce:testId>
                        </td>
                        <td>
                            <ycommerce:testId code="subscriptions_status_label">
                                <c:if test= "${not empty paymentInfoMap[subscription.paymentMethodId]}">
                                    <c:set value="${paymentInfoMap[subscription.paymentMethodId]}" var="paymentMethod"></c:set>
                                    <c:set value="${fn:length(paymentMethod.expiryMonth) lt 2 ? '0'.concat(paymentMethod.expiryMonth) : paymentMetho
                                    <spring:theme code="text.account.subscriptions.payment.details" arguments="${fn:substring(paymentMethod.cardType
                                </c:if>
                            </ycommerce:testId>
                        </td>
                        <td>
                            <a href="${myAccountSubscriptionDetailsUrl}" class="function_btn">
                                <spring:theme code="text.manage" text="Manage"/>
                            </a>
                            <c:if test="${fn:toUpperCase(subscription.subscriptionStatus) ne 'PAUSED' and fn:toUpperCase(subscription.subscriptionSt
                            and !empty subscriptionFacade.getUpsellingOptionsForSubscription(subscription.productCode)}">
                                <c:url value="/my-account/subscription/upgrades-comparison?subscriptionId=${subscription.id}" var="upgradeSubscripti
                                <a type="button" href="${upgradeSubscriptionComparisionUrl}" class="button secondary">
                                    <spring:theme code="text.account.subscription.upgradeSubscription" text="Upgrade Subscription"/>
                                </a>
                            </c:if>
                            <c:if test="${fn:toUpperCase(subscription.subscriptionStatus) eq 'PAUSED' and fn:toUpperCase(subscription.subscriptionSt
                                <c:url value="/my-account/subscription/change-state" var="changeSubscriptionStateUrl" />
                                <form:form class="resume_subscription_form" id="resumeSubscriptionForm" action="${changeSubscriptionStateUrl}" metho
                                    <button type="submit" class="positive">
                                        <spring:theme code="text.account.subscription.resumeSubscription" text="Resume Subscription"/>
                                    </button>
                                    <input type="hidden" name="newState" value="ACTIVE"/>
                                    <input type="hidden" name="subscriptionId" value="${subscription.id}"/>
                                </form:form>
                            </c:if>
                        </td>
                    </tr>
                </c:forEach>
            </tbody>
        </table>
    </c:if>
    <c:if test="${empty subscriptions}">
        <p class="emptyMessage"><spring:theme code="text.account.subscriptions.noSubscriptions" text="You have no subscriptions"/></p>
    </c:if>
```

The second one for Subscription Details page (*<${addOnPageRoot}>*`/WEB-INF/views/desktop/pages/telco/account/accountSubscriptionPage.jsp`):

```
<%@ page trimDirectiveWhitespaces="true" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@ taglib prefix="template" tagdir="/WEB-INF/tags/desktop/template" %>
<%@ taglib prefix="theme" tagdir="/WEB-INF/tags/shared/theme" %>
<%@ taglib prefix="nav" tagdir="/WEB-INF/tags/desktop/nav" %>
<%@ taglib prefix="spring" uri="http://www.springframework.org/tags" %>
<%@ taglib prefix="cms" uri="http://hybris.com/tld/cmstags" %>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>
<%@ taglib prefix="fn" uri="http://java.sun.com/jsp/jstl/functions" %>
<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form" %>
<%@ taglib prefix="common" tagdir="/WEB-INF/tags/desktop/common" %>
<%@ taglib prefix="ycommerce" uri="http://hybris.com/tld/ycommercetags" %>
<%@ taglib prefix="breadcrumb" tagdir="/WEB-INF/tags/desktop/nav/breadcrumb" %>
<%@ taglib prefix="product" tagdir="/WEB-INF/tags/addons/subscriptionstorefront/desktop/product" %>
                <c:url value="${subscriptionData.productUrl}" var="productDetailsUrl"/>
                <c:url value="/my-account/order/${subscriptionData.orderNumber}" var="orderDetailsUrl"/>

        <ycommerce:testId code="subscription_productName_link">
            <a href="${productDetailsUrl}">
                <div class="headline">${subscriptionData.name}</div>
            </a>
        </ycommerce:testId>
        <c:if test="${fn:toUpperCase(subscriptionData.subscriptionStatus) ne 'CANCELLED' and upgradable}">
            <c:url value="/my-account/subscription/upgrades-comparison?subscriptionId=${subscriptionData.id}" var="upgradeSubscriptionComparisionUrl
            <button type="button" onclick="window.location='${upgradeSubscriptionComparisionUrl}'" class="positive right p_action_btn"><spring:theme
        </c:if>
        <div class="description">
            <ycommerce:testId code="subscription_description_label">
                ${subscriptionData.description}
            </ycommerce:testId>
        </div>
        <div class="manageSubscription">
            <h3><spring:theme code="text.account.subscription.detail" text="Subscription detail"/></h3>
            <table id="subscription_details" class="manageSubscriptionTable">
                <tr>
                    <td>
                        <span><spring:theme code="text.account.subscription.billingFrequency" text="Billing Frequency"/>:</span>
                    </td>
                    <td>
                        <ycommerce:testId code="subscription_billingFrequency_label">
                            ${subscriptionData.billingFrequency}
                        </ycommerce:testId>
                    </td>
                </tr>
                <tr>
                    <td>
                        <span><spring:theme code="product.list.viewplans.price" text="Price"/>:</span>
                    </td>
                    <td>
                        <span class="bold_text"><product:subscriptionPricesLister subscriptionData="${subscriptionProductData}"/></span>
                    </td>
                </tr>
                <tr>
                    <td>
                        <span><spring:theme code="text.account.subscription.contractDuration" text="Contract Duration"/>:</span>
                    </td>
                    <td>
                        <ycommerce:testId code="subscription_contractDuration_label">
                            ${subscriptionData.contractDuration}&nbsp;${subscriptionData.contractFrequency}
                        </ycommerce:testId>
                    </td>
                </tr>
                <tr>
                    <td>
                        <span><spring:theme code="text.account.subscription.Cancellable" text="Cancellable"/>:</span>
                    </td>
                    <td>
                        <ycommerce:testId code="subscription_status_label">
                            <c:choose>
                                <c:when test="${subscriptionData.cancellable}">
                                    <spring:theme code="text.account.subscription.cancellable.yes" text="yes"/>
                                </c:when>
                                <c:otherwise>
                                    <spring:theme code="text.account.subscription.cancellable.no" text="no"/>
                                </c:otherwise>
                            </c:choose>
                        </ycommerce:testId>
                    </td>
                </tr>
                <c:catch>
                    <c:if test="${subscriptionProductData.entitlements ne null}">
                <tr>
                    <td>
                        <span><spring:theme code="product.list.viewplans.entitlements" text="Included"/>:</span>
                    </td>
                    <td>
                        <span><product:entitlementLister subscriptionData="${subscriptionProductData}"/></span>
                    </td>
                </tr>
                <tr>
                    <td>
                        <span><spring:theme code="product.list.viewplans.usage.charges" text="Usage Charges"/>:</span>
                    </td>
                    <td>
                        <span><product:usageChargesLister subscriptionData="${subscriptionProductData}"/></span>
                    </td>
                </tr>
                    </c:if>
                </c:catch>
            </table>
```

```
</div>
<div class="manageSubscription">
    <h3><spring:theme code="text.account.subscription.manageSubscription" text="Manage this subscription"/></h3>
    <table id="subscription_manage" class="manageSubscriptionTable">
        <c:if test="${fn:toUpperCase(subscriptionData.subscriptionStatus) ne 'CANCELLED' }">
        <tr>
            <td>
                <span>
                    <c:url value="/my-account/subscription/change-state" var="changeSubscriptionStateUrl" />
                    <c:if test="${fn:toUpperCase(subscriptionData.subscriptionStatus) eq 'PAUSED'}">
                        <form:form class="resume_subscription_form" id="resumeSubscriptionForm" action="${changeSubscriptionStateUrl}" metho
                            <button type="submit" class="r_function_btn negative play">
                                <spring:theme code="text.account.subscription.resumeSubscription" text="Resume Subscription"/>
                            </button>
                            <input type="hidden" name="newState" value="ACTIVE"/>
                            <input type="hidden" name="subscriptionId" value="${subscriptionData.id}"/>
                        </form:form>
                    </c:if>
                    <c:if test="${fn:toUpperCase(subscriptionData.subscriptionStatus) eq 'ACTIVE'}">
                    <form:form class="pause_subscription_form" id="pauseSubscriptionForm" action="${changeSubscriptionStateUrl}" method="pos
                        <button type="submit" class="r_function_btn negative pause">
                            <spring:theme code="text.account.subscription.pauseSubscription" text="Pause Subscription"/>
                        </button>
                        <input type="hidden" name="newState" value="PAUSED"/>
                        <input type="hidden" name="subscriptionId" value="${subscriptionData.id}"/>
                    </form:form>
                    </c:if>
                </span>
            </td>
            <td>
                <c:if test="${subscriptionData.cancellable}">
                    <div>
                        <c:url value="/my-account/subscription/cancel/${subscriptionData.id}" var="cancelUrl"/>
                        <button type="button" data-url="${cancelUrl}" class="function_btn js-cancel-subscription negative cancel"><spring:th
                        <div class="hidden">
                            <div id="cancel-subscription-confirm" >
                                <div class="small_popup_headline"><spring:theme code="text.account.subscription.confirm.cancellation" text="
                                <div class="small_popup_content">
                                    <p><spring:theme code="text.account.subscription.cancellation.message" text="Are you sure you would like
                                    <div class="actions">
                                        <a href="#" class="r_action_btn cancel"><spring:theme code="text.account.subscription.cancellable.no
                                        <a href="${cancelUrl}" class="confirm positive button"><spring:theme code="text.account.subscription
                                    </div>
                                </div>
                            </div>
                        </div>
                    </div>
                </c:if>
            </td>
        </tr>
        </c:if>
        <tr>
            <td>
                <span><spring:theme code="text.account.subscription.status" text="Status"/>:</span>
            </td>
            <td>
                <ycommerce:testId code="subscription_status_label">
                    ${subscriptionData.subscriptionStatus}
                </ycommerce:testId>
            </td>
        </tr>
        <c:if test="${fn:toUpperCase(subscriptionData.subscriptionStatus) ne 'CANCELLED' }">
        <tr>
            <td colspan="2">
                <span>
                    <c:url value="/my-account/subscription/set-autorenewal-status" var="setAutorenewStatusUrl" />
                    <c:if test="${!subscriptionData.renewalType}">
                        <form:form class="activate_autorenewal_form" id="activateAutorenewalForm" action="${setAutorenewStatusUrl}" method="
                            <button type="submit" class="r_function_btn negative play">
                                <spring:theme code="text.account.subscription.activateAutorenew" text="Activate Auto-Renew"/>
                            </button>
                            <input type="hidden" name="autorenew" value="true"/>
                            <input type="hidden" name="subscriptionId" value="${subscriptionData.id}"/>
                        </form:form>
                    </c:if>
                    <c:if test="${subscriptionData.renewalType}">
                        <form:form class="deactivate_autorenewal_form" id="deactivateAutorenewalForm" action="${setAutorenewStatusUrl}" meth
                            <button type="submit" class="r_function_btn negative stop">
                                <spring:theme code="text.account.subscription.deactivateAutorenew" text="Deactivate Auto-Renew"/>
                            </button>
                            <input type="hidden" name="autorenew" value="false"/>
                            <input type="hidden" name="subscriptionId" value="${subscriptionData.id}"/>
                        </form:form>
                    </c:if>
                </span>
            </td>
        </tr>
        <tr>
            <td>
                <span><spring:theme code="text.account.subscription.renewalType" text="Renewal Type"/>:</span>
            </td>
            <td>
                <ycommerce:testId code="subscription_renewalType_label">
                    ${subscriptionData.renewalType}
                </ycommerce:testId>
            </td>
        </tr>
        <tr>
            <td>
                <span><spring:theme code="text.account.subscription.endDate" text="End Date"/>:</span>
            </td>
            <td>
                <ycommerce:testId code="subscription_endDate_label">
                    <fmt:formatDate value="${subscriptionData.endDate}" dateStyle="long" timeStyle="short" type="both"/>
                </ycommerce:testId>
            </td>
        </tr>
        <tr>
            <td colspan="2">
                <c:url value="/my-account/subscription/extend-term-duration" var="extendSubscriptionTermDuration" />
                <form:form class="extend_subscription_form" id="extendSubscriptionForm" action="${extendSubscriptionTermDuration}" method="p
                    <select id="contractDurationExtensionOptions" name="contractDurationExtension" >
```

```
                        <c:forEach items="${extensionOptions}" var="extensionOption" varStatus="extensionOptionCounter">
                            <option value="${extensionOption.code}">${extensionOption.name}</option>
                        </c:forEach>
                    </select>
                    <button type="submit" class="function_btn secondary">
                        <spring:theme code="text.account.subscription.updateTermDuration" text="Update"/>
                    </button>
                    <input type="hidden" name="subscriptionId" value="${subscriptionData.id}"/>
                </form:form>
            </td>
        </tr>
        <tr>
            <td colspan="2">
                <c:url value="/my-account/subscription/billing-activity/payment-method/replace" var="replaceSubscriptionPaymentMethodUrl" />
                <form:form class="update_paymentmethod_form" id="updatePaymentmethodForm" action="${replaceSubscriptionPaymentMethodUrl}" me
                    <select name="paymentMethodId" id="paymentMethods" <c:if test="${fn:length(paymentInfos) lt 2}">disabled</c:if>>
                        <c:forEach items="${paymentInfos}" var="paymentInfo">
                            <option value="${paymentInfo.subscriptionId}" <c:if test="${paymentInfo.subscriptionId eq subscriptionData.paymen
                        </c:forEach>
                    </select>
                    <button type="submit" class="function_btn secondary" <c:if test="${fn:length(paymentInfos) lt 2}">disabled</c:if>>
                        <spring:theme code="text.account.subscription.updatePaymentMethod" text="Update"/>
                    </button>
                    <input type="hidden" name="effectiveFrom" value="NOW"/>
                    <input type="hidden" name="subscriptionId" value="${subscriptionData.id}"/>
                </form:form>
            </td>
        </tr>
        </c:if>
    </table>
    <div class="subscriptionActivity">
        <h3><spring:theme code="text.account.subscription.activity" text="Activity"/></h3>
        <div class="order_number">
            <span><spring:theme code="text.account.subscription.orderNumber" text="Order Number"/>:</span>
            <span><ycommerce:testId code="subscription_orderNumber_link">
                <a href="${orderDetailsUrl}">${subscriptionData.orderNumber}</a>
                </ycommerce:testId>
            </span>
        </div>
    </div>
    <table id="subscription_activity" class="manageSubscriptionTable">
        <tr>
            <td colspan="2">
                <c:url value="/my-account/subscription/billing-activity" var="subscriptionBillingActivityUrl" />
                <form:form class="view_billing_activity_form" id="viewBillingActivityForm" action="${subscriptionBillingActivityUrl}" method=
                    <button type="submit" class="function_btn negative">
                        <spring:theme code="text.account.subscription.viewBillingActivity" text="View Billing Activity"/>
                    </button>
                    <input type="hidden" name="subscriptionId" value="${subscriptionData.id}"/>
                </form:form>
            </td>
        </tr>
        <c:if test="${fn:toUpperCase(subscriptionData.subscriptionStatus) eq 'CANCELLED'}">
        <tr>
            <td>
                <span><spring:theme code="text.account.subscription.cancelledDate" text="Cancelled Date"/>:</span>
            </td>
            <td>
                <ycommerce:testId code="subscription_cancelledDate_label">
                    <fmt:formatDate value="${subscriptionData.cancelledDate}" dateStyle="long" timeStyle="short" type="both"/>
                </ycommerce:testId>
            </td>
        </tr>
        </c:if>
        <tr>
            <td>
                <span><spring:theme code="text.account.subscription.startDate" text="Start Date"/>:</span>
            </td>
            <td>
                <ycommerce:testId code="subscription_startDate_label">
                    <fmt:formatDate value="${subscriptionData.startDate}" dateStyle="long" timeStyle="short" type="both"/>
                </ycommerce:testId>
            </td>
        </tr>
        <tr>
            <td>
                <span><spring:theme code="text.account.subscription.placedOn" text="Placed On"/>:</span>
            </td>
            <td>
                <ycommerce:testId code="subscription_placedOn_label">
                    <fmt:formatDate value="${subscriptionData.placedOn}" dateStyle="long" timeStyle="short" type="both"/>
                </ycommerce:testId>
            </td>
        </tr>
    </table>
</div>
<div class="clear"></div>
<div class="manageSubscriptionActions">
    <c:url value="/my-account/subscription" var="backToSubscriptions"/>
    <a href="${backToSubscriptions}" class="r_action_btn"><spring:theme code="text.account.subscription.returntosubscriptions" text="Return 
</div>
```

Customization of Tag Files

File Names: *<${addOnPageRoot}>*`/WEB-INF/tags/desktop/product/*.tag`

Initial Value	Customization Possibilities
N/A	Customize all tags here to meet your requirements.

Refer to the tags by a directive, such as `<%@ taglib prefix="product" tagdir="/WEB-INF/tags/addons/subscriptionstorefront/desktop/product" %>`

Define Page and Components via ImpEx

Define the new CMS page and its components via ImpEx, such as the following eample from the *<${addOnName}>*`/resources/${addOnName}/import/contentCatalogs/${catalogName}/cms-content.impex` file:

```
#
# Import the CMS content for the MAnage Subscriptions
#
$contentCatalog=b2ctelcoContentCatalog
$contentCV=catalogVersion(CatalogVersion.catalog(Catalog.id[default=$contentCatalog]),CatalogVersion.version[default=Staged])[default=$contentCatalo
$jarResourceCms=jar:de.hybris.platform.subscriptionstorefront.constants.SubscriptionstorefrontConstants&/subscriptionstorefront/import/cockpits/cmsc
$addonExtensionName=subscriptionstorefront

# My Subscriptions Page Preview
# Here we create a preview of the page for the cms cockpit.
INSERT_UPDATE Media;$contentCV[unique=true];code[unique=true];realfilename;@media[translator=de.hybris.platform.impex.jalo.media.MediaDataTranslator
;;subscriptions-preview;SubscriptionsPage.png;$jarResourceCms/SubscriptionsPage.png;image/png;hybris Accelerator;

## Configure page for My Subscriptions / Subscription details / ...
INSERT_UPDATE ContentPage;$contentCV[unique=true];uid[unique=true];name;masterTemplate(uid,$contentCV);label;defaultPage[default='true'];approvalSta
;;subscriptions;Manage My Subscriptions;AccountPageTemplate;/my-account/subscription
;;subscription;Subscription Details Page;AccountPageTemplate;subscription

## here we link the preview image with the page for cmscockpit
UPDATE ContentPage;$contentCV[unique=true];uid[unique=true];previewImage(code, $contentCV)
;;subscriptions;subscriptions-preview
;;subscription;subscription-preview

# CMS Link Component for the link on the main account page
INSERT_UPDATE CMSLinkComponent;$contentCV[unique=true];uid[unique=true];name;url;&linkRef;&componentRef;target(code)[default='sameWindow']
;;AccountSubscriptionsLink;AccountSubscriptionsLink;/my-account/subscription;AccountSubscriptionsLink;AccountSubscriptionsLink;

# CMS Navigation Nodes (used in the main account page)

INSERT_UPDATE CMSNavigationNode;uid[unique=true];$contentCV[unique=true];name;parent(uid, $contentCV);links(&linkRef)[mode=append];&nodeRef
;AccountLeftNavNode;;My Subscriptions;SiteRootNode;AccountSubscriptionsLink;AccountLeftNavNode

# the jsp pages we defined earlier
INSERT_UPDATE JspIncludeComponent;$contentCV[unique=true];uid[unique=true];name;page;actions(uid,$contentCV);&componentRef
;;AccountSubscriptionComponent;Account My Subscriptions Component;/WEB-INF/views/addons/$addonExtensionName/desktop/pages/account/accountSubscriptio
;;AccountSubscriptionDetailsComponent;Account Subscription Component;/WEB-INF/views/addons/$addonExtensionName/desktop/pages/account/accountSubscrip

# the content slots for the Subscriptions list page
INSERT_UPDATE ContentSlot;$contentCV[unique=true];uid[unique=true];name;active;cmsComponents(uid,$contentCV);;;
;;SideContent-subscriptions;Side Content Slot for My Subscriptions Details;true;AccountLeftNavigationComponent;;;
;;BodyContent-subscriptions;Body Content Slot for My Subscriptions Details;true;AccountSubscriptionComponent;;;
...
INSERT_UPDATE ContentSlotForPage;$contentCV[unique=true];uid[unique=true];position[unique=true];page(uid,$contentCV)[unique=true][default='subscript
;;SideContent-subscriptions;SideContent;;SideContent-subscriptions;;;
;;BodyContent-subscriptions;BodyContent;;BodyContent-subscriptions;;;
...
## Subscription Details page as Content Slot
INSERT_UPDATE ContentSlot;$contentCV[unique=true];uid[unique=true];name;active;cmsComponents(uid,$contentCV);;;
;;SideContent-subscription;Side Content Slot for Subscription Details;true;AccountLeftNavigationComponent;;;
;;BodyContent-subscription;Body Content Slot for Subscription Details;true;AccountSubscriptionDetailsComponent;;;
...
INSERT_UPDATE ContentSlotForPage;$contentCV[unique=true];uid[unique=true];position[unique=true];page(uid,$contentCV)[unique=true][default='subscript
;;SideContent-subscription;SideContent;;SideContent-subscription;;;
;;BodyContent-subscription;BodyContent;;BodyContent-subscription;;;
```

Controller Configuration

Add a new controller to serve the newly created page as shown by the example from the `AccountSubscriptionsPageController`:

```
/*
 * [y] hybris Platform
 *
 * Copyright (c) 2000-2015 hybris AG
 * All rights reserved.
 *
 * This software is the confidential and proprietary information of hybris
 * ("Confidential Information"). You shall not disclose such Confidential
 * Information and shall use it only in accordance with the terms of the
 * license agreement you entered into with hybris.
 *
 *
 */
package de.hybris.platform.subscriptionstorefront.controllers.pages;
import de.hybris.platform.acceleratorstorefrontcommons.breadcrumb.Breadcrumb;
import de.hybris.platform.acceleratorstorefrontcommons.breadcrumb.ResourceBreadcrumbBuilder;
import de.hybris.platform.acceleratorstorefrontcommons.controllers.pages.AbstractSearchPageController;
import de.hybris.platform.acceleratorstorefrontcommons.controllers.util.GlobalMessages;
import de.hybris.platform.cms2.exceptions.CMSItemNotFoundException;
import de.hybris.platform.commercefacades.order.OrderFacade;
import de.hybris.platform.commercefacades.order.data.CCPaymentInfoData;
import de.hybris.platform.commercefacades.order.data.CartData;
import de.hybris.platform.commercefacades.order.data.CartModificationData;
import de.hybris.platform.commercefacades.order.data.OrderData;
import de.hybris.platform.commercefacades.order.data.OrderEntryData;
import de.hybris.platform.commercefacades.product.ProductFacade;
import de.hybris.platform.commercefacades.product.ProductOption;
import de.hybris.platform.commercefacades.product.data.ProductData;
import de.hybris.platform.commercefacades.user.UserFacade;
import de.hybris.platform.commerceservices.order.CommerceCartModificationException;
import de.hybris.platform.servicelayer.exceptions.UnknownIdentifierException;
import de.hybris.platform.subscriptionfacades.SubscriptionFacade;
import de.hybris.platform.subscriptionfacades.action.SubscriptionUpdateActionEnum;
import de.hybris.platform.subscriptionfacades.data.SubscriptionBillingData;
import de.hybris.platform.subscriptionfacades.data.SubscriptionBillingDetailFileStream;
import de.hybris.platform.subscriptionfacades.data.SubscriptionData;
import de.hybris.platform.subscriptionfacades.exceptions.SubscriptionFacadeException;
import de.hybris.platform.subscriptionfacades.order.SubscriptionCartFacade;
import de.hybris.platform.subscrptionstorefront.util.MessageHandler;
import java.io.IOException;
import java.text.DateFormat;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Collection;
import java.util.Collections;
import java.util.Date;
import java.util.HashMap;
import java.util.List;
import java.util.Locale;
import java.util.Map;
import javax.annotation.Nonnull;
import javax.annotation.Resource;
import javax.servlet.http.HttpServletResponse;
import org.apache.commons.collections.CollectionUtils;
import org.apache.commons.io.IOUtils;
import org.apache.log4j.Logger;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.servlet.mvc.support.RedirectAttributes;

/**
 * Controller for home page.
 */
@Controller
@RequestMapping("/my-account/subscription")
public class AccountSubscriptionsPageController extends AbstractSearchPageController
{

        // Internal Redirects
        private static final String REDIRECT_MY_ACCOUNT = REDIRECT_PREFIX + "/my-account";
        private static final String REDIRECT_URL_CART = REDIRECT_PREFIX + "/cart";
        private static final String REDIRECT_MY_ACCOUNT_SUBSCRIPTION = REDIRECT_PREFIX + "/my-account/subscription/";

        // CMS Pages
        private static final String SUBSCRIPTIONS_CMS_PAGE = "subscriptions";
        private static final String SUBSCRIPTION_COMPARISON_CMS_PAGE = "subscriptionComparison";
        private static final String SUBSCRIPTION_BILLING_ACTIVITY_CMS_PAGE = "subscriptionBillingActivity";
        private static final String SUBSCRIPTION_DETAILS_CMS_PAGE = "subscription";

        /**
         * We use this suffix pattern because of an issue with Spring 3.1 where a Uri value is incorrectly extracted if it
         * contains on or more '.' characters. Please see https://jira.springsource.org/browse/SPR-6164 for a discussion on
         * the issue and future resolution.
         */
        private static final String SUBSCRIPTION_ID_PATH_VARIABLE_PATTERN = "{subscriptionId:.*}";

        private static final Logger LOG = Logger.getLogger(AccountSubscriptionsPageController.class);

        @Resource(name = "userFacade")
        protected UserFacade userFacade;

        @Resource(name = "accountBreadcrumbBuilder")
        private ResourceBreadcrumbBuilder accountBreadcrumbBuilder;

        @Resource(name = "subscriptionFacade")
        private SubscriptionFacade subscriptionFacade;

        @Resource(name = "cartFacade")
        private SubscriptionCartFacade cartFacade;

        @Resource(name = "orderFacade")
        private OrderFacade orderFacade;

        @Resource(name = "productFacade")
        private ProductFacade productFacade;

        @Resource(name = "messageHandler")
        private MessageHandler messageHandler;

    // handles the path as /my-account/subscription and returns view corresponding to identifier SUBSCRIPTIONS_CMS_PAGE ("subscriptions" - see the i
        @RequestMapping(method = RequestMethod.GET)
        public String subscriptions(@Nonnull final Model model) throws CMSItemNotFoundException
        {
                List<SubscriptionData> sortedSubscriptions = new ArrayList<>();
                try
                {
                        final Collection<SubscriptionData> subscriptions = subscriptionFacade.getSubscriptions();
                        if (subscriptions != null)
                        {
                                sortedSubscriptions = new ArrayList<>(subscriptions);
                        }
                }
                catch (final SubscriptionFacadeException e)
                {
                        LOG.error("Error while retrieving subscriptions", e);
                }

                final List<CCPaymentInfoData> paymentInfoData = userFacade.getCCPaymentInfos(true);
                final Map<String, CCPaymentInfoData> paymentInfoMap = new HashMap<>();

                for (final CCPaymentInfoData paymentInfo : paymentInfoData)
                {
                        paymentInfoMap.put(paymentInfo.getSubscriptionId(), paymentInfo);
                }
                storeCmsPageInModel(model, getContentPageForLabelOrId(SUBSCRIPTIONS_CMS_PAGE));
                setUpMetaDataForContentPage(model, getContentPageForLabelOrId(SUBSCRIPTIONS_CMS_PAGE));
                model.addAttribute("subscriptionFacade", subscriptionFacade);
                model.addAttribute("paymentInfoMap", paymentInfoMap);
                model.addAttribute("subscriptions", sortedSubscriptions);
                model.addAttribute("breadcrumbs", accountBreadcrumbBuilder.getBreadcrumbs("text.account.subscriptions"));
                model.addAttribute("metaRobots", "no-index,no-follow");
                messageHandler.supplementModelWithMessages(model);
                return getViewForPage(model);
        }
    // handles the path as /my-account/subscription/{subscriptionId:.*} and returns view corresponding to identifier SUBSCRIPTION_DETAILS_CMS_PAGE (
    @RequestMapping(value = "/" + SUBSCRIPTION_ID_PATH_VARIABLE_PATTERN, method = RequestMethod.GET)
        public String subscription(@PathVariable("subscriptionId") final String subscriptionId, final Model model)
                        throws CMSItemNotFoundException
        {
                try
                {
                        model.addAttribute("paymentInfos", userFacade.getCCPaymentInfos(true));
                        final SubscriptionData subscriptionDetails = subscriptionFacade.getSubscription(subscriptionId);
                        model.addAttribute("subscriptionData", subscriptionDetails);
                        final ProductData subscriptionProductData = getProductForSubscription(subscriptionDetails);
                        model.addAttribute("subscriptionProductData", subscriptionProductData);
                        final List<Breadcrumb> breadcrumbs = accountBreadcrumbBuilder.getBreadcrumbs(null);
                        breadcrumbs.addAll(accountBreadcrumbBuilder.getBreadcrumbs("text.account.subscriptions"));
                        breadcrumbs.add(new Breadcrumb("#", getMessageSource().getMessage("text.account.subscription.subscriptionBreadcrumb",
                                        new Object[]
                                        { subscriptionDetails.getId() }, "Subscription {0}", getI18nService().getCurrentLocale()), null));
                        model.addAttribute("breadcrumbs", breadcrumbs);
                        final List<SelectOption> extensionOptions = new ArrayList<SelectOption>();
                        for (int i = 1; i <= 3; i++)
                        {
                                final String counter = String.valueOf(i);
                                extensionOptions.add(new SelectOption(counter, getMessageSource().getMessage("text.account.subscription.extendTerm",
                                                new Object[]
                                                { subscriptionDetails.getContractFrequency(), counter }, null, getI18nService().getCurrentLocale()))
                        }
                        model.addAttribute("extensionOptions", extensionOptions);
                        final List<CCPaymentInfoData> paymentMethods = userFacade.getCCPaymentInfos(true);
                        model.addAttribute("paymentInfos", paymentMethods);
                        final List<ProductData> upsellingOptions = subscriptionFacade.getUpsellingOptionsForSubscription(subscriptionDetails
                                        .getProductCode());
                        model.addAttribute("upgradable", Boolean.valueOf(CollectionUtils.isNotEmpty(upsellingOptions)));
                }
                catch (final UnknownIdentifierException | SubscriptionFacadeException e)
                {
                        LOG.warn("Attempted to load a subscription that does not exist or is not visible", e);
                        return REDIRECT_MY_ACCOUNT;
                }
                storeCmsPageInModel(model, getContentPageForLabelOrId(SUBSCRIPTION_DETAILS_CMS_PAGE));
                model.addAttribute("metaRobots", "no-index,no-follow");
                setUpMetaDataForContentPage(model, getContentPageForLabelOrId(SUBSCRIPTION_DETAILS_CMS_PAGE));
                return getViewForPage(model);
        }
}
```

Spring Configuration

Declare the newly created controller in the *<${addOnName}>*`-web-spring.xml` file.

Customizing the Category Page

This document describes how to create a customized Category page.

Pages, Tags, and Templates

Variables

Variable Name	Description	Value
<${addOnName}>	The name of the AddOn	`b2ctelcostorefront`
<${addOnPageRoot}>	The location where custom page fragments must go	`b2ctelcostorefront/acceleratoraddon/web/webroot/WEB-INF`
<${addOnStoreExt}>	The name of the addon's store extension	`b2ctelcostore`
<${customCategoryJSP}>	Custom category file name	`servicePlanCategoryOverviewPage`

Replace Category Page with a Customized Page

File Name: *<${addOnStoreExt}>*/resources/b2ctelcostore/import/coredata/contentCatalogs/b2ctelcoContentCatalog/cms-content.impex

```
INSERT_UPDATE PageTemplate;$contentCV[unique=true];uid[unique=true];name;frontendTemplateName;restrictedPageTypes(code);active[default=true]
                                        ;;CategoryPageTemplate;Category Page Template;addon:${addOnName}/pages/telco/category/${customCategoryJSP};C
```

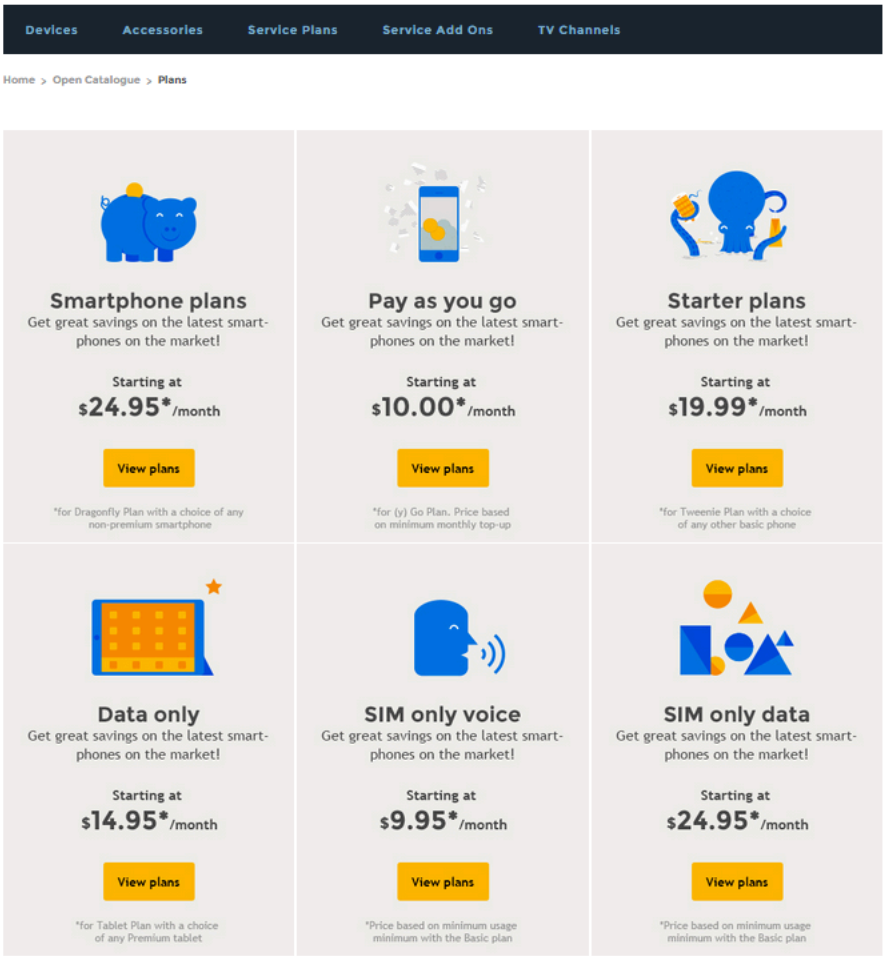
Customized the category page as shown in the example from the `servicePlanCategoryOverviewPage.jsp` file:

```
<%@ page trimDirectiveWhitespaces="true" %>
                                        <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
                                        <%@ taglib prefix="template" tagdir="/WEB-INF/tags/desktop/template" %>
                                        <%@ taglib prefix="nav" tagdir="/WEB-INF/tags/desktop/nav" %>
                                        <%@ taglib prefix="spring" uri="http://www.springframework.org/tags" %>
                                        <%@ taglib prefix="cms" uri="/WEB-INF/common/tld/cmstags.tld" %>
                                        <%@ taglib prefix="form" uri="http://www.springframework.org/tags/form" %>
                                        <%@ taglib prefix="ycommerce" uri="/WEB-INF/common/tld/ycommercetags.tld" %>
                                        <%@ taglib prefix="common" tagdir="/WEB-INF/tags/desktop/common" %>
                                        <%@ taglib prefix="breadcrumb" tagdir="/WEB-INF/tags/desktop/nav/breadcrumb" %>
                                        <template:page pageTitle="${pageTitle}">
                                        <div id="globalMessages">
                                        <common:globalMessages/>
                                        </div>

                                        <div class="span-20">
                                        <div id="productDetailUpdateable">
                                        <div class="span-24 plans-overview">
                                        <cms:pageSlot var="banner" position="Section2">
                                        <cms:component component="${banner}"/>
                                        </cms:pageSlot>
                                        </div>
                                        </div>
                                        </div>

                                        </template:page>
```

Add the Preview Page with a Preview Image to the CMS Cockpit

File Name: *<${addOnStoreExt}>*/resources/b2ctelcostore/import/sampledata/contentCatalogs/b2ctelcoContentCatalog/cms-content.impex

```
INSERT_UPDATE Media;$contentCV[unique=true];code[unique=true];mime;realfilename;@media[translator=de.hybris.platform.impex.jalo.media.MediaDataTrans
                                        ;;productDetailsForAddOnsPagePreview;text/png;ProductDetailsForAddOnsPage.png;$jarResourceCms/preview-images,
```



Smartphone plans
Get great savings on the latest smart-phones on the market!
Starting at
$24.95*/month
*for Dragonfly Plan with a choice of any non-premium smartphone

Pay as you go
Get great savings on the latest smart-phones on the market!
Starting at
$10.00*/month
*for (y) Go Plan. Price based on minimum monthly top-up

Starter plans
Get great savings on the latest smart-phones on the market!
Starting at
$19.99*/month
*for Tweenie Plan with a choice of any other basic phone

Data only
Get great savings on the latest smart-phones on the market!
Starting at
$14.95*/month
*for Tablet Plan with a choice of any Premium tablet

SIM only voice
Get great savings on the latest smart-phones on the market!
Starting at
$9.95*/month
*Price based on minimum usage minimum with the Basic plan

SIM only data
Get great savings on the latest smart-phones on the market!
Starting at
$24.95*/month
*Price based on minimum usage minimum with the Basic plan

Customizing the Product Details Pages

This document describes how to customize the Product Details pages.

Pages, Tags, and Templates

Variables

Templates and layout pages are taken from B2C except for the Add To Cart functionality, which is customized according to your needs. Due to the changes applied to the Add To Cart logic, the layout page would look as shown in the example from the `xxxLayoutPage.jsp` file:

```
<%@ taglib prefix="telcoProduct" tagdir="/WEB-INF/tags/addons/b2ctelcostorefront/desktop/product" %>
                                        ...
                                        <template:page pageTitle="${pageTitle}">
                                        ...
                                        <product:productDetailsPanel product="${product}"
                                        galleryImages="${galleryImages}" />

                                        <telcoProduct:productAddToCartPanel product="${product}" allowAddToCart="true"/>
                                        .....
                                        <product:productPageTabs />
                                        <cms:pageSlot position="Section4" var="feature" element="div"
                                        class="span-24 section4 cms_disp-img_slot">
                                        <cms:component component="${feature}" />
                                        </cms:pageSlot>
                                        </template:page>
```

Controller Configuration

Changes:

The ProductPageController is taken directly from B2C. Modify the **AddToCartController** according to your needs.

Customization of ImpEx

Import your page templates and your restrictions. There is a similar ImpEx example for creating a new page template on the Customizing the Header and Footer page.

Customization of XML

In the `XXXstorefront-web-spring.xml` file, you need to override the existing B2C product details controller to reference your AddOn implementation.

```
<bean name="addToCartController" class="de.hybris.platform.b2ctelcostorefront.controllers.misc.AddToCartController"/>
                                <bean name="productPageController" class="de.hybris.platform.b2ctelcostorefront.controllers.pages.ProductPageControl

                                <bean class="org.springframework.web.servlet.handler.SimpleUrlHandlerMapping">
                                <property name="mappings">
                                <props>
                                <prop key="/**/p">productPageController</prop>
                                <prop key="/cart/add">addToCartController</prop>
                                <prop key="/cart/addBundle">addToCartController</prop>
                                </props>
                                </property>
                                </bean>
```

Customizing the Order Details Page

Learn how to change the order totals and order detail components of the Order Details page.

Apart from layout changes, the main difference between the order details page, compared to the standard B2C Accelerator, is the display of a new order totals component and new order details component, containing price information for service plans and billing cycle details.

Pages, Tags, and Templates

Variables

Variable Name	Description	Value
<${addOnName}>	The name of the AddOn	`b2ctelcostorefront`
<${addOnPageRoot}>	The location where custom page fragments must go	`b2ctelcostorefront/acceleratoraddon/web/webroot/WEB-INF`

Files to Copy

File Name	Source Path	Target Path
Order Totals Component	`/yacceleratorstorefront/web/webroot/WEB-INF/views/desktop/pages/account/accountOrderDetailOrderTotals.jsp`	*<${addOnPageRoot}>*`/views/desktop/pages/account/accountOrderDetailOrderTo`
Order Totals Tag	`/yacceleratorstorefront/web/webroot/WEB-INF/tags/desktop/order/orderTotalsItem.tag`	*<${addOnPageRoot}>* `/tags/desktop/order/orderTotalsItem.tag`
Order Details Component	`/yacceleratorstorefront/web/webroot/WEB-INF/views/desktop/pages/account/accountOrderDetailItems.jsp`	*<${addOnPageRoot}>* `/views/desktop/pages/account/accountOrderDetailItems.`
Order Details Tag	`/yacceleratorstorefront/web/webroot/WEB-INF/tags/desktop/order/accountOrderDetailsItem.tag`	*<${addOnPageRoot}>*`/tags/desktop/order/accountOrderDetailsItem.tag`

Customization of JSP Files

File Name: *<${addOnPageRoot}>*`/views/desktop/pages/account/accountOrderDetailOrderTotals.jsp`

Sample Code	Notes

`<%@ taglib prefix="telcoOrder" tagdir="/WEB-INF/tags/addons/b2ctelcostorefront/desktop/order" %>`
`[...]`
 	
`<telcoOrder:orderTotalsItem order="${orderData}"/>`
`[...]`
	In order to reference tags from the AddOn's location, it is necessary to point to the target location where the AddOn's tags reside following installation. To do so, you have to declare a new prefix.
	Note that the taglib's directory is relative to the storefront extension, e.g. `yacceleratorstorefront`, while the `b2ctelcostorefront` in the path reflects your AddOn's name).
	With the new `taglib` declared, the tags can be used in the JSP using the new prefix, e.g. `telcoOrder`.

Customization of Tag Files

In this example, there is no specific customization necessary. For details look at the respective tag files mentioned above.

Customization of ImpEx Files

The order details page is defined as a `ContentPage` using the `AccountPageTemplate`.

For more details, see the `/b2ctelcostore/resources/b2ctelcostore/import/coredata/contentCatalogs/b2ctelcoContentCatalog/cms-content.impex` file.

The following example is an excerpt from the /yacceleratorinitialdata/resources/yacceleratorinitialdata/import/coredata/contentCatalogs/catalogName/cms-content.impex file:

```
# Create PageTemplates
# These define the layout for pages
# "FrontendTemplateName" is used to define the JSP that should be used to render the page for pages with multiple possible layouts.
# "RestrictedPageTypes" is used to restrict templates to page types
INSERT_UPDATE PageTemplate;$contentCV[unique=true];uid[unique=true];name;frontendTemplateName;restrictedPageTypes(code);active[default=true]
[...]
;;AccountPageTemplate;Account Page Template;account/accountLayoutPage;ContentPage;false;

[...]
INSERT_UPDATE ContentSlotForTemplate;$contentCV[unique=true];uid[unique=true];position[unique=true];pageTemplate(uid,$contentCV)[unique=true][defaul
;;SiteLogo-AccountPage;SiteLogo;;SiteLogoSlot;true
;;HomepageLink-AccountPage;HomepageNavLink;;HomepageNavLinkSlot;true
;;NavigationBar-AccountPage;NavigationBar;;NavigationBarSlot;true
;;MiniCart-AccountPage;MiniCart;;MiniCartSlot;true
;;Footer-AccountPage;Footer;;FooterSlot;true
;;HeaderLinks-AccountPage;HeaderLinks;;HeaderLinksSlot;true
;;SearchBox-AccountPage;SearchBox;;SearchBoxSlot;true
;;TopHeaderSlot-AccountPage;TopHeaderSlot;;TopHeaderSlot;true
;;BottomHeaderSlot-AccountPage;BottomHeaderSlot;;BottomHeaderSlot;true
;;PlaceholderContentSlot-AccountPage;PlaceholderContentSlot;;PlaceholderContentSlot;true

[...]
# Functional Content Pages
INSERT_UPDATE ContentPage;$contentCV[unique=true];uid[unique=true];name;masterTemplate(uid,$contentCV);label;defaultPage[default='true'];approvalSta
[...]
;;orders;Order History Page;AccountPageTemplate;orders
[...]
```

The content slots and related components are defined in the /b2ctelcostore/resources/b2ctelcostore/import/sampledata/contentCatalogs/b2ctelcoContentCatalog/cms-content.impex file.

The following is an excerpt from the /b2ctelcostore/resources/b2ctelcostore/import/sampledata/contentCatalogs/b2ctelcoContentCatalog/cms-content.impex file:

```
[...]
###### Account Order Details
# ContentSlot
INSERT_UPDATE ContentSlot;$contentCV[unique=true];uid[unique=true];name;active;cmsComponents(&componentRef);;;
;;SideContent-orderdetail;Side Content Slot for My Account Order Details;true;AccountLeftNavigationComponent;;;
;;BodyContent-orderdetail;Body Content Slot for My Account Order Details;true;AccountOrderDetailsHeadlineComponent,AccountOrderDetailsTotalsComponen

# ContentSlotForPage
INSERT_UPDATE ContentSlotForPage;$contentCV[unique=true];uid[unique=true];position[unique=true];page(uid,$contentCV)[unique=true][default='order'];c
;;SideContent-orderdetail;SideContent;;SideContent-orderdetail;;;
;;BodyContent-orderdetail;BodyContent;;BodyContent-orderdetail;;;
[...]
```

In order to modify the default order totals and order details components, copy and customize the existing JSPs (for their respective CMS components) and the TAG files referenced in these components into the AddOn, and then adjust the component definitions via ImpEx.

Overwriting the Existing Components in ImpEx

In order to reference the newly-created JSP files, you need to redefine the JspIncludeComponents via ImpEx and point to the JSP files in their respective locations.

Furthermore, the content slots using these components need to be defined as well. Note that the AccountLeftNavigationComponent referenced here is omitted from the following sample file, /b2ctelcostore/resources/b2ctelcostore/import/sampledata/contentCatalogs/b2ctelcoContentCatalog/cms-content.impex

Initial Value

```
[...]
$addonExtensionName=b2ctelcostorefront

[...]
INSERT_UPDATE JspIncludeComponent;$contentCV[unique=true];uid[unique=true];name;page;actions(uid,$contentCV);&componentRef
;;AccountOrderDetailsTotalsComponent;Account Order Details Order Totals Component;/WEB-INF/views/addons/$addonExtensionName/desktop/pages/account/accountOrderD
;;AccountOrderDetailsItemsComponent;Account Order Details Items Info Component;/WEB-INF/views/addons/$addonExtensionName/desktop/pages/account/accountOrderDeta

[...]
###### Account Order Details
# ContentSlot
INSERT_UPDATE ContentSlot;$contentCV[unique=true];uid[unique=true];name;active;cmsComponents(&componentRef);;;
;;SideContent-orderdetail;Side Content Slot for My Account Order Details;true;AccountLeftNavigationComponent;;;
;;BodyContent-orderdetail;Body Content Slot for My Account Order Details;true;AccountOrderDetailsHeadlineComponent,AccountOrderDetailsTotalsComponent,AccountOr

# ContentSlotForPage
INSERT_UPDATE ContentSlotForPage;$contentCV[unique=true];uid[unique=true];position[unique=true];page(uid,$contentCV)[unique=true][default='order'];contentSlot(
;;SideContent-orderdetail;SideContent;;SideContent-orderdetail;;;
;;BodyContent-orderdetail;BodyContent;;BodyContent-orderdetail;;;

[...]
```

Controller Configuration

The controller logic for the order details page is defined in /yacceleratorstorefront/web/src/de/hybris/platform/yacceleratorstorefront/controllers/pages/AccountPageController.java file.

In this example, there is no customization necessary as the additional data displayed in the order details and order totals components is already added into the DTO objects in the facade layer.

Spring Configuration

No additional spring configuration is necessary in this example.

Customizing the Mini Cart

Learn how to customize the Mini Cart.

Pages, Tags, and Templates

Variables

Variable Name	Description	Value
<${addOnName}>	The name of the AddOn	`b2ctelcostorefront`
<${addOnPageRoot}>	The location where custom page fragments must go	`b2ctelcostorefront/acceleratoraddon/web/webroot/WEB-INF`
<${addOnSource}>	The source path for an AddOn	`b2ctelcostorefront/acceleratoraddon/web/src`

Files to Copy

File Name	Source Path	Target Path
Controller	`/yacceleratorstorefront/web/src/de/hybris/platform/yacceleratorstorefront/controllers/misc/MiniCartController.java`	*<${addOnSource}>*`/de/hyb`
Controller Supporting Class	`/yacceleratorstorefront/web/src/de/hybris/platform/yacceleratorstorefront/controllers/ControllerConstants.java`	*<${addOnSource}>*`/de/hyb`
Mini Cart Popup Page	`/yacceleratorstorefront/web/webroot/WEB-INF/views/desktop/fragments/cart/cartPopup.jsp`	*<${addOnPageRoot}>*`/views`
Mini Cart Panel Page	`/yacceleratorstorefront/web/webroot/WEB-INF/views/desktop/fragments/cart/miniCartPanel.jsp`	*<${addOnPageRoot}>*`/views`

Customization of JSP Files

File Name: *<${addOnPageRoot}>*`/views/desktop/fragments/cart/cartPopup.jsp`

Initial Value	Customization Possibilities
New logic to only display the `cartURL` when there are more than 0 items in the cart.	You can customize this fragment as required by your project.

Customization of Tag Files

File Name: *<${addOnPageRoot}>*`/views/desktop/fragments/cart/miniCartPanel.jsp`

Initial Value	Customization Possibilities
No changes by default	You can customize this fragment as required by your project.

Controller Configuration

Class Name	Source Path	Target Path
MiniCartController	`/yacceleratorstorefront/web/src/de/hybris/platform/yacceleratorstorefront/controllers/misc/MiniCartController.java`	*<${addOnSource}>*`/d`

Changes:

- Change the reference from `ControllerConstants` to `TelcoControllerConstants`
- Change the return value for `de.hybris.platform.b2ctelcostorefront.controllers.misc.MiniCartController.getMiniCart(String, Model)` to return JSP in the AddOn (see above).
- Change the return value for `de.hybris.platform.b2ctelcostorefront.controllers.misc.MiniCartController.rolloverMiniCartPopup(String, Model)` to return JSP from the AddOn (see above).

Class Name	Source Path	Target Path
ControllerConstants	`/yacceleratorstorefront/web/src/de/hybris/platform/yacceleratorstorefront/controllers/ControllerConstants.java`	*<${addOnSource}>*`/de/h`

Changes

- Change to reference the customized page view names. For example, for the page view for the `cartPopUp`:
  - Old: `String CartPopup = "fragments/cart/cartPopup";`

- New: `String CartPopup = ADDON_PREFIX + "fragments/cart/cartPopup";`
- Where: `String ADDON_PREFIX = "addon:/b2ctelcostorefront/";`

Spring Configuration

File Name	Source Path	Target Path
<${addOnName}>-web-spring.xml	N/A	N/A

Bean Changes:

```
<!-- Add this new bean configuration -->
                                <bean name="miniCartController" class="de.hybris.platform.b2ctelcostorefront.controllers.misc.MiniCartController"/>

                                <!-- add the url to controller mapping to the existing SimpleUrlHandlerMapping bean -->
                                <bean class="org.springframework.web.servlet.handler.SimpleUrlHandlerMapping">
                                <property name="mappings">
                                <props>
                                ...
                                <prop key="/cart/miniCart/**">miniCartController</prop>
                                <prop key="/cart/rollover/**">miniCartController</prop>
                                </props>
                                </property>
                                </bean>
```

Customizing the Cart Page

Learn how to customize the Cart page.

Pages, Tags, and Templates

Variables

Variable Name	Description	Value
<${addOnName}>	The name of the AddOn	b2ctelcostorefront
${addOnPageRoot}	The location where custom page fragments must go	b2ctelcostorefront/acceleratoraddon/web/webroot/WEB-INF
${addOnWebSource}	The location where custom web logic (controllers) must go	b2ctelcostorefront/acceleratoraddon/web/src

Files to Copy

Copy these files to customize your Accelerator implementation.

File Name	Source Path	Target Path
Cart Page	/yacceleratorstorefront/web/webroot/WEB-INF/views/desktop/pages/cart/cartPage.jsp	*<${addOnPageRoot}>*/views/desktop/pages/cart/cartPage.jsp
Cart Page Tags	/b2ctelcostorefront/acceleratoraddon/web/webroot/WEB-INF/tags/desktop/cart/*.tag	*<${addOnPageRoot}>*/tags/desktop/cart/
ImpEx File	N/A	N/A

Customization of JSP Files

File Name: *<${addOnPageRoot}>*/views/desktop/pages/cart/cartPage.jsp

Initial Value	Customization Possibilities
`<%@ taglib prefix="cart" tagdir="/WEB-INF/tags/desktop/cart" %>`	`<%@ taglib prefix="cart" tagdir="/WEB-INF/tags/addons/b2ctelcostorefront/desktop/cart" %>`

Customization of Tag Files

File Name: *<${addOnPageRoot}>*/tags/desktop/cart/*.tag

Initial Value	Customization Possibilities
N/A	• Customize all tags here to meet your requirements. • Customize product populator components (in b2ctelcofacades extension) to change the ProductData objects, which are then used by the tags.

Controller Configuration

Class Name	Source Path	Target Pa
CartPageController	/yacceleratorstorefront/web/src/de/hybris/platform/yacceleratorstorefront/controllers/pages/CartPageController.java	*<${addu*
TelcoControllerConstants	/b2ctelcostorefront/src/de/hybris/platform/b2ctelcostorefront/controllers/TelcoControllerConstants.java	N/A

Changes:

- Change the value returned by method CartPageController.showCart(Model) method to point to the AddOn-specific customization. Remember that your new controllers cannot be configured through annotations due to the pre-existing controllers (pre-registered in the spring context) from yacceleratorstorefront. You must configure your new controller in a custom UrlHandlerMapping in the AddOn's spring.xml configuration file (see the web-spring.xml example below).
- The b2ctelcostorefront AddOn's overriding controller returns the *<TelcoControllerConstants.Views.Pages.Cart.CartPage>* value, which points to the *<${addOnPageRoot}>*/views/desktop/pages/cart/cartPage.jsp file.
- This JSP is not fully WCMS enabled. It uses some slots for layout, but the main cart page tag is inlined logic, not a CMS component.

Spring Configuration

File Name	Source Path	Target Path
<${addOnName}>-web-spring.xml	N/A	N/A

Bean Changes:

```
<bean name="cartPageController" class="de.hybris.platform.b2ctelcostorefront.controllers.pages.CartPageController"/>

                                <bean class="org.springframework.web.servlet.handler.SimpleUrlHandlerMapping">
                                <property name="mappings">
                                <props>
                                ...
                                <prop key="/cart/**">cartPageController</prop>
                                ...
                                </props>
                                </property>
                                </bean>
```

Customizing the Checkout Registration/Login Process

Learn how to customize the checkout registration and login process.

All on-page sections of the the Registration/Login part of the Accelerator checkout process have been broken down into components, which simplifies customizing the Registration/Login pages. Now you only need to remove the existing components from the page, create new components based on the originals, then add your new components to the page.

To implement the b2ctelcostorefront AddOn, you must remove the guest checkout component and replace the controller logic with your custom code, following the standard method for customizing controllers.

Pages, Tags, and Templates

Variables

Variable Name	Description	Value
<${addOnName}>	The name of the AddOn	b2ctelcostorefront
<${addOnPageRoot}>	The location where custom page fragments must go	b2ctelcostorefront/acceleratoraddon/web/webroot/WEB-INF
<${addOnSource}>	Source path for the AddOn	b2ctelcostorefront/acceleratoraddon/web/src
<${addOnResources}>	The base resource path for your AddOn	b2ctelcostorefront/resources/b2ctelcostorefront
<${projectData}>	The project data path for your project	b2ctelcostore/resources/b2ctelcostore

Files to Copy

File Name	Source Path	Target Path
CMS Content ImpEx File	/b2ctelcostore/resources/b2ctelcostore/import/sampledata/contentCatalogs/b2ctelcoContentCatalog/cms-content.impex	*<${projectData}>*/import/coredata/content content.impex

Customization of ImpEx Files

Disabling the GuestCheckoutLoginComponent using ImpEx

File Name: *<${addOnPageRoot}>*`/resources/b2ctelcostore/import/coredata/contentCatalogs/b2ctelcoContentCatalog/cms-content.impex`

Initial Value	Customization Possibilities

`# file: /b2ctelcostore/resources/b2ctelcostore/import/coredata/contentCatalogs/b2ctelcoContentCatalog/cms-content.impex`
`# Lines 599-600`
 	
`#Remove the GuestCheckoutLoginComponent by making it invisible (visible=false)`
`INSERT_UPDATE JspIncludeComponent;$contentCV[unique=true];uid[unique=true];name;page;visible;actions(uid,$contentCV);&componentRef`
`;;GuestCheckoutLoginComponent;Guest Checkout Login Component;checkoutGuestLogin.jsp;false;;GuestCheckoutLoginComponent`
	ImpEx syntax and content have to obey ImpEx rules. For more information, see:
	• impex Extension
	• cms2 Extension Tutorial

Controller Configuration

The controller configuration is only necessary if you need to change the default registration/login process at the controller level.

Class Name	Source Path
AbstractLoginPageController	`/b2ctelcostorefront/acceleratoraddon/web/src/de/hybris/platform/b2ctelcostorefront/controllers/pages/AbstractLoginPageContro`
AbstractRegisterPageController	`/b2ctelcostorefront/acceleratoraddon/web/src/de/hybris/platform/b2ctelcostorefront/controllers/pages/AbstractRegisterPageCon`
TelcoCheckoutLoginController	`/b2ctelcostorefront/acceleratoraddon/web/src/de/hybris/platform/b2ctelcostorefront/controllers/pages/TelcoCheckoutLoginContr`
TelcoCheckoutRegisterController	`/b2ctelcostorefront/acceleratoraddon/web/src/de/hybris/platform/b2ctelcostorefront/controllers/pages/TelcoCheckoutRegisterCo`
TelcoLoginPageController	`/b2ctelcostorefront/acceleratoraddon/web/src/de/hybris/platform/b2ctelcostorefront/controllers/pages/TelcoLoginPageControlle`
TelcoRegisterPageController	`/b2ctelcostorefront/acceleratoraddon/web/src/de/hybris/platform/b2ctelcostorefront/controllers/pages/TelcoRegisterPageContro`

Changes:

Customize your checkout process as necessary.

Spring Configuration

This Spring configuration is only necessary if you have changed the default registration/login controllers.

File Name	Source Path	Target Path
<${addOnName}>`-web-spring.xml`	`/b2ctelcostorefront/resources/b2ctelcostorefront/web/spring/b2ctelcostorefront-web-spring.xml`	*<${addOnResources}>*`/web/spring/${addOnName}-web-spring.xml`

Bean Changes:

```
<!-- Use your custom classnames here -->
        <bean name="checkoutLoginController" class="de.hybris.platform.${addOnName}.controllers.pages.TelcoCheckoutLoginController"/>
        <bean name="checkoutRegisterController" class="de.hybris.platform.${addOnName}.controllers.pages.TelcoCheckoutRegisterController"/>
        <bean name="loginPageController" class="de.hybris.platform.${addOnName}.controllers.pages.TelcoLoginPageController"/>
        <bean name="registerPageController" class="de.hybris.platform.${addOnName}.controllers.pages.TelcoRegisterPageController"/>

        <!-- Add your new components to your URL handler mapping -->
        <bean class="org.springframework.web.servlet.handler.SimpleUrlHandlerMapping">
                <property name="mappings">
                    <props>
                                        ...
                        <prop key="/login/checkout">checkoutLoginController</prop>
                        <prop key="/register/checkout">checkoutRegisterController</prop>
                        <prop key="/login">loginPageController</prop>
                        <prop key="/register">registerPageController</prop>
                                        ...
                    </props>
                </property>
        </bean>
```

Customizing the Account - Payment Page

This document describes how to customize the Payment page.

i Note

Prerequisite: For this example to work the `b2ctelcocheckoutaddon` module must be installed as the controller re-uses the silent order post JSP page from that addon module.

Pages, Tags, and Templates

Variables

Variable Name	Description	Value
<${addOnName}>	The name of the Addon	`b2ctelcostorefront`

Variable Name	Description	Value
<${addOnPageRoot}>	The location where custom page fragments must go	`b2ctelcostorefront/acceleratoraddon/web/webroot/WEB-INF`

Files to Copy

File Name	Source Path	Target Path
Account Payment Info Page	`/yacceleratorstorefront/web/webroot/WEB-INF/views/desktop/pages/account/accountPaymentInfoPage.jsp`	*<${addOnPageRoot}>*`/views/desktop/pages/account/accountPaymentInfoPage.jsp`

Customization of JSP Files

In this example, there is no specific customization necessary. For further details, look at the JSP file mentioned above.

Customization of Tag Files

In this example, there is no specific customization necessary.

Customization of ImpEx Files

Overwriting the Existing Components in ImpEx

In order to reference the newly created JSP files, you need to redefine the `JspIncludeComponents` via ImpEx and point to the JSP files in their respective location.

Furthermore, the content slots using these components need to be defined as well.

File Name: `/b2ctelcostore/resources/b2ctelcostore/import/sampledata/contentCatalogs/b2ctelcoContentCatalog/cms-content.impex`

Initial Value

```
# Functional Content Pages
                                                              UPDATE ContentPage;$contentCV[unique=true];uid[unique=true];previewImage(code, $contentCV)
                                                              ;;payment-details;paymentDetailsPagePreview

                                                              INSERT_UPDATE JspIncludeComponent;$contentCV[unique=true];uid[unique=true];name;page;actions(ui
                                                              ;;AccountPaymentDetailsComponent;Account Payment Details Component;/WEB-INF/views/addons/b2ctel

                                                              ######  Account payment detail page
                                                              # ContentSlot
                                                              INSERT_UPDATE ContentSlot;$contentCV[unique=true];uid[unique=true];name;active;cmsComponents(&c
                                                              ;;SideContent-payment-details;Side Content Slot for My Account payment-details;true;AccountLeft
                                                              ;;BodyContent-payment-details;Body Content Slot for My Account payment-details;true;AccountPaym
                                                              # ContentSlotForPage
                                                              INSERT_UPDATE ContentSlotForPage;$contentCV[unique=true];uid[unique=true];position[unique=true]
                                                              ;;SideContent-payment-details;SideContent;;SideContent-payment-details;;;
                                                              ;;BodyContent-payment-details;BodyContent;;BodyContent-payment-details;;;

                                                              # Navigation Bar Component
                                                              INSERT_UPDATE AccountNavigationComponent;$contentCV[unique=true];uid[unique=true];name;navigati
                                                              ;;AccountPaymentDetailsNavigationComponent;Account Payment Details Navigation Component;Account

                                                              # CMS Navigation Nodes
                                                              INSERT_UPDATE CMSNavigationNode;uid[unique=true];$contentCV[unique=true];name;parent(uid, $cont
                                                              ;AccountPaymentDetailsNavNode;;Payment Details;SiteRootNode;AccountManagePaymentDetailsLink;Acc

                                                              # CMS Link Components
                                                              INSERT_UPDATE CMSLinkComponent;$contentCV[unique=true];uid[unique=true];name;url;&linkRef;&comp
                                                              ;;AccountPaymentDetailsLink;AccountPaymentDetailsLink;/my-account/my-payment-details;AccountPay
                                                              ;;AccountManagePaymentDetailsLink;AccountManagePaymentDetailsLink;/my-account/my-payment-detail
```

Controller Configuration

The controller handles `/my-account/my-payment-details` path using the page identified as `PAYMENT_DETAILS_CMS_PAGE = "my-payment-details"` (see the identifier in the `cms-content.impex`).

`/b2ctelcostorefront/acceleratoraddon/web/src/de/hybris/platform/b2ctelcostorefront/controllers/pages/myaccount/PaymentDetailsPageController.`

```
package de.hybris.platform.b2ctelcostorefront.controllers.pages.myaccount;
                                  ....
                                  @Controller
                                  @RequestMapping(PaymentDetailsPageController.CONTROLLER_CTXT)
                                  public class PaymentDetailsPageController extends AbstractAddOnPageController
                                  {
                                  /**
                                  * This controller's default base context URI.
                                  */
                                  static final String CONTROLLER_CTXT = "/my-account/my-payment-details";
                                  private static final Logger LOG = Logger.getLogger(PaymentDetailsPageController.class);
                                  private static final String PAYMENT_DETAILS_CMS_PAGE = "payment-details";
                                  ....
@RequestMapping(method = RequestMethod.GET)
@RequireHardLogIn
public String paymentDetails(final Model model) throws CMSItemNotFoundException
{
model.addAttribute("customerData", customerFacade.getCurrentCustomer());
final List<CCPaymentInfoData> ccPaymentInfos = userFacade.getCCPaymentInfos(true);
model.addAttribute("paymentInfoData", ccPaymentInfos);
setUpSubscriptionsPerPaymentMethod(model, ccPaymentInfos);
storeCmsPageInModel(model, getContentPageForLabelOrId(PAYMENT_DETAILS_CMS_PAGE));
setUpMetaDataForContentPage(model, getContentPageForLabelOrId(ADD_EDIT_ADDRESS_CMS_PAGE));
model.addAttribute("breadcrumbs", accountBreadcrumbBuilder.getBreadcrumbs("text.account.paymentDetails"));
model.addAttribute("metaRobots", "noindex,nofollow");
return getViewForPage(model);
}

....

}
```

Spring Configuration

The following excerpt from the `b2ctelcostorefront/resources/b2ctelcostorefront/web/spring/b2ctelcostorefront-web-spring.xml` file shows the Spring configuration:

```
...
                                   <bean name="paymentDetailsPageController" class="de.hybris.platform.b2ctelcostorefront.controllers.pages.myaccount.Pa
                                   ...
                                   <bean class="org.springframework.web.servlet.handler.SimpleUrlHandlerMapping">
                                   <property name="mappings">
                                   <props>
                                   ...
                                   <prop key="/my-account/my-payment-details/**">paymentDetailsPageController</prop>
                                   </props>
                                   </property>
                                   </bean>
                                   ...
```

Overwriting an Existing Email Template

Learn how to overwrite an existing Email template.

Pages, Tags, and Templates

Variables

Variable Name	Description	Example Value
<${projectName}>	The name of the project	`b2ctelco`
<${dataModule}>	The name of the project's data module	`b2ctelcostore`

Files to Copy

File Name	Source Path
Email Velocity Templates	`/b2ctelcostore/resources/b2ctelcostore/import/cockpits/cmscockpit/acc/structure-view/structure_orderConfirmationEmailTemplate.vm`
Email Velocity Templates	`/b2ctelcostore/resources/b2ctelcostore/import/emails/email-orderConfirmationBody.vm`

```
## messageSource=classpath:/b2ctelcostore/messages/email-orderConfirmation_$lang.properties
#macro( genHtmlBoldFont $text )
<font color="#414a4f" size="2" face="Arial, Helvetica, sans-serif"><b>$text</b></font>
#end
#macro(genHtmlLinkStartTag $url)
<a href="$url"><font color="#484848">
#end
#macro(genHtmlLinkEndTag)
</font></a>
#end
#macro(genHtmlLink $url $textColor $bodyContent)
<a href="$url"><font color="$textColor">$bodyContent</font></a>
#end
<html>
    <head>
    </head>
    <body>
    <table width="100%" border="0" align="center" cellpadding="0" cellspacing="0">
        <tr>
            <td>&nbsp;</td>
        </tr>
        <tr>
            <td align="center" valign="top">
                <table width="1024" border="0" align="center" cellpadding="0" cellspacing="0" style="font-size:16px; line-height:16px">
                    <tr>
                        <td align="center" valign="top">
                            <table width="800" cellpadding="0" cellspacing="0" border="0" align="center">
```

File Name	Source Path

```
<tr>
    <td valign="middle">&nbsp;</td>
</tr>
<tr>
    <td valign="middle">
        ${ctx.cmsSlotContents.SiteLogo}
        <img src="${ctx.themeResourceUrl}/images/header_01.png" alt="" width="229" height="72" border="0" align="righ
    </td>
</tr>
<tr>
    <td height="30" align="right" valign="middle" bgcolor="#000000">
        <font color="#FFFFFF" size="2" face="Arial, Helvetica, sans-serif"><a href="${ctx.secureBaseUrl}/my-account">
    </td>
</tr>
<tr>
    <td align="center" valign="middle">
        <a href="${ctx.baseUrl}" style="display:block; margin-top:10px;margin-bottom:10px;">${ctx.cmsSlotContents.Top
    </td>
</tr>
<tr>
    <td>&nbsp;</td>
</tr>
<tr>
    <td align="left" valign="top">
        <p><font color="#484848" size="2" face="Arial, Helvetica, sans-serif"><b>${ctx.messages.getMessage('salutatio
        <p><font color="#484848" size="2" face="Arial, Helvetica, sans-serif">${ctx.messages.getMessage('thankYouForC
        <table width="100%" border="0" align="center" cellpadding="0" cellspacing="0" style="padding:10px 0 50px">
            <tr>
                <td background="#FFFFFF">
                    <table width="100%" cellpadding="0" cellspacing="0" background="#FFFFFF" style="table-layout: fix
                        #set ($firstBundleNo=-99)
                        #set ($displayPackageNo=false)
                        #foreach( $ordEntry in ${ctx.order.entries} )
                            #if($ordEntry.bundleNo > 0 && $firstBundleNo < 0)
                                #set ($firstBundleNo=$ordEntry.bundleNo)
                            #end
                            #if($ordEntry.bundleNo > 0 && $firstBundleNo > 0 && $firstBundleNo != $ordEntry.bundleNo)
                                #set ($displayPackageNo=true)
                            #end
                        #end
                        #set ($lastBundleNo=-99)
                        #foreach( $entry in ${ctx.order.entries} )
                            #if($lastBundleNo != $entry.bundleNo)
                                #set ($lastBundleNo=$entry.bundleNo)
                                <tr>
                                    <td width="auto" style="width:120px; font-size:16px; font-family:Arial, Helvetica
                                        <font color="#484848" size="2" face="Arial, Helvetica, sans-serif">
                                            <b>
                                                #if($entry.rootBundleTemplate)
                                                    $entry.rootBundleTemplate.name
                                                    #if($displayPackageNo)
                                                        (${ctx.messages.bundleNo}. $entry.bundleNo)
                                                    #end
                                                    #else
                                                        ${ctx.messages.productsAndAccessories}
                                                    #end
                                                </b>
                                        </font>
                                    </td>
                                    <td width="auto" style="width:280px; font-size:16px; font-family:Arial, Helvetica
                                        <font color="#484848" size="2" face="Arial, Helvetica, sans-serif"><b>${ctx.m
                                    </td>
                                    <td width="auto" style="font-size:16px; font-family:Arial, Helvetica, sans-serif;
                                        <font color="#484848" size="2" face="Arial, Helvetica, sans-serif"><b>${ctx.m
                                    </td>
                                    #foreach( $orderPrice in ${ctx.order.orderPrices} )
                                        <td width="auto" style="font-size:16px; font-family:Arial, Helvetica, sans-se
                                            <font color="#484848" size="2" face="Arial, Helvetica, sans-serif"><b>#if
                                        </td>
                                    #end
                               </tr>
                            #end
                            #set ($displayBackground="")
                            #if($velocityCount%2 eq 0)
                                #set ($displayBackground="background-color:#f8f8f7;")
                            #end
                            <tr>
                                <td valign="top" style="padding:0 20px 25px 0; $displayBackground">
                                    <p>
                                        <font color="#484848" size="2" face="Arial, Helvetica, sans-serif">
                                            <b>
                                                #if($entry.bundleTemplate)
                                                    $entry.bundleTemplate.name
                                                #else
                                                    &nbsp;
                                                #end
```

File Name	Source Path

```
            </b>
        </font>
    </p>
    <a href="${ctx.baseUrl}$entry.product.url">
        #foreach($image in $entry.product.images)
            #if($image.imageType == "PRIMARY" && $image.format == "thumbnail" )
                <img src="${ctx.mediaBaseUrl}$image.url" alt="" title="$entry.product
            #end
        #end
    </a>
</td>
<td valign="middle" style="padding:5px 20px 5px 0; $displayBackground">
    <table width="100%" border="0" cellpadding="0" cellspacing="0" >
        <tr>
            <td valign="top">
                <font color="#484848" size="2" face="Arial, Helvetica, sans-serif">
                    <a href="${ctx.baseUrl}$entry.product.url" style="display:block;
                            <b>$entry.product.name</b>
                    </a>
                </font>
                #if ($!entry.entryMessage)
                    <font color="#484848" size="1" face="Arial, Helvetica, sans-serif
                        ${entry.entryMessage}
                    </font>
                #end
                <table cellpadding="0" cellspacing="0" >
                    #if (!$entry.product.baseOptions.isEmpty())
                        #foreach ($option in $entry.product.baseOptions)
                            #if ($option.selected && ($option.selected.url == $entry.
                                <tr>
                                    <td>
                                        <table width="100%" cellpadding="0" cellspaci
                                            #foreach ($selectedOption in $option.sele
                                                <tr>
                                                    <td width="30%" style="padding:0
                                                    <td width="70%" style="padding:0
                                                </tr>
                                            #end
                                        </table>
                                    </td>
                                </tr>
                            #end
                        #end
                    #end
                    #foreach( $orderPrice in $ctx.order.orderPrices )
                        <tr>
                           #foreach( $promotion in $orderPrice.appliedProductPromotio
                               <td valign="top" style="padding:0 0 5px 0">
                                  #set ($displayed = false)
                                  #foreach ($consumedEntry in $promotion.consumedEntr
                                     #if (!$displayed && ($consumedEntry.orderEntryNu
                                        #set ($displayed = true)
                                        <font color="#414a4f" size="2" face="Arial, H
                                            #if($orderPrice.billingTime.nameInOrder)
                                                <b>$orderPrice.billingTime.nameInOrde
                                            #else
                                                <b>$orderPrice.billingTime.name:</b>
                                            #end
                                            ${promotion.description}
                                        </font>
                                     #end
                                  #end
                               </td>
                           #end
                        </tr>
                    #end
                </table>
            </td>
        </tr>
    </table>
</td>
<td valign="middle" style="padding:5px 20px 5px 0; $displayBackground">
    <p><font color="#484848" size="2" face="Arial, Helvetica, sans-serif">$entry.quan
</td>
#foreach($orderEntryPrice in $entry.orderEntryPrices)
    <td valign="middle" style="padding:5px 20px 5px 0; $displayBackground">
        <p><font color="#414a4f" size="2" face="Arial, Helvetica, sans-serif">
                #if($orderEntryPrice.basePrice.value - $orderEntryPrice.totalPrice.va
                    <s><del>$orderEntryPrice.basePrice.formattedValue</del></s>
                    <br>
                #end
                #if($orderEntryPrice.totalPrice.value)
                    <b>$orderEntryPrice.totalPrice.formattedValue</b>
                #end
            </font>
        </p>
```

File Name	Source Path

```
                            </td>
                        #end
                    </tr>
                #end
            </table>
        </td>
    </tr>
</table>
<br/>
<table width="100%" border="0" align="center" cellpadding="0" cellspacing="0" style="padding-bottom:87px">
    <tr>
        <td>
            <table cellpadding="0" cellspacing="0">
                <tr>
                    <td width="40%" style="border-bottom:2px solid #ffbc0e; color:#484848; font-weight:700; l
                        <font color="#484848" size="2" face="Arial, Helvetica, sans-serif">
                            ${ctx.messages.deliveryMethod}
                        </font>
                    </td>
                    <td width="10%" style="border-bottom:2px solid #ffbc0e; color:#484848; font-weight:700; l
                        &nbsp;
                    </td>
                    <td width="50%" style="border-bottom:2px solid #ffbc0e; color:#484848; font-weight:700; l
                        <font color="#484848" size="2" face="Arial, Helvetica, sans-serif">
                            ${ctx.messages.deliveryAddress}
                        </font>
                    </td>
                </tr>
                <tr>
                    <td>&nbsp;</td>
                    <td>&nbsp;</td>
                    <td>&nbsp;</td>
                </tr>
                <tr>
                    <td style="padding-left:5px" valign="top">
                        #if(${ctx.order.deliveryCost.value} > 0)
                        <p><font color="#484848" size="2" face="Arial, Helvetica, sans-serif">${ctx.order.del
                        #else
                        <p><font color="#484848" size="2" face="Arial, Helvetica, sans-serif">${ctx.messages.
                        #end
                        <p><font color="#484848" size="2" face="Arial, Helvetica, sans-serif">${ctx.messages.
                    </td>
                    <td>&nbsp;</td>
                    <td valign="top">
                        <p>
                            <font color="#484848" size="2" face="Arial, Helvetica, sans-serif">
                                ${ctx.order.deliveryAddress.title} ${ctx.order.deliveryAddress.firstName} ${c
                                ${ctx.order.deliveryAddress.line1} ${ctx.order.deliveryAddress.line2}<br/>
                                ${ctx.order.deliveryAddress.town}<br/>
                                ${ctx.order.deliveryAddress.postalCode}<br/>
                                ${ctx.order.deliveryAddress.country.name}
                            </font>
                        </p>
                    </td>
                </tr>
            </table>
        </td>
    </tr>
</table>
<br/>
<table width="100%" border="0" align="center" cellpadding="0" cellspacing="0" bordercolor="#bfc1c0" style="pa
    <tr>
        <td valign="top">
            <table width="100%" cellpadding="0" cellspacing="0">
                <tr>
                    <td width="30%" style="border-bottom:2px solid #ffbc0e; color:#484848; font-weight:700; l
                        <font color="#484848" size="2" face="Arial, Helvetica, sans-serif">
                            ${ctx.messages.paymentDetails}
                        </font>
                    </td>
                    <td width="20%" style="border-bottom:2px solid #ffbc0e; color:#484848; font-weight:700; l
                        &nbsp;
                    </td>
                    <td width="50%" style="border-bottom:2px solid #ffbc0e; color:#484848; font-weight:700; l
                        <font color="#484848" size="2" face="Arial, Helvetica, sans-serif">
                            ${ctx.messages.orderTotals}
                        </font>
                    </td>
                </tr>
                <tr>
                    <td>&nbsp;</td>
                    <td>&nbsp;</td>
                    <td>&nbsp;</td>
                </tr>
                <tr>
                    <td valign="top" style="padding-left:5px">
```

File Name	Source Path

```
                        #if(${ctx.order.paymentInfo})
                            <p><font color="#484848" size="2" face="Arial, Helvetica, sans-serif">${ctx.messa
                            <p><font color="#484848" size="2" face="Arial, Helvetica, sans-serif">
                                <b>${ctx.messages.billingAddress}:</b><br/>
                                ${ctx.order.paymentInfo.billingAddress.firstName} ${ctx.order.paymentInfo.bil
                                ${ctx.order.paymentInfo.billingAddress.line1} #evaluate(${ctx.order.paymentIn
                                ${ctx.order.paymentInfo.billingAddress.town}<br/>
                                ${ctx.order.paymentInfo.billingAddress.postalCode}<br/>
                                ${ctx.order.paymentInfo.billingAddress.country.name}
                            </font></p>
                        #end
                    </td>
                    <td>&nbsp;</td>
                    <td valign="top">
                        <table width="100%" cellpadding="0" cellspacing="0" >
                            <tr>
                                <td valign="top"><p><font color="#444444" size="2" face="Arial, Helvetica, sa
                                #foreach( $orderPrice in ${ctx.order.orderPrices} )
                                    <td valign="top"><p><font color="#444444" size="2" face="Arial, Helvetica
                                #end
                            </tr>
                            <tr>
                                <td valign="top"><p><font color="#444444" size="2" face="Arial, Helvetica, sa
                                #foreach( $orderPrice in ${ctx.order.orderPrices} )
                                    <td valign="top"><p><font color="#444444" size="2" face="Arial, Helvetica
                                #end
                            </tr>
                            <tr>
                                <td valign="top"><p><font color="#444444" size="2" face="Arial, Helvetica, sa
                                #foreach( $orderPrice in ${ctx.order.orderPrices} )
                                    <td valign="top"><p><font color="#444444" size="2" face="Arial, Helvetica
                                #end
                            </tr>
                            #if ( !${ctx.order.net} )
                                <tr>
                                    <td valign="top"><p><font color="#444444" size="2" face="Arial, Helvetica
                                    #foreach( $orderPrice in ${ctx.order.orderPrices} )
                                        <td valign="top"><p><font color="#444444" size="2" face="Arial, Helve
                                    #end
                                </tr>
                            #end
                            <tr>
                                <td valign="top" style="border-top:1px solid #000"><p><font color="#444444" s
                                #foreach( $orderPrice in ${ctx.order.orderPrices} )
                                    <td valign="top" style="border-top:1px solid #000"><p><font color="#44444
                                #end
                            </tr>
                            #if ( ${ctx.order.net} )
                                <tr>
                                    <td valign="top"><p><font color="#444444" size="1" face="Arial, Helvetica
                                    #foreach( $orderPrice in ${ctx.order.orderPrices} )
                                        <td valign="top"><p><font color="#444444" size="1" face="Arial, Helve
                                    #end
                                </tr>
                            #end
                            <tr>
                                <td valign="top" width="auto" style="background-color:#000; color:#fff; paddi
                                #foreach( $orderPrice in ${ctx.order.orderPrices} )
                                    <td valign="top" width="auto" style="background-color:#000; color:#fff; p
                                #end
                            </tr>
                        </table>
                    </td>
                </tr>
            </table>
        </td>
    </tr>
</table>
#set ($accountUrl = "${ctx.baseUrl}/my-account/orders")
#if ( $ctx.order.guestCustomer )
    #set ($deliveryInfoUrl = "${ctx.baseUrl}/guest/order/${ctx.order.guid}")
#else
    #set ($deliveryInfoUrl = "${ctx.baseUrl}/my-account/order/${ctx.order.code}")
#end
#set ($mailToUrl = "mailto:${ctx.messages.contactUsEmailAddress}")
#if(${ctx.baseSite.Uid} == "telco")
#set ( $paragraphContactUs = ${ctx.messages.getMessage('paragraphContactUs_telco', "#genHtmlLinkStartTag(${ct
#else
#set ($faqPage = "${ctx.baseUrl}/faq")
#set ( $paragraphContactUs = ${ctx.messages.getMessage('paragraphContactUs', "#genHtmlLinkStartTag($faqPage)"
#end
<p><font color="#484848" size="2" face="Arial, Helvetica, sans-serif">${ctx.messages.getMessage('paragraphAcc
<p><font color="#484848" size="2" face="Arial, Helvetica, sans-serif">${ctx.messages.paragraphCreateAccount}<
<p><font color="#484848" size="2" face="Arial, Helvetica, sans-serif">${ctx.messages.getMessage('paragraphDel
<p><font color="#484848" size="2" face="Arial, Helvetica, sans-serif">$paragraphContactUs</font></p>
<p><font color="#484848" size="2" face="Arial, Helvetica, sans-serif">${ctx.messages.complimentaryClosing}</f
```

File Name	Source Path

```
                                            <p><font color="#484848" size="2" face="Arial, Helvetica, sans-serif">${ctx.messages.signature}</font></p>
                                        </td>
                                    </tr>
                                    <tr>
                                        <td>&nbsp;</td>
                                    </tr>
                                    <tr>
                                        <td align="center" valign="middle">
                                            <a href="${ctx.baseUrl}" style="display:block; margin-top:10px;margin-bottom:10px;">${ctx.cmsSlotContents.Bot
                                        </td>
                                    </tr>
                                    <tr>
                                        <td height="30" align="right" valign="middle" bgcolor="#000000" style="padding-right:10px">
                                            <font color="#FFFFFF" size="2" face="Arial, Helvetica, sans-serif"><a href="${ctx.baseUrl}"><font color="#FFF
                                        </td>
                                    </tr>
                                    <tr>
                                        <td>&nbsp;</td>
                                    </tr>
                                </table>
                            </td>
                        </tr>
                    </table>
                </td>
            </tr>
            <tr>
                <td>&nbsp;</td>
            </tr>
        </table>
    </body>
    </html>
```

Email Velocity Templates

```
/b2ctelcostore/resources/b2ctelcostore/import/emails/email-orderConfirmationSubject.vm

 ${ctx.messages.getMessage('emailSubject', ${ctx.order.code})}
```

Email Locale Messages (one per supported language)

```
/b2ctelcostore/resources/b2ctelcostore/messages/email-orderConfirmation_en.properties

 ## Please note that the syntax of the property name should be as follows:
 ## propertyName=some string
 ## Please DO NOT USE, property.name=some string
 emailSubject=Order Confirmation {0}
 myAccount=My Account
 storeFinder=Store Finder
 salutation=Dear {0} {1}
 thankYouForOrder=Thank you for your order. Your order number is {0} and the total cost was {1}.
 quantity=Quantity
 itemPrice=Item Price
 total=Total
 free=FREE
 deliveryMethod=Delivery Method
 deliveryAddress=Delivery Address
 freeDelivery=Free Delivery
 deliveryETA=Your delivery should arrive in {0} following dispatch
 paymentDetails=Payment Details
 orderTotals=Order Totals
 paymentCardType=Paid by {0} card
 billingAddress=Billing Address
 subtotal=Subtotal
 savings=Savings
 delivery=Delivery
 taxes=Tax
 taxesInc=Tax Inc.
 includesTax=Your order includes {0} tax
 paragraphAccountLocation=If you've registered an account with us, the details of your order can also be found in your {0}Order History.{1}
 paragraphCreateAccount=If you do not have an account, why not create an account today? You will check out faster next time by using saved delivery ad
 paragraphDeliveryInfo=Please see the {0}Delivery Information{1} for further details about receiving your order.
 paragraphContactUs=If we can help you with any enquiry, please check our {0}FAQ page{1}, use our {2}Contact Us{3} page, or contact our customer servi
 paragraphContactUs_telco=If we can help you with any enquiry, please use our {0}Contact Us{1} page, or contact our customer services team directly vi
 complimentaryClosing=Many Thanks
 signature=Customer Services
 help=Help
 contactUs=Contact Us
 contactUsPage=http://www.hybris.com/hybris/en/Contact.html
 contactUsEmailAddress=customerservices@hybris.com
 termsAndCondition=Terms &amp; Conditions
 pickupInStore=Pick Up in Store
 pickupItems={0} Items for Pick Up in Store
 pickupPoints={0} Store Pick Up Destination(s)
 noPickup=No Pick Up required for this Order
 deliveryOptions=Delivery Options
 noDelivery=No Delivery Required for this Order
 bundleNo=Bundle No
 productsAndAccessories=Products &amp; Accessories
 productName=Product Name
```

File Name	Source Path
Email Context Class	`/b2ctelcostore/src/de/hybris/platform/b2ctelcostore/process/email/context/OrderNotificationEmailContext.java`
Email CMS ImpEx file	`/b2ctelcostore/resources/b2ctelcostore/import/sampledata/contentCatalogs/b2ctelcoContentCatalog/email-content.impex`

Customization of ImpEx Files

Defining the Email Template using ImpEx

You must define both the generic email content and the localized email content. We recommend using separate files: one generic, then one for each locale.

File Name: *<${dataModule}>*`/resources/`*<${dataModule}>*`/impex/sampledata/contentCatalogs/`*<${projectName}>*`ContentCatalog/email-content.impex`

Initial Value

```
#
# Import the CMS content for the B2C Telco site emails.
# For clarity (to avoid confusion with the ImpEx syntax) the Variable Values (e.g., "b2ctelco") have been used in this example,
# not the Variable Names (e.g., "${projectName}").
# Replace them as required in your project.
#
$contentCatalog=b2ctelcoContentCatalog
$contentCV=catalogVersion(CatalogVersion.catalog(Catalog.id[default=$contentCatalog]),CatalogVersion.version[default=Staged])[default=$contentCatalog:Staged]
$wideContent=CMSImageComponent,BannerComponent
# Import modulegen config properties into impex macros
UPDATE GenericItem[processor=de.hybris.platform.commerceservices.impex.impl.ConfigPropertyImportProcessor];pk[unique=true]
$jarResourceCms=$config-jarResourceCmsValue
$emailPackageName=$config-emailContextPackageName
$jarProjectResourceCms=jar:de.hybris.platform.b2ctelcostore.setup.B2cTelcoStoreSystemSetup&/b2ctelcostore/import/cockpits/cmscockpit/acc

# Email page Template
INSERT_UPDATE EmailPageTemplate;$contentCV[unique=true];uid[unique=true];name;active;frontendTemplateName;subject(code);htmlTemplate(code);restrictedPageTypes(
;;OrderConfirmationEmailTemplate;Order Confirmation Email Template;true;orderConfirmationEmail;b2ctelco_Email_Order_Confirmation_Subject;b2ctelco_Email_Order_C

# Templates for CMS Cockpit Page Edit
UPDATE EmailPageTemplate;$contentCV[unique=true];uid[unique=true];velocityTemplate[translator=de.hybris.platform.commerceservices.impex.impl.FileLoaderValueTra
;;OrderConfirmationEmailTemplate;$jarProjectResourceCms/structure-view/structure_orderConfirmationEmailTemplate.vm

INSERT_UPDATE ContentSlotName;name[unique=true];template(uid,$contentCV)[unique=true][default='OrderConfirmationEmailTemplate'];validComponentTypes(code)
;SiteLogo;;;logo
;TopContent;;$wideContent;
;BottomContent;;$wideContent;

# Create Content Slots
INSERT_UPDATE ContentSlot;$contentCV[unique=true];uid[unique=true];name;active
;;EmailTopSlot;Default Email Top Slot;true
;;EmailBottomSlot;Default Email Bottom Slot;true
;;EmailSiteLogoSlot;Default Email Site Slot;true

INSERT_UPDATE ContentSlotForTemplate;$contentCV[unique=true];uid[unique=true];position[unique=true];pageTemplate(uid,$contentCV)[unique=true][default='OrderCon
;;SiteLogo-OrderConfirmationEmail;SiteLogo;;EmailSiteLogoSlot;true
;;TopContent-OrderConfirmationEmail;TopContent;;EmailTopSlot;true
;;BottomContent-OrderConfirmationEmail;BottomContent;;EmailBottomSlot;true

# Email Pages
INSERT_UPDATE EmailPage;$contentCV[unique=true];uid[unique=true];name;masterTemplate(uid,$contentCV);defaultPage;approvalStatus(code)[default='approved']
;;OrderConfirmationEmail;Order Confirmation Email;OrderConfirmationEmailTemplate;true;

# Email velocity templates (b2ctelco customized)
INSERT_UPDATE RendererTemplate;code[unique=true];contextClass;rendererType(code)[default='velocity']
;b2ctelco_Email_Order_Confirmation_Body;de.hybris.platform.b2ctelcostore.process.email.context.OrderNotificationEmailContext
;b2ctelco_Email_Order_Confirmation_Subject;de.hybris.platform.b2ctelcostore.process.email.context.OrderNotificationEmailContext

# Assumes that the EmailPageModel_preview image is already defined:
UPDATE EmailPage;$contentCV[unique=true];uid[unique=true];previewImage(code, $contentCV)
;;OrderConfirmationEmail;EmailPageModel_preview
```

File Name: *<${dataModule}>* `/resources/` *<${dataModule}>* `/impex/sampledata/contentCatalogs/` *<${projectName}>* `ContentCatalog/email-content_en.impex`

Initial Value

```
#
# Import the CMS content for the B2C Telco site emails
#
$contentCatalog=b2ctelcoContentCatalog
$contentCV=catalogVersion(CatalogVersion.catalog(Catalog.id[default=$contentCatalog]),CatalogVersion.version[default=Staged])[default=$contentCatalog:Staged]
# Import config properties into impex macros for modulegen
```

Initial Value

```
UPDATE GenericItem[processor=de.hybris.platform.commerceservices.impex.impl.ConfigPropertyImportProcessor];pk[unique=true]
$emailResource=$config-emailResourceValue
# Language
$lang=en

# Email Pages
UPDATE EmailPage;$contentCV[unique=true];uid[unique=true];fromEmail[lang=$lang];fromName[lang=$lang]
;;OrderConfirmationEmail;"customerservices@hybris.com";"Customer Services Team"

# CMS components and Email velocity templates
UPDATE RendererTemplate;code[unique=true];description[lang=$lang];templateScript[lang=$lang,translator=de.hybris.platform.commerceservices.impex.impl.FileLoade
;b2ctelco_Email_Order_Confirmation_Body;"Order Confirmation Email Body";$emailResource/email-orderConfirmationBody.vm
;b2ctelco_Email_Order_Confirmation_Subject;"Order Confirmation Email Subject";$emailResource/email-orderConfirmationSubject.vm
```

Customizing the Bundle Guided Selling Pages

This document provides an example for how to customize the Bundle Guided Selling pages.

Pages, Tags, and Templates

Variables

The following lists all the variables with its corresponding description and value.

Variable Name	Description	Value
<${addOnName}>	The name of the AddOn	`b2ctelcostorefront`
<${addOnPageRoot}>	The location where custom page fragments must go	`b2ctelcostorefront/acceleratoraddon/web/webroot/WEB-INF`
<${targetRoot}>	The location where source files should be copied	`yacceleratorstorefront/web/webroot/WEB-INF`

Files to Copy

The following lists the files needed to be copied into a target path. The table also shows where the file can be found.

File Name	Source Path	Target Patl
Controller	*<${addOnName}>*/acceleratoraddon/web/src/de/hybris/platform/b2ctelcostorefront/controllers/pages/TelcoGuidedSellingController.java	yacceler
Styling	*<${addOnName}>*/acceleratoraddon/web/webroot/_ui/desktop/common/css/telco_guidedSelling.css	
JavaScript	*<${addOnName}>*/acceleratoraddon/web/webroot/_ui/desktop/common/js/acc.guidedSelling.js	
Localization	*<${addOnPageRoot}>*/messages/base.properties	
Tags	*<${addOnPageRoot}>*/tags/desktop/guidedselling/dashboard.tag	*<${targe*
Tags	*<${addOnPageRoot}>*/tags/desktop/guidedselling/dashboardEmpty.tag	*<${targe*
Tags	*<${addOnPageRoot}>*/tags/desktop/guidedselling/editComponentAccordeonStyle.tag	*<${targe*
Tags	*<${addOnPageRoot}>*/tags/desktop/guidedselling/viewAllServicePlans.tag	*<${targe*
Tags	*<${addOnPageRoot}>*/tags/desktop/product/productListerGridItemForGuidedSelling.tag	*<${targe*
JSP	*<${addOnPageRoot}>*/views/desktop/pages/guidedselling/editComponentAccordeonStylePage.jsp	*<${targe*
JSP	*<${addOnPageRoot}>*/views/desktop/pages/guidedselling/editComponentSolrStylePage.jsp	*<${targe*
JSP	*<${addOnPageRoot}>*/views/desktop/pages/guidedselling/viewAllServicePlansPage.jsp	*<${targe*
JSP	*<${addOnPageRoot}>*/views/desktop/pages/product/deviceLayoutPage.jsp	*<${targe*
JSP	*<${addOnPageRoot}>*/views/desktop/pages/product/serviceAddOnLayoutPage.jsp	*<${targe*
JSP	*<${addOnPageRoot}>*/views/desktop/pages/product/servicePlanLayoutPage.jsp	*<${targe*

Customization of JSP Files

The changes are mostly related to adding new `taglibs` and using these tags. The following shows an example JSP file:

File Name: `${addOnPageRoot}/views/desktop/pages/product/deviceLayoutPage.jsp`

Initial Value	Customization Possibilities

`[...]`
`<%@ taglib prefix="telcoProduct" tagdir="/WEB-INF/tags/addons/b2ctelcostorefront/desktop/product" %>`
`[...]`
`<%@ taglib prefix="guidedselling" tagdir="/WEB-INF/tags/addons/b2ctelcostorefront/desktop/guidedselling" %>`
`[...]`
	Added new `taglibs` and new sections for the Guided Selling dashboard and b2ctelco-specific device details panel in the `deviceLayoutPage`.

Initial Value	Customization Possibilities

`<guidedselling:dashboardEmpty type="DeviceModel"/>`
`<telcoProduct:deviceProductDetailsPanel product="${product}"`
`   galleryImages="${galleryImages}" />`
`[...]`	

Customization of Tag Files

In this example, there is no specific customization necessary. For details, refer to the respective tag files mentioned above.

Customization of ImpEx Files

Defining the New Page Template using ImpEx

File Name: `b2ctelcostore/resources/b2ctelcostore/import/coredata/contentCatalogs/b2ctelcoContentCatalog/cms-content.impex`

Initial Value

```
$contentCatalog=b2ctelcoContentCatalog
[...]
$addonExtensionName=b2ctelcostorefront

# Create PageTemplates
# These define the layout for pages
# "FrontendTemplateName" is used to define the JSP that should be used to render the page for pages with multiple possible layouts.
# "RestrictedPageTypes" is used to restrict templates to page types
INSERT_UPDATE PageTemplate;$contentCV[unique=true];uid[unique=true];name;frontendTemplateName;restrictedPageTypes(code);active[default=true]
[...]
;;BundleSelectionAddOnPageTemplate;Extras Page Template;addon:/$addonExtensionName/pages/guidedselling/editComponentAccordeonStylePage;ContentPage;false;
;;BundleSelectionDevicePageTemplate;Device Page Template;addon:/$addonExtensionName/pages/guidedselling/editComponentSolrStylePage;ContentPage;false;
;;BundleSelectionPlanPageTemplate;Plans Page Template;;ContentPage;false;

# Add Velocity templates that are in the CMS Cockpit. These give a better layout for editing pages
# The FileLoaderValueTranslator loads a File into a String property. The templates could also be inserted in-line in this file.
UPDATE PageTemplate;$contentCV[unique=true];uid[unique=true];velocityTemplate[translator=de.hybris.platform.commerceservices.impex.impl.FileLoaderValueTranslat
[...]
;;BundleSelectionAddOnPageTemplate         ;$jarResourceCmsTelco/structure-view/structure_bundleSelectionPageTemplate.vm
;;BundleSelectionPlanPageTemplate         ;$jarResourceCmsTelco/structure-view/structure_bundleSelectionPageTemplate.vm

[...]

INSERT_UPDATE ContentSlotName;name[unique=true];template(uid,$contentCV)[unique=true][default='BundleSelectionPlanPageTemplate'];validComponentTypes(code)
;SiteLogo;;CMSImageComponent,BannerComponent
;HeaderLinks;;CMSLinkComponent,CMSParagraphComponent
;MiniCart;;MiniCartComponent
;NavigationBar;;NavigationComponent
;Footer;;CMSLinkComponent,CMSParagraphComponent,FooterNavigationComponent

INSERT_UPDATE ContentSlotName;name[unique=true];template(uid,$contentCV)[unique=true][default='BundleSelectionAddOnPageTemplate'];validComponentTypes(code)
;SiteLogo;;CMSImageComponent,BannerComponent
;HeaderLinks;;CMSLinkComponent,CMSParagraphComponent
;MiniCart;;MiniCartComponent
;NavigationBar;;NavigationComponent
;Footer;;CMSLinkComponent,CMSParagraphComponent,FooterNavigationComponent

INSERT_UPDATE ContentSlotForTemplate;$contentCV[unique=true];uid[unique=true];position[unique=true];pageTemplate(uid,$contentCV)[unique=true][default='BundleSe
;;SiteLogo-BundleSelectionPlan;SiteLogo;;SiteLogoSlot;true
;;HomepageLink-BundleSelectionPlan;HomepageNavLink;;HomepageNavLinkSlot;true
;;MiniCart-BundleSelectionPlan;MiniCart;;MiniCartSlot;true
;;NavigationBar-BundleSelectionPlan;NavigationBar;;NavigationBarSlot;true
;;Tabs-BundleSelectionPlan;Tabs;;TabsSlot;true
;;Footer-BundleSelectionPlan;Footer;;FooterSlot;true
;;HeaderLinks-BundleSelectionPlan;HeaderLinks;;HeaderLinksSlot;true

[...]

INSERT_UPDATE ContentSlotForTemplate;$contentCV[unique=true];uid[unique=true];position[unique=true];pageTemplate(uid,$contentCV)[unique=true][default='BundleSe
;;SiteLogo-BundleSelectionAddOn;SiteLogo;;SiteLogoSlot;true
;;HomepageLink-BundleSelectionAddOn;HomepageNavLink;;HomepageNavLinkSlot;true
;;NavigationBar-BundleSelectionAddOn;NavigationBar;;NavigationBarSlot;true
;;MiniCart-BundleSelectionAddOn;MiniCart;;MiniCartSlot;true
;;Footer-BundleSelectionAddOn;Footer;;FooterSlot;true
;;HeaderLinks-BundleSelectionAddOn;HeaderLinks;;HeaderLinksSlot;true
;;SearchBox-BundleSelectionAddOn;SearchBox;;SearchBoxSlot;true
;;TopHeaderSlot-BundleSelectionAddOn;TopHeaderSlot;;TopHeaderSlot;true
;;BottomHeaderSlot-BundleSelectionAddOn;BottomHeaderSlot;;BottomHeaderSlot;true
;;PlaceholderContentSlot-BundleSelectionAddOn;PlaceholderContentSlot;;PlaceholderContentSlot;true

INSERT_UPDATE ContentSlotForTemplate;$contentCV[unique=true];uid[unique=true];position[unique=true];pageTemplate(uid,$contentCV)[unique=true][default='BundleSe
;;SiteLogo-BundleSelectionPlan;SiteLogo;;SiteLogoSlot;true
;;HomepageLink-BundleSelectionPlan;HomepageNavLink;;HomepageNavLinkSlot;true
;;NavigationBar-BundleSelectionPlan;NavigationBar;;NavigationBarSlot;true
;;MiniCart-BundleSelectionPlan;MiniCart;;MiniCartSlot;true
```

Initial Value

```
;;Footer-BundleSelectionPlan;Footer;;FooterSlot;true
;;HeaderLinks-BundleSelectionPlan;HeaderLinks;;HeaderLinksSlot;true
;;SearchBox-BundleSelectionPlan;SearchBox;;SearchBoxSlot;true
;;TopHeaderSlot-BundleSelectionPlan;TopHeaderSlot;;TopHeaderSlot;true
;;BottomHeaderSlot-BundleSelectionPlan;BottomHeaderSlot;;BottomHeaderSlot;true
;;PlaceholderContentSlot-BundleSelectionPlan;PlaceholderContentSlot;;PlaceholderContentSlot;true

# Content slots for template
INSERT_UPDATE ContentSlotName;name[unique=true];template(uid,$contentCV)[unique=true][default='SBGProductsLandingPageTemplate'];validComponentTypes(code);compT
;SiteLogo;;CMSImageComponent,BannerComponent;
;HeaderLinks;;CMSLinkComponent,CMSParagraphComponent;
;SearchBox;;;searchbox
;MiniCart;;MiniCartComponent;
;NavigationBar;;NavigationComponent;
;Section1;;$wideContent;
;Section2;;$wideContent;
;Section3;;$narrowContent;
;Section4;;$narrowContent;
;Footer;;CMSLinkComponent,CMSParagraphComponent,FooterNavigationComponent;
;TopHeaderSlot;;;wide

INSERT_UPDATE ContentSlotForTemplate;$contentCV[unique=true];uid[unique=true];position[unique=true];pageTemplate(uid,$contentCV)[unique=true][default='SBGProdu
;;TopHeaderSlot-SBGProductsLandingPage;TopHeaderSlot;;TopHeaderSlot;true
;;SiteLogo-SBGProductsLandingPage;SiteLogo;;SiteLogoSlot;true
;;HomepageLink-SBGProductsLandingPage;HomepageNavLink;;HomepageNavLinkSlot;true
;;SearchBox-SBGProductsLandingPage;SearchBox;;SearchBoxSlot;true
;;MiniCart-SBGProductsLandingPage;MiniCart;;MiniCartSlot;true
;;NavigationBar-SBGProductsLandingPage;NavigationBar;;NavigationBarSlot;true
;;Footer-SBGProductsLandingPage;Footer;;FooterSlot;true
;;HeaderLinks-SBGProductsLandingPage;HeaderLinks;;HeaderLinksSlot;true

INSERT_UPDATE ContentSlotForTemplate;$contentCV[unique=true];uid[unique=true];position[unique=true];pageTemplate(uid,$contentCV)[unique=true][default='PlanDeta
;;SiteLogo-PlanDetails;SiteLogo;;SiteLogoSlot;true
;;HomepageLink-PlanDetails;HomepageNavLink;;HomepageNavLinkSlot;true
;;MiniCart-PlanDetails;MiniCart;;MiniCartSlot;true
;;NavigationBar-PlanDetails;NavigationBar;;NavigationBarSlot;true
;;Tabs-PlanDetails;Tabs;;TabsSlot;true
;;Footer-PlanDetails;Footer;;FooterSlot;true
;;HeaderLinks-PlanDetails;HeaderLinks;;HeaderLinksSlot;true

INSERT_UPDATE ContentSlotForTemplate;$contentCV[unique=true];uid[unique=true];position[unique=true];pageTemplate(uid,$contentCV)[unique=true][default='AddOnDet
;;TopHeaderSlot-AddOnDetails;TopHeaderSlot;;TopHeaderSlot;true
;;SiteLogo-AddOnDetails;SiteLogo;;SiteLogoSlot;true
;;HomepageLink-AddOnDetails;HomepageNavLink;;HomepageNavLinkSlot;true
;;MiniCart-AddOnDetails;MiniCart;;MiniCartSlot;true
;;NavigationBar-AddOnDetails;NavigationBar;;NavigationBarSlot;true
;;Tabs-AddOnDetails;Tabs;;TabsSlot;true
;;Footer-AddOnDetails;Footer;;FooterSlot;true
;;HeaderLinks-AddOnDetails;HeaderLinks;;HeaderLinksSlot;true

[...]
```

Localization of Pages using ImpEx

File Name: `b2ctelcostore/resources/b2ctelcostore/import/coredata/contentCatalogs/b2ctelcoContentCatalog/cms-content_en.impex`

Initial Value

```
$contentCatalog=b2ctelcoContentCatalog
$contentCV=catalogVersion(CatalogVersion.catalog(Catalog.id[default=$contentCatalog]),CatalogVersion.version[default=Staged])[default=$contentCatalog:Staged]
# Language
$lang=en
# Functional Content Pages
INSERT_UPDATE ContentPage;$contentCV[unique=true];uid[unique=true]                              ;name                           ;masterTemplate(uid,$contentCV)
                                                 ;                                               ;guidedselling-select-device    ;Select Device Page     ;Bundle
# Content Pages
UPDATE ContentPage;$contentCV[unique=true];uid[unique=true];title[lang=$lang]
[...]
 ;;bundleselection-plan;"Service Plan Selection"
 ;;bundleselection-extra;"Service Add Ons Selection"
 ;;guidedselling-select-device;"Product Selection"

[...]
```

Setting Preview Images for Guided Selling Pages using ImpEx

File Name: `b2ctelcostore/resources/b2ctelcostore/import/sampledata/contentCatalogs/b2ctelcoContentCatalog/cms-content.impex`

Initial Value

`$contentCatalog=b2ctelcoContentCatalog`
`$contentCV=catalogVersion(CatalogVersion.catalog(Catalog.id[default=$contentCatalog]),CatalogVersion.version[default=Staged])[default=$contentCatalog:Staged]`
`[...]`
`$media=media(code, $contentCV);`
`$contentPage=contentPage(uid, $contentCV);`
`$addonExtensionName=b2ctelcostorefront`
 	
`[...]`
 	
`# Preview Image for use in the CMS Cockpit for special ContentPages`
`INSERT_UPDATE Media;$contentCV[unique=true];code[unique=true];mime;realfilename;@media[translator=de.hybris.platform.impex.jalo.media.MediaDataTranslator][forc`
`[...]`
`;;guidedSellingSelectDevicePagePreview;text/png;GuidedSellingSelectDevicePage.png;$jarResourceCms/preview-images/b2ctelco/GuidedSellingSelectDevicePage.png`
`;;guidedSellingSelectPlanPagePreview;text/png;GuidedSellingSelectPlanPage.png;$jarResourceCms/preview-images/b2ctelco/GuidedSellingSelectPlanPage.png`
 	
`# Functional Content Pages`
`UPDATE ContentPage;$contentCV[unique=true];uid[unique=true];previewImage(code, $contentCV)`
`[...]`
`;;guidedselling-select-device;guidedSellingSelectDevicePagePreview`
`;;bundleselection-plan;guidedSellingSelectPlanPagePreview`
 	
`[...]`

Controller Configuration

The special controller is created to implement the Guided Selling functionality.

Class Name	Source Path
`TelcoGuidedSellingController`	*<${addOnName}>*/acceleratoraddon/web/src/de/hybris/platform/b2ctelcostorefront/controllers/pages/TelcoGuidedSellingContro

`TelcoGuidedSellingController`

```
/*
 * [y] hybris Platform
 *
 * Copyright (c) 2000-2015 hybris AG
 * All rights reserved.
 *
 * This software is the confidential and proprietary information of hybris
 * ("Confidential Information"). You shall not disclose such Confidential
 * Information and shall use it only in accordance with the terms of the
 * license agreement you entered into with hybris.
 *
 *
 */
package de.hybris.platform.b2ctelcostorefront.controllers.pages;
import de.hybris.platform.acceleratorservices.controllers.page.PageType;
import de.hybris.platform.acceleratorstorefrontcommons.controllers.util.GlobalMessages;
import de.hybris.platform.acceleratorstorefrontcommons.forms.UpdateQuantityForm;
import de.hybris.platform.b2ctelcofacades.bundle.GuidedSellingFacade;
import de.hybris.platform.b2ctelcofacades.data.BundleTabData;
import de.hybris.platform.b2ctelcofacades.product.TelcoProductFacade;
import de.hybris.platform.b2ctelcoservices.model.DeviceModel;
import de.hybris.platform.b2ctelcoservices.model.ServiceAddOnModel;
import de.hybris.platform.b2ctelcoservices.model.ServicePlanModel;
import de.hybris.platform.b2ctelcostorefront.controllers.TelcoControllerConstants;
import de.hybris.platform.cms2.exceptions.CMSItemNotFoundException;
import de.hybris.platform.commercefacades.order.data.CartModificationData;
import de.hybris.platform.commercefacades.product.ProductOption;
import de.hybris.platform.commercefacades.product.data.ProductData;
import de.hybris.platform.commercefacades.search.data.SearchStateData;
import de.hybris.platform.commerceservices.order.CommerceCartModificationException;
import de.hybris.platform.commerceservices.search.facetdata.ProductSearchPageData;
import de.hybris.platform.commerceservices.search.pagedata.PageableData;
import de.hybris.platform.configurablebundlefacades.order.BundleCartFacade;
import de.hybris.platform.configurablebundleservices.bundle.BundleTemplateService;
import de.hybris.platform.configurablebundleservices.enums.BundleTemplateStatusEnum;
import de.hybris.platform.configurablebundleservices.model.BundleTemplateModel;
import java.util.Arrays;
import java.util.List;
import javax.annotation.Resource;
import javax.servlet.http.HttpServletRequest;
import javax.validation.Valid;
import org.apache.log4j.Logger;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.validation.BindingResult;
import org.springframework.validation.ObjectError;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.servlet.mvc.support.RedirectAttributes;

/**
 * Controller for guided selling flows like the extras page.
 */
@Controller
@RequestMapping(value = "/bundle")
public class TelcoGuidedSellingController extends AbstractSearchPageController
{
        private static final String GUIDEDSELLING_SELECT_ADDONS = "bundleselection-extra";
        private static final String GUIDEDSELLING_SELECT_DEVICE = "guidedselling-select-device";
        private static final String GUIDEDSELLING_BUNDLE_PLAN = "bundleselection-plan";
```

```
protected static final Logger LOG = Logger.getLogger(TelcoGuidedSellingController.class);
/**
 * Component navigation.
 */
public static enum ComponentNavigation
{
        CURRENT(""), NEXT("next"), PREVIOUS("prev");
        final String code;
        private ComponentNavigation(final String code)
        {
                this.code = code;
        }
}
@Resource(name = "bundleTemplateService")
private BundleTemplateService bundleTemplateService;
@Resource(name = "guidedSellingFacade")
private GuidedSellingFacade guidedSellingFacade;
@Resource(name = "cartFacade")
private BundleCartFacade cartFacade;
@Resource(name = "telcoProductFacade")
private TelcoProductFacade productFacade;
@RequestMapping(value = "/view-plans/{bundleTemplateId}", method = RequestMethod.GET)
public String viewPlans(@PathVariable("bundleTemplateId") final String bundleTemplateId, final Model model)
                throws CMSItemNotFoundException
{
        BundleTemplateModel bundleTemplateModel = bundleTemplateService.getBundleTemplateForCode(bundleTemplateId);
        if (bundleTemplateModel == null || !BundleTemplateStatusEnum.APPROVED.equals(bundleTemplateModel.getStatus().getStatus()))
        {
                if (bundleTemplateModel != null)
                {
                        bundleTemplateModel = bundleTemplateService.getAllApprovedRootBundleTemplates(bundleTemplateModel.getCatalogVersion(
                                        .iterator().next();
                }
                GlobalMessages.addErrorMessage(model, "guidedselling.viewplans.packagenotavailable");
        }
        final BundleTemplateModel firstComponentModel = bundleTemplateService
                        .getAllComponentsOfType(bundleTemplateModel, ServicePlanModel.class).iterator().next();
        final ServicePlanModel firstPlan = (ServicePlanModel) firstComponentModel.getProducts().iterator().next();
        final ProductData productData = productFacade.getProductForOptions(firstPlan, Arrays.asList(ProductOption.BASIC,
                        ProductOption.PRICE, ProductOption.SUMMARY, ProductOption.DESCRIPTION, ProductOption.GALLERY,
                        ProductOption.CATEGORIES, ProductOption.REVIEW, ProductOption.PROMOTIONS, ProductOption.CLASSIFICATION,
                        ProductOption.VARIANT_FULL, ProductOption.STOCK, ProductOption.SERVICE_PLAN_BUNDLE_TABS));
        // change selected package to the one that is passed as a parameter
        final List<BundleTabData> bundleTabs = productData.getBundleTabs();
        for (final BundleTabData bundleTab : bundleTabs)
        {
                bundleTab.setPreselected(bundleTemplateId.equals(bundleTab.getParentBundleTemplate().getId()));
        }
        model.addAttribute("bundleTabs", bundleTabs);
        storeCmsPageInModel(model, getContentPageForLabelOrId(GUIDEDSELLING_BUNDLE_PLAN));
        setUpMetaDataForContentPage(model, getContentPageForLabelOrId(GUIDEDSELLING_BUNDLE_PLAN));
        return TelcoControllerConstants.Views.Pages.GuidedSelling.ViewAllServicePlansPage;
}
/**
 * Edit component page.
 *
 * @param bundleNo
 * @param componentId
 * @param searchQuery
 * @param page
 * @param showMode
 * @param sortCode
 * @param request
 * @param model
 * @return
 * @throws CMSItemNotFoundException
 */
@RequestMapping(value = "/edit-component/{bundleNo}/component/{componentId}", method = RequestMethod.GET)
public String editComponent(@PathVariable("bundleNo") final String bundleNo,
                @PathVariable("componentId") final String componentId,
                @RequestParam(value = "q", required = false) final String searchQuery,
                @RequestParam(value = "page", defaultValue = "0") final int page,
                @RequestParam(value = "show", defaultValue = "Page") final ShowMode showMode,
                            @RequestParam(value = "sort", required = false) final String sortCode,
                            final HttpServletRequest request, final Model model)
                throws CMSItemNotFoundException
{
        return internalEditComponent(bundleNo, componentId, searchQuery, page, showMode, sortCode, request, model);
}
/**
 * Edit component page.
 *
 * @param bundleNo
 * @param componentId
 * @param searchQuery
 * @param page
 * @param showMode
 * @param sortCode
 * @param request
 * @param model
 * @param redirectAttributes
 * @return
 * @throws CMSItemNotFoundException
 */
@RequestMapping(value = "/edit-component/{bundleNo}/nextcomponent/{componentId}", method = RequestMethod.GET)
public String editNextComponent(@PathVariable("bundleNo") final String bundleNo,
                @PathVariable("componentId") final String componentId,
                @RequestParam(value = "q", required = false) final String searchQuery,
                @RequestParam(value = "page", defaultValue = "0") final int page,
                @RequestParam(value = "show", defaultValue = "Page") final ShowMode showMode,
                                @RequestParam(value = "sort", required = false) final String sortCode,
                                final HttpServletRequest request,
                                final Model model, final RedirectAttributes redirectAttributes)
                throws CMSItemNotFoundException
{
        return editPositionalComponent(bundleNo, componentId, searchQuery,
                        page, showMode, sortCode, request, model, 1, redirectAttributes);
}
/**
 * Preview component page.
 *
 * @param bundleNo
 * @param componentId
 * @param searchQuery
 * @param page
 * @param showMode
 * @param sortCode
 * @param request
 * @param model
 * @param redirectAttributes
 * @return
 * @throws CMSItemNotFoundException
 */
@RequestMapping(value = "/edit-component/{bundleNo}/prevcomponent/{componentId}", method = RequestMethod.GET)
public String editPreviousComponent(@PathVariable("bundleNo") final String bundleNo,
                @PathVariable("componentId") final String componentId,
                @RequestParam(value = "q", required = false) final String searchQuery,
                @RequestParam(value = "page", defaultValue = "0") final int page,
                @RequestParam(value = "show", defaultValue = "Page") final ShowMode showMode,
                                    @RequestParam(value = "sort", required = false) final String sortCode,
                                    final HttpServletRequest request,
                                    final Model model, final RedirectAttributes redirectAttributes)
                throws CMSItemNotFoundException
{
        return editPositionalComponent(bundleNo, componentId, searchQuery,
                        page, showMode, sortCode, request, model, -1, redirectAttributes);
}
/**
 * Generic method to edit a component at a relative position. It resolves the relative component and checks if it is
 * valid to navigate there.
 */
protected String editPositionalComponent(final String bundleNo, final String componentId, final String searchQuery,
                                         final int page, final ShowMode showMode, final String sortCode,
                                         final HttpServletRequest request, final Model model,
                                         final int relativeposition, final RedirectAttributes redirectAttributes)
                throws CMSItemNotFoundException
{
        String componentIdToNavigateTo = null;
        if (guidedSellingFacade.checkIsComponentSelectionCriteriaMet(bundleNo, componentId))
        {
                componentIdToNavigateTo = guidedSellingFacade.getRelativeComponentId(bundleNo, componentId, relativeposition);
        }
        else
        {
                GlobalMessages.addFlashMessage(redirectAttributes, GlobalMessages.ERROR_MESSAGES_HOLDER,
                                "basket.next.selection.criteria.not.met");
                componentIdToNavigateTo = componentId;
                final HttpServletRequest previousRequest = (HttpServletRequest) model.asMap().get("request");
                final String previousUrl = previousRequest.getHeader("referer");
                final int pos = previousUrl.indexOf(previousRequest.getContextPath()) + previousRequest.getContextPath().length();
                final String redirectUrl = previousUrl.substring(pos, previousUrl.length());
                return REDIRECT_PREFIX + redirectUrl;
        }
        if (componentIdToNavigateTo == null)
        {
                return REDIRECT_PREFIX + "/cart";
        }
        return internalEditComponent(bundleNo, componentIdToNavigateTo, searchQuery, page, showMode, sortCode, request, model);
}
/**
 * Populates the dashboard and the appropriate step of the guided selling flow.
 */
protected String internalEditComponent(final String bundleNo, final String componentId, final String searchQuery,
                                       final int page, final ShowMode showMode, final String sortCode,
                                       final HttpServletRequest request, final Model model)
                throws CMSItemNotFoundException
{
        model.addAttribute("dashboard", guidedSellingFacade.getDashboard(Integer.parseInt(bundleNo), componentId));
        final String productType = guidedSellingFacade.getComponentProductType(componentId);
        if (productType == null)
        {
                // we have an invalid (empty?) component here: do not try to edit it but go to cart
                return REDIRECT_PREFIX + "/cart";
        }
        model.addAttribute("productType", productType);
        switch (productType)
        {
                case DeviceModel._TYPECODE:
                case ServicePlanModel._TYPECODE:
                        final String urlPrefix = "/bundle/edit-component/" + bundleNo + "/component/" + componentId + "?q=";
                        final PageableData pageableData = createPageableData(page, getSearchPageSize(), sortCode, showMode);
                        final ProductSearchPageData<SearchStateData, ProductData> searchPageData = guidedSellingFacade.bundleSearch(
                                        pageableData, searchQuery, urlPrefix, componentId, Integer.valueOf(bundleNo));
                        model.addAttribute("urlPrefix", urlPrefix);
                        model.addAttribute("pageType", PageType.PRODUCTSEARCH);
                        model.addAttribute("componentId", componentId);
                        populateModel(model, searchPageData, showMode);
                        storeCmsPageInModel(model, getContentPageForLabelOrId(GUIDEDSELLING_SELECT_DEVICE));
                        setUpMetaDataForContentPage(model, getContentPageForLabelOrId(GUIDEDSELLING_SELECT_DEVICE));
                        return TelcoControllerConstants.Views.Pages.GuidedSelling.EditComponentSolrStylePage;
                case ServiceAddOnModel._TYPECODE:
                        model.addAttribute("bundleTemplateData", guidedSellingFacade.getComponentToEdit(bundleNo, componentId));
                        storeCmsPageInModel(model, getContentPageForLabelOrId(GUIDEDSELLING_SELECT_ADDONS));
                        setUpMetaDataForContentPage(model, getContentPageForLabelOrId(GUIDEDSELLING_SELECT_ADDONS));
                        return TelcoControllerConstants.Views.Pages.GuidedSelling.EditComponentAccordeonStylePage;
                default:
                        return null;
        }
}
/**
 * Add product to cart.
 *
 * @param code
 * @param model
 * @param form
 * @param bundleNo
 * @param bundleTemplateId
 * @param removeCurrentProducts
 * @param navigation
 * @param bindingResult
 * @param redirectModel
 * @return
 */
@RequestMapping(value = "/addEntry", method = RequestMethod.POST)
```

```
        public String addToCart(
                        @RequestParam("productCodePost") final String code,
                        final Model model,
                        @Valid final UpdateQuantityForm form,
                        @RequestParam(value = "bundleNo", required = false) final Integer bundleNo,
                        @RequestParam(value = "bundleTemplateId", required = false) final String bundleTemplateId,
                        @RequestParam(value = "removeCurrentProducts", required = false, defaultValue = "false")
                                        final boolean removeCurrentProducts,
                        @RequestParam("navigation") final ComponentNavigation navigation, final BindingResult bindingResult,
                        final RedirectAttributes redirectModel)
        {
                if (bindingResult.hasErrors())
                {
                        for (final ObjectError error : bindingResult.getAllErrors())
                        {
                                if (error.getCode().equals("typeMismatch"))
                                {
                                        GlobalMessages.addFlashMessage(redirectModel, GlobalMessages.ERROR_MESSAGES_HOLDER,
                                                        "basket.error.quantity.invalid");
                                }
                                else
                                {
                                        GlobalMessages.addFlashMessage(redirectModel, GlobalMessages.ERROR_MESSAGES_HOLDER,
                                                        error.getDefaultMessage());
                                }
                        }
                }
                int finalBundleNo = addProduct(code, model, form.getQuantity().intValue(), bundleNo, bundleTemplateId,
                                removeCurrentProducts, redirectModel);
                if (finalBundleNo < 0)
                {
                        finalBundleNo = bundleNo == null ? 0 : bundleNo.intValue();
                }
                return REDIRECT_PREFIX + "/bundle/edit-component/" + finalBundleNo + "/" + navigation.code + "component/"
                                + bundleTemplateId;
        }
        protected int addProduct(final String code, final Model model, final long qty, final Integer bundleNo,
                                 final String bundleTemplateId, final boolean removeCurrentProducts,
                                 final RedirectAttributes redirectModel)
        {
                try
                {
                        final List<CartModificationData> cartModifications = cartFacade.addToCart(code, qty, bundleNo.intValue(),
                                        bundleTemplateId, removeCurrentProducts);
                        model.addAttribute("modifiedCartData", cartModifications);
                        for (final CartModificationData cartModification : cartModifications)
                        {
                                if (cartModification.getQuantityAdded() == 0L)
                                {
                                        GlobalMessages.addFlashMessage(redirectModel, GlobalMessages.ERROR_MESSAGES_HOLDER,
                                                        "basket.information.quantity.noItemsAdded." + cartModification.getStatusCode());
                                }
                                else if (cartModification.getQuantityAdded() < qty)
                                {
                                        GlobalMessages.addFlashMessage(redirectModel, GlobalMessages.ERROR_MESSAGES_HOLDER,
                                                        "basket.information.quantity.reducedNumberOfItemsAdded." + cartModification.getStatusCode())
                                }
                                else if (cartModification.getQuantityAdded() > 0)
                                {
                                        GlobalMessages.addFlashMessage(redirectModel, GlobalMessages.CONF_MESSAGES_HOLDER,
                                                        "guidedselling.basket.page.message.added");
                                        return cartModification.getEntry().getBundleNo();
                                }
                        }
                }
                catch (final CommerceCartModificationException ex)
                {
                        GlobalMessages.addFlashMessage(redirectModel, GlobalMessages.ERROR_MESSAGES_HOLDER,
                                        "basket.error.occurred");
                        LOG.warn("Couldn't add product of code " + code + " to cart.", ex);
                }
                return -2;
        }

        /**
         * Remove product from cart.
         *
         * @param entryNumber
         * @param model
         * @param form
         * @param bundleNo
         * @param componentId
         * @param bindingResult
         * @param redirectModel
         * @return
         * @throws CMSItemNotFoundException
         */
        @RequestMapping(value = "/removeEntry", method = RequestMethod.POST)
        public String updateCartQuantities(@RequestParam("entryNumber") final long entryNumber, final Model model,
                        @Valid final UpdateQuantityForm form, @RequestParam("bundleNo") final String bundleNo,
                        @RequestParam("componentId") final String componentId, final BindingResult bindingResult,
                        final RedirectAttributes redirectModel) throws CMSItemNotFoundException
        {
                if (bindingResult.hasErrors())
                {
                        for (final ObjectError error : bindingResult.getAllErrors())
                        {
                                if (error.getCode().equals("typeMismatch"))
                                {
                                        GlobalMessages.addErrorMessage(model, "basket.error.quantity.invalid");
                                }
                                else
                                {
                                        GlobalMessages.addErrorMessage(model, error.getDefaultMessage());
                                }
                        }
                }
                else if (cartFacade.getSessionCart().getEntries() != null)
                {
                        try
                        {
                                final CartModificationData cartModification = cartFacade.updateCartEntry(entryNumber, form.getQuantity().longValue()
                                if (cartModification.getQuantity() == form.getQuantity().longValue())
                                {
                                        // Success
                                        if (cartModification.getQuantity() == 0)
                                        {
                                                // Success in removing entry
                                                GlobalMessages.addFlashMessage(redirectModel, GlobalMessages.CONF_MESSAGES_HOLDER,
                                                                "guidedselling.basket.page.message.removed");
                                        }
                                        else
                                        {
                                                // Success in update quantity
                                                GlobalMessages.addFlashMessage(redirectModel, GlobalMessages.CONF_MESSAGES_HOLDER,
                                                                "basket.page.message.update");
                                        }
                                }
                                else
                                {
                                        // Less than successful
                                        if (form.getQuantity().longValue() == 0)
                                        {
                                                // Failed to remove entry
                                                GlobalMessages.addFlashMessage(redirectModel, GlobalMessages.ERROR_MESSAGES_HOLDER,
                                                                "basket.information.quantity.reducedNumberOfItemsAdded." + cartModification.getStatu
                                        }
                                        else
                                        {
                                                // Failed to update quantity
                                                GlobalMessages.addFlashMessage(redirectModel, GlobalMessages.ERROR_MESSAGES_HOLDER,
                                                                "basket.information.quantity.reducedNumberOfItemsAdded." + cartModification.getStatu
                                        }
                                }
                                return REDIRECT_PREFIX + "/bundle/edit-component/" + bundleNo + "/component/" + componentId;
                        }
                        catch (final CommerceCartModificationException ex)
                        {
                                LOG.warn("Couldn't update product with the entry number: " + entryNumber + ".", ex);
                                return REDIRECT_PREFIX + "/bundle/edit-component/" + bundleNo + "/component/" + componentId;
                        }
                }
                return REDIRECT_PREFIX + "/cart";
        }
}
```

Spring Configuration

File Name	Source Path
b2ctelcostorefront-web-spring.xml	*<${addOnName}>*/resources/b2ctelcostorefront/web/spring/b2ctelcostorefront-web-spring.xml

Bean Changes:

b2ctelcostorefront-web-spring.xml

```
[...]
        <bean name="telcoGuidedSellingController" class="de.hybris.platform.b2ctelcostorefront.controllers.pages.TelcoGuidedSellingController"/>
[...]
```

Customizing the Product Results Page

Learn how to customize the Product Results Page.

Pages, Tags, and Templates

Variables

The following table displays the variables with its corresponding description and value:

Variable Name	Description	Value
<${addOnName}>	The name of the AddOn	b2ctelcostorefront
<${addOnPageRoot}>	The location where the custom page fragments must go	b2ctelcostorefront/acceleratoraddon/web/webroot/WEB-INF
<${addOnStoreExt}>	The name of the AddOn's store extension	b2ctelcostore

Files to Copy

The following table displays the files that you need to copy to the appropriate target path and provides the location where you can find the files:

File Name	Source Path	Target Path
Layout	/yacceleratorstorefront/web/webroot/WEB-INF/views/addons/b2ctelcostorefront/desktop/pages/search/searchResultsListPage.jsp /yacceleratorstorefront/web/webroot/WEB-INF/views/addons/b2ctelcostorefront/desktop/pages/search/searchResultsGridPage.jsp	*<${addOnPageRoot}>*/views/desktop/pages/ *<${addOnPageRoot}>*/acceleratoraddon/web INF/views/desktop/pages/search/searchRe
Page Template	/yacceleratorstorefront/web/webroot/WEB-INF/tags/desktop/product/productListerGridItem.tag	*<${addOnPageRoot}>*/tags/desktop/product

File Name	Source Path	Target Path
Some ImpEx file	/b2ctelcostore/resources/b2ctelcostore/import/sampledata/contentCatalogs/b2ctelcoContentCatalog/cms-content.impex	*<${addOnStoreExt}>*/resources/b2ctelcost content.impex

Customization of JSP Files

Changes

`searchResultsGridPage .jsp` and `searchResultsListPage.jsp` contain the `b2cTelcoProduct` tag:

```
<%@ taglib prefix="b2ctelcoProduct" tagdir="/WEB-INF/tags/addons/b2ctelcostorefront/desktop/product" %>
                                                        ...
                                                        <b2ctelcoProduct:productListerGridItem product="${product}"/>
```

Customization of Tag Files

Changes

`productListerGridIte.tag` contains list of `product.classifications`:

```
<c:if test="${not empty product.classifications}">
                                                        ...
```

And special part for bundle prices:

CB

```
<div class="grid-product-price">
                                                        ... special code for bundle prices ...
```

Customization of ImpEx Files

Defining the JSP Components in WCMS using ImpEx

File Name: `/b2ctelcostore/resources/b2ctelcostore/import/sampledata/contentCatalogs/b2ctelcoContentCatalog/cms-content.impex`

Changes

```
$contentCatalog=b2ctelcoContentCatalog
                                                        $contentCatalogName=B2C Telco Content Catalog
                                                        $contentCV=catalogVersion(CatalogVersion.catalog(Catalog.id[default=$contentCatalog]),CatalogVe
                                                        $productCatalog=b2ctelcoProductCatalog
                                                        $productCatalogName=B2C Telco Product Catalog
                                                        $productCV=catalogVersion(catalog(id[default=$productCatalog]),version[default='Staged'])[uniqu
                                                        $picture=media(code, $contentCV);
                                                        $image=image(code, $contentCV);
                                                        $media=media(code, $contentCV);
                                                        $page=page(uid, $contentCV);
                                                        $contentPage=contentPage(uid, $contentCV);
                                                        $product=product(code, $productCV)
                                                        $category=category(code, $productCV)
                                                        $siteResource=jar:de.hybris.platform.b2ctelcostore.constants.B2ctelcostoreConstants&/b2ctelcost
                                                        $productResource=jar:de.hybris.platform.b2ctelcostore.constants.B2ctelcostoreConstants&/b2ctelc
                                                        $jarResourceCms=jar:de.hybris.platform.b2ctelcostore.constants.B2ctelcostoreConstants&/b2ctelco
                                                        $addonExtensionName=b2ctelcostorefront

                                                        INSERT_UPDATE JspIncludeComponent;$contentCV[unique=true];uid[unique=true];name;page;actions(ui
                                                        ;;SearchResultsList;Search Result List Component;/WEB-INF/views/addons/$addonExtensionName/desk
                                                        ;;SearchResultsGrid;Search Result Grid Component;/WEB-INF/views/addons/$addonExtensionName/desk

                                                        #### SearchResultListPage
                                                        INSERT_UPDATE ContentSlot;$contentCV[unique=true];uid[unique=true];name;active;cmsComponents(&c
                                                        ;;SearchResultsListSlot;Search Result List Slot for Search List;true;SearchResultsList;;;

                                                        #### SearchResultGridPage
                                                        INSERT_UPDATE ContentSlot;$contentCV[unique=true];uid[unique=true];name;active;cmsComponents(&c
                                                        ;;SearchResultsGridSlot;Search Result Grid Slot for Search List;true;SearchResultsGrid;;;
```