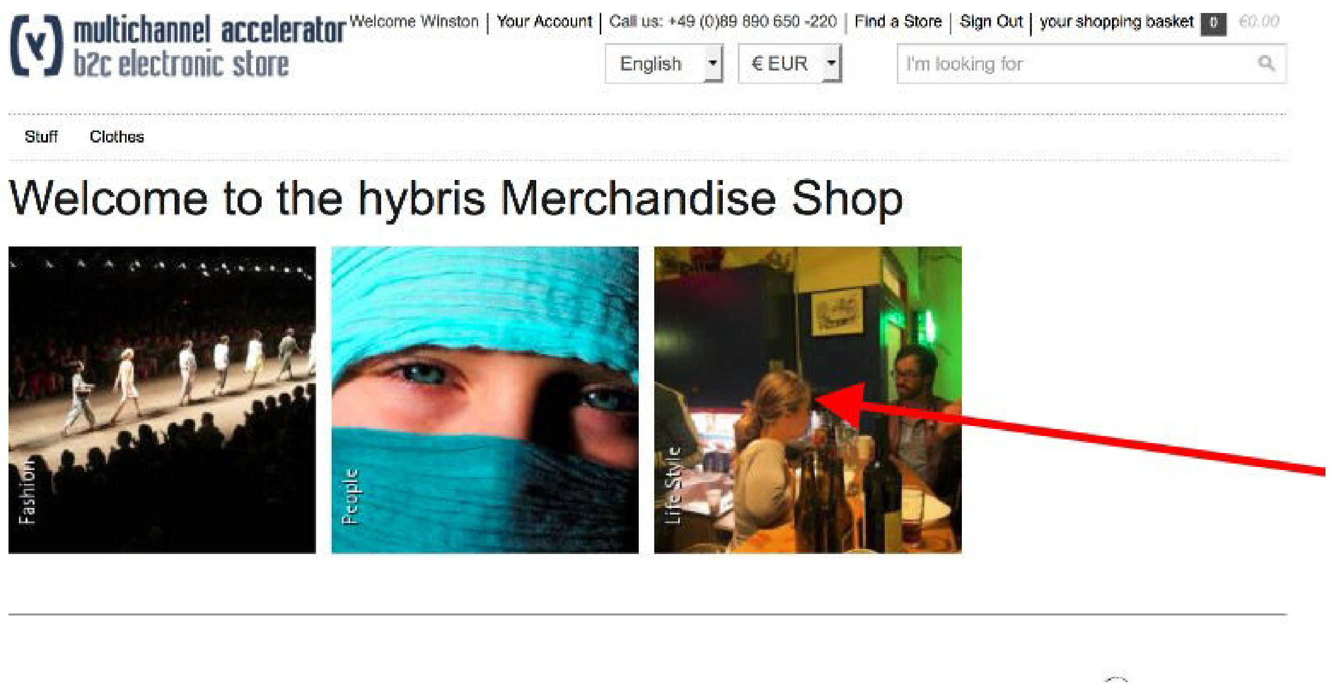
How to create New CMS Components by using AddOn extension

The advantage of this approach is that you don't have to touch the Accelerator codebase, making your code both cleaner and easier to migrate / re-use in other projects.

Let's create a new CMS component type for displaying random, text-bearing images of a specific size and topic. To simplify, we'll be using http://www.lorempixel.com as a source for these images;


multichannel accelerator b2c electronic store
Welcome Winston | Your Account | Call us: +49 (0)89 890 650 -220 | Find a Store | Sign Out | your shopping basket 0 €0.00
English
€ EUR
I'm looking for
Stuff
Clothes
Welcome to the hybris Merchandise Shop
Fashion
People
Life Style

1.Create the AddOn extension:

Step1: To create a new AddOn use the **yaddon** template with ant extgen:

ex cmd> ant extgen

1.Template name: yaddon

2.extension name:randomimageaddon

3.package: de.hybris.platform.addons.randomimageaddon

The new extension (randomiamgeaddon)is automatically placed in the custom folder.

Step2:Add the extension in config/localextensions.xml

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

Step3: Import the new extension to your Eclipse / STS workspace.

By with this Steps we Have created new Addon Extension in our proj

2.**Creating new cms component in RandomimageAddon extension:**

Step1: *Modify* ***randomimageaddon-items.xml*** *to add the definition for the new CMS component and a new enum type for the topic attribute:.*

```
<enumtypes>
<enumtype code="RandomPictureTopic" autocreate="true" generate="true"
dynamic="true">
<value code="fashion"/>
<value code="sports"/>
<value code="abstract"/>
<value code="animals"/>
<value code="city"/>
<value code="food"/>
<value code="nightlife"/>
<value code="people"/>
<value code="nature"/>
<value code="technics"/>
<value code="transport"/>
</enumtype>
</enumtypes>
```

```
<itemtypes>
<itemtype code="RandomImageParagraphComponent" autocreate="true"
generate="true" extends="CMSParagraphComponent">
<attributes>
<attribute qualifier="topic" type="RandomPictureTopic">
<description>Topic of the Image</description>
<persistence type="property" />
</attribute>
<attribute type="int" qualifier="width">
<modifiers read="true" write="true" search="true" optional="false" />
<defaultvalue>Integer.valueOf(228)</defaultvalue>
<persistence type="property" />
</attribute>
<attribute type="int" qualifier="height">
<modifiers read="true" write="true" search="true" optional="false" />
<defaultvalue>Integer.valueOf(228)</defaultvalue>
<persistence type="property" />
</attribute>
<attribute type="localized:java.lang.String" qualifier="text">
<modifiers read="true" write="true" search="true" optional="true" />
<persistence type="property" />
/attribute>
</attributes>
</itemtype>
</itemtypes>
```

Step2: Localize this new type and its attributes:

randomimageaddon/resources/localization/randomimageaddon-locales_en.properties:

```
type.RandomImageParagraphComponent.name=Random Image Paragraph
Component
type.RandomImageParagraphComponent.topic.name=Topic of the picture
type.RandomImageParagraphComponent.height.name=Height
type.RandomImageParagraphComponent.width.name=Width
type.RandomImageParagraphComponent.text.name=Text in the Picture
```

Step3: **Specify the dependency. The new component type extends CMSParagraphComponent defined in the cms2extension. You need to set these dependencies in Eclipse andextensioninfo.xmlrespectively.**

randomimageaddon/extensioninfo.xml:

```
<------------>
<requires-extension name="cms2"/>
```

Step4: Run ant all to rebuild your system.

Step5:   Do a system update-->Update Running system and localized Types

Step6:**Defining the view for the component**

For the view part, you usually need to do 2 things -

1. define the component's:

Create JSP file :

randomimageaddon/acceleratoraddon/web/webroot/WEB-INF/views/desktop/cms/randomimageparagraphcomponent.jsp

```
<img src="http://lorempixel.com/${width}/${height}/${topic}/${text}" />
```

2.renderer and the view :

Inside merchandisestorefront set dependency to randomimageaddon's build path in Eclipse.

ex-> merchandisestorefront ->buildpath->Add class folder->randomimageAddon folder

3.Declare the bean of renderer and mapping to the AddOn CMS Component Renderer registry:

randomimageaddon/resources/randomimageaddon/web/spring/randomimageaddon-web-spring.xml

```
<bean id="randomImageParagraphComponentRenderer"
parent="addOnJspIncludeCMSComponentRenderer" />
<bean id="randomImageParagraphComponentRendererMapping"
parent="addonCmsComponentRendererMapping">
<property name="typeCode" value="RandomImageParagraphComponent" />
<property name="renderer" ref="randomImageParagraphComponentRenderer" />
</bean>
```

3.Building the AddOn:

1.To make the AddOn available for the storefront extension you should run:

```
cmd>:ant addoninstall -Daddonnames="randomimageaddon"
DaddonStorefront.yacceleratorstorefront="merchandisestorefront"
```

note:A build callback will now copy the web application content to the storefront.

2.Run ant all

After refreshing your workspace you should find the view JSP duplicated to **merchandisestorefront/web/webroot/WEB-INF/views/addons/randomimageaddon/desktop/cms**

Add the new component

1.Add the RandomImageParagraphComponent as a valid type for the three sections:

merchandiseinitialdata/resources/merchandiseinitialdata/import/coredata/contentCatalogs/hybrisContentCatalog/cms-content.impex

```
INSERT_UPDATE ContentSlotName;name[unique=true];template(uid,$contentCV)
[unique=true]
[default='LandingPage2Template'];validComponentTypes(code);compTypeGroup(code
)
;Section2A;;RandomImageParagraphComponent;narrow
;Section2B;;RandomImageParagraphComponent;narrow
;Section2C;;RandomImageParagraphComponent;wide
```

2.Create and assign new components to the content slots:

merchandiseinitialdata/resources/merchandiseinitialdata/import/sampledata/contentCatalogs/hybrisContentCatalog/cms-content.impex

## NEW CMS COMPONENT - RANDOM PICTURE

```
INSERT_UPDATE RandomImageParagraphComponent;
$contentCV[unique=true];uid[unique=true];name;&componentRef;topic(code);width;height
;;RandomImage1;Random Image;RandomImage1;fashion;228;228
;;RandomImage2;Random Image;RandomImage2;people;228;228
;;RandomImage3;Random Image;RandomImage3;nightlife;228;228
```

```
INSERT_UPDATE ContentSlot;
$contentCV[unique=true];uid[unique=true];name;active;cmsComponents(&componentRef)
;;Section2ASlot-Homepage;Section2A Slot for Homepage;true;RandomImage1
;;Section2BSlot-Homepage;Section2B Slot for Homepage;true;RandomImage2
;;Section2CSlot-Homepage;Section2C Slot for Homepage;true;RandomImage3
```

3.Localize the new CMS components:

Update the system, selection project data for merchandiseinitialdata in the hybris Admin Console.->coredataaa and sample data

TASK2:Creating Custom video component in yaddon

Step1:Creating new cms component in RandomimageAddon extension:

```
  <itemtype code="CMSvideoParagraphComponent" generate="true"
extends="CMSParagraphComponent" autocreate="true">
    <description>It represents paragraph component with an additional url
attribute.</description>
    <attributes>
        <attribute qualifier="url" generate="true" autocreate="true"
type="localized:java.lang.String">
          <persistence type="property" />
          <description>Attribute that stores the localized video  url of the
paragraph.</description>
        </attribute>
        <attribute type="int" qualifier="width">
                    <modifiers read="true" write="true" search="true" optional="false" />
                    <defaultvalue>Integer.valueOf(300)</defaultvalue>
                    <persistence type="property" />
                </attribute>
                <attribute type="int" qualifier="height">
                    <modifiers read="true" write="true" search="true" optional="false" />
                    <defaultvalue>Integer.valueOf(300)</defaultvalue>
                    <persistence type="property" />
                </attribute>
    </attributes>
```

```
</itemtype>
```

Step2: Localize this new type and its attributes:

```
type.CMSvideoParagraphComponent.name=CMS video Paragraph component
type.CMSvideoParagraphComponent.url.name= video url
type.CMSvideoParagraphComponent.width.name=video widt
type.CMSvideoParagraphComponent.height.name=video hight
```

Step3: Run ant all to rebuild your system.

Step4:Do a system update-->Update Running system and localized Types

Step5:Defining the view for the component:

1.Create JSP file :

randomimageaddon/acceleratoraddon/web/webroot/WEB-INF/views/desktop/cms/cmsvideoparagraphcomponent.jsp

Note:Here We are using iframe tag in jsp to diplay youtube videos

```
 <iframe width='${width}' height='${height}' src='${url}' frameborder='0'
allowfullscreen></iframe>${height}
```

After refreshing your workspace you should find the view JSP duplicated to ***merchandisestorefront/web/webroot/WEB-INF/views/addons/randomimageaddon/desktop/cms***

2.Declare the bean of renderer and mapping to the AddOn CMS Component Renderer registry:

randomimageaddon/resources/randomimageaddon/web/spring/randomimageaddon-web-spring.xml

```
<bean id="CMSvideoParagraphComponentRenderer"
parent="addOnJspIncludeCMSComponentRenderer" />

<bean id="CMSvideoParagraphComponentRendererMapping"
parent="addonCmsComponentRendererMapping">
```

```
        <property name="typeCode" value="CMSvideoParagraphComponent" />
        <property name="renderer" ref="CMSvideoParagraphComponentRenderer" />
```

Preparing Impex For Video Component

2.Create and assign new components to the content slots:

```
INSERT_UPDATE CMSVedioParagraphComponent;
$contentCV[unique=true];uid[unique=true];name;&componentRef;url;width;height
```

```
;;vedioc1;cms vedio component;vediocomp1;https://www.youtube.com/embed/sm7NgB57lKQ;230;330
;;vedioc2;cms vedio component;vediocomp2;https://www.youtube.com/embed/uCVTg3h9B6s;330;330
```

*Note:url=*https://www.youtube.com/embed/sm7NgB57lKQ
Here We should Give Embeded Url of youtube youtube ->share ->Embeded ->copy url

```
INSERT_UPDATE ContentSlot;
$contentCV[unique=true];uid[unique=true];name;active;cmsComponents(&componentRef)
```

```
;;2ALycaSlot;Section2A vedio slot;true;vedioc1
;;2BLycaSlot;Section2B Vedio slot;true;vedioc2
```

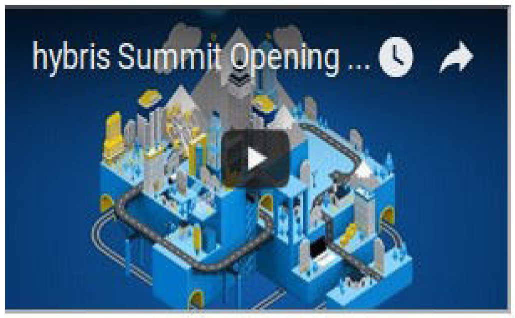
Step3:Assign Content slot to our ContentSlotForPage

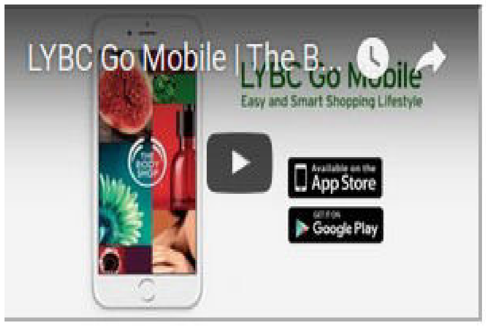
```
INSERT_UPDATE ContentSlotForPage;
$contentCV[unique=true];uid[unique=true];position[unique=true];page(uid,$contentCV)[unique=true][default='lycahomepage'];contentSlot(uid,$contentCV)[unique=true]
```

```
;;2ALycapage;Section2A;;2ALycaSlot
;;2BLycapage;Section2B;;2BLycaSlot
```

Note:Here Section2A, Section2B Represents position of our page

output:


hybris Summit Opening ...


LYBC Go Mobile | The B...
LYBC Go Mobile
Easy and Smart Shopping Lifestyle
App Store
Google Play