

Expose a new CMS Component type

Tuesday, January 20, 2015 11:12 AM

wiki address : <https://wiki.hybris.com/display/tr51/Expose+a+new+CMS+Component+type+through+an+AddOn>

The sample code:

Project Name : yviewmodels

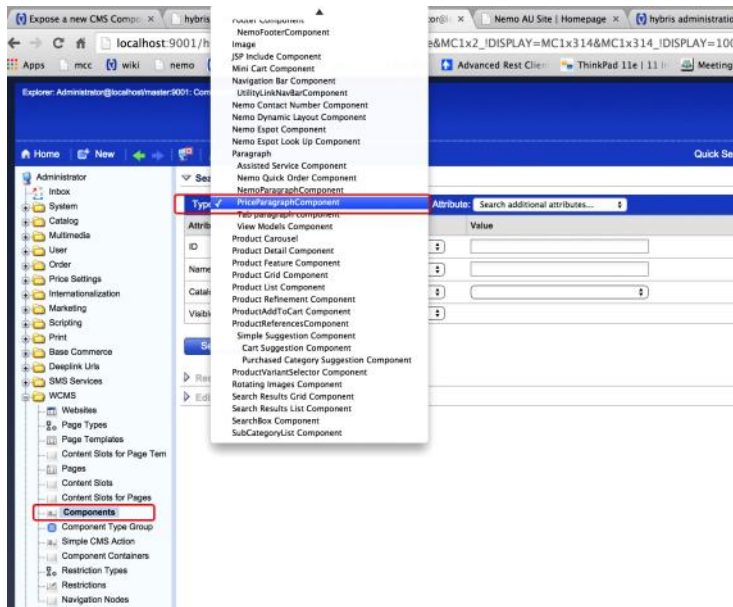
1. add a new itemType: PriceParagraphComponent -->/yviewmodels/resources/yviewmodels-items.xml

```
<itemtype code="PriceParagraphComponent" extends="CMSParagraphComponent"
autocreate="true" generate="true">
<attributes>
<attribute qualifier="webPrice" type="java.lang.Boolean"
generate="true">
<persistence type="property" />
<modifiers read="true" write="true" optional="true"
search="true" />
</attribute>
<attribute qualifier="salesPrice" type="java.lang.Boolean"
generate="true">
<persistence type="property" />
<modifiers read="true" write="true" optional="true"
search="true" />
</attribute>
<attribute qualifier="promoPrice" type="java.lang.Boolean"
generate="true">
<persistence type="property" />
<modifiers read="true" write="true" optional="true"
search="true" />
</attribute>
<attribute qualifier="couponCode" type="java.lang.Boolean"
generate="true">
<persistence type="property" />
<modifiers read="true" write="true" optional="true"
search="true" />
</attribute>
<attribute qualifier="savePrice" type="java.lang.Boolean"
generate="true">
<persistence type="property" />
<modifiers read="true" write="true" optional="true"
search="true" />
</attribute>
</attributes>
</itemtype>
```

2. Localize this new type and its attributes -->/yviewmodels/resources/localization/yviewmodels-locales_en.properties

```
type.PriceParagraphComponent.name=PriceParagraphComponent
type.PriceParagraphComponent.webPrice.name=Web Price
type.PriceParagraphComponent.salesPrice.name=Sales Price
type.PriceParagraphComponent.promoPrice.name=Promo Price
type.PriceParagraphComponent.couponCode.name=Coupon Code
type.PriceParagraphComponent.savePrice.name=Save Price
```

3. Run ant all to rebuild your system.
4. Do a system update



5. Create `/yviewmodels/acceleratoraddon/web/webroot/WEB-INF/views/desktop/cms/priceparagraphcomponent.jsp` 和 `/yviewmodels/src/com/novarto/yviewmodels/component/renderer/PriceParagraphComponentRenderer.java`

code:

```
<dl class="pricingSummary-details">
<c:choose>
<c:when test="{webPrice}">
222
</c:when>
<c:otherwise>
333
</c:otherwise>
</c:choose>
</dl>
```

Code : PriceParagraphComponentRenderer.java

```
public class PriceParagraphComponentRenderer<C extends PriceParagraphComponentModel> extends DefaultAddOnCMSComponentRenderer<C>
{
```

```
CMSComponentService cmsComponentService;
ModelService modelService;
```

```
@Override
@Required
public void setCmsComponentService(final CMSComponentService cmsComponentService)
{
this.cmsComponentService = cmsComponentService;
}
```

```
@Override
@Required
public void setModelService(final ModelService modelService)
{
this.modelService = modelService;
}
```

```
@Override
protected Map<String, Object> getVariablesToExpose(final PageContext pageContext, final C component)
{
final Map<String, Object> variables = new HashMap<String, Object>();

for (final String property : cmsComponentService.getEditorProperties(component))
{
try
```

```

{
    final Object value = modelService.getAttributeValue(component, property);

    variables.put(property, value);

}
catch (final AttributeNotSupportedException ignore)
{
    // ignore
}
}

return variables;
}

```

6. Create the renderer bean:

```

<bean id="priceParagraphComponentRenderer"
class="com.novarto.yviewmodels.component.renderer.PriceParagraphComponentRenderer">
<property name="typeService" ref="typeService" />
<property name="uiExperienceService" ref="defaultUiExperienceService" />
<property name="cmsComponentService" ref="cmsComponentService" />
<property name="modelService" ref="modelService" />
</bean>

```

7. Add new renderer to CMS Component Renderer registry

→ /nemostorefront/web/webroot/WEB-INF/config/spring-cms-config.xml

```

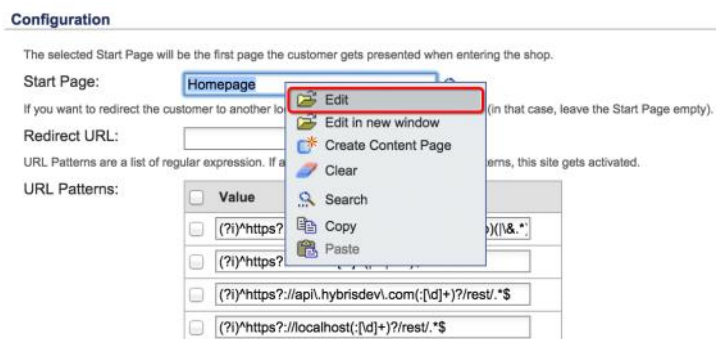
<alias alias="cmsComponentRendererRegistry" name="acceleratorCMSComponentRendererRegistry" />
<bean id="acceleratorCMSComponentRendererRegistry" parent="defaultCMSComponentRendererRegistry">
<property name="renderers">
<map>
<entry key="CMSLinkComponent" value-ref="cmsLinkComponentRenderer"/>
<entry key="YviewmodelsParagraphComponent" value-ref="viewModelsParagraphComponentRenderer"/>
<entry key="PriceParagraphComponent" value-ref="priceParagraphComponentRenderer"/>
<!-- POC uncomment to use <entry key="CustomCMSComponent" value-ref="customCMSComponentRenderer"/> -->
</map>
</property>
</bean>

```

8. ant clean all

9. Add the new component in the hmc

- In the hMC, expand the WCMS menu and click on Websites.
- Search for and select the hybris website.
- Edit the Start Page property by right-clicking in the field:



- In the Content Page editor, edit the Page Template property:

Properties

Page Title in the Frontend. This value is localized.

Page Title:

Page Template:

If you have multiple pages for a resource, at least one page needs to be the default page.

Is default page: ☒

Edit
 Edit in new window
 Create new
 Clear
 Search
 Copy
 Paste

- In the Page Template editor, right-click in the Section2A content slot and select Edit in new window or click on the blue box next to the field:

Content Slots

This is the list of all section names which will be available for this Page Template. Each section can be assigned with one Content Slot.

Available Content Slots:

Name
<input type="checkbox"/> HeaderLinks
<input type="checkbox"/> SiteLogo
<input type="checkbox"/> SearchBox
<input type="checkbox"/> MiniCart
<input type="checkbox"/> Section2B
<input type="checkbox"/> NavigationBar
<input checked="" type="checkbox"/> Section2A
<input type="checkbox"/> Section1
<input type="checkbox"/> Section2C
<input type="checkbox"/> Section3
<input type="checkbox"/> Section4
<input type="checkbox"/> Section5
<input type="checkbox"/> Footer

Edit in new window
 Edit
 Create Content Slot Name
 Remove
 Select all
 Deselect all
 Copy

-
- In the Valid Component Types table, right-click and select Add Component Type:
- Search for and select PriceParagraphComponent.

Properties

A Content Slot Name defines a section/place in a Page Template. To each section there can be assigned a Content Slot.

Name:

Component Type Group:

Valid Component Types:

Identifier	Name	Extensionname
<input checked="" type="checkbox"/> PriceParagraphComponent	PriceParagraphComponent	yviewmodels

- Click Save and Close in the top bar of the pop-out window.

Using the new component type

- Login to the [cmscockpit](https://cmscockpit.com).
- In the left navigation menu, select hybris Site and hybris Content Catalog / Staged.
- In the browser area, double-click on Homepage to activate it.
- In Section2A Slot for Homepage, click on Create an item:

Nemo Content Catalog >>> Online >>> Homepage - NEMO

Site Logo

Search

Navigation Bar

Not defined [Section1]

Default Section2A for Homepage

PriceParagraphComponent

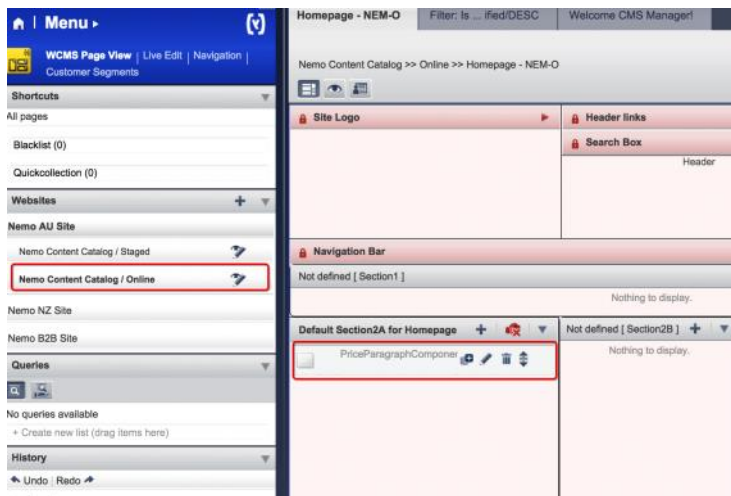
Not defined [Section2]

Create/Add New Component - Select desired type

- NemoLinkComponent
- NemoParagraphComponent
- Paragraph
- PriceParagraphComponent
- Product Carousel
- Product Feature Component
- ProductAddToCart Component
- ProductReferencesComponent
- ProductVariantSelector Component
- Purchased Category Suggestion Component

Cancel Done

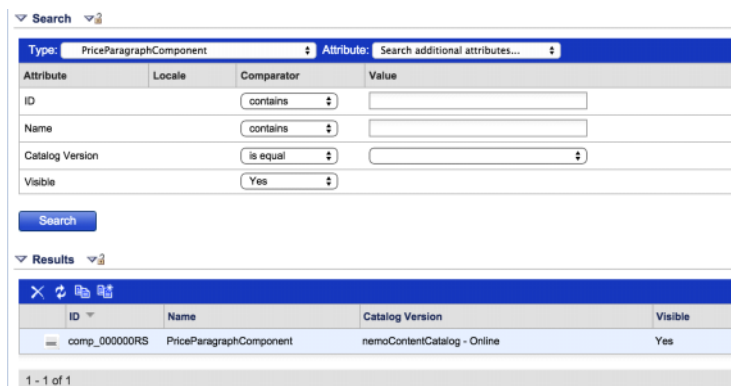
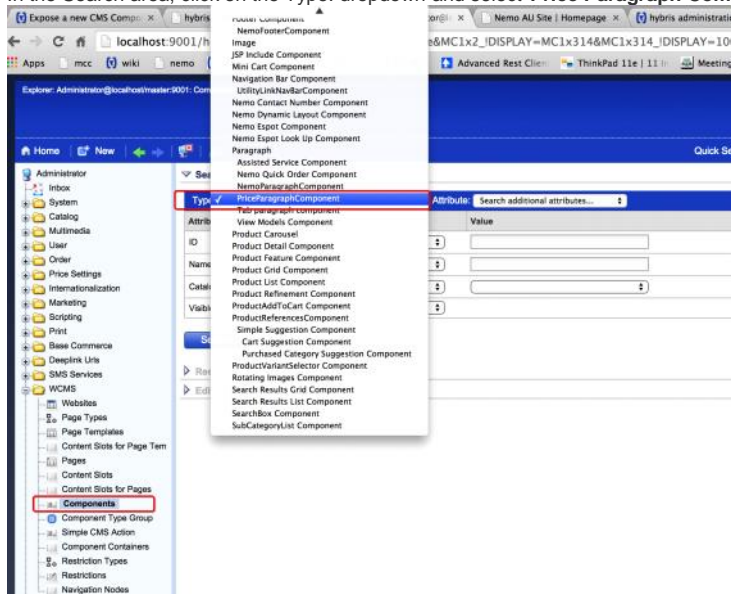
- Select Random Image Paragraph Component as the type and Create a new item.
- Enter a name for your new component and click on Done.
- Right-click on the hybris Content Catalog / onLine option in the left navigation menu and click on Synchronize content catalog



Select Topics for the components

Right now, if you view the home page, you will see three identical images. Ideally, we would use the CMS cockpit to select each image's topic. Since we haven't yet customized the cockpit configuration, we'll have to use the hMC to set each image's topic. (Customizing the cockpit to set these values is the subject of the next, optional part of this trail.)

- In the hMC, expand the WCMS menu and click on Components.
- In the Search area, click on the Type: dropdown and select **Price Paragraph Component**



- Each component you created will appear twice (in the Staging and Online catalogs). Select one of the components in the Online catalog, go to the Administration tab, and select the **web Price**

Coupon Code: ☐ Yes ☒ No ☐ n/a

Promo Price: ☐ Yes ☒ No ☐ n/a

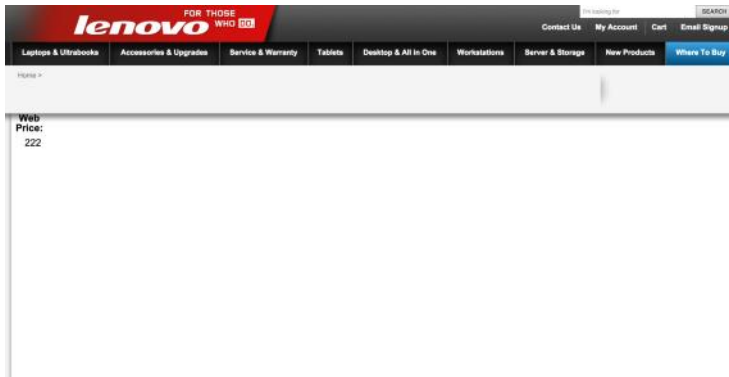
Is restricted: ☐

Sales Price: ☐ Yes ☒ No ☐ n/a

Save Price: ☐ Yes ☒ No ☐ n/a

Web Price: ☒ Yes ☐ No ☐ n/a

Save the component, and navigate to (or refresh) the homepage.



```
$contentCatalog=nemoContentCatalog
```

```
$contentCatalogName=Nemo Content Catalog
```

```
$contentCV=catalogVersion(CatalogVersion.catalog(Catalog.id[default=$contentCatalog]),CatalogVersion.version[default=Staged])[default=$contentCatalog:Staged]
```

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

```
INSERT_UPDATE PriceParagraphComponent;$contentCV[unique=true];uid[unique=true];webPrice;salesPrice;promoPrice;couponCode;save Price;
;;price1;true;true;true;true;true
```

Hybris Syntax for Impex

Tuesday, April 14, 2015 11:22 AM

Hi All,

Hybris Trail

<https://wiki.hybris.com/display/R5T/Trail+~+FAQ#Trail~FAQ-Howtoinitialisethetesttenant>

Hybris Syntax for Impex

<https://wiki.hybris.com/display/release5/ImpEx+Syntax>

<https://wiki.hybris.com/display/release5/How+to+Add+a+New+CMS+Component>

example

<https://wiki.hybris.com/display/R5T/hybris+5+Developer+Training+Trails+--+Part+I+--+Core>

<https://wiki.hybris.com/display/R5T/hybris+5+Developer+Training+Trails+--+Part+II+--+Commerce>

WCMS <http://localhost:9001/cmscockpit/login.zul>

HMC <http://localhost:9001/hmc/hybris>

HAC <http://localhost:9001/>

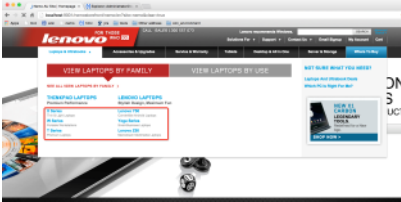
Regards,

Bo

PS: please focus on the impex part, that is important.

NA-5399 menu的导入

Monday, April 20, 2015 2:56 PM



menu的导入

```
$contentCatalog=nemoContentCatalog
$contentCatalogName=Nemo Content Catalog
$contentCV=catalogVersion(catalogVersion.catalog[catalog.id[default=$contentCatalog]],catalogVersion.version[default=Staged]) [default=$contentCatalog:Staged]
$productCatalog=auProductCatalog
$lang=en
$productCatalogName=Nemo Australia Catalog
$productCV=catalogVersion(catalog[id[default=$productCatalog]],version[default=Staged])[unique=true,default=$productCatalog:Staged]
$image=media(code,$contentCV)
$media=media(code,$contentCV)
$page=page(uid,$contentCV)
$contentPage=contentPage(uid,$contentCV)
$product=product(code,$productCV)
$category=category(code,$productCV)
$siteResource=jar:com.nemo.initialdata.constants.NemoInitialDataConstants!/nemoinitialdata/import/sampledata/contentCatalog/$contentCatalog
$productResource=jar:com.nemo.initialdata.constants.NemoInitialDataConstants!/nemoinitialdata/import/sampledata/productCatalogs/$productCatalog
$jarResourceCms=jar:com.nemo.initialdata.constants.NemoInitialDataConstants!/nemoinitialdata/import/sampledata/cockpits/cmscockpit
$products=products(code,$productCV)
$masterProductCatalog=masterProductCatalog
$catalogVersion=catalogVersion(catalog[id[default=$masterProductCatalog]],version[default=Default])[unique=true,default=$masterProductCatalog:Default]
```

```
INSERT_UPDATE CMSLinkComponent ; $contentCV[unique=true] ; uid[unique=true] ; name ; uri ; &linkRef ;
&componentRef ; target[code][default=SameWindow] ; Category ; Product ; external ;
; YogaTabletLink ; YogaTabletLink ; YogaTabletLink ;
YogaTabletLink ; ; ; false ; YogaTabletLink ;
INSERT_UPDATE CMSNavigationNode ; uid[unique=true] ; $contentCV[unique=true] ; name ; parent[uid,$contentCV] ;
links(&linkRef) ; &nodeRef ;
YogaTabletNode ; YogaTabletNode ; YogaTabletNode ; UltrabooksNavNode ;
YogaTabletLink ; YogaTabletNode ; YogaTabletNode ;
UPDATE CMSNavigationNode ; $contentCV[unique=true] ; uid[unique=true] ; title[lang=$lang] ;
; YogaTabletNode ; "YogaTablet Series"
```