styla integration for hybris

A full integration of styla with your online shop consists of:

- adding a content page to your online shop where the styla content is rendered
- adding styla’s javascript and stylesheet definition to the content page that renders the styla content
- provide endpoints which respond to requests coming from styla in order to expose information about your category structure, productimages in a category and detailed product information
- integrate with styla’s SEO endpoint, which returns for a requested styla content information relevant for search engine optimization
- provide a javascript function or endpoint for adding products to the user’s cart when called from a magazine page

Solution

The styla integration for hybris consists of two extensions, which simplify the integration of styla with your online shop. The extensions use hyris’ Service Layer API which let you easily implement the interfaces on your own or extend the default implementations provided with this solution.

extension “styla”

The styla extension is essential. It

- defines the interfaces of service classes which are used by the implementation of the endpoints responding to requests coming from styla
- provides classes for convenient conversion of Java objects to JSON and vice versa
- defines an interface and provides a default implementation for requesting SEO content from styla

extension “stylaaddon”

The stylaaddon extension is optional and is a sample implementation adding the styla magazine to the hybris B2C Accelerator. It depends on the styla extension, as it implements its interfaces. Features of this extension:

- provides a controller implementing the endpoint for exporting data to styla
- provides default implementations of the service interfaces which are used for exporting data to styla
- provides a controller responsible for rendering the magazine page in your online shop
- html templates (*jsp and *tag) that render the html for the styla page and add the javascript/stylesheet declarations
- ImpEX files for the creation of CMS items (pages, contentslots, etc.) for the magazine page

If your online shop doesn’t use the standard hybris accelerator (B2C) you won’t use this extension in your system.

Get started

Disclaimer:

Before deploying this solution to your production environment, you should have installed and tested the proper functionality in your testing environment. Make sure you have a backup of your working system.

Adding styla extension

1. Unzip hybris-styla-extension.zip to {HYBRIS_HOME}/hybris. It will add the styla extension to {HYBRIS_HOME}/hybris/bin/custom/styla

2. Add to {HYBRIS_HOME}/hybris/config/localextensions.xml this line within the extensions tag:
   …
   <extension name='styla' />
   ...
3. Edit {HYBRIS_HOME}/hybris/config/local.properties and override the properties from {HYBRIS_HOME}/hybris/bin/custom/styla/project.properties with the custom values provided to you by styla.

4. Change directory to {HYBRIS_HOME}/hybris/bin/platform/ and execute

   ant all

As this extension doesn't modify the hybris type system, there is no need to perform a platform update.

Adding stylaaddon extension

1. Unzip hybris-stylaaddon-extension.zip to {HYBRIS_HOME}/hybris. It will add the stylaaddon extension to {HYBRIS_HOME}/hybris/bin/custom/stylaaddon

2. Add to {HYBRIS_HOME}/hybris/config/localextensions.xml this line within the extensions tag:
   …
   <extension name='stylaaddon' />
   ...
3. Change directory to {HYBRIS_HOME}/hybris/bin/platform/ and execute

   ant addoninstall -Daddonnames="stylaaddon"
   -DaddonStorefront.yacceleratorstorefront="yacceleratorstorefront"

   If your storefrontname is different, change "yacceleratorstorefront" accordingly.

4. Copy {HYBRIS_HOME}/hybris/bin/custom/stylaaddon/acceleratoraddon/web/webroot/WEB-INF/views/desktop/pages/layout/* to

   {HYBRIS_HOME}/hybris/bin/ext-template/yacceleratorstorefront/web/webroot/WEB-INF/views/desktop/pages/layout/ and

   {HYBRIS_HOME}/hybris/bin/acceleratoraddon/web/webroot/WEB-INF/tags/shared/styla to

   {HYBRIS_HOME}/hybris/bin/ext-template/yacceleratorstorefront/web/webroot/WEB-INF/tags/shared/

   This will copy the html template files.

5. Edit {yacceleratorstorefront}/web/webroot/WEB-INF/tags/desktop/template/master.tag adding

   <%@ taglib prefix="styla" tagdir="/WEB-INF/tags/shared/styla" %> to the taglib list and

   <styla:styla/> within the <head> element

6. Change directory to {HYBRIS_HOME}/hybris/bin/platform/ and execute

   ant all

7. Start hybris and perform an impex of the cms-content.impex file located under

   {HYBRIS_HOME}/hybris/bin/custom/stylaaddon/resources/stylaaddon/import/contentCatalogs/<storename>

   This will create the CMS items for the styla content page.

As this extension doesn’t modify the hybris type system, there is no need to perform a platform update.

Customizations

Custom frontend

If you don’t use the hybris accelerator or a web application framework other than Spring MVC, you have to implement the endpoints that render the styla magazine page (requests for /magazine) and expose online shop data to styla (requests for /styla/*), respectively, on your own.

In order to get the appropriate SEO information from styla for rendering the magazine page you can use the DefaultStylaSeoService class.

In order to expose your online shop data (categories, productimages and product detail information) you have to implement the interfaces:

- StylaCategoryProvider: Return the category/navigation menu as shown in the frontend of your online shop.
- StylaProductProvider: Return a product for a given product id.
- StylaSearchService: Return for a given category and a optional searchterm the list of products that satisfy these criterias. The results should correspond to the result set done by the same search in the frontend of your online shop, i.e. as if you would use the text search or category search.