

Unit 5. Product Comparison

Backend steps:

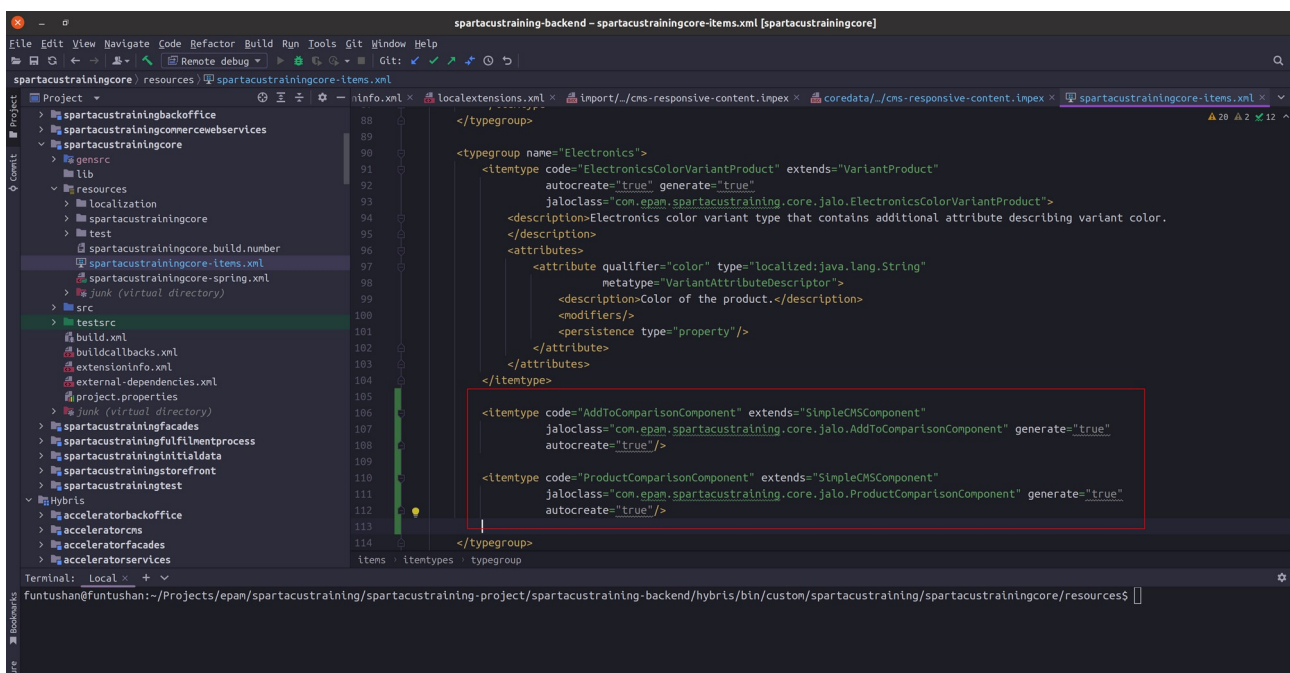
1. in spartacustrainingcore/resources open spartacustrainingcore-items.xml and add next code to electronics typegroup:

```
<itemtype code="AddToComparisonComponent"
extends="SimpleCMSComponent"
```

```
    jaloclass="com.epam.spartacustraining.core.jalo.AddToComparisonComponent"
    generate="true"
    autocreate="true"/>
```

```
<itemtype code="ProductComparisonComponent"
extends="SimpleCMSComponent"
```

```
    jaloclass="com.epam.spartacustraining.core.jalo.ProductComparisonComponent"
    generate="true"
    autocreate="true"/>
```



2. in /spartacustraining-backend/hybris/bin/custom/spartacustrainingcommercewebservices/resources open spartacustrainingcommercewebservices-beans.xml and add next code

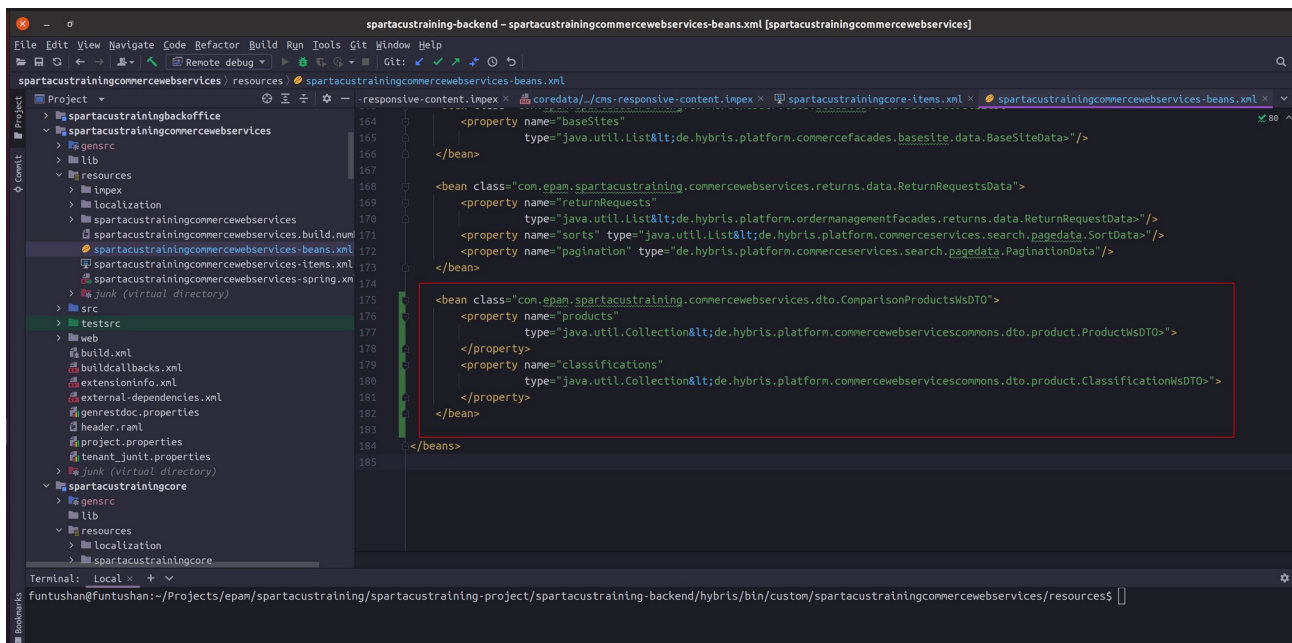
```

<bean
class="com.epam.spartacustraining.commercewebservices.dto.ComparisonProductsWsDTO">
    <property name="products"

type="java.util.Collection<de.hybris.platform.commercewebservicescommons.dto.product.ProductWsDTO>">
    </property>
    <property name="classifications"

type="java.util.Collection<de.hybris.platform.commercewebservicescommons.dto.product.ClassificationWsDTO>">
    </property>
</bean>

```



2. in hybris/bin/platform run . ./setantenv.sh (bat)
 3. run **ant all** (clean all if you have a errors)
 4. In spartacustraining-backend/hybris/bin/custom/spartacustrainingcommercewebservices/web/src/com/epam/spartacustrainingcommercewebservices/v2/controller create new java class - ProductComparisonRestController.java file
- and add this code to file:

```
package
com.epam.spartacustraining.commerceservices.v2.controller
;

import javax.annotation.Resource;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.HashSet;
import java.util.List;
import java.util.Objects;
import java.util.Set;
import java.util.stream.Collectors;

import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;

import
de.hybris.platform.commerceservices.product.ProductFacade;
import
de.hybris.platform.commerceservices.product.ProductOption;
import
de.hybris.platform.commerceservices.product.data.ClassificationData;
import
de.hybris.platform.commerceservices.product.data.ProductData;
import
de.hybris.platform.commerceservicescommons.dto.product.ClassificationWsDTO;
import
de.hybris.platform.commerceservicescommons.dto.product.ProductWsDTO;
import
de.hybris.platform.webservicescommons.mapping.DataMapper;

import
com.epam.spartacustraining.commerceservices.dto.ComparisonProductsWsDTO;

@RestController
@RequestMapping(value = "/{baseSiteId}/comparison")
public class ProductComparisonRestController {
```

```

    private final static String fields =
"DEFAULT,images,classifications";
    private final static Set<ProductOption> OPTIONS = new
HashSet<>(Arrays.asList(ProductOption.values()));

    @Resource(name = "cwsProductFacade")
    private ProductFacade productFacade;
    @Resource(name = "dataMapper")
    private DataMapper dataMapper;

    @GetMapping("/products/{products}")
    public ResponseEntity<ComparisonProductsWsDTO>
getComparisonProductsWsDTO(@PathVariable final String
products)
    {
        List<ProductData> productData =
getProducts(products).stream()
            .map(code ->
productFacade.getProductForCodeAndOptions(code, OPTIONS))
            .collect(Collectors.toList());

        return
ResponseEntity.status(HttpStatus.OK).body(getComparisonProdu
ctsDto(productData));
    }

    private List<String> getProducts(String s)
    {
        return Arrays.asList(s.split(","));
    }

    private ComparisonProductsWsDTO
getComparisonProductsDto(List<ProductData> productData)
    {
        ComparisonProductsWsDTO comparisonProductsDto = new
ComparisonProductsWsDTO();

        comparisonProductsDto.setProducts(dataMapper.mapAsList(produ
ctData, ProductWsDTO.class, fields));

        comparisonProductsDto.setClassifications(dataMapper.mapAsLis
t(getClassificationData(productData),
ClassificationWsDTO.class, "code,name,features"));
        return comparisonProductsDto;
    }

```

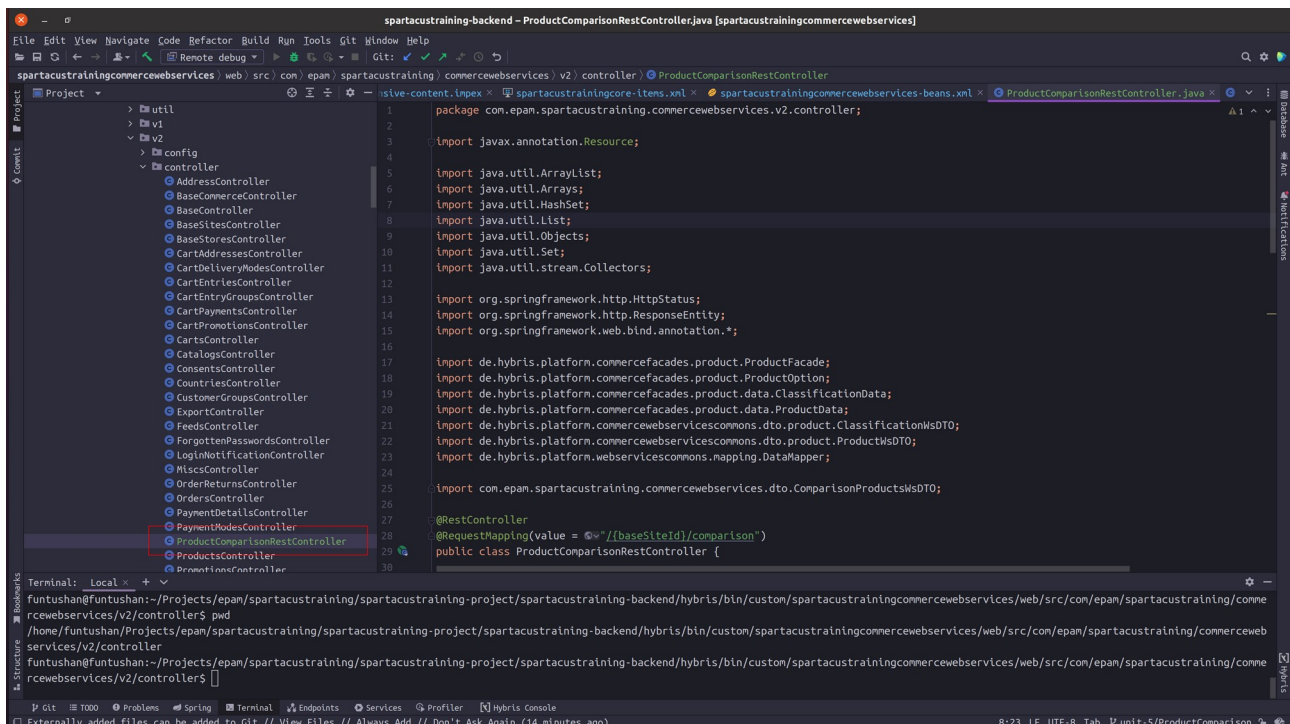
```

    private List<ClassificationData>
getClassificationData(List<ProductData> productData)
    {
        List<ClassificationData> allClassifications = new
ArrayList<>();
        productData.stream()
            .filter(Objects::nonNull)
            .map(ProductData::getClassifications)
            .filter(Objects::nonNull)
            .forEach(classifications -> classifications
                .stream()
                .filter(Objects::nonNull)
                .forEach(classification -> {
                    if (!contains(allClassifications,
classification)) {
                        allClassifications.add(classification);
                    }
                }));

        return allClassifications;
    }

    private boolean contains(List<ClassificationData> list,
ClassificationData classification)
    {
        return list.stream()
            .anyMatch(x ->
x.getCode().equals(classification.getCode()));
    }
}

```



5. Run **ant all**
6. Run **server ./hybrisserver.sh**
7. Open <https://localhost:9002/hac> and navigate to console/impex/import

and run next code:

```
$contentCatalog=electronics-spaContentCatalog
$contentCV=catalogVersion(CatalogVersion.catalog(Catalog.id[
default=$contentCatalog]),CatalogVersion.version[default=Sta
ged])[default=$contentCatalog:Staged]
$siteResource=jar:de.hybris.platform.spartacussampledataby
stants.Spartacussampledataby
constants.Spartacussampledataby
constants.Spartacussampledataby
import/contentCatalogs/electronicsContentCatalog
```

```
INSERT_UPDATE AddToComparisonComponent; uid[unique = true]
; name ; $contentCV[unique = true]
;
```

```
AddToComparisonComponent; AddToComparisonComponent;
```

```
INSERT_UPDATE ContentSlot; uid[unique = true] ;
cmsComponents(uid, $contentCV); $contentCV[unique = true]
; ProductSummarySlot ;
```

```
ProductIntroComponent,ProductImagesComponent,ProductSummaryC
omponent,VariantSelector,AddToCart,AddToWishListComponent,St
ockNotificationComponent,AddToComparisonComponent;
```

```

INSERT_UPDATE ProductComparisonComponent; uid[unique = true]
; name ; $contentCV[unique = true]
;
productComparisonComponent ; productComparisonComponent ;

INSERT_UPDATE ContentPage; uid[unique = true]; name
; title[lan = en]; masterTemplate(uid, $contentCV); label
; defaultPage[default = 'true']; approvalStatus(code)
[default = 'approved']; homepage[default = 'false'];
$contentCV[unique = true]
; productComparison ; Product
Comparison ; Comparison Products ; ContentPage1Template;
/comparison/products ;

INSERT_UPDATE ContentSlot; $contentCV[unique = true];
uid[unique = true] ; name ; active;
cmsComponents(uid, $contentCV)
;
Section2AComparison ; Slot Product Comparison ; true ;
productComparisonComponent

INSERT_UPDATE ContentSlotForPage; $contentCV[unique = true];
uid[unique = true] ; position[unique = true]; page(uid,
$contentCV)[unique = true]; contentSlot(uid, $contentCV)
[unique = true]
;
Section2AComparison ; Section2A ;
productComparison ; Section2AComparison

#add link
INSERT_UPDATE CMSLinkComponent; $contentCV[unique = true];
uid[unique = true]; name ; linkName ; url
; target(code)[default = 'sameWindow']
;
ComparisonLink ; Comparison Link ; Comparison page ;
/comparison/products

INSERT_UPDATE ContentSlot; $contentCV[unique = true];
uid[unique = true]; name ; active;
cmsComponents(uid, $contentCV)
;
SiteLinksSlot ; Slot contains some links ; true ;
ComparisonLink,OrdersLink,WishListLink,StoreFinderLink,Conta
ctUsLink,HelpLink

```

And run this code by online catalog

The screenshot shows the 'hybris administration console' interface. The 'ImpEx Import' page is active, with the 'Import content' tab selected. The code area contains the following script:

```
1 $contentCatalog=electronics-spaContentCatalog
2 $contentCV=catalogVersion(CatalogVersion.catalog(Catalog.id[default=$contentCatalog]),CatalogVersion.version[default=Staged])[default=$contentCatalog:Staged]
3 $siteResource=jar:de.hybris.platform.spartacus.sampledata.constants.SpartacusSampledataConstants$SpartacusSampledata$ImportContentCatalogs/electronicContentCatalog
4
5
6 INSERT_UPDATE AddToComparisonComponent; uid[unique = true] ; name ; $contentCV[unique = true]
7 ; AddToComparisonComponent; AddToComparisonComponent;
8
9 INSERT_UPDATE ContentSlot; uid[unique = true] ; cmsComponents(uid, $contentCV); $contentCV[unique = true]
10 ; ProductSummarySlot ; ProductIntroComponent,ProductImagesComponent,ProductSummaryComponent,VariantSelector,AddToCart,AddToWishListComponent,StockNotificationComponent,AddToComparisonCompon
11
12 INSERT_UPDATE ProductComparisonComponent; uid[unique = true] ; name ; $contentCV[unique = true]
13 ; productComparisonComponent ; productComparisonComponent ;
14
15 INSERT_UPDATE ContentPage; uid[unique = true]; name ; title[lan = en]; masterTemplate(uid, $contentCV); label ; defaultPage[default = 'true']; approvalStatus(code)[default = 'approved']; homepage
16 ; productComparison ; Product Comparison ; Comparison Products ; ContentPageTemplate /comparison/products ;
17
18
19 INSERT_UPDATE ContentSlot; $contentCV[unique = true]; uid[unique = true] ; name ; active; cmsComponents(uid, $contentCV)
```

Below the code, there are buttons for 'Clear content', 'Import content', and 'Validate content'. The 'Settings' section is expanded, showing options for 'Import validation mode', 'Script encoding' (set to UTF-8), 'Enable code execution' (checked), 'Distributed mode' (unchecked), 'Max. threads' (32), 'Legacy mode' (unchecked), 'Syntax Highlighting' (checked), and 'Direct persistence' (unchecked).

The screenshot shows the same 'hybris administration console' interface, but with the 'Import script' tab selected. The code area contains the following script:

```
1 $contentCatalog=electronics-spaContentCatalog
2 $contentCV=catalogVersion(CatalogVersion.catalog(Catalog.id[default=$contentCatalog]),CatalogVersion.version[default=Online])[default=$contentCatalog:Online]
3 $siteResource=jar:de.hybris.platform.spartacus.sampledata.constants.SpartacusSampledataConstants$SpartacusSampledata$ImportContentCatalogs/electronicContentCatalog
4
5
6 INSERT_UPDATE AddToComparisonComponent; uid[unique = true] ; name ; $contentCV[unique = true]
7 ; AddToComparisonComponent; AddToComparisonComponent;
8
9 INSERT_UPDATE ContentSlot; uid[unique = true] ; cmsComponents(uid, $contentCV); $contentCV[unique = true]
10 ; ProductSummarySlot ; ProductIntroComponent,ProductImagesComponent,ProductSummaryComponent,VariantSelector,AddToCart,AddToWishListComponent,StockNotificationComponent,AddToComparisonCompon
11
12 INSERT_UPDATE ProductComparisonComponent; uid[unique = true] ; name ; $contentCV[unique = true]
13 ; productComparisonComponent ; productComparisonComponent ;
14
15 INSERT_UPDATE ContentPage; uid[unique = true]; name ; title[lan = en]; masterTemplate(uid, $contentCV); label ; defaultPage[default = 'true']; approvalStatus(code)[default = 'approved']; homepage
16 ; productComparison ; Product Comparison ; Comparison Products ; ContentPageTemplate /comparison/products ;
17
18
19 INSERT_UPDATE ContentSlot; $contentCV[unique = true]; uid[unique = true] ; name ; active; cmsComponents(uid, $contentCV)
```

The 'Settings' section remains the same as in the previous screenshot, with 'Import validation mode' set to 'import_script', 'Script encoding' set to UTF-8, 'Enable code execution' checked, 'Distributed mode' unchecked, 'Max. threads' set to 32, 'Legacy mode' unchecked, 'Syntax Highlighting' checked, and 'Direct persistence' unchecked.