



SUMMIT

THE DIGITAL MARKETING CONFERENCE

March 9-13, 2015 | Salt Lake City, Utah

Twitter : #AdobeSummit



Adobe Experience Manager - Commerce Integration Framework - Workbook

Varun Venkataraman | Senior Technical Consultant

SUMMIT
THE DIGITAL MARKETING CONFERENCE

Agenda

- 1 | Exercise 1 : Commerce Product Import Management
- 2 | Exercise 2 : Commerce Page Rollout Mechanism Management
- 3 | Exercise 3 : Using & Extending Commerce Integration Framework
- 4 | Exercise 4 : Extending ElasticPath Commerce Framework

Housekeeping - Your Machine

Home	- /Users/l716
Location of EP	- /Users/l716/elasticpath
AEM 6.1_L17_EP	- /Users/l716/Desktop/AEM6.1-L17_EP
Eclipse WS	- /Users/l716/Documents/workspace
GIT WS	- /Users/l716/Desktop/AEM-Commerce-Lab-WS/aem-commerce-framework-extensions

GitHub – References

GitHub Project:

<https://github.com/Adobe-Marketing-Cloud/aem-commerce-framework-extensions>

Exercise 1 – Base Structure and Solutions:

<https://github.com/Adobe-Marketing-Cloud/aem-commerce-framework-extensions/tree/exercise1>

<https://github.com/Adobe-Marketing-Cloud/aem-commerce-framework-extensions/tree/exercise1-solution>

Exercise 2 – Base Structure and Solutions:

<https://github.com/Adobe-Marketing-Cloud/aem-commerce-framework-extensions/tree/exercise2>

<https://github.com/Adobe-Marketing-Cloud/aem-commerce-framework-extensions/tree/exercise2-solutions>

GitHub – References

Exercise 3 – Base Structure and Solutions:

<https://github.com/Adobe-Marketing-Cloud/aem-commerce-framework-extensions/tree/exercise3-task1>

<https://github.com/Adobe-Marketing-Cloud/aem-commerce-framework-extensions/tree/exercise3-task1-solutions>

<https://github.com/Adobe-Marketing-Cloud/aem-commerce-framework-extensions/tree/exercise3-task2>

<https://github.com/Adobe-Marketing-Cloud/aem-commerce-framework-extensions/tree/exercise3-task2-solutions>

Exercise 4 – Base Structure and Solutions:

<https://github.com/Adobe-Marketing-Cloud/aem-commerce-framework-extensions/tree/exercise4-task2>

<https://github.com/Adobe-Marketing-Cloud/aem-commerce-framework-extensions/tree/exercise4-task2-solutions>

Agenda

- 1 | Exercise 1 : Commerce Product Import Management**
- 2 | Exercise 2 : Commerce Page Rollout Mechanism Management
- 3 | Exercise 3 : Using & Extending Commerce Integration Framework
- 4 | Exercise 4 : Extending ElasticPath Commerce Framework



Exercise 1

Commerce Product Import Management

Exercise 1 – Extending Product Import Management (PIM) Console

- **Task :**

Add an additional properties for a product in PIM called **summit** and extending the scaffolding framework to add additional properties to be edited on the PIM from Touch UI.

- **Time:**

10 min

- **Note**

Solution installation given (with notes on how to get to solutions)

Exercise 1 -Setup

Setup

1. In your terminal
2. Enter the following

```
sh install-exercise1-task1-setup.sh
```

Note: The above script is a simple git branch change and a maven install of the setup for Lab 1.

```
git checkout exercise1  
mvn clean install -PautoInstallPackage
```

Exercise 1 -Observation 1

Observation 1

1. Navigate to (copy & paste in browser)

<http://localhost:4502/libs/commerce/gui/content/products/properties.html/etc/commerce/products/summit-geometrixx-outdoors/equipment/running/eqsmmr>

2. Take note of all the properties in Edit mode.

Note: This covers all the properties editable for the product catalog.

Exercise 1 -Observation 2

Observation 2

1. Now, navigate to (copy & paste in browser)

<http://localhost:4502/crx/de/index.jsp#/etc/commerce/products/summit-geometrixx-outdoors/equipment/running/eqsmmr>

2. Take note of all the properties in the JCR

Name	Type	Value	Protected	Mandatory	Multiple
1 cq:commerceType	String	product	false	false	false
2 cq:productSkuCode	String	eqsmmr	false	false	false
3 cq:productVariantAxes	String[]	geo-color, size	false	false	true
4 cq:tags	String[]	summit-geometrixx-outdoors:gender/women, ...	[+] false	false	true
5 description	String	Women's Cross-Trainers	false	false	false
6 features	String	 Extra lateral support Quick fa...	[+] false	false	false
7 identifier	String	eqsmmr	false	false	false
8 jcr:lastModified	Date	2014-12-09T15:04:51.847-08:00	false	false	false
9 jcr:mixinTypes	Name[]	cq:Taggable	true	false	true
10 jcr:primaryType	Name	nt:unstructured	true	true	false
11 jcr:title	String	Mombassa Runners	false	false	false
12 sling:resourceType	String	commerce/components/product	false	false	false
13 summary	String	<p>Specially engineered for a woman's foot. ...	[+] false	false	false

Exercise 1 -Solution

Solution

1. In your terminal
2. Enter the following

```
sh install-exercise1-task1-solution.sh
```

Note: The above script is a simple git branch change and a maven install of the solution for Lab 1.

```
git checkout exercise1-solution  
mvn clean install -PautoInstallPackage
```

Exercise 1 – Result 1

Commerce

Catalogs

Products

Collections

Orders

Payment Methods

Shipping Methods

Edit

Product thumbnail

Nairobi Runners

BASIC PRODUCT TABS

Title *

Nairobi Runners

Tags

Summit Geometrixx Outdoors : Season / Summer

Summit Geometrixx Outdoors : Activity / Running

Summit Geometrixx Outdoors : Apparel / Footwear

Summit Description

My Summit Description

ERP Data

SKU *

eqsmnr

A screenshot of the Adobe Commerce interface. On the left is a dark sidebar with navigation links: Commerce, Catalogs, Products, Collections, Orders, Payment Methods, and Shipping Methods. Above these is an 'Edit' button. The main area shows a product card for 'Nairobi Runners'. The card has a placeholder 'Product thumbnail' and the product name 'Nairobi Runners'. Below the card are two tabs: 'BASIC' (which is selected) and 'PRODUCT TABS'. Under 'BASIC', there are fields for 'Title *' (set to 'Nairobi Runners'), 'Tags' (empty), and three descriptive text boxes: 'Summit Geometrixx Outdoors : Season / Summer', 'Summit Geometrixx Outdoors : Activity / Running', and 'Summit Geometrixx Outdoors : Apparel / Footwear'. Under 'PRODUCT TABS', there is a field for 'Summit Description' (set to 'My Summit Description'). At the bottom is a section titled 'ERP Data' with a single field for 'SKU *' (set to 'eqsmnr').

Exercise 1 – Result 2

CRXDE Lite

Save All | Create ... | Delete | Copy Paste Move ... | Rename ... | Overlay Node ... | Team | Mixins ... | Tools | admin@crx.default | /etc/commerce/products/summit-geometrixx-outdoors/equipment/running/eqsmnr

cloudsettings
commerce
catalogblueprintimporter
collections
locations
orders
payment-methods
productimporter
products
geometrixx-outdoors
geometrixx-outdoors.csv
rep:policy
summit-geometrixx-outdoors
equipment
biking
hiking
jcr:content
running
eqsmmr
image
purple
pink
eqsmnr
image
blue
pink
geo-green
jcr:content
skiing

Home

CRXDE Lite Content Repository Extreme Development Environment Lite

Properties		Access Control	Replication	Console	Build Info					
Name	Type	Value				Protected	Mandatory	Multiple	Auto Created	
1	cq:commerceType	String	product			false	false	false	false	
2	cq:productSkuCode	String	eqsmnr			false	false	false	false	
3	cq:productVariantAxes	String[]	geo-color, size			false	false	true	false	
4	cq:tags	String[]	summit-geometrixx-outdoors.season/summer, summit-geometrixx-out... Edit	View		false	false	true	false	
5	description	String	Men's Sport Footwear			false	false	false	false	
6	features	String	 Lightweight upper with mesh windows for ventilation ... Edit	View		false	false	false	false	
7	identifier	String	eqsmnr			false	false	false	false	
8	jcr:lastModified	Date	2015-02-22T20:52:27.354+11:00			false	false	false	false	
9	jcr:mixinTypes	Name[]	cq:Taggable			true	false	true	false	
10	jcr:primaryType	Name	nt:unstructured			true	true	false	true	
11	jcr:summit	String	My Summit Description			false	false	false	false	
12	jcr:title	String	Nairobi Runners			false	false	false	false	
13	sling:resourceType	String	commerce/components/product			false	false	false	false	
14	summary	String	<p>An everyday runner that provides excellent support and comfort. Fe... Edit	View		false	false	false	false	

Name Type Value Multi Add Clear

© 2015 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

15

Adobe

Exercise 1 - Task Undertaken to get solution

- **Task :** Add an additional property to the product and make sure this is available for edit in scaffolding UI.
- Steps to follow
 1. Go to <http://localhost:4502/crx/de/index.jsp>
 2. Locate in tree **/etc/scaffolding/summit-geometrixx-outdoors/product/jcr:content/cq:dialog/content/items/tabs/items/basic/items/titles/items**
 3. Right click on items node and create a node with following properties (or copy paste **title** node and change accordingly)
- 4. Now navigate to any product for summit-geometrixx in product management window. Example: <http://localhost:4502/libs/commerce/gui/content/products/properties.html/etc/commerce/products/summit-geometrixx-outdoors/equipment/running/eqsmmr>
- 5. This will have the field created for edit and on filling the data, this is available for management.

Properties		Access Control		Replication		Console		Build Info						
	Name ▲	Type	Value							Protected	Mandatory	Multiple	Auto Created	
1	fieldLabel	String	Summit Description							false	false	false	false	
2	jcr:primaryType	Name	nt:unstructured							true	true	false	true	
3	name	String	.jcr:summit							false	false	false	false	
4	renderReadOnly	Boolean	true							false	false	false	false	
5	sling:resourceType	String	granite/ui/components/foundation/form/textfield							false	false	false	false	

Exercise 1 – Summary and Key Takeaways

- Import and Manage Products in Adobe Experience Manager !!
- Use Cases : As a user, I would like to manage my organisation's products in my content management system, so I will be able to use this information to present to my customers.
- Example :
 1. Product management.
 2. Product related pages.
 3. Compare feature.

Agenda

- 1 | Exercise 1 : Commerce Product Import Management
- 2 | **Exercise 2 : Commerce Page Rollout Mechanism Management**
- 3 | Exercise 3 : Using & Extending Commerce Integration Framework
- 4 | Exercise 4 : Extending ElasticPath Commerce Framework

Exercise 2

Commerce Page Rollout Mechanism Management

Exercise 2 – Commerce Page Rollout Mechanism Management

- **Tasks :**

Task 1: Create Website using catalog

Task 2: Add Price and see the change Reflected

Task 3 (Optional – Bonus Points) : Modify Section Template and Rollout

- **Time :**

15 min

Exercise 2 -Setup

Setup

1. In your terminal
2. Enter the following

```
sh install-exercise2-setup.sh
```

Exercise 2 – Observation 1 – Rollout Framework

Observation 1 – Rollout Framework

1. Go to <http://localhost:4502/projects.html>
2. Navigate to Commerce > Catalogs
3. Click Summit Geometrixx Outdoors
4. **Observe:**
 1. **Template Pages**
 - Catalog Page Template
 - Section Page Template
 - Product Page Template
 2. **Base Catalog**
 - Equipment
 - Biking
 - Hiking
 - Running
 - Skiing
 - Sunglasses
 - Surfing
 - swimming

The screenshot shows the Adobe Marketing Cloud Commerce interface. On the left, a sidebar lists Commerce, Catalogs, Products, Collections, Orders, Payment Methods, and Shipping Methods. The main area is titled 'Summit Geometrixx Outdoors' and shows a 'Template Pages' section with a placeholder icon and a 'Base Catalog' section featuring a thumbnail of a website page for 'Summit Geometrixx Outdoors'.

Exercise 2 – Observation 2 – Template and Properties

Observation 2 – Template and Properties

1. Go to
<http://localhost:4502/aem/catalogs.html/content/catalogs/summit-geometrixx-outdoors/template-pages>
2. Click on Edit on Catalog Page.
 Observation: Page Template and components
3. Click on Edit on Section Page
 Observation: Page Template and components
4. Click on Edit on Product Page
 Observation: Page Template and components

This represents how the page template which will be used for roll out.

Exercise 2 – Observation 3 – Catalog Blueprint and Properties

Observation 3 - Catalog Blueprint and Properties

1. Go to <http://localhost:4502/aem/catalogs.html/content/catalogs/summit-geometrixx-outdoors>

2. Click on View Properties on Base Catalog

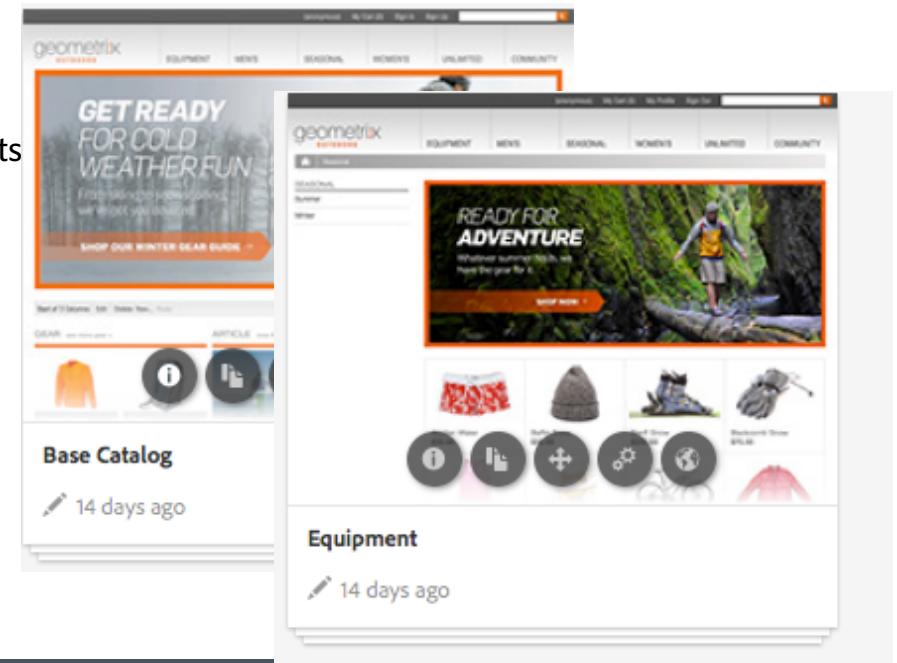
Observation: Template pages associated to Blueprints

3. Press Back and

Click on "Equipments" and subsequent subnodes

Observation: Note the Products tab and

tag specification.

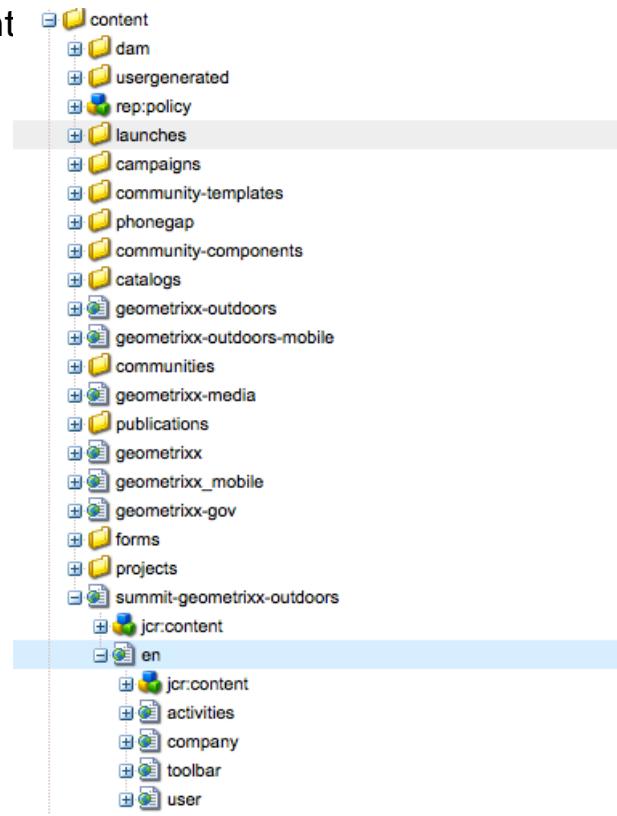


This represents blue print for the pages roll out.

Exercise 2 – Observation 4 – Notice the pages for the summit-outdoors content

Observation 4: Notice the pages for the summit-outdoors content

1. Go to <http://localhost:4502/crx/de/index.jsp>
2. Go to node - /content/summit-geometrixx-outdoors/en
3. Observer there is no **equipments** node

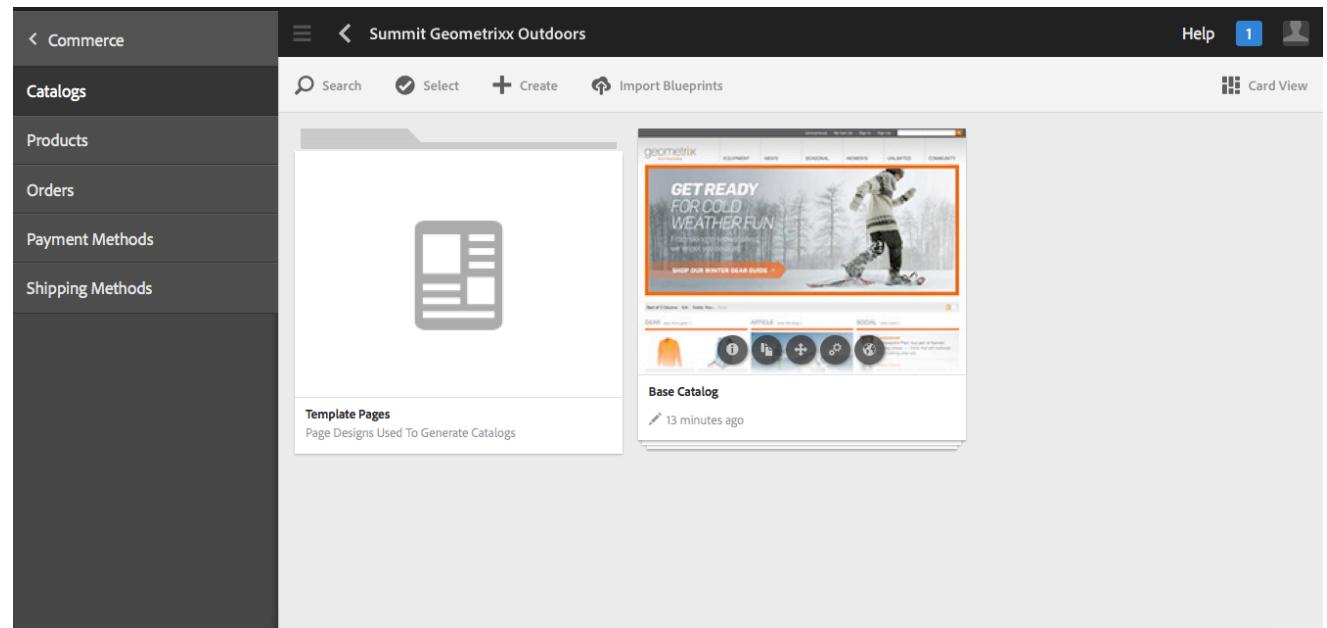


Exercise 2 – Task 1

Create Website using catalog

Exercise 2 - Task 1 - Create Website using catalog

- Go to <http://localhost:4502/aem/catalogs.html/content/catalogs/summit-geometrixx-outdoors>
- Select Rollout Pages on Base Catalog
- Press Rollout Changes



Exercise 2 – Task 1 – Result – Rolled out website

Result: Rolled out Website from Exercise2-Task1

- Go to <http://localhost:4502/crx/de/index.jsp#/content/summit-geometrixx-outdoors/en>
 - New node "equipment"
 - Double click "en" . This will reflect the template for the catalog
<http://localhost:4502/editor.html/content/catalogs/summit-geometrixx-outdoors/template-pages/catalog-page-template.html>
 - Double click on "equipment". This will reflect the template for the section
<http://localhost:4502/editor.html/content/catalogs/summit-geometrixx-outdoors/template-pages/section-page-template.html>
 - Double click on a product under "equipment". This will reflect the template for the product
<http://localhost:4502/editor.html/content/catalogs/summit-geometrixx-outdoors/template-pages/product-page-template.html>

Exercise 2 – Task 2

Add Price and see the change
Reflected.

Exercise 2 – Task 2 – Observation 1 – Price on products

- Go to <http://localhost:4502/content/summit-geometrixx-outdoors/en/equipment/running.html>

- **Observation 1:**
No Price for the products.

RUNNING
Mombassa Runners
Nairobi Runners



Exercise 2 - Task 2 - Action - Add Price

- **Go to**
<http://localhost:4502/libs/commerce/gui/content/products/properties.html/etc/commerce/products/summit-geometrixx-outdoors/equipment/running/eqsmnr>
- **Edit** Product and add price (Example: 50.00)
- **Go to** <http://localhost:4502/content/summit-geometrixx-outdoors/en/equipment/running.html>

Exercise 2 – Task 2 – Result – Price reflected as changed through PIM

- **Result:** Price appears for Nairobi Runner

The screenshot shows a website interface for running gear. At the top, there's a navigation bar with icons for Home, equipment, and Running. Below the navigation, a sidebar menu is open under the 'RUNNING' category, listing 'Mombassa Runners' and 'Nairobi Runners'. The main content area features a large promotional banner with the text 'HIT THE GROUND RUNNING' and 'Our annual 50% sale means you can gear up for you run from head to toe.' A blue button labeled 'SEE WHAT'S ON SALE →' is visible. To the right of the banner, a male runner in a purple tank top and black shorts is shown stretching his right leg. Below the banner, there are two product cards. The first card on the left shows a pair of pink running shoes with the text 'Mombassa Runners \$0.00'. The second card on the right shows a pair of grey and blue running shoes with the text 'Nairobi Runners \$40.00'.

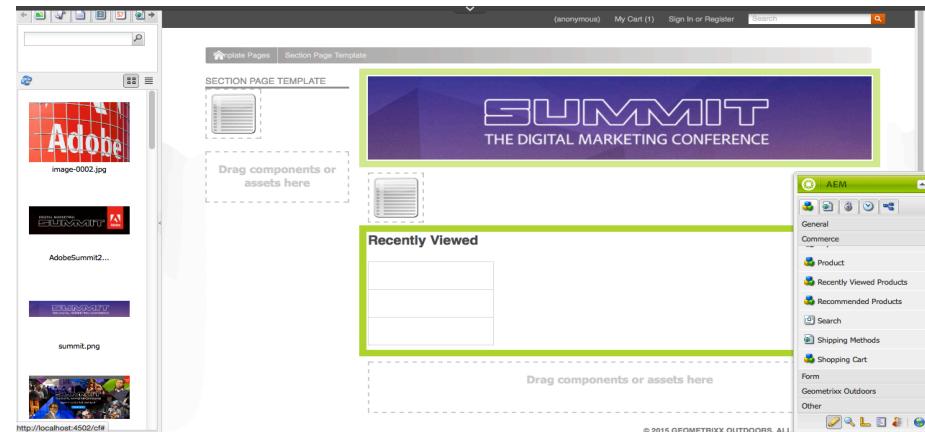
Exercise 2 - Task 3 (Optional)

Modify Section
Template and Rollout

Exercise 2 - Task 3 - Modify Section Template and Rollout

Task

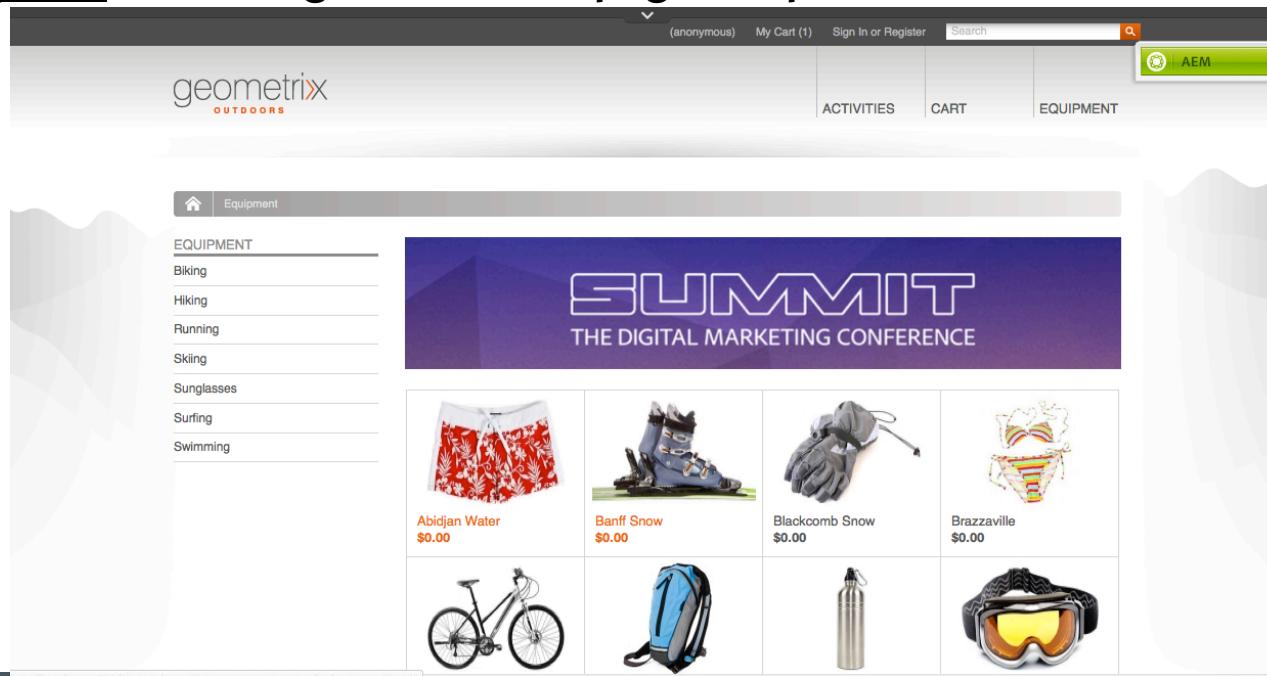
- Go to <http://localhost:4502/editor.html/content/catalogs/summit-geometrixx-outdoors/template-pages/section-page-template.html>
- Drag and Drop another Banner image
- Go to <http://localhost:4502/aem/catalogs.html/content/catalogs/summit-geometrixx-outdoors>
- Select Rollout Pages on Base Catalog
- Press Rollout Changes



Exercise 2 - Task 3 - Result - Section page change reflected

- **Result:** Go to

<http://localhost:4502/content/summit-geometrixx-outdoors/en/equipment/running.html> The changes to section page template is now reflected on equipments.



Exercise 2 -Solution

Setup

1. In your terminal
2. Enter the following

```
sh install-exercise2-solution.sh
```

Exercise 2 – Summary and Key Takeaway

The PIM managed from Exercise 1 is rolled out as pages using Page Rollout Hook in AEM Commerce Integration framework.

Use Cases : As a user, I would like to create consistent pages using blueprint template for product categories, sections and product pages, so I will be able to reduce effort in replicating page creation process for each page in authoring interface.

Example :

Product management.

Product related pages.

Compare feature.

Agenda

- 1 | Exercise 1 : Commerce Product Import Management
- 2 | Exercise 2 : Commerce Page Rollout Mechanism Management
- 3 | Exercise 3 : Using & Extending Commerce Integration Framework**
- 4 | Exercise 4 : Extending ElasticPath Commerce Framework

Exercise 3

Commerce Integration Framework and Custom implementation

Exercise 3 – Commerce Integration Framework and Custom implementation

- **Tasks :**

Task 1: Build a simple product component.

Task 2 (Optional - Advanced): Extending Commerce Integration Framework (CIF)

- **Total Time :**

20 min

WARNING: Cleanup required and done through script in setup.



Exercise 3 – Task 1

Using CIF for Commerce Components

Exercise 3 - Task1 -Setup

Setup

1. In your terminal
2. Enter the following

```
sh install-exercise3-task1-setup.sh
```

Exercise 3 - Task1 - Solution

Solution - (Optional - If you don't want to follow the steps in following slides for Exercise 3 - Task1)

1. In your terminal
2. Enter the following

```
sh install-exercise3-task1-solution.sh
```

Exercise 3 – Task 1 – Custom Product component

The product.jsp implementation contains several tasks, which you should complete to get your first own product component running (you'll find the Tasks as comments inside the JSP file – **Please uncomment them !!**):

Open [aem-commerce-framework-extensions/content/src/main/content/jcr_root/apps/commerce-extentions/components/product/product.jsp](#)

- **Step 1:** Get a CommerceService instance

```
CommerceService commerceService = resource.adaptTo(CommerceService.class);
```

- **Step 2:** Get the CommerceSession of the current user

```
CommerceSession session = commerceService.login(slingRequest, slingResponse);
```

- **Step 3:** Load the base Product

```
Node node = resource.adaptTo(Node.class);
String productDataPath = node.getProperty("productData").getString();
Product baseProduct = resourceResolver.resolve(productDataPath).adaptTo(Product.class);
```

Exercise 3 – Task 1 – Custom Product component

- **Step 4:** Get product description to be from Summit Description. Open `aem-commerce-framework-extensions/content/src/main/content/jcr_root/apps/commerce-extentions/components/product/product_header.jsp`
- **Step 5:** Add

```
<%= XSSAPI.encodeForHTML(product.getProperty("jcr:summit", String.class)) %>
```

under

```
<h2 class="product-description"><%= XSSAPI.encodeForHTML(product.getDescription()) %></h2>
```

Exercise 3 – Task 1 – Custom Product component

- **Step 6:** Save the products.jsp and product_header.jsp file

- **Step 7:** Repoint Template product to new components.

So go into `src/main/content/jcr_root/content/catalogs/summit-geometrixx-outdoors/template-pages/product-page-template/.content.xml`

Change

`commerce/components/product`

to

`commerce-extentions/components/product`

Exercise 3 – Task 1 – Custom Product component

- **Step 8:** Run

```
mvn clean install -PautoInstallPackage
```

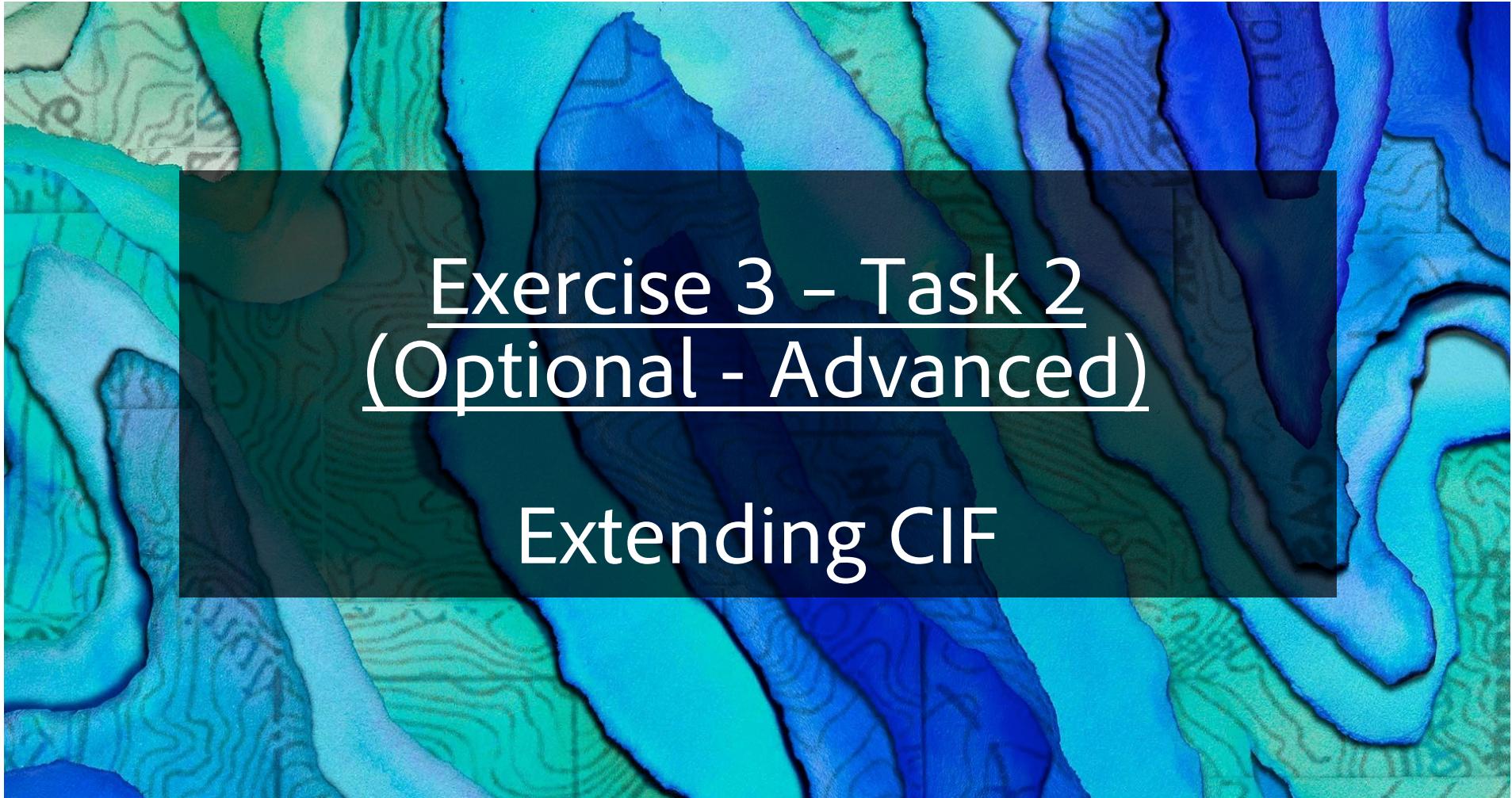
- **Step 9:** Go to <http://localhost:4502/aem/catalogs.html/content/catalogs/summit-geometrixx-outdoors>
- **Step 10:** Select Rollout Pages on Base Catalog
- **Step 11:** Press Rollout Changes
- **Step 12:** Go here to view change -
<http://localhost:4502/content/summit-geometrixx-outdoors/en/equipment/running.html>

Exercise 3 - Task 1 - Result - Product page with Summit Description form Exercise1

- **Result:** Go to

<http://localhost:4502/content/summit-geometrixx-outdoors/en/equipment/running/nairobi-runners.html> : The Summit Description appears on product page ☺

The screenshot shows a product page for 'Nairobi Runners' running shoes. At the top, there's a navigation bar with icons for Home, Equipment, Running, and Nairobi Runners. Below the navigation is a large image of two grey and blue running shoes. To the right of the image, the product title 'Nairobi Runners' is displayed in bold, followed by 'My Summit Description'. The price '\$40.00' is shown with a small note 'eqsmnr.b-7.5-7.5'. Below the price is a 'Rating' section with a 5-star icon and a link to 'Tweet'. A note indicates 'Average of 0 ratings: 0.00 stars'. On the right side, there are two sections: 'Size' and 'Quantity'. The 'Size' section contains a grid of buttons for sizes 7.5, 11, 10, 9, 8, 7, 12, 13, 8.5, 9.5, 11.5, 10.5, 5.5, 10.5, 6.5, 9, 7.5, 8.5, geo-5, and 6. The 'Quantity' section has a single input field.



Exercise 3 – Task 2 (Optional - Advanced)

Extending CIF

Exercise 3 - Task 2 - Extend and create own Commerce Framework

- **Task :** Extend and create own Commerce Framework to create custom Product model to display **SummitDescription** from a Custom Product Model

WARNING: Cleanup required and done through script in setup.

Exercise 3 - Task2 -Setup

Setup

1. In your terminal
2. Enter the following

```
sh install-exercise3-task2-setup.sh
```

Exercise 3 – Task2 – Solution

Solution (Optional – If you don't want to follow the steps in following slides for Exercise 3 – Task2)

1. In your terminal
2. Enter the following

```
sh install-exercise3-task2-solution.sh
```

Exercise 3 - Task 2 - Observation 1 - pom.xml (Dependencies)

- **Observation 1 :** Review the commerce and other related maven dependencies in
 - aem-commerce-framework-extensions/bundle/pom.xml
 - aem-commerce-framework-extensions/content/pom.xml
 - aem-commerce-framework-extensions/pom.xml

```
<dependency>
    <groupId>com.day.cq.wcm</groupId>
    <artifactId>cq-wcm-api</artifactId>
    <version>5.6.6</version>
    <scope>provided</scope>
</dependency>

<dependency>
    <groupId>com.day.cq</groupId>
    <artifactId>cq-commons</artifactId>
    <version>5.8.12</version>
    <scope>provided</scope>
</dependency>

<!-- Commerce specific dependencies -->

<dependency>
    <groupId>com.adobe.cq.commerce</groupId>
    <artifactId>cq-commerce-core</artifactId>
    <version>5.12.32</version>
    <scope>provided</scope>
</dependency>
<dependency>
    <groupId>com.adobe.cq.commerce</groupId>
    <artifactId>cq-commerce-pim</artifactId>
    <version>5.9.60</version>
    <scope>provided</scope>
</dependency>
<dependency>
    <groupId>com.adobe.granite</groupId>
    <artifactId>com.adobe.granite.workflow.api</artifactId>
    <version>1.0.2</version>
    <scope>provided</scope>
</dependency>
```

Exercise 3 – Task 2 – Observation 2 – Key extension points

- **Observation 2 :** Review the commerce extension interfaces
 - SummitCommerceService
 - SummitCommerceServiceFactory
 - SummitCommerceSession
 - DefaultSummitCommerceServiceFactory (Note the commerce provider name)
 - SummitCommerceServiceImpl.java
 - SummitCommerceSessionImpl.java

Exercise 3 – Task 2 – Setup

- **Step 1: Clean Project and Refresh Eclipse workspace (Clean Project and Refresh)**

Exercise 3 - Task 2 - Extend and create own Commerce Framework

- **Step 1:** Open aem-commerce-framework-extensions/bundle/src/main/java/com/adobe/cq/commerce/custom/impl/DefaultSummitCommerceServiceFactory

Uncomment

```
return new SummitCommerceServiceImpl(getServiceContext(), resource); // Step 1
```

Exercise 3 - Task 2 - Extend and create own Commerce Framework

- **Step 2:** Open `aem-commerce-framework-extensions/bundle/src/main/java/com/adobe/cq/commerce/custom/impl/SummitCommerceServiceImpl`

Uncomment section commented as Step2

```
@Override  
public Product getProduct(final String path) throws CommerceException {  
    /*  
     * Step 2  
     */  
    Resource resource = resolver.getResource(path);  
    if (resource != null && SummitProduct.isAProductOrVariant(resource)) {  
        return new SummitProduct(resource);  
    }  
    return null;  
}
```

Exercise 3 – Task 2 – Extend and create own Commerce Framework

- **Step 3:** Open `aem-commerce-framework-extensions/bundle/src/main/java/com/adobe/cq/commerce/custom/impl/SummitCommerceServiceImpl`

Uncomment section marked between Step 3

```
/* Step 3 – Uncomment
@Override
public CommerceSession login(SlingHttpServletRequest request,
    SlingHttpServletResponse response) throws CommerceException {
    return new SummitCommerceSessionImpl(this, request, response, resource);
}
*/
```

Exercise 3 – Task 2 – Extend and create own Commerce Framework

- **Step 4:** Add custom method to pull Summit Description [aem-commerce-framework-extensions/bundle/src/main/java/com/adobe/cq/commerce/custom/SummitProduct.java](#)

Uncomment

```
/* Step 4
public String getSummitDescription() {
    return getProperty("jcr:summit", String.class);
}
*/
```

Exercise 3 – Task 2 – Extend and create own Commerce Framework

- **Step 5:** Get product description to be from Summit Description from new Product domain model.

Open **aem-commerce-framework-extensions/content/src/main/content/jcr_root/apps/commerce-extentions/components/product/product_header.jsp**

Observe

```
<!-- Step 5 -->
<%= XSSAPI.encodeForHTML(((SummitProduct)product).getSummitDescription()) %>
```

Exercise 3 - Task 2 - Extend and create own Commerce Framework

- **Step 7:** Run following in terminal

```
mvn clean install -PautoInstallPackage
```

- **Step 8:** Go to <http://localhost:4502/aem/catalogs.html/content/catalogs/summit-geometrixx-outdoors>
- **Step 9:** Select Rollout Pages on Base Catalog
- **Step 10:** Press Rollout Changes
- **Step 11:** Go here to view change -
<http://localhost:4502/content/summit-geometrixx-outdoors/en/equipment/running/nairobi-runners.html>

Exercise 3 - Task 2 - Result - Product page with Summit Description form Exercise1

- **Result:** Go to

<http://localhost:4502/content/summit-geometrixx-outdoors/en/equipment/running/nairobi-runners.html> : The Summit Description appears on product page ☺

The screenshot shows a product page for 'Nairobi Runners' running shoes. At the top, there's a navigation bar with icons for Home, Equipment, Running, and Nairobi Runners. Below the navigation is a large image of two grey and blue running shoes. To the right of the image, the product title 'Nairobi Runners' is displayed in bold, followed by 'My Summit Description'. The price '\$40.00' is shown with a small note 'eqsmnr.b-7.5-7.5'. Below the price is a 'Rating' section with a 5-star icon and a 'Tweet' button. A note indicates 'Average of 0 ratings: 0.00 stars'. On the right side, there are two sections: 'Size' and 'Quantity'. The 'Size' section contains a grid of buttons for sizes 7.5, 11, 10, 9, 8, 7, 12, 13, 8.5, 9.5, 11.5, 10.5, 5.5, 10.5, 6.5, 9, 7.5, 8.5, geo-5, and 6. The 'Quantity' section has a single input field.

Exercise 3 – Task2 – Solution

Solution (Optional – If you don't want to follow the steps in following slides for Exercise 3 – Task2)

1. In your terminal
2. Enter the following

```
sh install-exercise3-task2-solution.sh
```

Exercise 3 – Summary and Key Takeaway

Custom Commerce Implementation based on AEM Commerce Integration framework.

Use Cases : As a user, I would like to create my own commerce implementation, so that I can use my own commerce engine or extend and manage the products from JCR.

Example :

Custom Implementations

Custom Products

Agenda

- 1 | Exercise 1 : Commerce Product Import Management
- 2 | Exercise 2 : Commerce Page Rollout Mechanism Management
- 3 | Exercise 3 : Using & Extending Commerce Integration Framework
- 4 | Exercise 4 : Extending ElasticPath Commerce Framework**

Exercise 4

Extending Elastic Path Commerce Framework

Exercise 4 – Extending Elastic Path Commerce Framework

- **Tasks :**

Task 1: Add a new attribute to EP Commerce Catalog and make it reflect in AEM Product Page

Task 2 (Advanced - Optional): Extend and create own Elastic Path Commerce Framework

Task 3 (Advanced - Optional): Dynamic Price

- **Time :**

20 min

- **WARNING:** Cleanup (Script Provided - `/Users/l716/Desktop/AEM-Commerce-Lab-WS/aem-commerce-framework-extensions`) and restart

Exercise 4 – Setup 1/2 – Elastic Path

Setup 1/2 (No need to start Elastic Path if already running)

- Open new terminal.
- Navigate to /Users/l716/elasticpath

```
cd /Users/l716/elasticpath
```

- Start Elastic Path servers

```
sh start.sh
```

- View image to confirm startup successful

```
Waiting for ep-cmserver to start...
Cortex Studio: http://localhost:9080/studio
Search Server: http://localhost:8082/searchserver/product/select?q=*&wt=json
CM Server: http://localhost:8781/cmserver
ActiveMQ: http://localhost:18081/jms
```

Exercise 4 – Setup 2/2 – Elastic Path

Setup 2/2

1. In your terminal
2. Enter the following

```
sh install-exercise4-setup.sh
```

Note: The above script is a simple git branch change, cleanup of aem instance and a maven install of the setup for Lab 4.

Exercise 4 – Task 1

View and Change
Product details in EP
to reflect in AEM

Exercise 4 – Task 1 – View and Change Product details in EP to reflect in AEM

- **Task :** Add a new attribute to EP Commerce Catalog and make it reflect in AEM Product Page

Exercise 4 – Task 1a – View the existing product attributes

Open Commerce Manager

1. Click Commerce Manager Icon from DOCK
2. Switch to Catalog Manager Activity
3. Open Geometrixx Outdoors catalog
4. Navigate to women -> shirts
5. Open Tupai Summer
6. Note attributes on attribute tab



Exercise 4 – Task 1a – Observation 1 - View the existing product attributes – Result Screenshot

Observation 1: Product in EP Commerce Manager

The screenshot shows the EP Commerce Manager interface. On the left, there is a navigation sidebar with various catalog entries. In the center, there is a 'Product Listing' window showing five products. Below it, a specific product's attributes are displayed.

Product Listing:

Product Code	Product Name	Product Type	Brand	Default Category	Active
wmapbb	Bora Bora	geometrixxWithSiz	Geometrixx	shirts	Yes
wmapmm	Maui Marine	geometrixxWithSiz	Geometrixx	shirts	Yes
wmapnh	Norfolk Highland	geometrixxWithSiz	Geometrixx	shirts	Yes
wmappl	Palau Summer	geometrixxWithCol	Geometrixx	shirts	Yes
wmaptp	Tupai Summer	geometrixxWithSizi	Geometrixx	shirts	Yes

Attributes:

Name	Type	Required	Multi-Language	Value
Features	Long Text	No	Yes	 • Women's Frock • Lightweight cotton in a variety
Description	Short Text	No	No	
BBSummary	Long Text	No	No	
Tag	Short Text (Mul)	No	No	geometrixx-outdoors:gender/wom

Buttons: Language: English | Edit Attribute Value... | Clear Attribute Value

Bottom Navigation: Summary | Pricing | Image | Attributes | SKU Details | Category Assignment | SEO | Merchandising Associations

Logged in as: admin Server URL: http://localhost:8781/cmserver/

Exercise 4 – Task 1b – Add a new product attributes – **SummitDescription**

In Commerce Manager

1. Switch to Catalog Manager Activity
2. Right-click Geometrixx Outdoors catalog and open
3. Edit attributes and add new **SummitDescription** product attribute (short-text)
4. Save
5. Click Product Types
6. Edit the GeometrixxWithSize product type, add **SummitDescription** attribute to assigned attributes.
7. Save

Exercise 4 – Task 1b – Add a new product attributes – Add an attribute value to the product

Switch to Catalog Manager Activity

1. Open Geometrixx Outdoors catalog
2. Navigate to women -> shirts
3. Open Tupai Summer (or click the Refresh button if it's already open)
4. Note new **SummitDescription** attribute
5. Set a value for **SummitDescription**
6. Save

Exercise 4 - Task 1b - Add a new product attributes - Screenshot

Observation 2: Product in EP Commerce Manager

The screenshot shows the EP Commerce Manager interface. On the left, there is a navigation sidebar with various catalog and virtual catalog entries. The main area displays a 'Product Listing' grid with columns: Product Code, Product Name, Product Type, Brand, Default Category, and Active. Five products are listed: Bora Bora, Maui Marine, Norfolk Highland, Palau Summer, and Tupai Summer, all categorized under shirts. Below the listing, a specific product's attributes are shown in a detailed view. The 'Attributes' table has columns: Name, Type, Required, Multi-Language, and Value. The 'Value' column contains several descriptive text entries. At the bottom of the screen, there are tabs for Summary, Pricing, Image, Attributes, SKU Details, Category Assignment, SEO, and Merchandising Associations. A status bar at the bottom indicates the user is logged in as 'admin' and provides the server URL.

Name	Type	Required	Multi-Language	Value
Features	Long Text	No	Yes	 Women's Frock <>Lightweight cotton in a variety geometrixx-outdoors:gender/wom Varun's Summit Description
Description	Short Text	No	No	
BBSummary	Long Text	No	No	
Tag	Short Text (Mul)	No	No	
SummitDescription	Short Text	No	No	

Summary | Pricing | Image | **Attributes** | SKU Details | Category Assignment | SEO | Merchandising Associations

Logged in as: admin Server URL: http://localhost:8781/cmserver/

Exercise 4 – Task 1c Import the new attribute to AEM

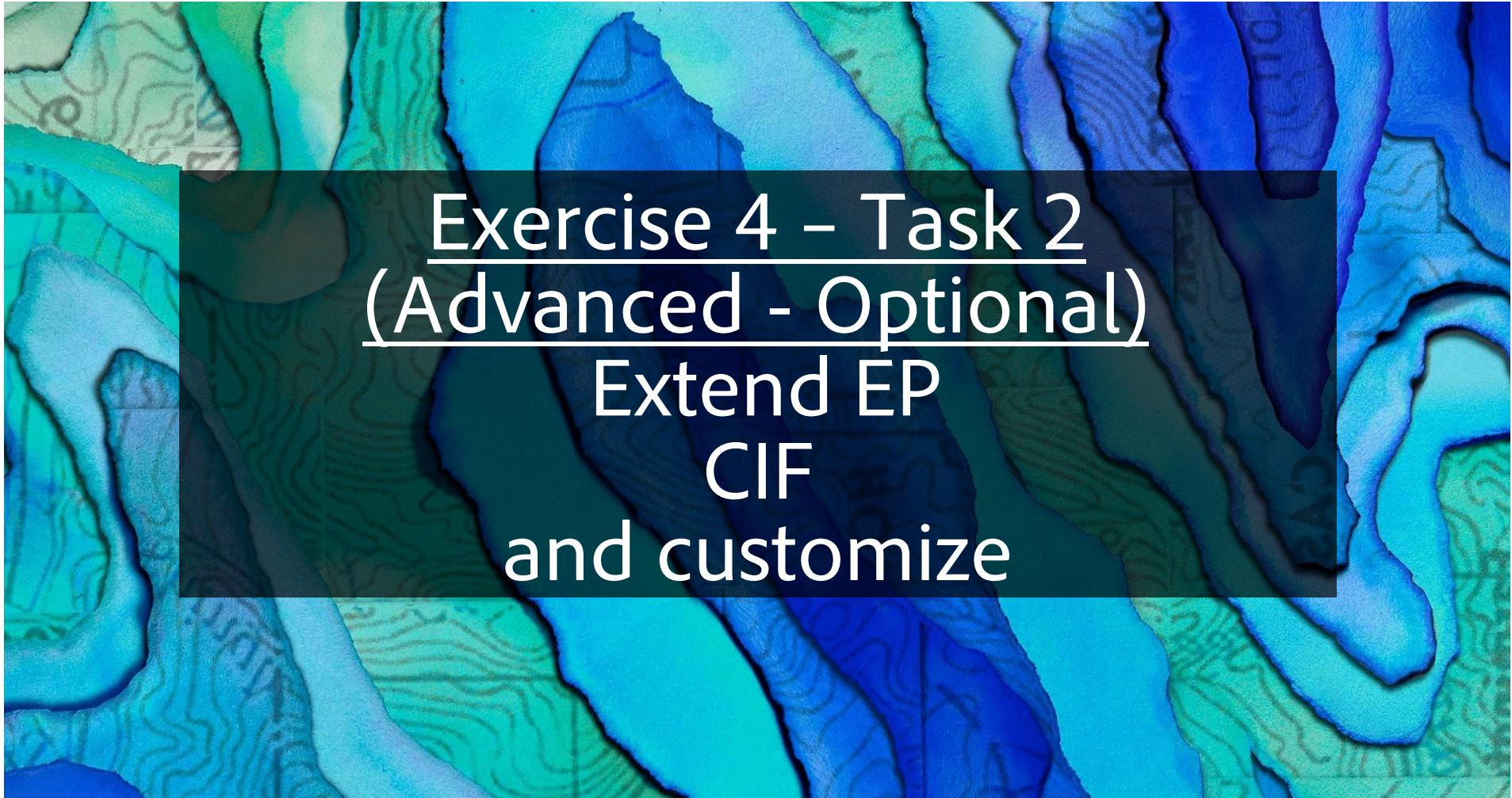
1. Go to <http://localhost:4502/etc/importers/elasticpath.html> (sorry new Touch UI importer isn't ready yet)
2. Set Catalog to geometrixx
3. Click Import Products
4. View the product in CRX. Note new attribute has been imported as a property:
5. <http://localhost:4502/crx/de/index.jsp#/etc/commerce/products/geometrixx/women/shirtsw/wmapt>

Exercise 4 - Task 1 - Result - Import the new attribute to AEM - Screenshot

Result 1: **SummitDescription** attribute on Product in EP is now in AEM 😊

The screenshot shows the CRXDE Lite interface with the path `/etc/commerce/products/geometrixx/women/shirtsw/wmapt` selected in the left navigation tree. The right panel displays the properties of this node, specifically the 'Properties' tab. The 'SummitDescription' property is listed with a value of "Varun's Summit Description". Other properties shown include `cq:commerceType` (product), `cq:productSkuCode` (wmapt), `cq:productVariantAxes` (size), `cq:tags` (geometrixx-outdoors:gender/women, geometrixx-out...), `description` (Women's Frock), `features` (Lightweight cotton twill Made in Tha...), `identifier` (wmapt), `jcr:lastModified` (2015-02-20T22:02:51.610+11:00), `jcr:mixinTypes` (cq:Taggable), `jcr:primaryType` (nt:unstructured), `jcr:title` (Tupai Summer), `sling:resourceType` (commerce/components/product), and `summary` (cool, lightweight cotton in a variety of colors. Perfect...).

Name	Type	Value	Protected	Mandatory
1 SummitDescription	String	Varun's Summit Description	false	false
2 cq:commerceType	String	product	false	false
3 cq:productSkuCode	String	wmapt	false	false
4 cq:productVariantAxes	String[]	size	false	false
5 cq:tags	String[]	geometrixx-outdoors:gender/women, geometrixx-out...	false	false
6 description	String	Women's Frock	false	false
7 features	String	 Lightweight cotton twill Made in Tha...	false	false
8 identifier	String	wmapt	false	false
9 jcr:lastModified	Date	2015-02-20T22:02:51.610+11:00	false	false
10 jcr:mixinTypes	Name[]	cq:Taggable	true	false
11 jcr:primaryType	Name	nt:unstructured	true	true
12 jcr:title	String	Tupai Summer	false	false
13 sling:resourceType	String	commerce/components/product	false	false
14 summary	String	cool, lightweight cotton in a variety of colors. Perfect...	false	false



Exercise 4 – Task 2 (Advanced - Optional)

Extend EP CIF and customize

Exercise 4 – Task 2 – Extend CIF with Elastic Path CIF

- **Task :** Extend EP Commerce Integration Framework and customize to show property using Product domain model

Exercise 4 – Task2 -Setup

Setup

1. In your terminal
2. Enter the following

```
sh install-exercise4-task2-setup.sh
```

Note: The above script is a simple git branch change, cleanup of aem instance and a maven install of the setup for Lab 4.

Exercise 4 – Task2 – Solution

Solution (Optional – If you don't want to follow the steps in following slides or we run out of time)

1. In your terminal
2. Enter the following

```
sh install-exercise4-task2-solution.sh
```

3. Refresh workspace (Clean Project and Refresh)

Exercise 4 - Task 2 - Observation 1 - pom.xml (Dependencies)

Observation 1 :

Review the Elasticpath CIF extention related maven dependencies in

- /bundle/pom.xml
- /content/pom.xml
- /pom.xml

```
<!-- Elasticpath Commerce Integration Framework related dependencies -->
<dependency>
    <groupId>com.elasticpath.aem.commerce</groupId>
    <artifactId>cq-commerce-elasticpath-impl</artifactId>
    <version>0-SNAPSHOT</version>
    <scope>provided</scope>
</dependency>

<dependency>
    <groupId>com.elasticpath.rest</groupId>
    <artifactId>cortex-jaxrs-client</artifactId>
    <version>0-SNAPSHOT</version>
    <scope>provided</scope>
</dependency>

<!-- EP - CIF - Rest client to communicate with EP server -->
<dependency>
    <groupId>javax.ws.rs</groupId>
    <artifactId>javax.ws.rs-api</artifactId>
    <version>2.0.1</version>
    <scope>provided</scope>
</dependency>

<!-- EP - CIF - Sling Adaptor to communicate with EP server -->
<dependency>
    <groupId>org.apache.sling</groupId>
    <artifactId>org.apache.sling.adapter</artifactId>
    <version>2.0.4</version>
    <scope>provided</scope>
</dependency>

<!-- EP - CIF - CQ Address to communicate with EP server -->
<dependency>
    <groupId>com.adobe.cq</groupId>
    <artifactId>cq-address</artifactId>
    <version>1.1.6</version>
    <scope>provided</scope>
</dependency>

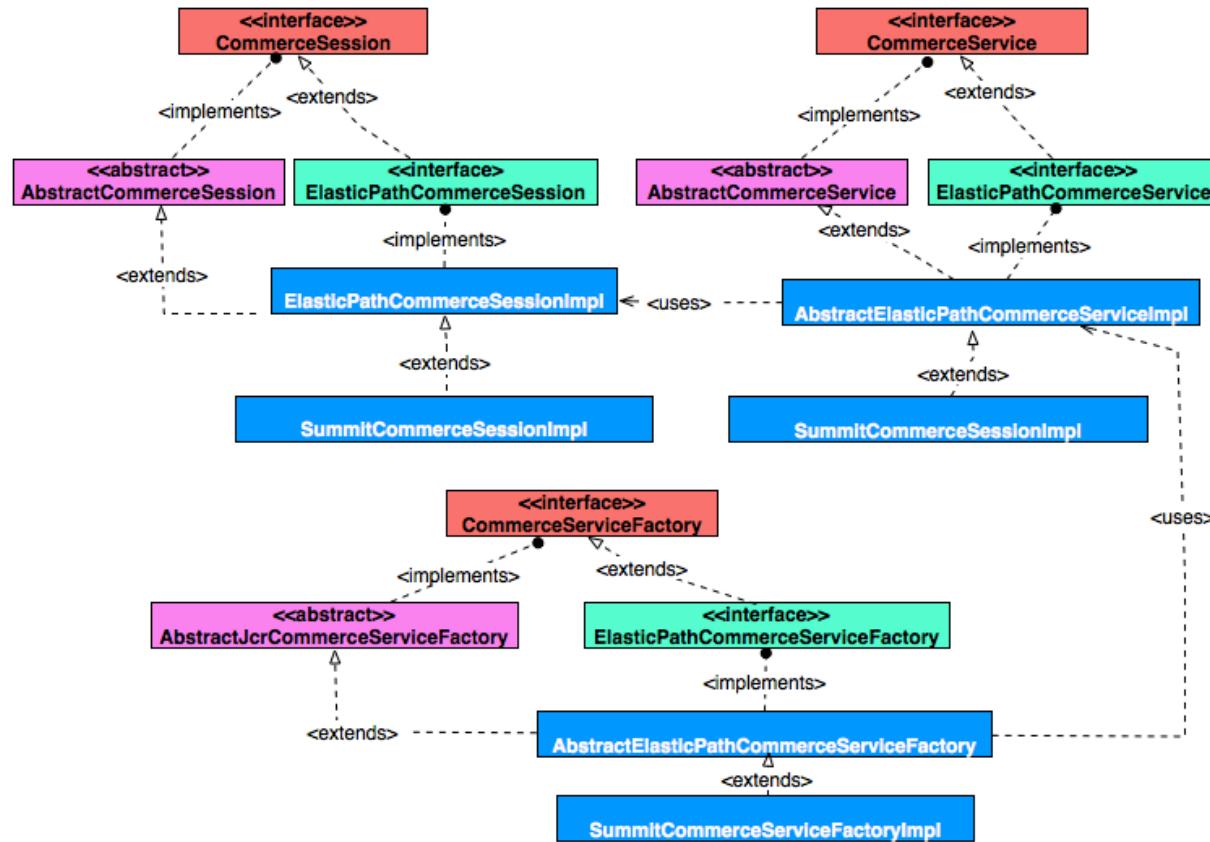
<!-- EP - CIF - Utility services for communication with EP server -->
<dependency>
    <groupId>com.google.guava</groupId>
    <artifactId>guava</artifactId>
    <version>15.0</version>
    <scope>provided</scope>
</dependency>
<dependency>
    <groupId>commons-validator</groupId>
    <artifactId>commons-validator</artifactId>
    <version>1.4.0</version>
    <scope>provided</scope>
</dependency>
<dependency>
    <groupId>org.apache.commons</groupId>
    <artifactId>commons-lang3</artifactId>
    <version>3.3.2</version>
    <scope>provided</scope>
</dependency>
<dependency>
    <groupId>com.adobe.granite</groupId>
    <artifactId>com.adobe.granite.ui.commons</artifactId>
    <version>5.5.138</version>
    <scope>provided</scope>
</dependency>
<dependency>
    <groupId>com.day.cq</groupId>
    <artifactId>cq-personalization</artifactId>
    <version>5.6.6</version>
    <scope>provided</scope>
</dependency>
```

Exercise 4 – Task 2 – Observation 2 – Key extension points

Observation 2:

- `SummitCommerceServiceFactoryImpl`
- `SummitCommerceServiceImpl`
- `SummitCommerceSessionImpl`

Exercise 4 - Task 2 - Observation 2 - Key Classes Extension



Exercise 4 – Task 2 – Observation 3 – Key extension points

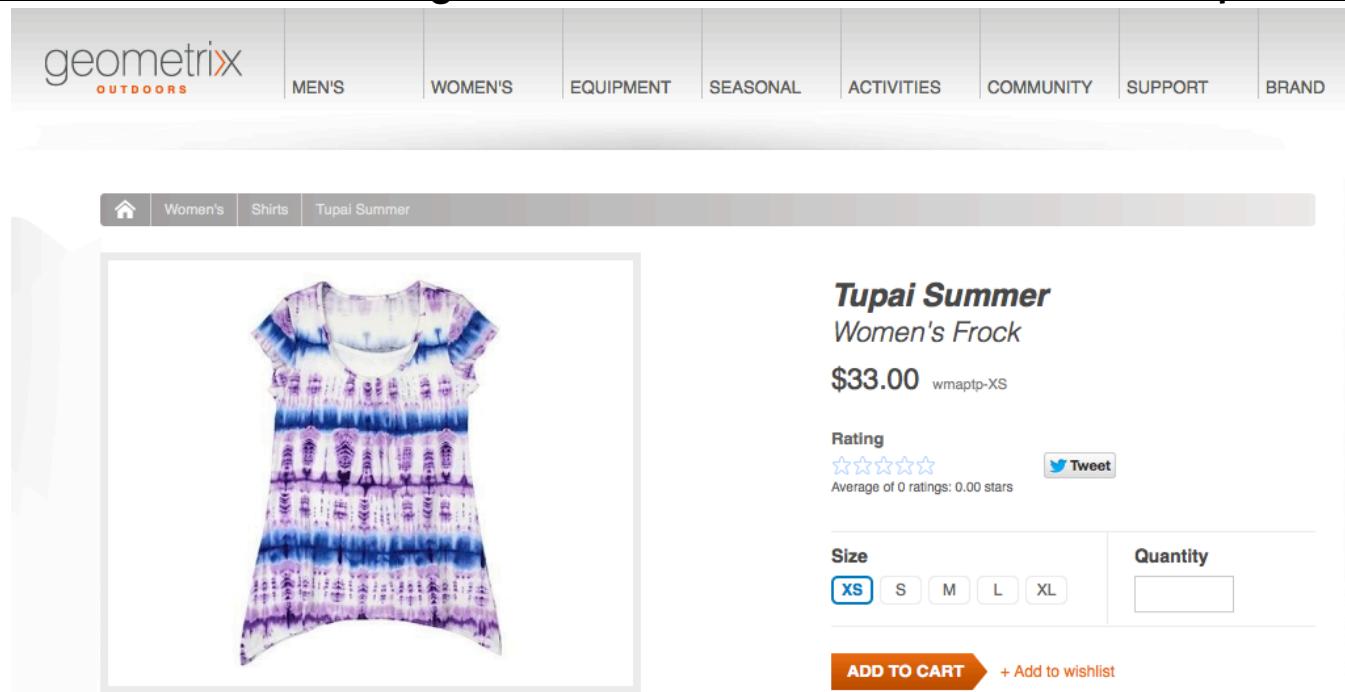
Observation 3: Associated Packages

- com.adobe.summit.commerce.custom.rest.sdk.impl
- com.adobe.summit.commerce.custom.rest.sdk.internal
- com.adobe.summit.commerce.custom.views
- com.adobe.summit.commerce.custom.impl.views.components

Exercise 4 – Task 2 – Observation 4 – Key extension points

- **Observation 4: Description on Product**

Go to http://localhost:4502/content/geometrixx-outdoors/en_US/women/shirts/tupai-summer.html



Exercise 4 - Task 2 - Create custom Elastic Path Product for Summit

- **Step 1:** Modify Custom Product for Summit

- **Name and extention:** SummitProduct extends ElasticPathProductImpl

- **Location:**

- /Users/l716/Desktop/AEM-Commerce-Lab-WS/aem-commerce-framework-extensions/bundle/src/main/java/com/adobe/cq/commerce/custom/impl

- **Uncomment :**

```
/* Step 1
@Override
public String getDescription() {
    String description = getProperty(SUMMIT_DESCRIPTION,
String.class);
    if (StringUtils.isEmpty(description)) {
        return StringUtils.EMPTY;
    }
    return description;
}*/
```

Exercise 4 – Task 2 – Create custom Elastic Path Product for Summit

- **Step 2:** Make reference to Custom Product

- **Name :** `SummitCommerceServiceImpl`
- **Location :** `/Users/l716/Desktop/AEM-Commerce-Lab-WS/aem-commerce-framework-extensions/bundle/src/main/java/com/adobe/cq/commerce/custom/impl`
- **Snippet:** Uncomment following snippet to `SummitCommerceServiceImpl`

```
/* Step 2  @Override
public Product getProduct(final String path) throws CommerceException {
    Resource resource = resolver.getResource(path);
    if (resource != null &&
        SummitProduct.isAProductOrVariant(resource)) {
        return new SummitProduct(resource);
    }
    return null;
} */
```

Exercise 4 – Task 2 – Create custom Elastic Path Product for Summit

- **Step 3:** Make reference to Custom Commerce Service

- **Name :** .content.xml

- **Location:** /Users/l716/Desktop/AEM-Commerce-Lab-WS/aem-commerce-framework-extensions/content/src/main/content/jcr_root/content/geometrixx-outdoors/en_US/.content.xml

- **Snippet :**

Replace

cq:commerceProvider="geometrixx-elasticsearch"

with

cq:commerceProvider="summit-commerce"

Exercise 4 – Task 2 – Create custom Elastic Path Product for Summit

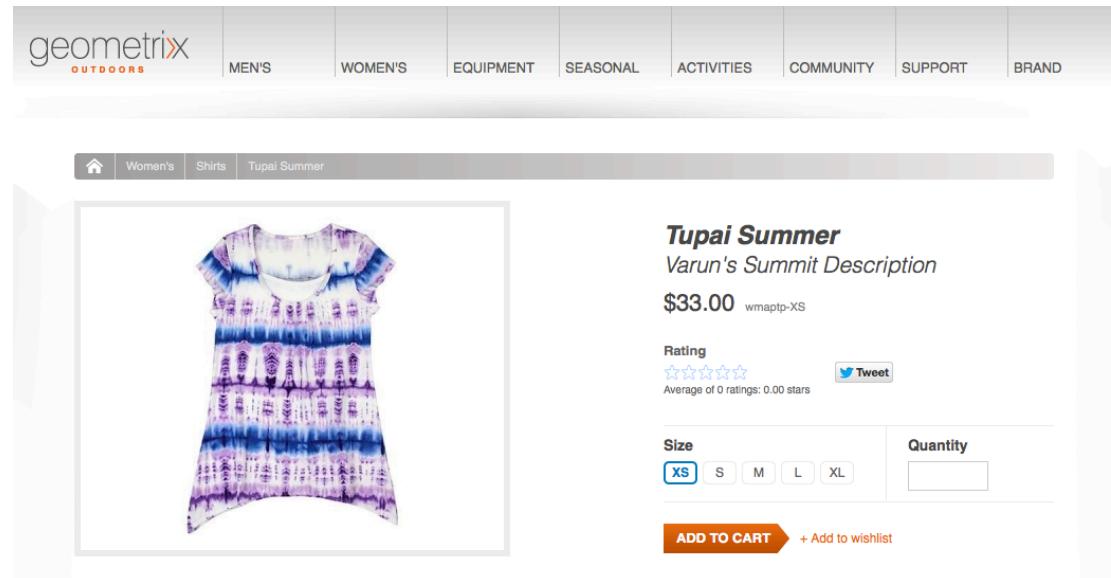
- **Step 4: Build deploy and test**

1. mvn clean install -PautoInstallPackage

Exercise 4 - Task 2 - Result - Create custom Elastic Path Product for Summit

- **Result:**

1. Go to http://localhost:4502/content/geometrixx-outdoors/en_US/women/shirts/tupai-summer.html
2. View the changes – Varun's Summit Description is actually from ElasticPath !!! 😊



Exercise 4 – Task2 – Solution

Solution (Optional – If you don't want to follow the steps in following slides or we run out of time)

1. In your terminal
2. Enter the following

```
sh install-exercise4-task2-solution.sh
```

3. Refresh workspace (Clean Project and Refresh)

Exercise 4 - Task 3 (Optional)

Dynamic Price - Extensibility of Cortex Java SDK

Exercise 4 – Task 3 – Add Sale Price for a Product in EP to reflect in AEM

- **Task :** Add Sale Price for a Product in EP to reflect in AEM
- **Time :** 5 min

Exercise 4 - Task 3 - Observation 1 - Current Price in Cortex SDK

Observation 1:

- Go to
[http://localhost:9080/studio/#/prices/items/geometrixx/m5yumvdrlbshiwkyiiyggqzrlfktmtcdoa3ugudfkjfgcobumnftmrtymuvxqv3biuyeuwq](http://localhost:9080/studio/#/prices/items/geometrixx/m5yumvdrlbshiwkyiiyggqzrlfktmtcdoa3ugudfkjfgcobumnftmcdoa3ugudfkjfgcobumnftmrtymuvxqv3biuyeuwq)
- View the pricing for the Tupai product in the Cortex API

```
{  
  self: {  
    type: "elasticpath.prices.item-price",  
    uri: "/prices/items/geometrixx/m5yumvdrlbshiwkyiiyggqzrlfktmtcdoa3ugudfkjfgcobumnftm...  
    href: "http://localhost:9080/cortex/prices/items/geometrixx/m5yumvdrlbshiwkyiiyggqzrlfktmtcdoa3ugudfkjfgcobumnftm...  
    max-age: 600  
  },  
  list-price: [  
    {  
      amount: 33,  
      currency: "USD",  
      display: "$33.00"  
    }  
  ],  
  purchase-price: [  
    {  
      amount: 33,  
      currency: "USD",  
      display: "$33.00"  
    }  
  ],  
  links: [  
    {  
      rel: "item",  
      rev: "price",  
      type: "elasticpath.items.item",  
      uri: "/items/geometrixx/m5yumvdrlbshiwkyiiyggqzrlfktmtcdoa3ugudfkjfgcobumnftmrtymuvx...  
      href: "http://localhost:9080/cortex/items/geometrixx/m5yumvdrlbshiwkyiiyggqzrlfktmtcdoa3ugudfkjfgcobumnftmrtymuvx...  
    }  
  ]  
}
```

Exercise 4 – Task 3 – Add Sale Price for a Product in EP to reflect in AEM

In Commerce Manager

1. Switch to Catalog Manager Activity
2. Right-click Geometrixx Outdoors catalog and open
3. Navigate to women -> shirts
4. Open Tupai Summer (or click the Refresh button if it's already open)
5. Click the Pricing tab
6. Set a sale price for each sku
7. Save

Exercise 4 - Task 3 - Observation 2 - Current Price in Cortex SDK

Observation 2:

- Go to
<http://localhost:9080/studio/#/prices/items/geometrixx/m5yumvdrlbshiwkyiiyggqzrlfktmtcdoa3ugudfkjfgcobumnftmrtymuvxqv3biuyeuwq>
- View the pricing for the Tupai product reflects new price.

```
{  
  self: {  
    type: "elasticpath.prices.item-price",  
    uri: "/prices/items/geometrixx/m5yumvdrlbshiwkyiiyggqzrlfktmtcdoa3ugudfkjfgcobumnftm...  
    href: "http://localhost:9080/cortex/prices/items/geometrixx/m5yumvdrlbshiwkyiiyggqzrlfktmtcdoa3ugudfkjfgcobumnftm..."  
    max-age: 600  
  },  
  list-price: [  
    {  
      amount: 33,  
      currency: "USD",  
      display: "$33.00"  
    }  
  ],  
  purchase-price: [  
    {  
      amount: 20,  
      currency: "USD",  
      display: "$20.00"  
    }  
  ],  
  links: [  
    {  
      rel: "item",  
      rev: "price",  
      type: "elasticpath.items.item",  
      uri: "/items/geometrixx/m5yumvdrlbshiwkyiiyggqzrlfktmtcdoa3ugudfkjfgcobumnftmrtymuvx...  
      href: "http://localhost:9080/cortex/items/geometrixx/m5yumvdrlbshiwkyiiyggqzrlfktmtcdoa3ugudfkjfgcobumnftmrtymuvx..."  
    }  
  ]  
}
```

Exercise 4 – Task 3 – Observation 3 – Current Price in Website

Observation 3:

- Go to
http://localhost:4502/content/geometrixx-outdoors/en_US/women/shirts/tupai-summer.html
- View the pricing for the Tupai product reflects new price.

Tupai Summer

Women's Frock

\$20.00 wmaptp-XS

Rating



[Tweet](#)

Average of 0 ratings: 0.00 stars

Size

XS S M L XL

Quantity

ADD TO CART

+ Add to wishlist

Exercise 4 – Summary and Key Takeaway

Abstract implementations for Elastic Path and Hybris in AEM Commerce Framework is a scalable framework for extension based on client use cases.

Use Cases : As a user, I would like to extend my commerce framework for my commerce servers (such as Hybris or Elastic Path) to create customizations, so I can address my customer requirements.

Example :

1. Create additional attributes in my commerce service for products and make sure this appears in AEM.
2. Customize cart data received from my commerce service to appear in AEM.

Q&A

Varun Venkataraman

vvenkata@adobe.com

Twitter: @varunstech

LinkedIn: <http://au.linkedin.com/in/varunv1>