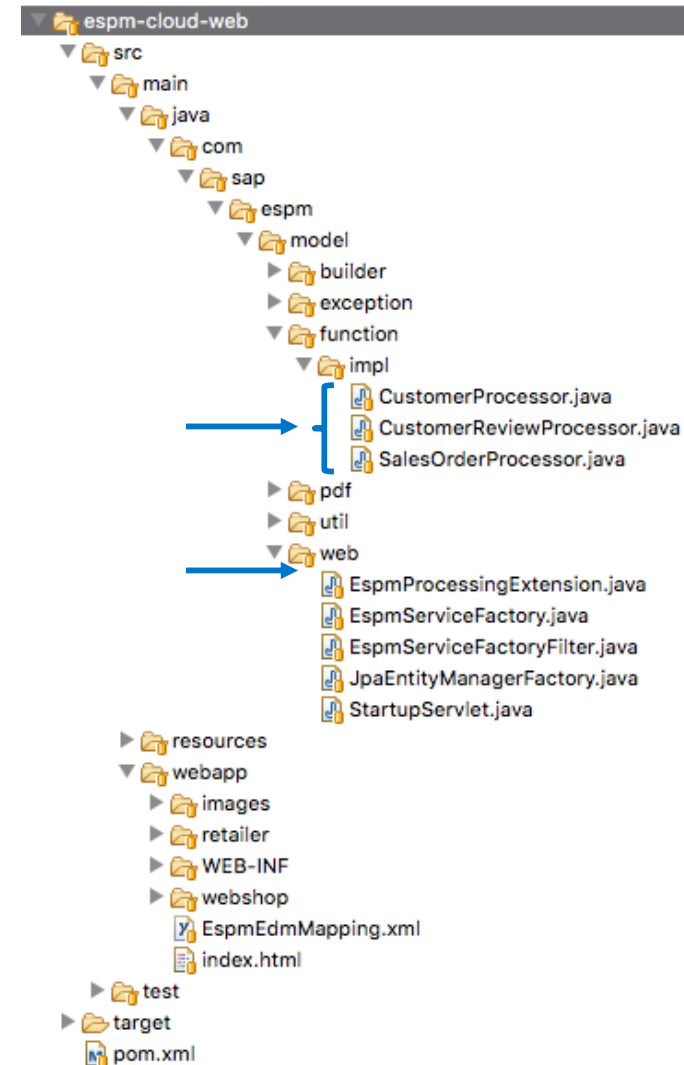
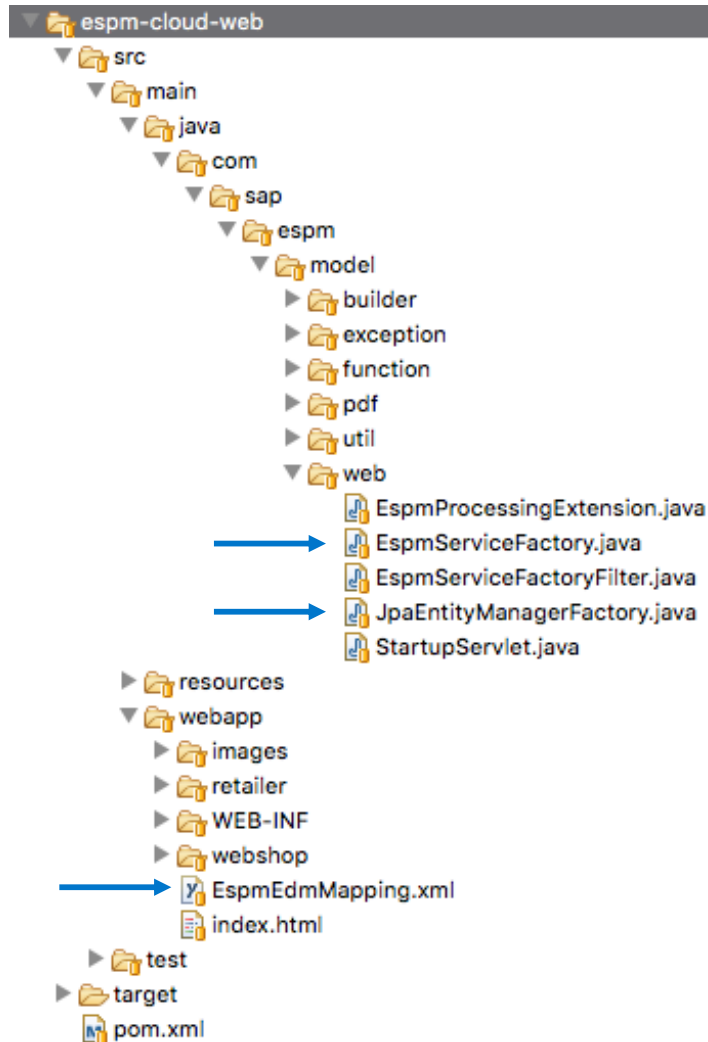


# Week 2 Unit 4: Generating the OData Services – Practice

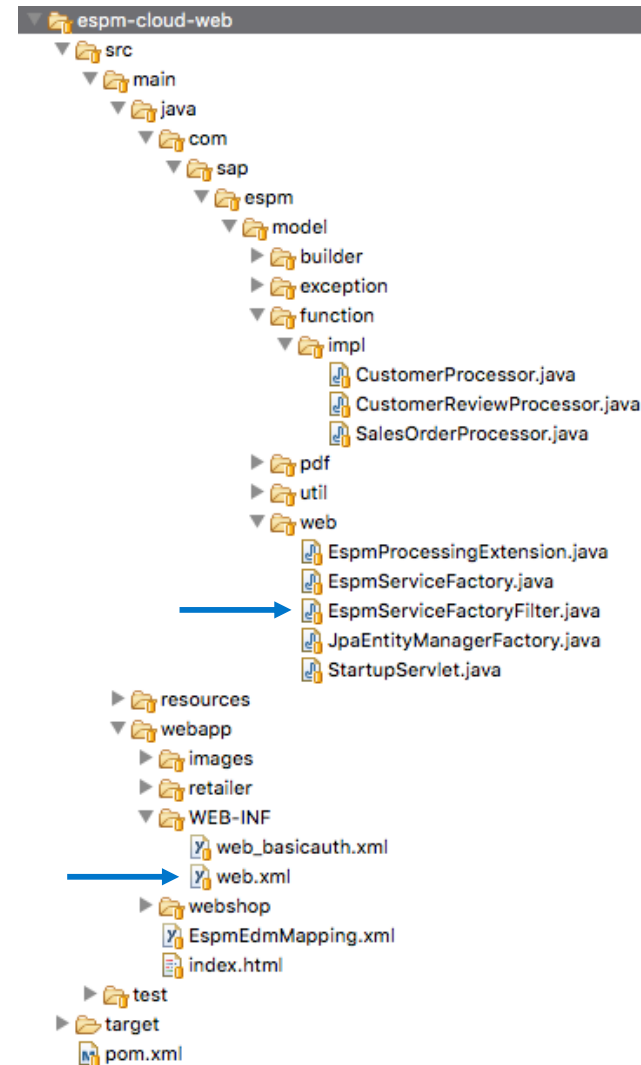
# Generating the OData Services – Practice

## Recap: The OData services of the ESPM application



## Recap: The OData services of the ESPM application

## Recap: The OData services of the ESPM application

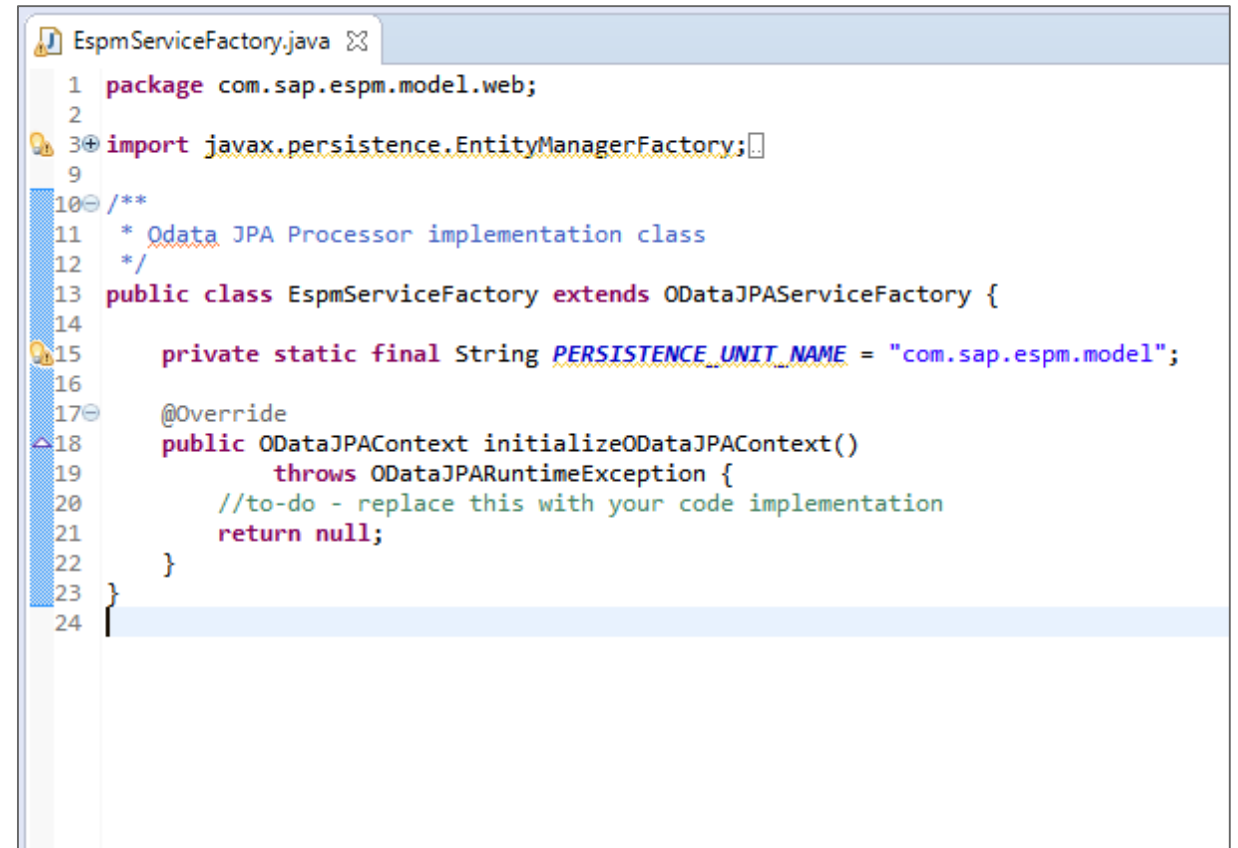


# Generating the OData Services – Practice

## Hands-on (1)

### Initialize the OData JPA Context

- In Eclipse, navigate to the project espm-cloud-web
- Navigate to src/main/java → com.sap.espm.model.web
- Open the EspmServiceFactory.java file
- Replace the to-do part as shown in the screenshot with code snippet 1 from the file openSAP\_hcp2\_Week\_2\_Unit\_4\_Exercise
- Save your changes



```
1 package com.sap.espm.model.web;
2
3 import javax.persistence.EntityManagerFactory;
4
5
6
7
8
9
10 /**
11  * Odata JPA Processor implementation class
12  */
13 public class EspmServiceFactory extends ODataJPAServiceFactory {
14
15     private static final String PERSISTENCE_UNIT_NAME = "com.sap.espm.model";
16
17     @Override
18     public ODataJPAContext initializeODataJPAContext()
19         throws ODataJPARuntimeException {
20         //to-do - replace this with your code implementation
21         return null;
22     }
23 }
24
```

# Generating the OData Services – Practice

## Hands-on (2)

### Define Function Imports

- Navigate to src/main/java → com.sap.espm.model.function.impl
- Open the CustomerProcessor.java file
- Replace the to-do part as shown in the screenshot with code snippet 2 from the file openSAP\_hcp2\_Week\_2\_Unit\_4\_Exercise
- Save your changes

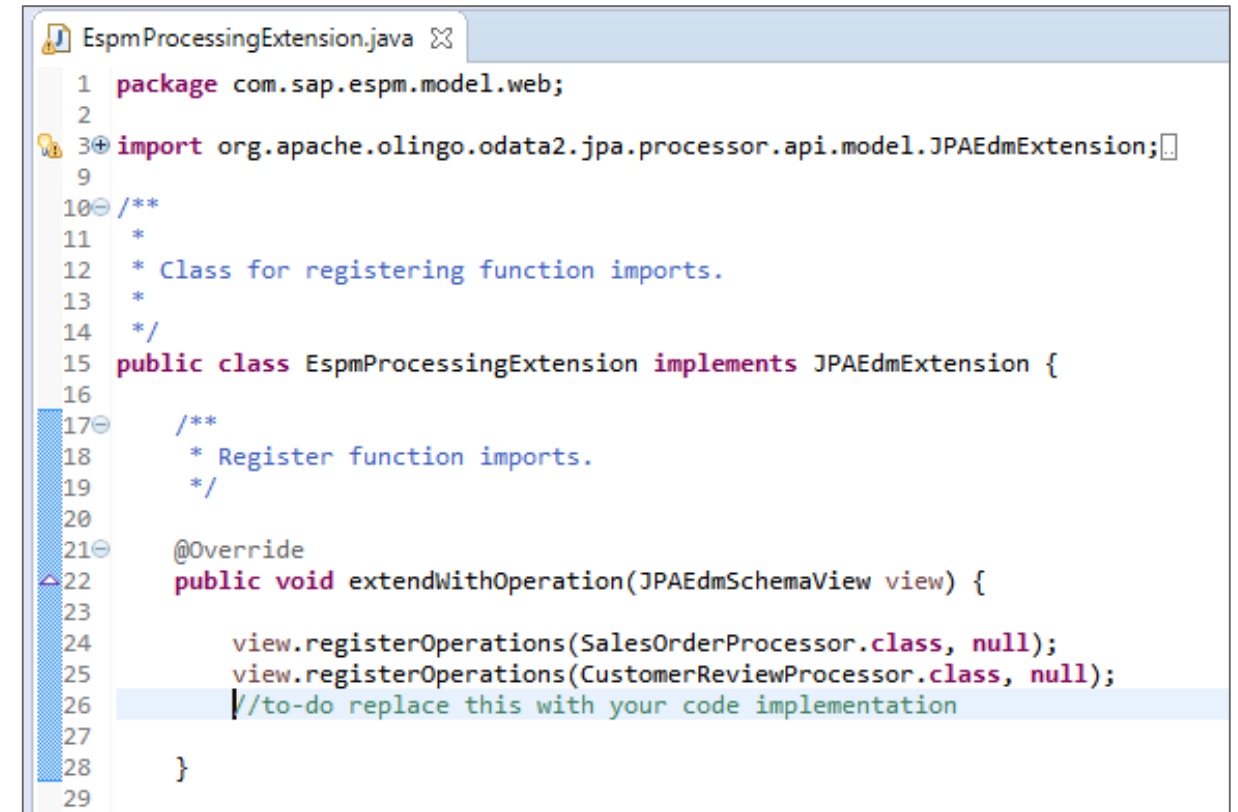


# Generating the OData Services – Practice

## Hands-on (3)

### Register Function Imports

- Navigate to src/main/java → com.sap.espm.model.web
- Open the EspmProcessingExtension.java file
- Replace the to-do part as shown in the screenshot with code snippet 3 from the file openSAP\_hcp2\_Week\_2\_Unit\_4\_Exercise
- Save your changes
- Build your application and deploy it
- You should be able to access the OData service in ESPM and view the metadata to see the getCustomerByEmailAddress function import registered



```
1 package com.sap.espm.model.web;
2
3 import org.apache.olingo.odata2.jpa.processor.api.model.JPAEdmExtension;
4
5
6
7
8
9
10 /**
11  *
12  * Class for registering function imports.
13  *
14  */
15 public class EspmProcessingExtension implements JPAEdmExtension {
16
17     /**
18      * Register function imports.
19      */
20
21     @Override
22     public void extendWithOperation(JPAEdmSchemaView view) {
23
24         view.registerOperations(SalesOrderProcessor.class, null);
25         view.registerOperations(CustomerReviewProcessor.class, null);
26         //to-do replace this with your code implementation
27
28     }
29 }
```

# Generating the OData Services – Practice

What you've learned in this unit

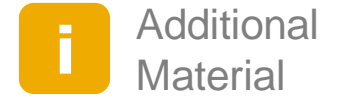
---

- Recap: The OData services of the ESPM application
- How to initialize the ODataJPAContext
- How to define function imports
- How to register function import

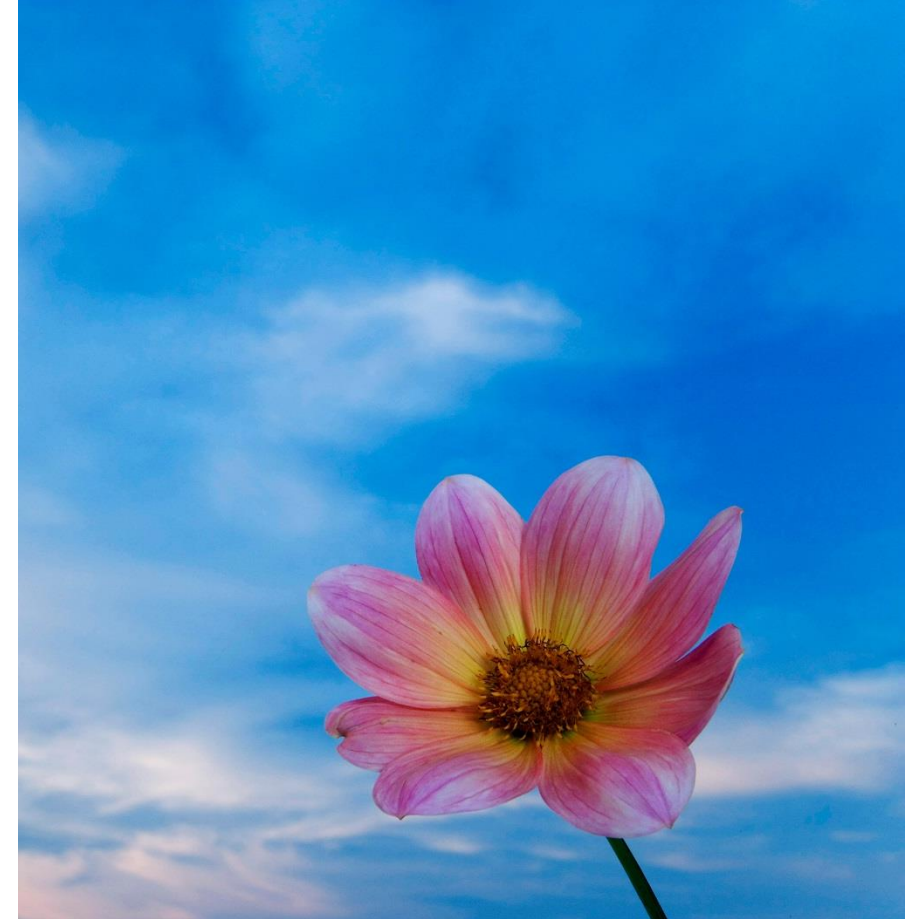


# Generating the OData Services – Practice

## Further reading



- Apache Olingo Project <https://olingo.apache.org/>







# Thank you

Contact information:

[open@sap.com](mailto:open@sap.com)

open**SAP**

# © 2016 SAP SE or an SAP affiliate company. All rights reserved.

---

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. Please see <http://global12.sap.com/corporate-en/legal/copyright/index.epx> for additional trademark information and notices.

Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors.

National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP SE or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP SE or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

In particular, SAP SE or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP SE's or its affiliated companies' strategy and possible future developments, products, and/or platform directions and functionality are all subject to change and may be changed by SAP SE or its affiliated companies at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, which speak only as of their dates, and they should not be relied upon in making purchasing decisions.