

Week 4 Unit 4:

Document Service – Practice

Document Service – Practice

Recap: How to connect

- Document service implements CMIS (JSON & AtomPub bindings)
- Apache OpenCMIS client libraries automatically provisioned on all HCP Java Runtimes
- Libraries automatically configured with document service endpoint and your account name as authentication means
- Initial service lookup either by JNDI (EcmService) or static class (EcmFactory):
 - JNDI: `(EcmService) new InitialContext().lookup("java:comp/env/EcmService");`
 - Static: e.g. `EcmFactory.connect(...)`
- Connection always from HCP VM. Connection from external requires setup of the ProxyBridge

Document Service – Practice

Recap: Creating & connecting

Create a new repository

```
RepositoryOptions options = new RepositoryOptions();
options.setUniqueName("com.foo.MyRepository");
options.setRepositoryKey("my_super_secret_key_123"); // (min. 10 chars)
options.setVisibility(Visibility.PROTECTED);
//optionally enable virus scanning on upload: options.setVirusScannerEnabled(true);
InitialContext ctx = new InitialContext();
EcmService ecmService = (EcmService) ctx.lookup("java:comp/env/EcmService");
ecmService.createRepository(options);
```

Connect & fetch the root folder

```
openCmisSession = ecmService.connect("com.foo.MyRepository", "my_super_secret_key_123");
Folder root = openCmisSession.getRootFolder();
```

Document Service – Practice

Recap: Listing & uploading

Get all children of root folder

```
ItemIterable<CmisObject> children = root.getChildren();  
for (CmisObject o : children) { ... }
```

Create a new document

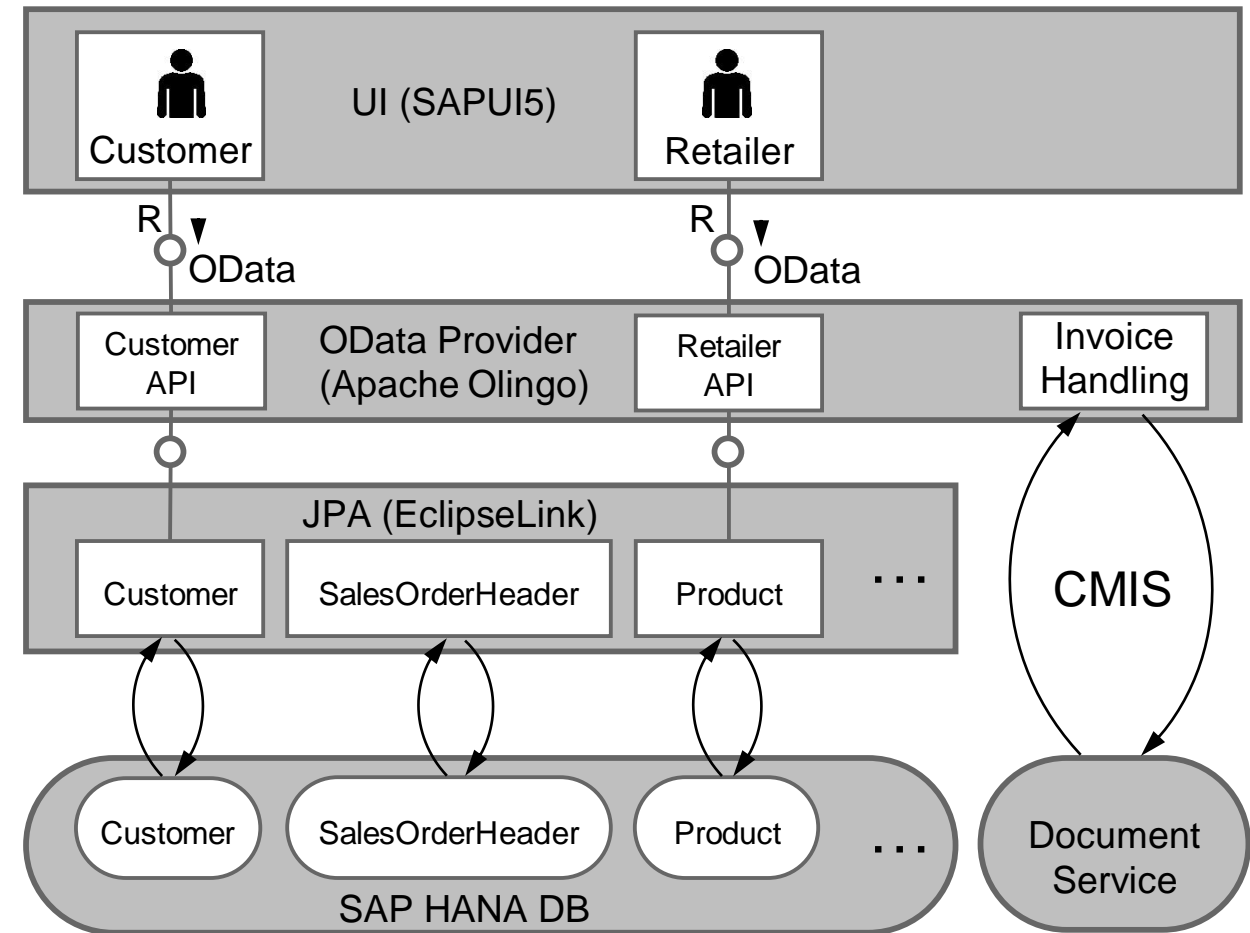
```
// define mandatory properties  
Map<String, Object> properties = new HashMap<String, Object>();  
properties.put(PropertyIds.OBJECT_TYPE_ID, "cmis:document");  
properties.put(PropertyIds.NAME, "HelloWorld.txt");  
// define the content  
byte[] helloContent = "Hello World!".getBytes("UTF-8");  
InputStream stream = new ByteArrayInputStream(helloContent);  
ContentStream contentStream = openCmisSession.getObjectFactory().createContentStream("HelloWorld.txt",  
helloContent.length, "text/plain; charset=UTF-8", stream);  
// create the document below the root node  
root.createDocument(properties, contentStream, VersioningState.NONE);
```

Document Service – Practice

Hands-on (1)

Reading and writing the code to handle the interaction of the ESPM application with the document service

- Goal is to store the created invoices in the document repository and retrieve them from there
- What are the necessary steps?
 - Connect to the document service
 - Create a repository
 - Store documents
 - Retrieve documents

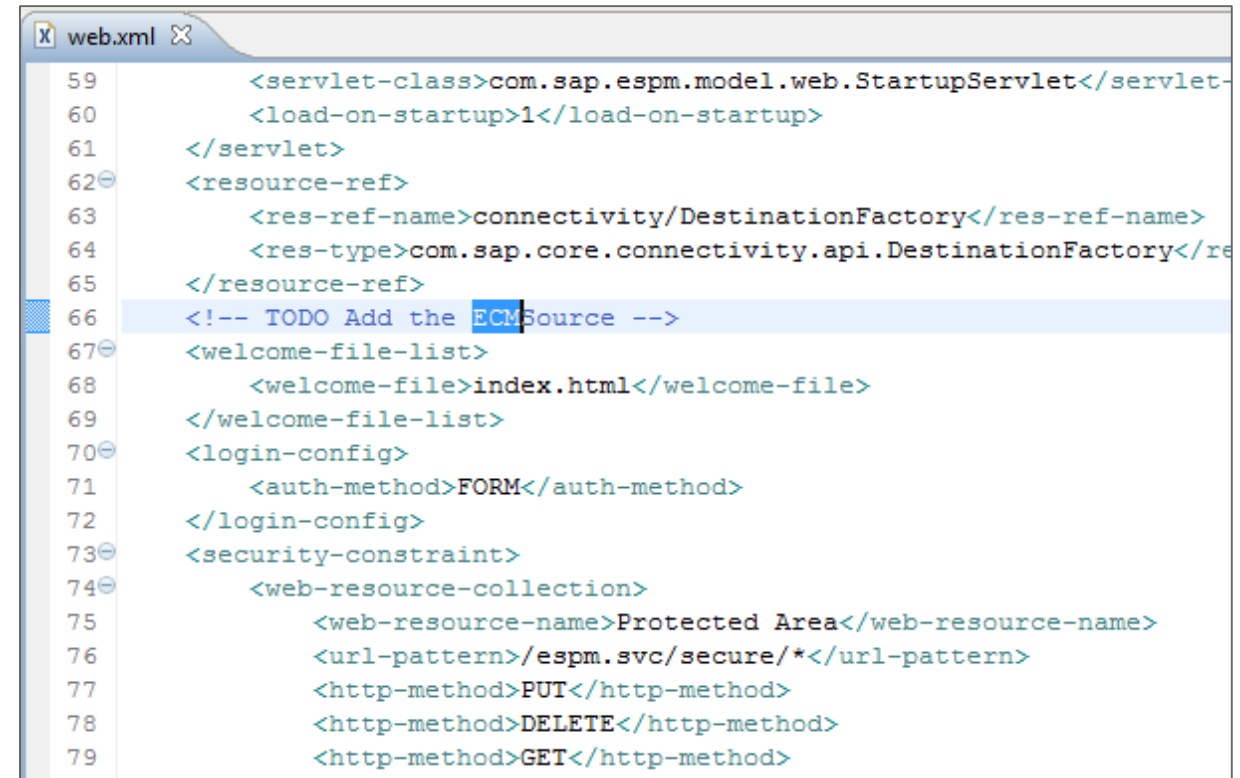


Document Service – Practice

Hands-on (2)

Prepare the Web application project for DocumentService

- In Eclipse, navigate to the project espm-cloud-web
- Navigate to src/main/webapp/WEB-INF
- Open the web.xml file
- Replace the to-do part as shown in the screenshot with code snippet 1 from the file openSAP_hcp2_Week_4_Unit_4_Exercise
- Save your changes



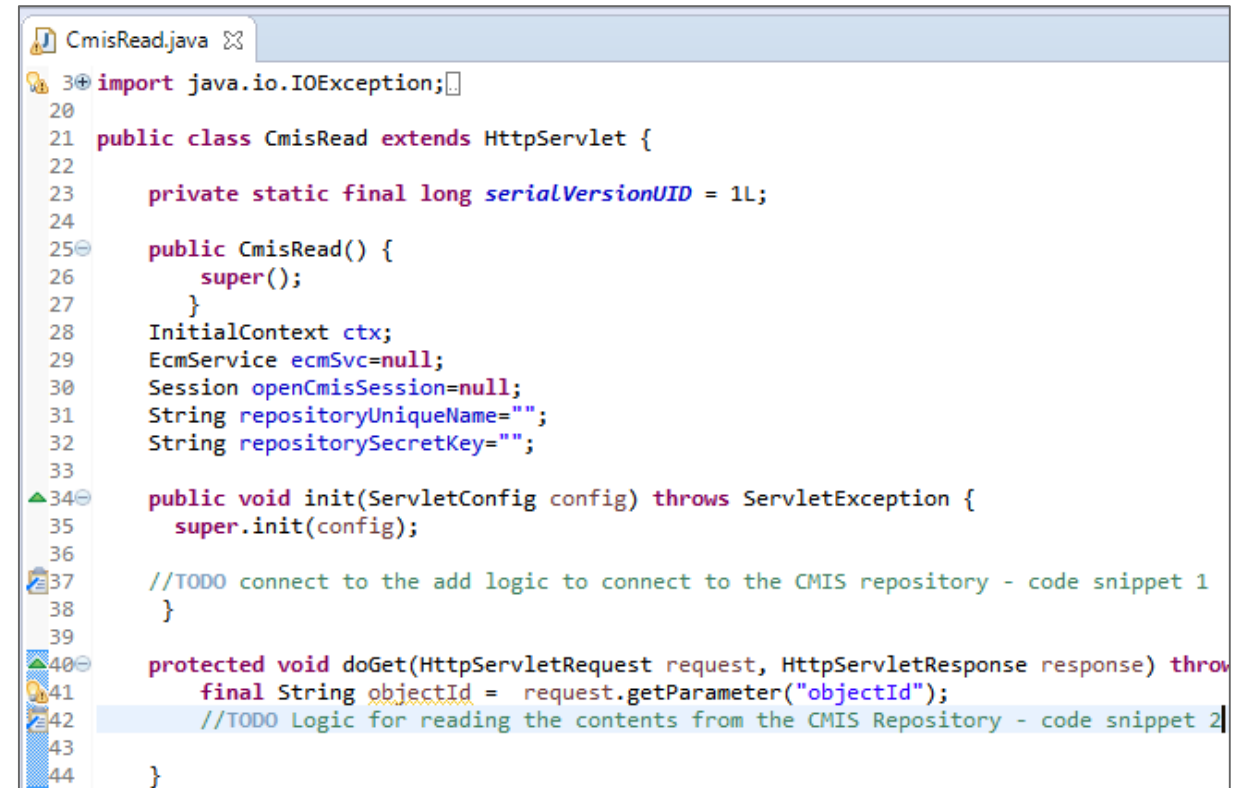
```
59     <servlet-class>com.sap.espm.model.web.StartupServlet</servlet-  
60     <load-on-startup>1</load-on-startup>  
61 </servlet>  
62 <resource-ref>  
63     <res-ref-name>connectivity/DestinationFactory</res-ref-name>  
64     <res-type>com.sap.core.connectivity.api.DestinationFactory</re  
65 </resource-ref>  
66 <!-- TODO Add the ECMSource -->  
67 <welcome-file-list>  
68     <welcome-file>index.html</welcome-file>  
69 </welcome-file-list>  
70 <login-config>  
71     <auth-method>FORM</auth-method>  
72 </login-config>  
73 <security-constraint>  
74     <web-resource-collection>  
75         <web-resource-name>Protected Area</web-resource-name>  
76         <url-pattern>/espm.svc/secure/*</url-pattern>  
77         <http-method>PUT</http-method>  
78         <http-method>DELETE</http-method>  
79         <http-method>GET</http-method>
```


Document Service – Practice

Hands-on (3)

Modify the CmisRepository initialization

- In Eclipse, navigate to the project espm-cloud-web
- Navigate to src/main/java → com.sap.espm.model.pdf.generator
- Open the CmisRead.java file
- Replace the to-do part – code snippet 1 as shown in the screenshot – with code snippet 2 from the file openSAP_hcp2_Week_4_Unit_4_Exercise
- Save your changes



```
CmisRead.java
import java.io.IOException;

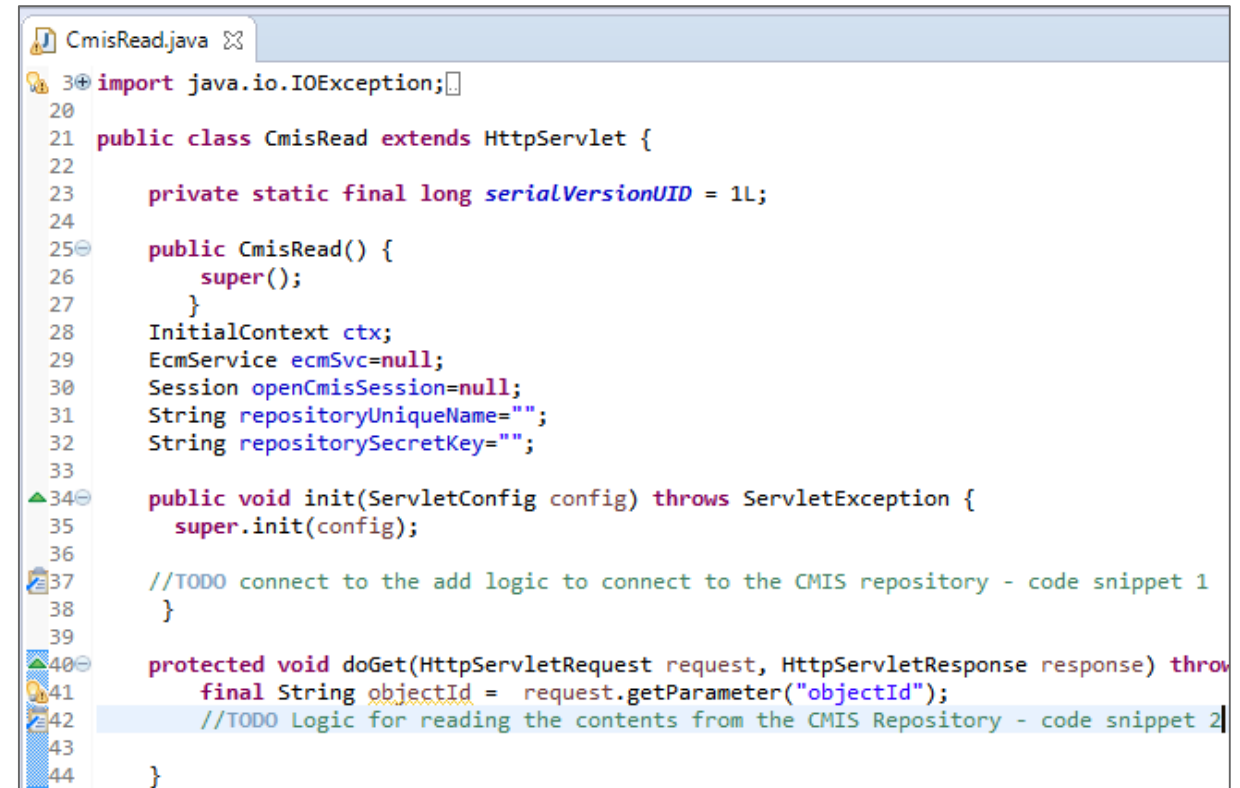
20
21 public class CmisRead extends HttpServlet {
22
23     private static final long serialVersionUID = 1L;
24
25     public CmisRead() {
26         super();
27     }
28     InitialContext ctx;
29     EcmService ecmSvc=null;
30     Session openCmisSession=null;
31     String repositoryUniqueName="";
32     String repositorySecretKey="";
33
34     public void init(ServletConfig config) throws ServletException {
35         super.init(config);
36
37         //TODO connect to the add logic to connect to the CMIS repository - code snippet 1
38     }
39
40     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
41         ServletException, IOException {
42         final String objectId = request.getParameter("objectId");
43         //TODO Logic for reading the contents from the CMIS Repository - code snippet 2
44     }
45 }
```

Document Service – Practice

Hands-on (4)

Modify the CmisRepository initialization

- In Eclipse, navigate to the project espm-cloud-web
- Navigate to src/main/java → com.sap.espm.model.pdf.generator
- Open the CmisRead.java file
- Replace the to-do part – code snippet 2 as shown in the screenshot – with code snippet 3 from the file openSAP_hcp2_Week_4_Unit_4_Exercise
- Save your changes
- Build your project and deploy the same



```
30 import java.io.IOException;
20
21 public class CmisRead extends HttpServlet {
22
23     private static final long serialVersionUID = 1L;
24
25     public CmisRead() {
26         super();
27     }
28     InitialContext ctx;
29     EcmService ecmSvc=null;
30     Session openCmisSession=null;
31     String repositoryUniqueName="";
32     String repositorySecretKey="";
33
34     public void init(ServletConfig config) throws ServletException {
35         super.init(config);
36
37         //TODO connect to the add logic to connect to the CMIS repository - code snippet 1
38     }
39
40     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
41         ServletException, IOException {
42         final String objectId = request.getParameter("objectId");
43         //TODO Logic for reading the contents from the CMIS Repository - code snippet 2
44     }
```


Document Service – Practice

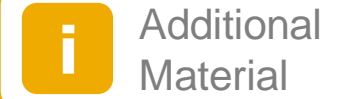
What you've learned in this unit

- Recap on how to interact with the document service via OpenCMIS
- Reading and writing the code to handle the interaction of the ESPM application with the document service
 - Connect to the document service
 - Create a repository
 - Store documents
 - Retrieve documents



Document Service – Practice

Further reading



- **SCN blog post:**

<http://scn.sap.com/community/developer-center/cloud-platform/blog/2013/02/28/coming-next-on-devtalks-document-service-on-sap-netweaver-cloud-and-the-borg>

- **Performance tips:**

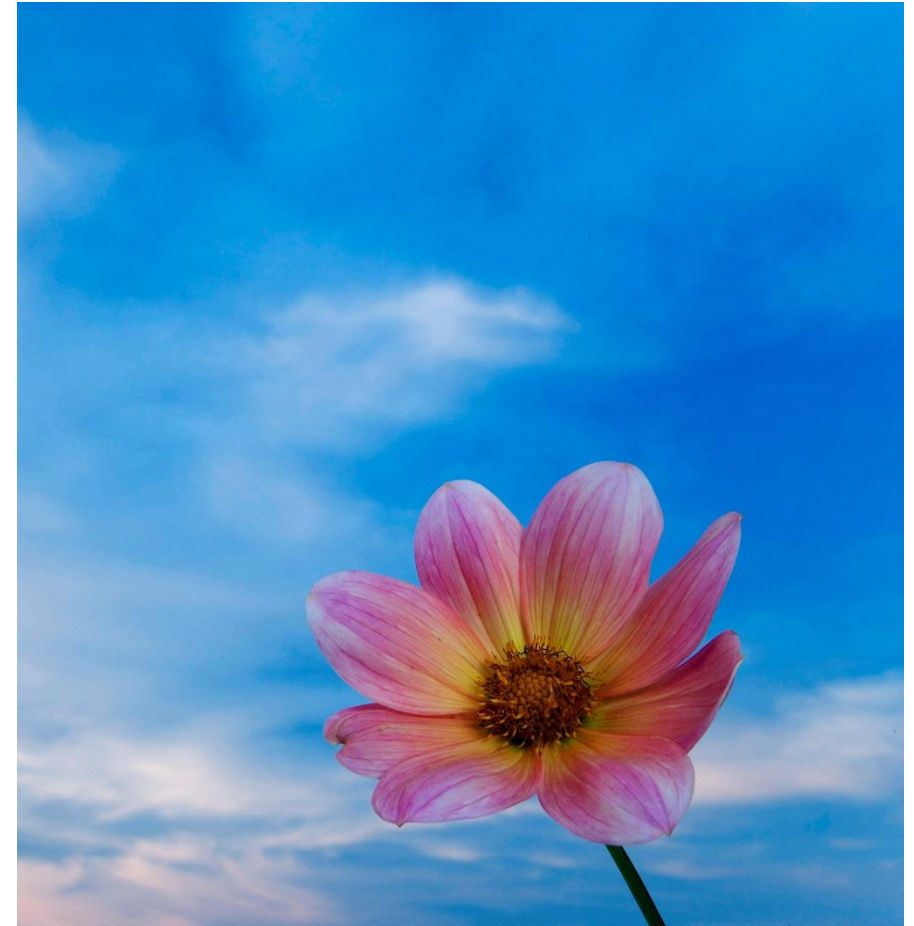
<https://help.hana.ondemand.com/help/frameset.htm?87b51b9fa8684995aa050277b26b7dcd.html>

- **Official documentation:**

<https://help.hana.ondemand.com/help/frameset.htm?e60b7e45bb57101487a881c7c5487778.html>

- **HelloWorld app:**

<https://help.hana.ondemand.com/help/frameset.htm?e62cd24abb571014b73792d85402f104.html>





Thank you

Contact information:

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.