# Chapter 2: Developing the Persistency with JPA and exposing it as OData Service

In this chapter, we shall develop the persistency and expose it as OData service, for the “personslist” UI which we developed earlier.

There are 3 steps for the above.

1. **Create the JPA Entity for the Person table.**
2. **Expose the JPA as OData service.**
3. **Deploy the application on to SAP Cloud Platform trial account.**

**1. Create the JPA Entity for the Person table**

Creating the JPA Entity consists of the following sub steps

a. Creating the Dynamic Web project in Eclipse.

b. set the Java package for the JPA Entity.

c. Creating the Person JPA Entity.

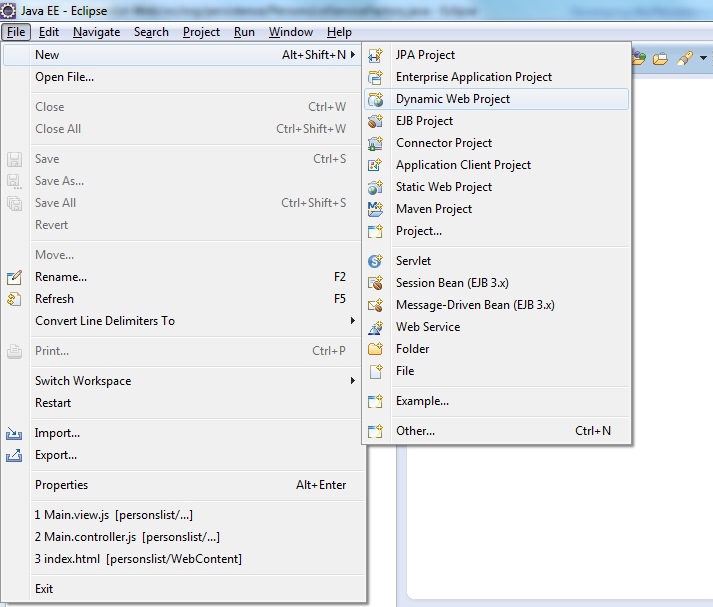
d. Set the details for Person Entity.

**a. Creating the Dynamic Web project in Eclipse**

1. Launch Eclipse by running the “eclipse.exe” file located in the *<eclipse-unzipped-directory>\eclipse\eclipse.exe.* This opens the eclipse workspace dialog

2. Enter *C:\Users\<username>\workspace* as the workspace for the application and click on the “Ok” button.

3. From the Eclipse menu bar, select File -> New -> Dynamic Web Project as shown below

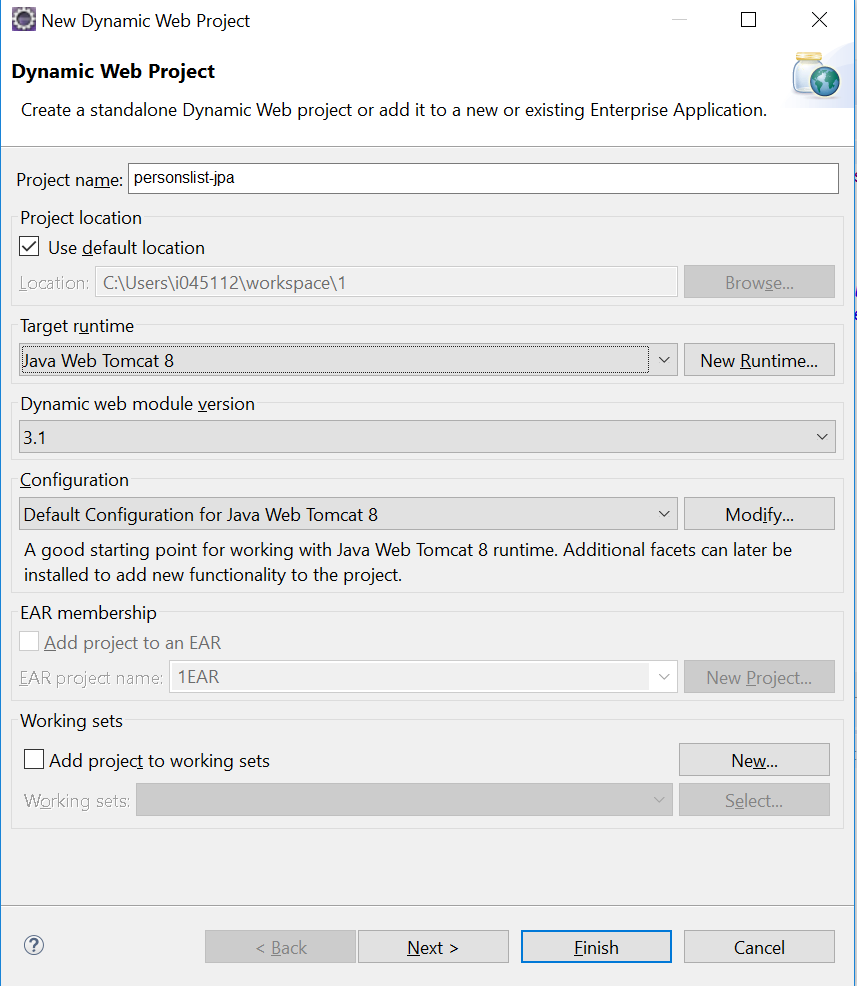


3. In the dialog opened, enter the project name as “personslist-jpa”. Make sure that the

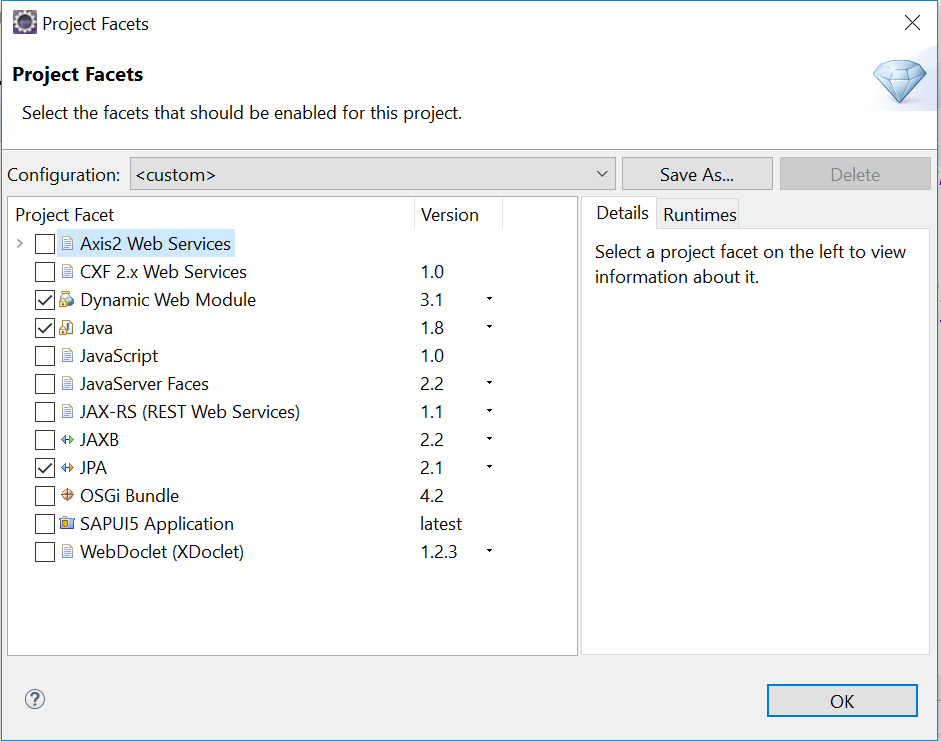
Target runtime = Java Web Tomcat 8 , The pre-requisite of this is you have already [set up the run time environment](https://help.sap.com/viewer/65de2977205c403bbc107264b8eccf4b/Cloud/en-US/7613f000711e1014839a8273b0e91070.html) inside the eclipse.

Dynamic web module version = 3.1

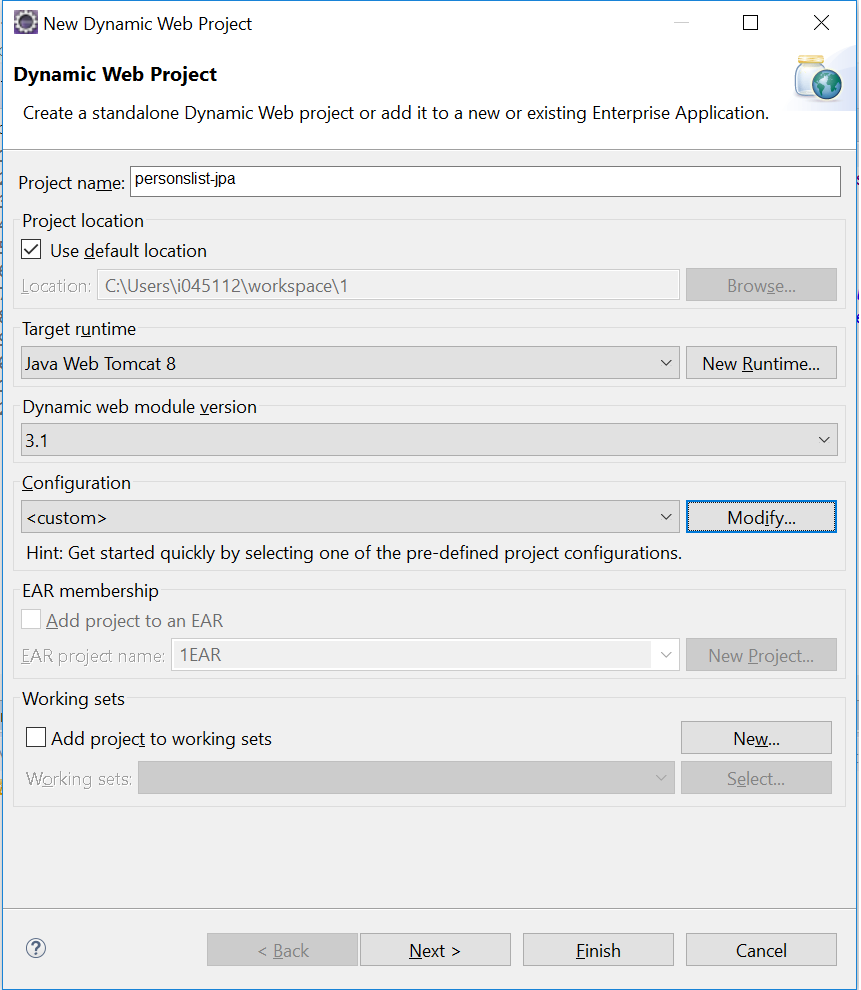
Click on the Modify button



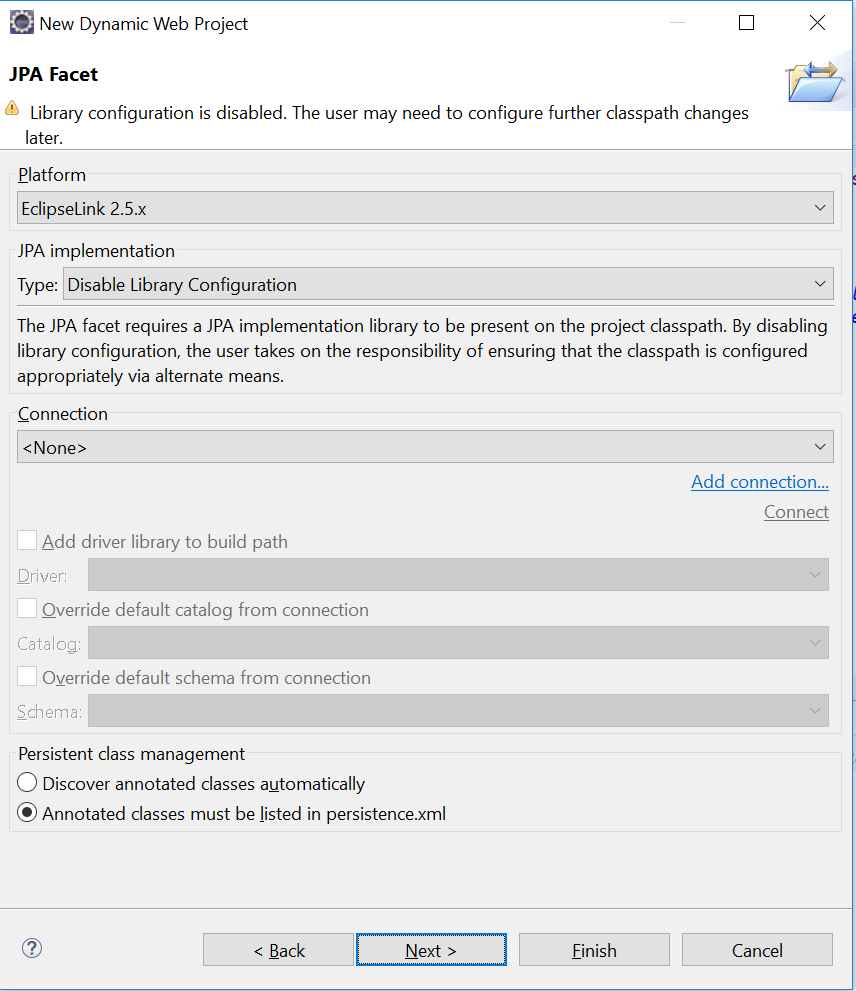
4. This opens the window “Project Facets”. Select the checkbox against the “JPA”.



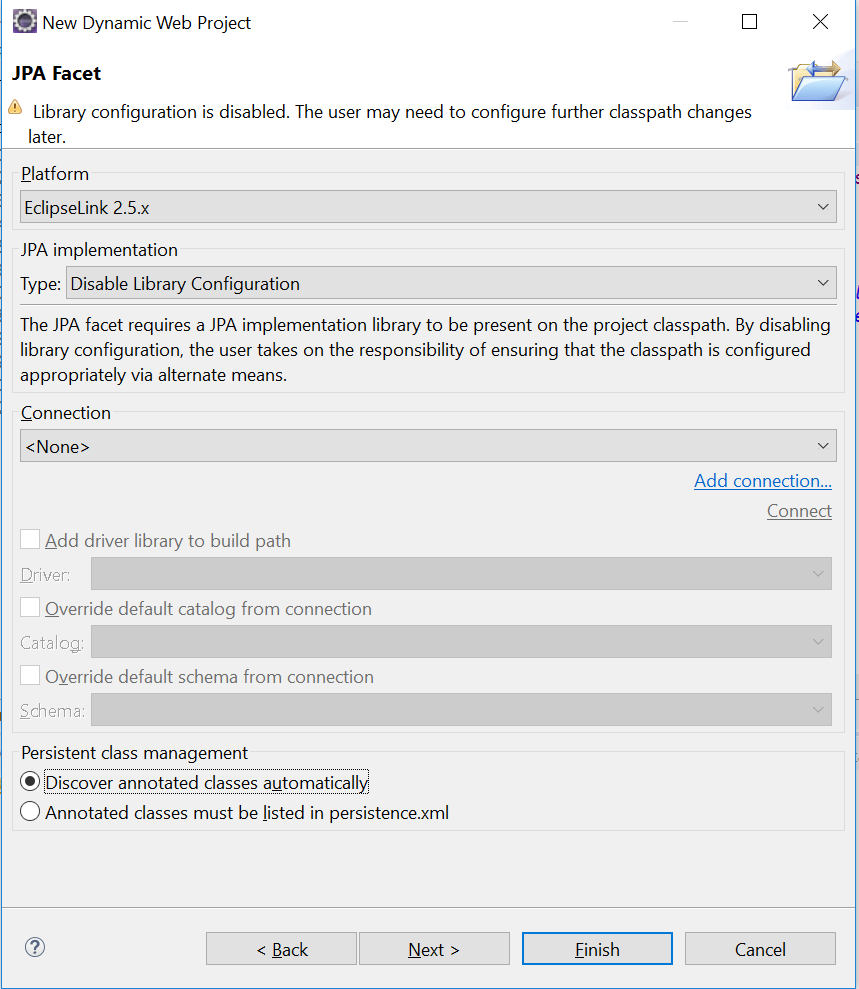
5. Click on the “OK” button to close the “Projects Facets” Window. The result is as shown in the following snapshot.



6. Click on the “Next >” button. Again, click on the “Next >” button. In the resulting screen, for the “Platform” content, you will select “EclipseLink 2.5.x”. For the JPA Implementation type, you will choose “Disable Library Configuration”.

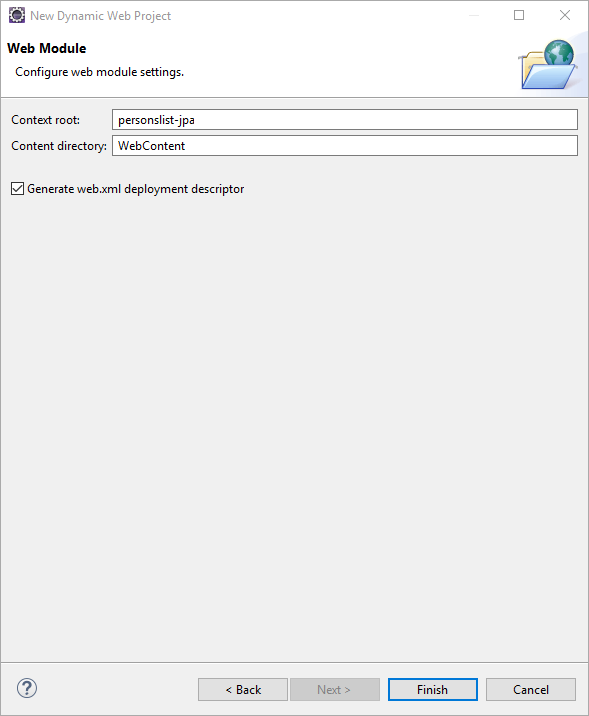


Also, select the Persistent class management as “Discover annotated classes automatically”

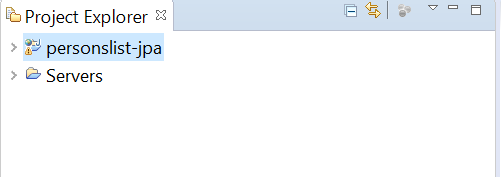


7. Choose Next>.

8. In the Web Module screen, select “Generate web.xml deployment descriptor”.



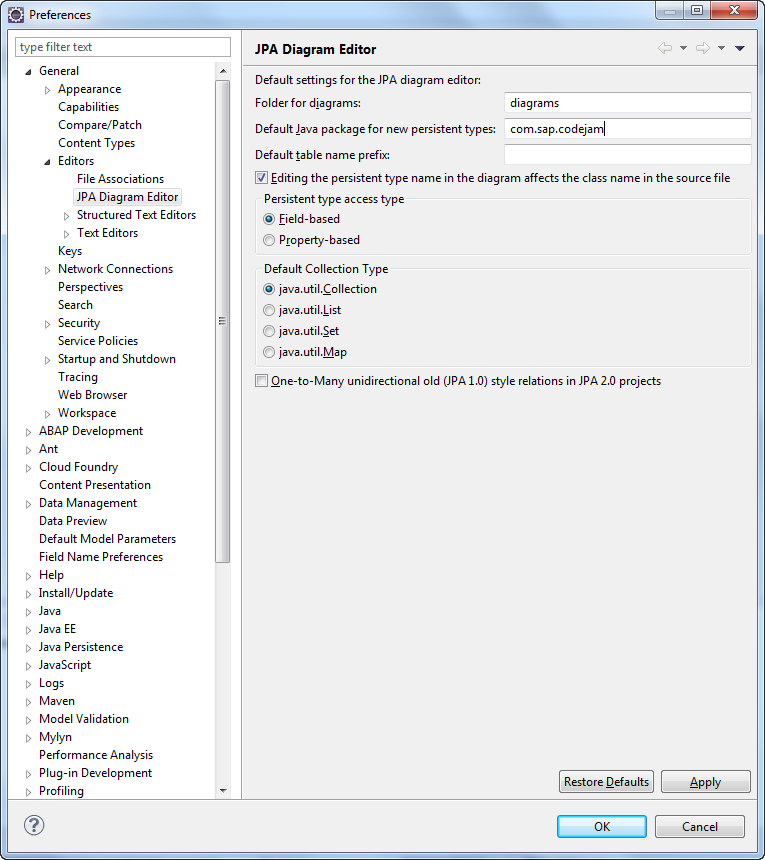
9. Click on “Finish” button to close the wizard. This creates the project “personslist-jpa” as seen in the project explorer.



**b. Set the Java package for the JPA Entity**

1. Select Window -> Preferences from the Eclipse Menu

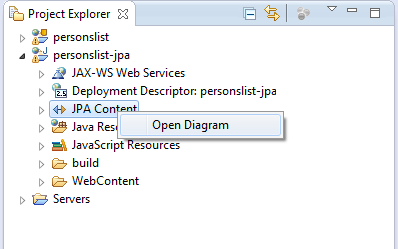
2. Select General -> Editors -> JPA Diagram Editor in the left pane. In the right pane, set the value for “Default Java package for new persistent types” = com.sap.codejam



3. Click on the “OK” button.

**c. Creating the Person JPA Entity**

1. In the Project Explorer, expand the “personslist-jpa” project. Right Click “JPA Content” and Select “Open Diagram”

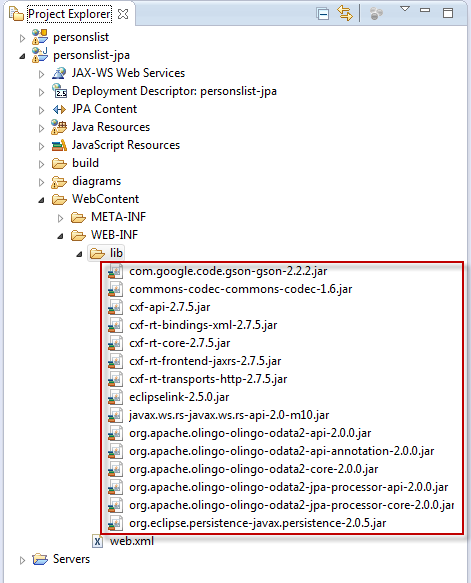


**Note: if you get an error when opening the JPA Diagram editor, please update your Eclipse version from the URL** [**http://archive.eclipse.org/graphiti/updates/0.11.4**](http://archive.eclipse.org/graphiti/updates/0.11.4)

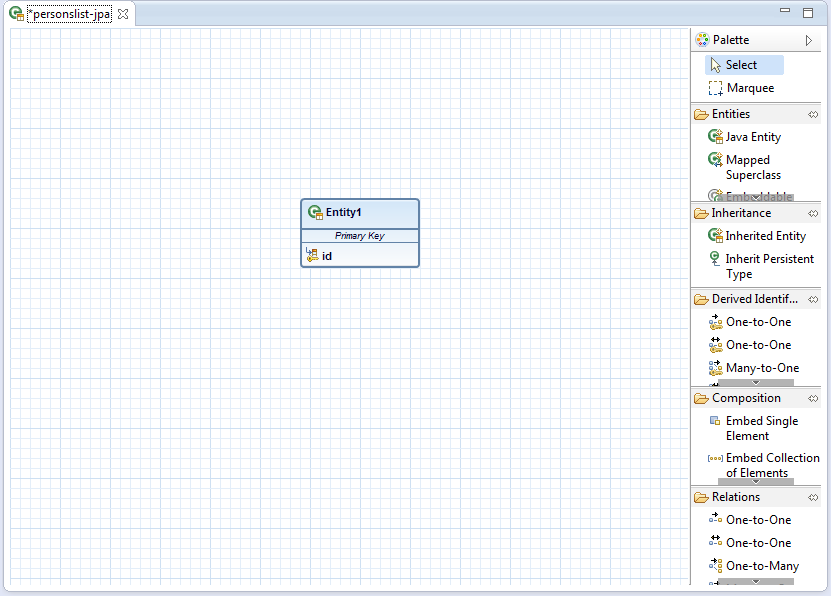
2. Extract the libraries from the “jar.zip” file attached here



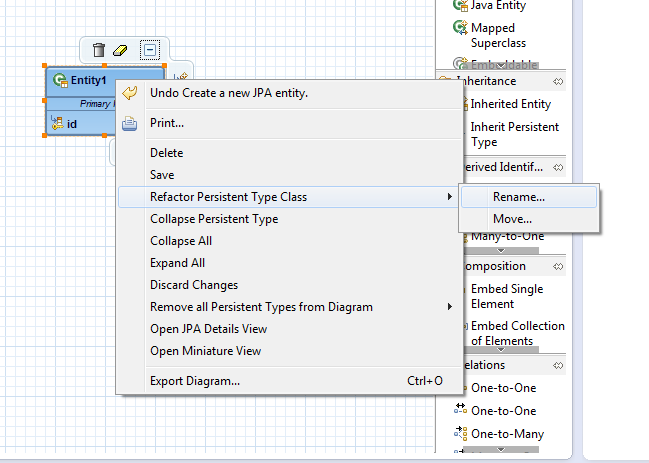
Copy and paste into the libs folder as shown in the below snapshot.



3. From the opened window, Select “Java Entity” from the Pallete, and drag and drop the same on to the designer window. This creates the Entity “Entity1” on the designer.



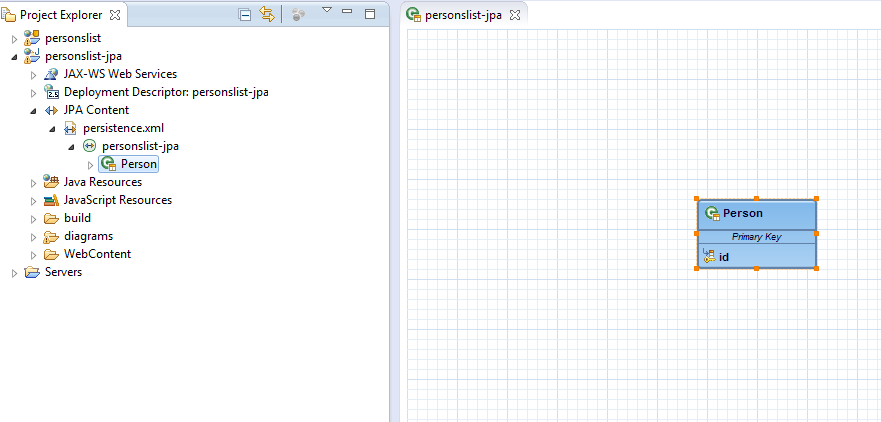
3. Right Click on the Entity1 and select “Refractor Persistent Type Class” -> “Rename…”



4. In the Opened Window, Enter the New name as “Person” and Click on Finish.

Press “Ctrl + SHIFT + S” to save the file.

This creates the Class “Person” as shown in the snapshot.

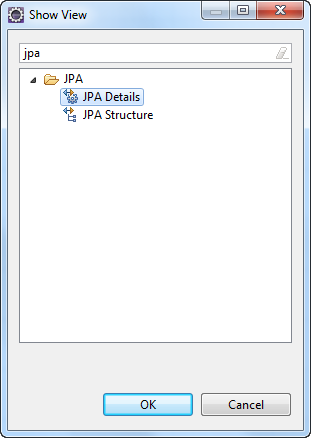


**d. Set the details for Person Entity**

1. In the Eclipse Menu, select Window -> Show View -> Other…

In the Opened window, search for “jpa”

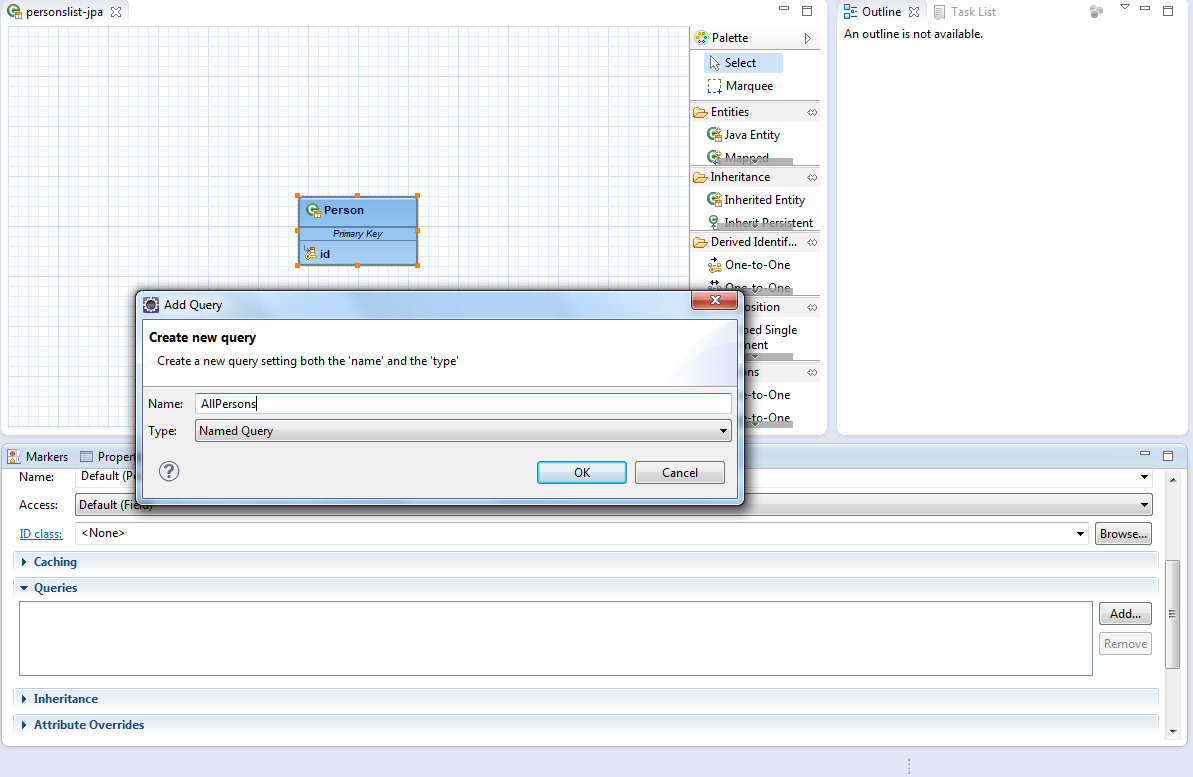
Select “JPA Details” from the Opened Window and Click on “OK” button



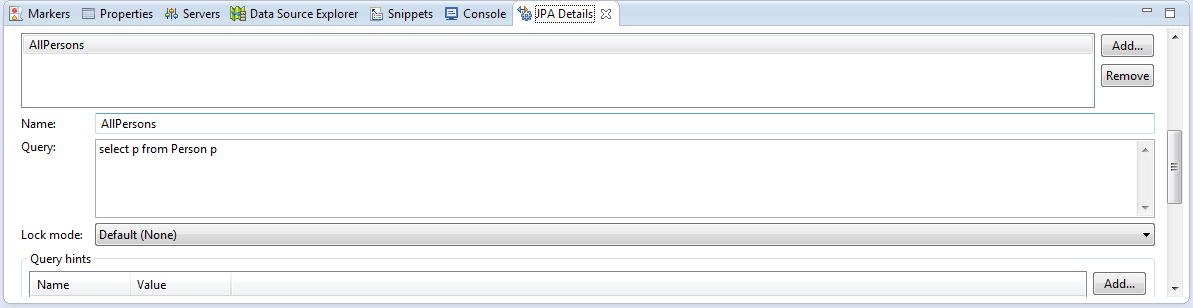
2. Select the “Person” Entity in the designer window.

In the JPA Details view navigate to Queries section, choose “Add”. The Add Query dialog box opens.

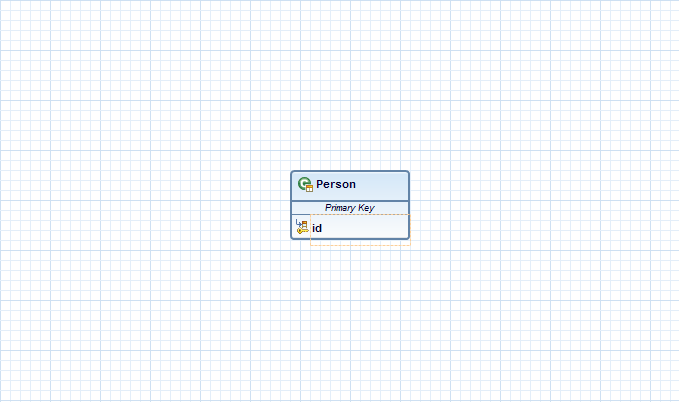
Enter the Name as “AllPersons”. Choose type as “Named Query”.



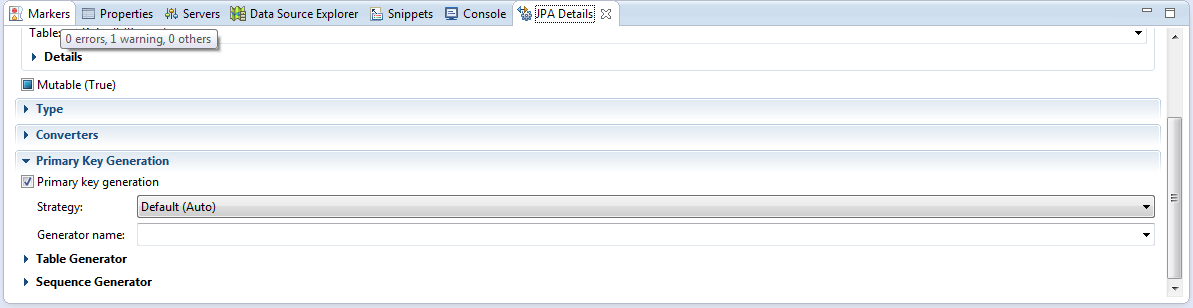
3. Click on the “OK” button. In the Query section, enter “select p from Person p”



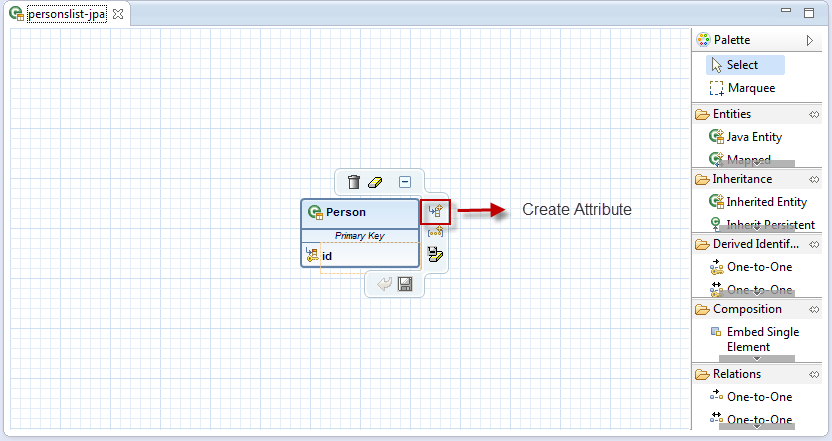
4. Select the “id” attribute for the “Person” Entity in the designer window



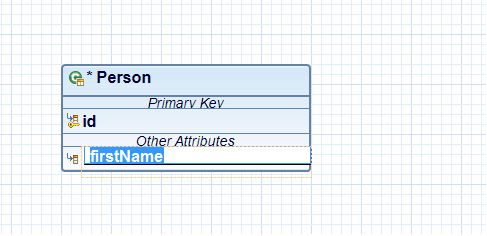
5. In the “Primary Key Generation” section in the JPA Details view, select the “Primary key generation” checkbox and retain “Default (Auto)” as the strategy.



6. Hover over the “Person” entity graphical box to display the hover buttons. Select the “Create Attribute” button. The “attribute1” element is highlighted.

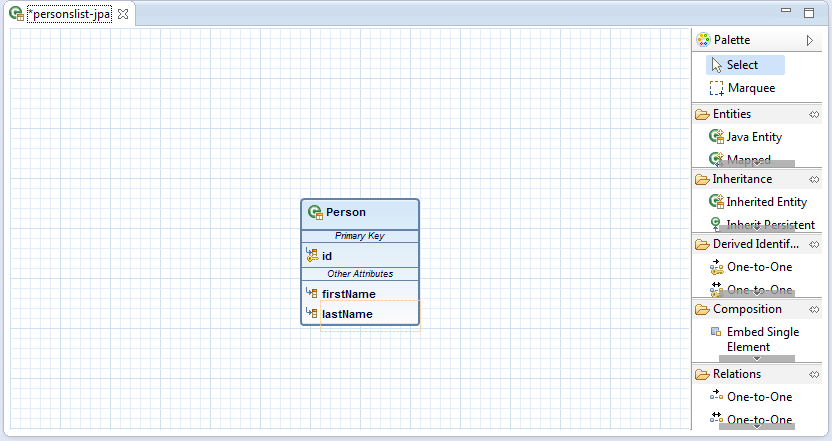


7. Enter the highlighted name attribute as “firstName”.

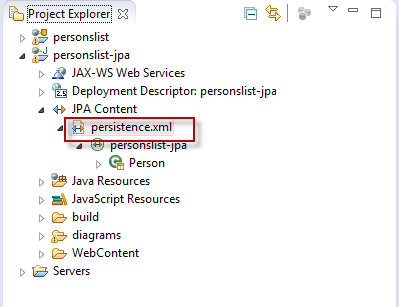


Similarly create the attribute name “lastName” as shown below.

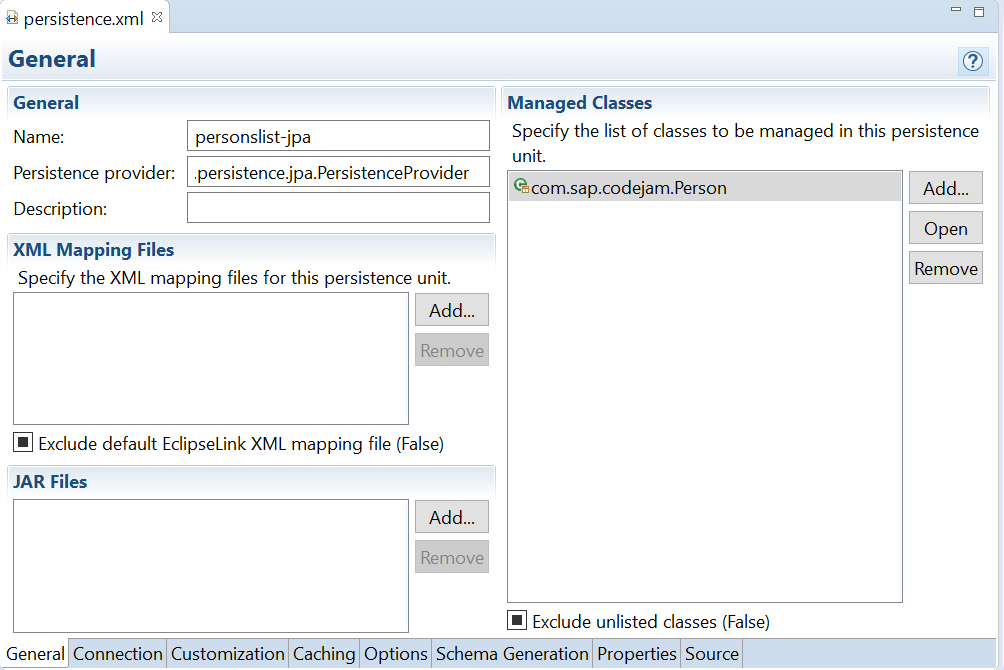
Press “Ctrl + SHIFT + S” to save it.



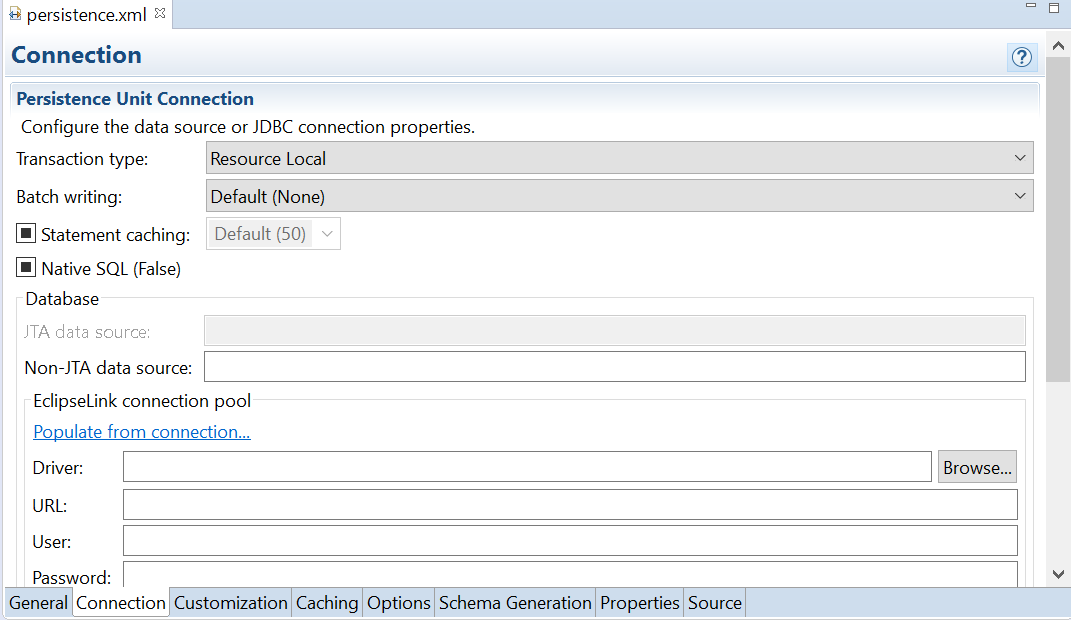
8. From the Project Explorer, double click on the “persistence.xml” to open it in the editor.



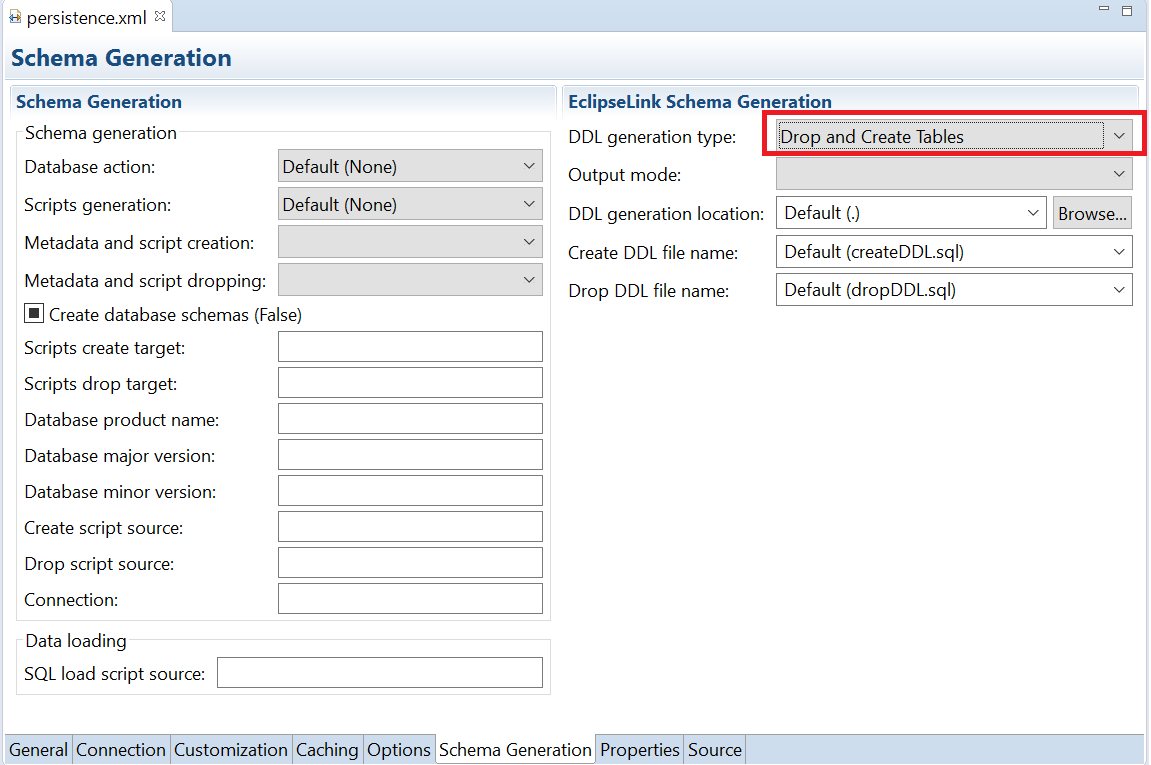
9. On the General tab, in the Presistence provider field, the value “org.eclipse.persistence.jpa.PersistenceProvider” appears. In the managed Classes add the “Person” class that we created.



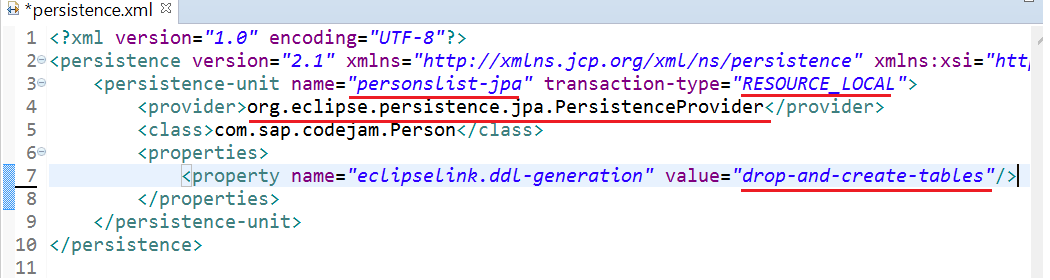
10. On Connection tab select “Resource Local” as the Transaction Type.



11. On “Schema Generation” tab, under “EclipseLink Schema Generation” section, select the value “Drop and Create Tables” as the DDL generation type.

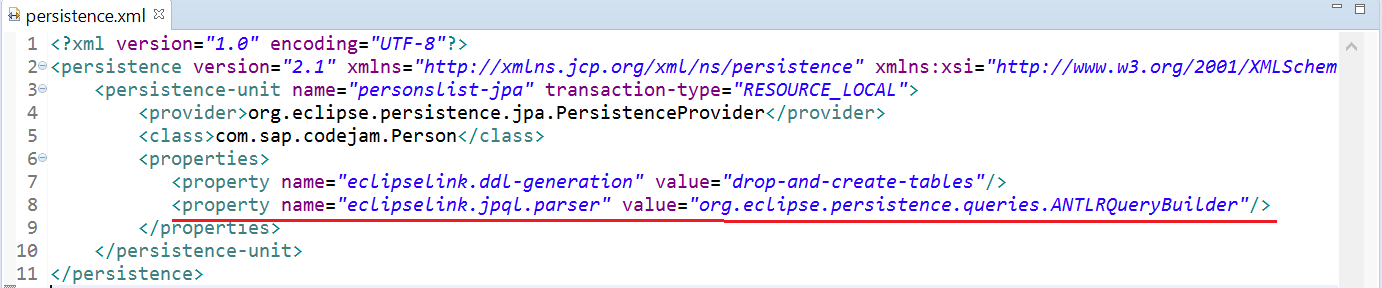


12. Save the file. After this the “Source” tab should look like the following screenshot.

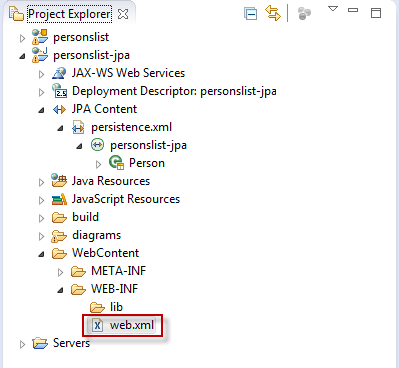


13. Add the following code in above file as shown in the snapshot and save it.

**<**property name**=**"eclipselink.jpql.parser" value**=**"org.eclipse.persistence.queries.ANTLRQueryBuilder"**/>**

****

14. Open the “web.xml” file located in the project explorer.



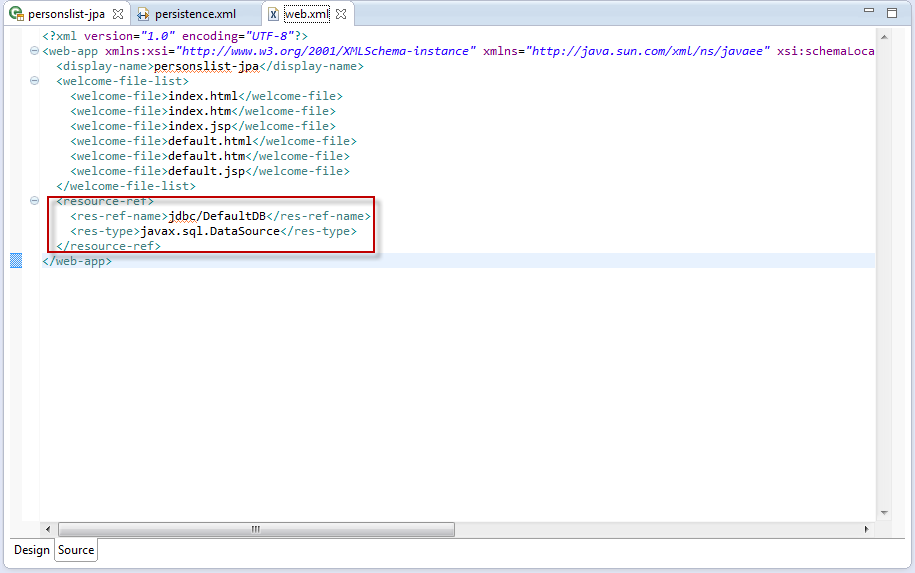
15. Add the following xml content in the editor just before the last line as shown in the snapshot. Save the file.

<resource-ref>

<res-ref-name>**jdbc/DefaultDB**</res-ref-name>

<res-type>**javax.sql.DataSource**</res-type>

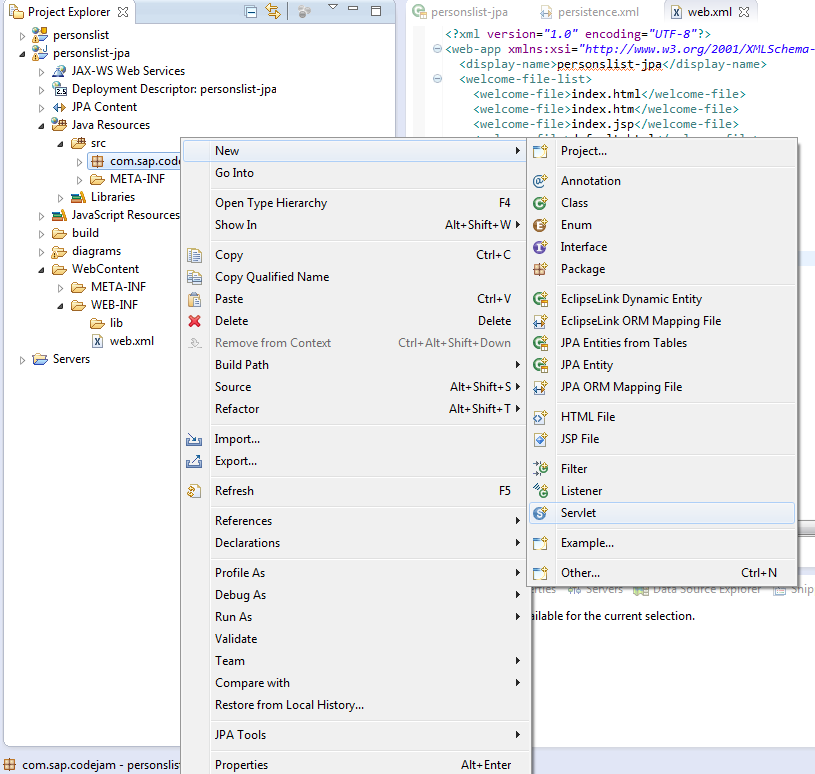
</resource-ref>



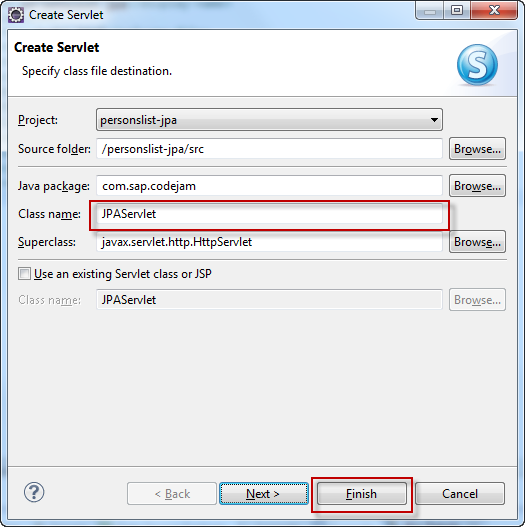
**2. Expose the JPA as OData service**

Follow the below steps to expose the JPA Entities as OData service.

1. Select the package “com.sap.codejam” from the Project Explorer. Right Click on the package and select “New” -> “Servlet” from the context menu.



2. Enter the name of the class as “JPAServlet” and click on the “Finish” button.



2. In the Web.xml file, copy and paste the following servlet information after the <resource-ref>……. </resource-ref> tag as shown in the below snapshot.

<servlet>

<description></description>

<display-name>JPAServlet</display-name>

<servlet-name> JPAServlet</servlet-name>

<servlet-class> com.sap.codejam.JPAServlet </servlet-class>

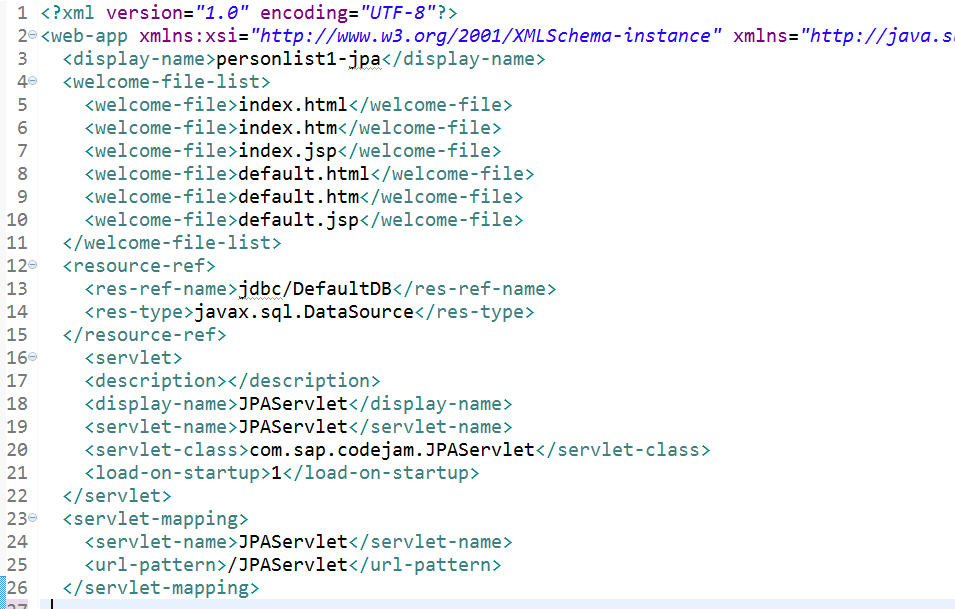
</servlet>

<servlet-mapping>

<servlet-name> JPAServlet</servlet-name>

<url-pattern>/JPAServlet</url-pattern>

</servlet-mapping>



3. Copy and paste the following code in the JPAServlet.java file. (Delete the existing code in the editor)

Save the file.

package com**.**sap**.**codejam**;**

**import** java**.**sql**.**Connection**;**

**import** java**.**sql**.**SQLException**;**

**import** java**.**util**.**HashMap**;**

**import** java**.**util**.**Map**;**

**import** javax**.**naming**.**InitialContext**;**

**import** javax**.**naming**.**NamingException**;**

**import** javax**.**persistence**.**EntityManagerFactory**;**

**import** javax**.**persistence**.**Persistence**;**

**import** javax**.**servlet**.**ServletConfig**;**

**import** javax**.**servlet**.**ServletException**;**

**import** javax**.**servlet**.**http**.**HttpServlet**;**

**import** javax**.**sql**.**DataSource**;**

**import** org**.**slf4j**.**Logger**;**

**import** org**.**slf4j**.**LoggerFactory**;**

**import** org**.**eclipse**.**persistence**.**config**.**PersistenceUnitProperties**;**

/\*\*

\* Servlet implementation class JPAServlet

\*/

public class JPAServlet **extends** HttpServlet **{**

private static final long serialVersionUID **=** 1L**;**

private DataSource ds**;**

EntityManagerFactory emf**;**

private static final Logger LOGGER **=** LoggerFactory

**.**getLogger**(**JPAServlet**.**class**);**

/\*\*

\* **@see** HttpServlet#HttpServlet()

\*/

public JPAServlet**()** **{**

**super();**

// TODO Auto-generated constructor stub

**}**

/\*\*

\* **@see** Servlet#init(ServletConfig)

\*/

public void init**(**ServletConfig config**)** **throws** ServletException **{**

// TODO Auto-generated method stub

Connection connection **=** **null;**

**try** **{**

InitialContext ctx **=** **new** InitialContext**();**

ds **=** **(**DataSource**)** ctx**.**lookup**(**"java:comp/env/jdbc/DefaultDB"**);**

connection **=** ds**.**getConnection**();**

Map properties **=** **new** HashMap**();**

properties**.**put**(**PersistenceUnitProperties**.**NON\_JTA\_DATASOURCE**,** ds**);**

emf **=** Persistence**.**createEntityManagerFactory**(**"personslist-jpa"**,** properties**);**

Utility**.**setEntityManagerFactory**(**emf**);**

**}** **catch** **(**NamingException e**)** **{**

**throw** **new** ServletException**(**e**);**

**}** **catch** **(**SQLException e**)** **{**

LOGGER**.**error**(**"Could not determine database product."**);**

**throw** **new** ServletException**(**e**);**

**}** **finally** **{**

**if** **(**connection **!=** **null)** **{**

**try** **{**

connection**.**close**();**

**}** **catch** **(**SQLException x**)** **{**

LOGGER**.**debug**(**"Unable to close connection."**,** x**);**

**}**

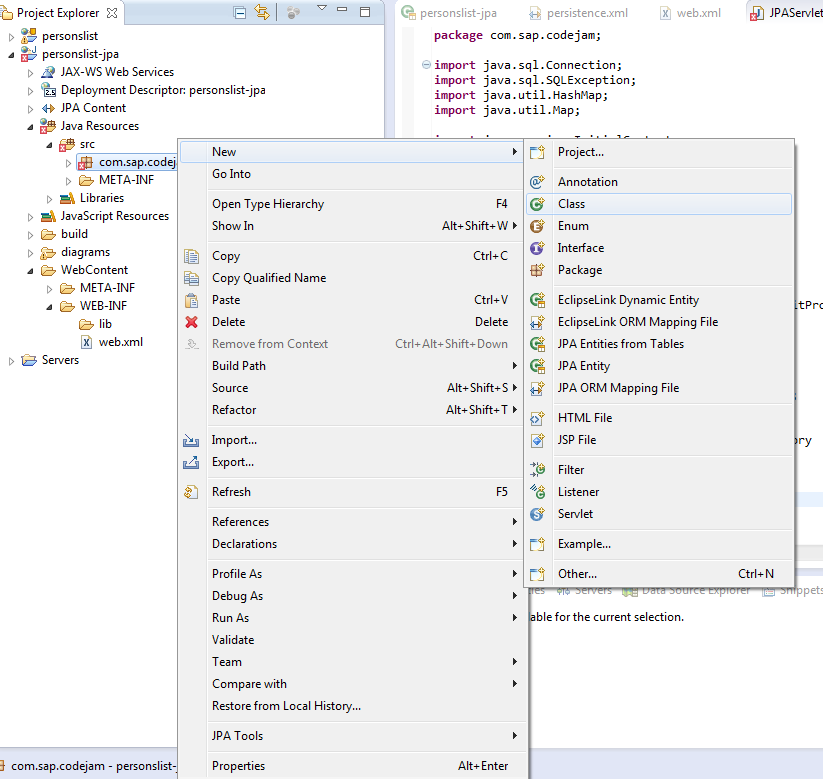
**}**

**}**

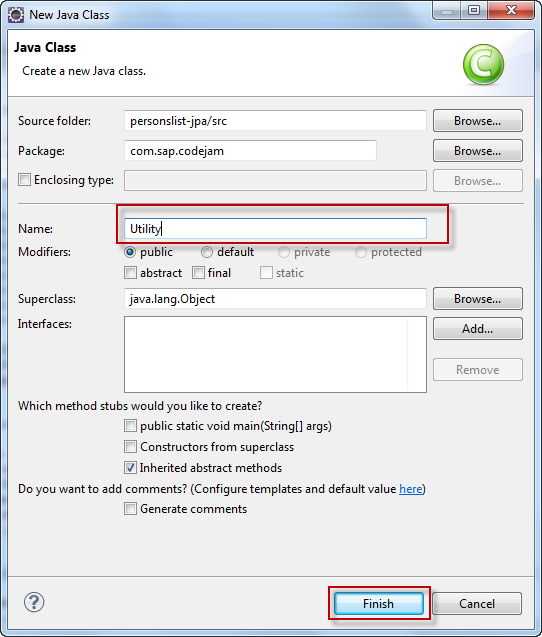
**}**

**}**

4. Select the package “com.sap.codejam” from the Project Explorer. Right Click on the package and select “New” -> “Class” from the context menu.



5. Enter the name of the class as “Utility” and click on the “Finish” button.



6. Copy and paste the following code in the “Utility”. (Delete the existing code in the editor)

Save the file.

package com**.**sap**.**codejam**;**

**import** javax**.**persistence**.**EntityManagerFactory**;**

public class Utility **{**

private static EntityManagerFactory emf**;**

public static EntityManagerFactory getEntityManagerFactory**()** **{**

**if** **(**emf **==** **null)** **{**

**throw** **new** IllegalArgumentException**(**

"EntityManagerfactory is not initialized!!!"**);**

**}**

**return** emf**;**

**}**

public static void setEntityManagerFactory**(**EntityManagerFactory emf**)** **{**

Utility**.**emf **=** emf**;**

**}**

**}**

7. Create a new class named “PersonsListServiceFactory” under the package “com.sap.codejam”.

Copy and Paste the following code in the class (Delete the existing content of the class)

Save the file.

package com.sap**.**codejam**;**

**import** javax**.**persistence**.**EntityManagerFactory**;**

**import** org**.**apache**.**olingo**.**odata2**.**jpa**.**processor**.**api**.**ODataJPAContext**;**

**import** org**.**apache**.**olingo**.**odata2**.**jpa**.**processor**.**api**.**ODataJPAServiceFactory**;**

**import** org**.**apache**.**olingo**.**odata2**.**jpa**.**processor**.**api**.**exception**.**ODataJPARuntimeException**;**

public class PersonsListServiceFactory **extends** ODataJPAServiceFactory **{**

private static final String PERSISTENCE\_UNIT\_NAME **=** "personslist-jpa"**;**

@Override

public ODataJPAContext initializeODataJPAContext**()**

**throws** ODataJPARuntimeException **{**

ODataJPAContext oDataJPAContext **=** **this.**getODataJPAContext**();**

**try** **{**

EntityManagerFactory emf **=** Utility**.**getEntityManagerFactory**();**

oDataJPAContext**.**setEntityManagerFactory**(**emf**);**

oDataJPAContext**.**setPersistenceUnitName**(**PERSISTENCE\_UNIT\_NAME**);**

**return** oDataJPAContext**;**

**}** **catch** **(**Exception e**)** **{**

**throw** **new** RuntimeException**(**e**);**

**}**

**}**

**}**

8. Add the following xml code to the web.xml file just above the last line as shown in the below snapshot.

Save the file.

<servlet>

<servlet-name>**ODataServlet**</servlet-name>

<servlet-class>**org.apache.cxf.jaxrs.servlet.CXFNonSpringJaxrsServlet**</servlet-class>

<init-param>

<param-name>**javax.ws.rs.Application**</param-name>

<param-value>**org.apache.olingo.odata2.core.rest.app.ODataApplication**</param-value>

</init-param>

<init-param>

<param-name>**org.apache.olingo.odata2.service.factory**</param-name>

<param-value>**com.sap.codejam.PersonsListServiceFactory**</param-value>

</init-param>

<load-on-startup>**2**</load-on-startup>

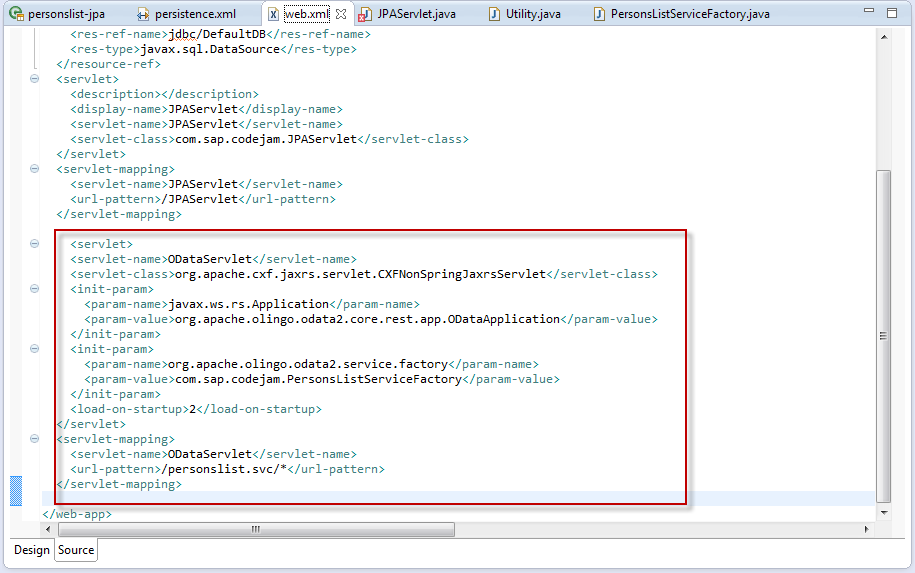
</servlet>

<servlet-mapping>

<servlet-name>**ODataServlet**</servlet-name>

<url-pattern>**/personslist.svc/\***</url-pattern>

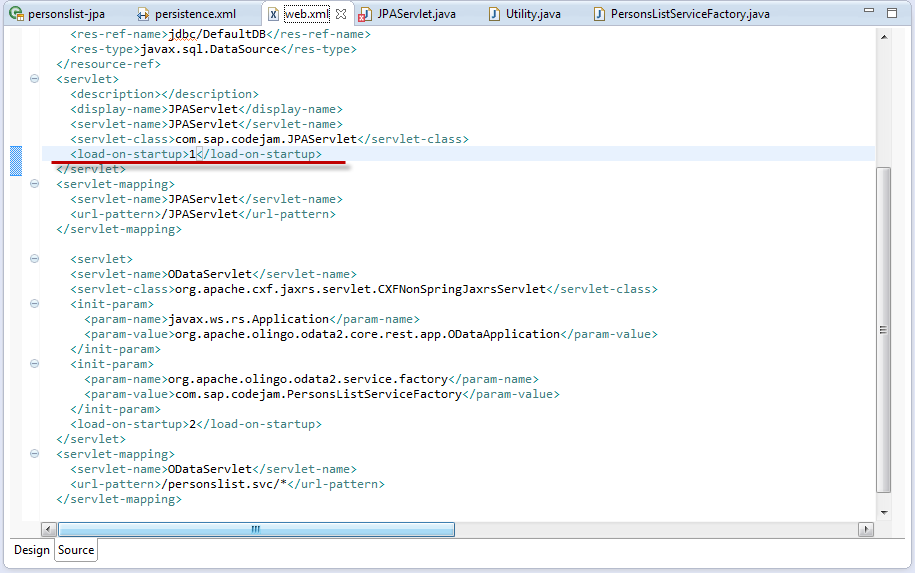
</servlet-mapping>



9. Add the below xml code for the “JPAServlet” as underlined in red in the below snapshot.

Save the file.

<load-on-startup>**1**</load-on-startup>

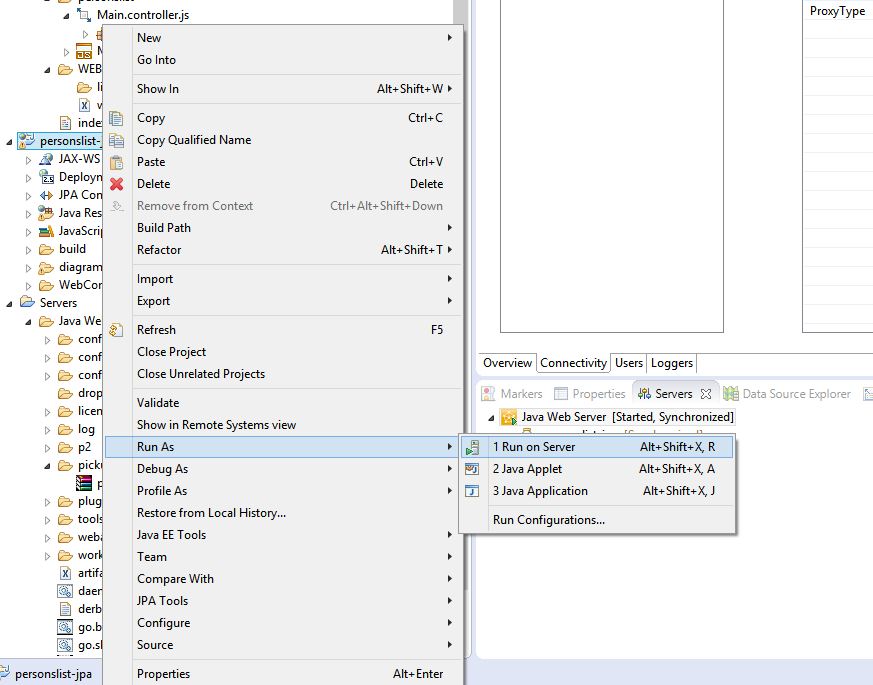


**3. Deploy the application to SAP Cloud platform trial account**

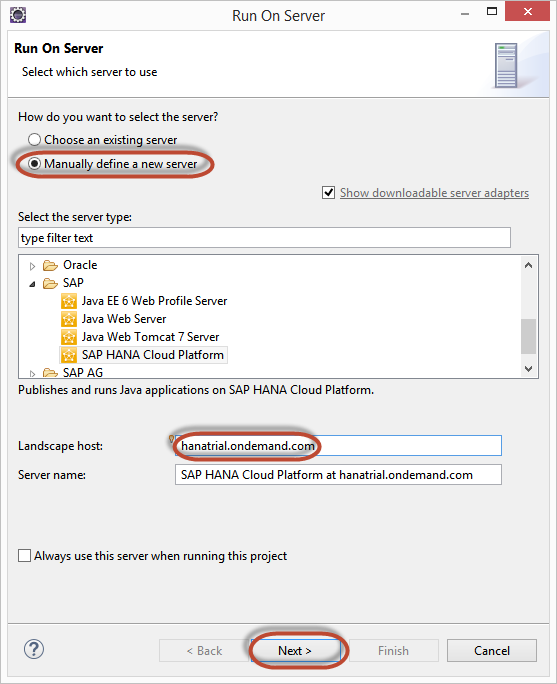
Now, we shall deploy the application to SAP HANA Cloud platform trial account.

Note: If you have not registered yet, please follow the steps mentioned [here to create an account](https://hcp.sap.com/developers.html?cq_ck=1470685205397#section_section_2)

1. Right click on the project and select “Run as” -> “Run on Server”



1. Enter the options as shown in the below screen shot



1. Enter your SAP HANA Cloud Platform details as shown below

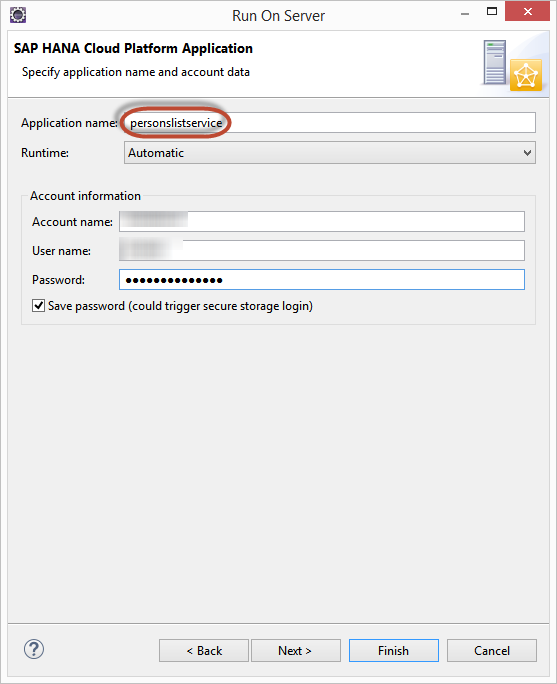
Application name = personslistservice

Runtime=Automatic

Account name=<username>trial

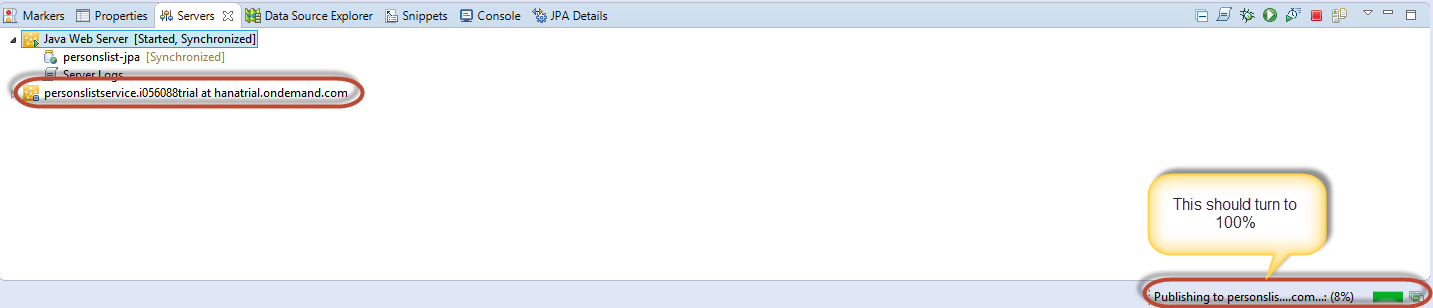
User name=<username>

Password=<password>



**Note: This is just for your information that Application name here is “personlistservice’ which is different from the project name. It is not mandatory to give different name here.**

1. Click on the Finish button. Wait for the application to get deployed successfully



**Note: you should be able to access the URL below, by replacing the account name.**

As soon as the application is published, copy the URL to your web browser and append the term “personslist.svc” to the URL. The URL appears as below:

**https://personslistservice<accountname>.hanatrial.ondemand.com/personslist-jpa/personslist.svc/**

The oData Service appears as below:

