
SAP NW CLOUD HANDS-ON WORKSHOP

CD261

Exercises

Sajjad Ahmed, Steven Taylor / SAP

Table of Contents

Introduction	P. 3
Application Description	P. 3
Workshop Agenda	P. 3
Exercise 1 – Configure Development Environment	P. 4
Exercise 2 – Import Code Files	P. 15
Exercise 3 – Develop Persistence Code	P. 19
Exercise 4 – Develop Business Logic Code	P. 26
Exercise 5 – Develop UI	P. 28
Exercise 6 – Run Application Locally	P. 31
Exercise 7 – Run Application in the Cloud	P. 38
Exercise 8 – Monitor Application	P. 44

Introduction

Welcome to the hands-on workshop for SAP NetWeaver Cloud. Over the next two hours, you will use this brand new platform to develop a web-based application, deploy it directly to the cloud and integrate it with both SuccessFactors Employee Central and on-premise SAP ERP system. Once you're done, you will be equipped with the knowledge and skills to quickly build your own applications in the cloud, now made much easier with the rich set of services provided by NetWeaver Cloud.

Application Description

In this workshop, you will build a simple application that allows a user to schedule events, choose venues and assign participants. The venues will be pulled from the on-premise ERP system, while the list of participants will come from SuccessFactors. The application will fully leverage the key services of the NetWeaver Cloud platform including identity management, persistence, unstructured storage and on-premise connectivity.

Workshop Agenda

This workshop is structured into 8 exercises, as follows:

1. Configure development environment;
2. Import skeleton and completed code files;
3. Develop persistence code;
4. Develop business logic code;
5. Develop UI;
6. Run application locally;
7. Run application in the cloud;
8. Monitor application.

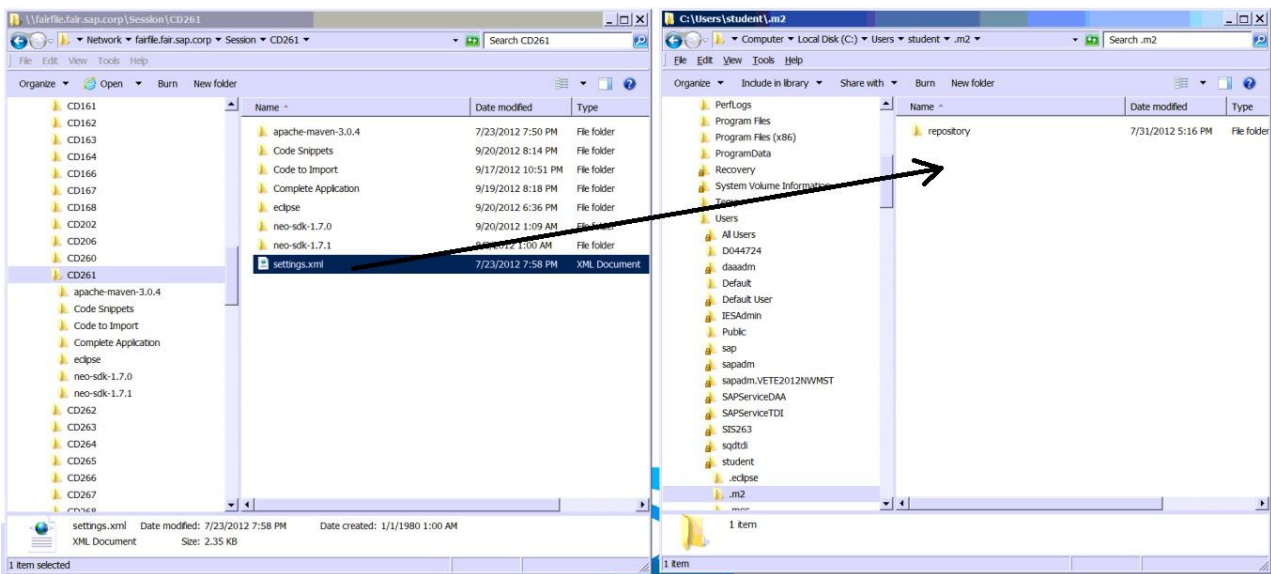
Exercise 1 – Configure Development Environment

In this exercise, you will start up Eclipse, do some basic setup for your development environment and create your application project.

1.1 Start Eclipse and Verify Settings

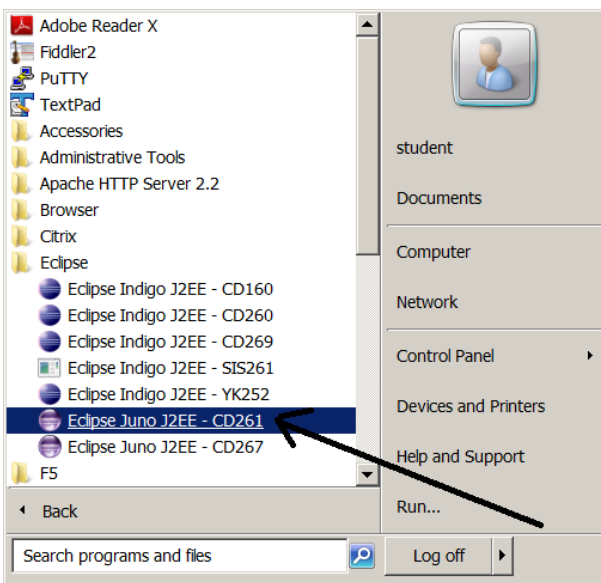
1.1.1 Copy Maven settings.xml File

- On your desktop, double-click on the **Session (TechEd FileServer)** shortcut.
- Open the **CD261** folder.
- Select the **settings.xml** file, right-click and hit **Copy**.
- Navigate to **C:\Users\student\m2**.
- Paste the file in this folder.



1.1.2 Start Eclipse

- Click on **Start → All Programs → Eclipse → Eclipse Juno J2EE – CD261**.
- If a security warning message appears, just click on **Run**.



1.1.3 Verify Proxy Settings

In Eclipse, click on **Window**→**Preferences**.

Open the **General** tab and click on **Network Connections**.

Verify that at least the entries listed below are checked. If not, then update accordingly by switching to *Active Provider* to **Manual** first.

Proxy entries:

HTTP, proxy, 8080, False

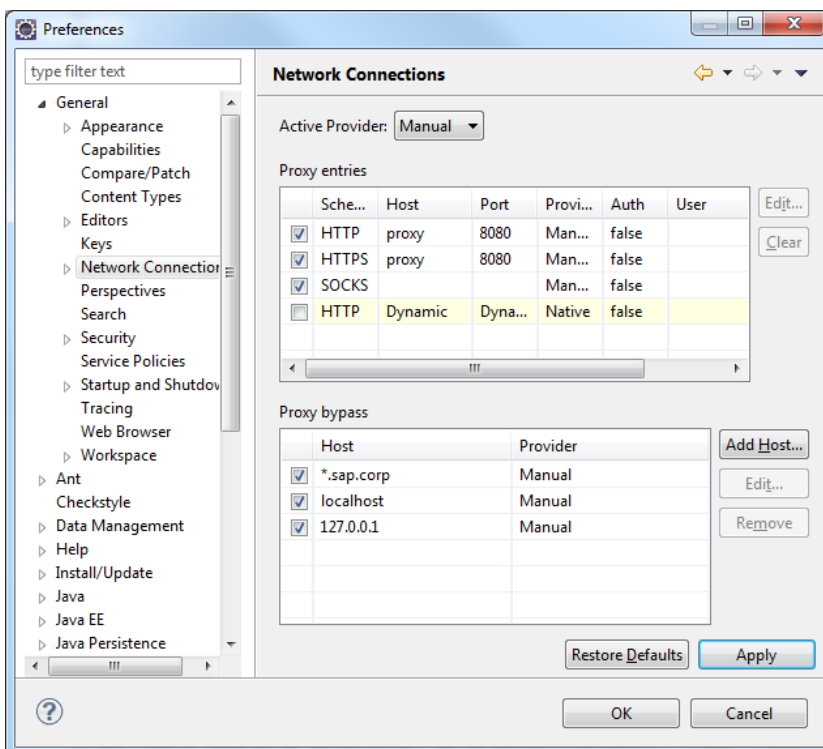
HTTPS, proxy, 8080, False

Proxy bypass:

*.sap.corp

localhost

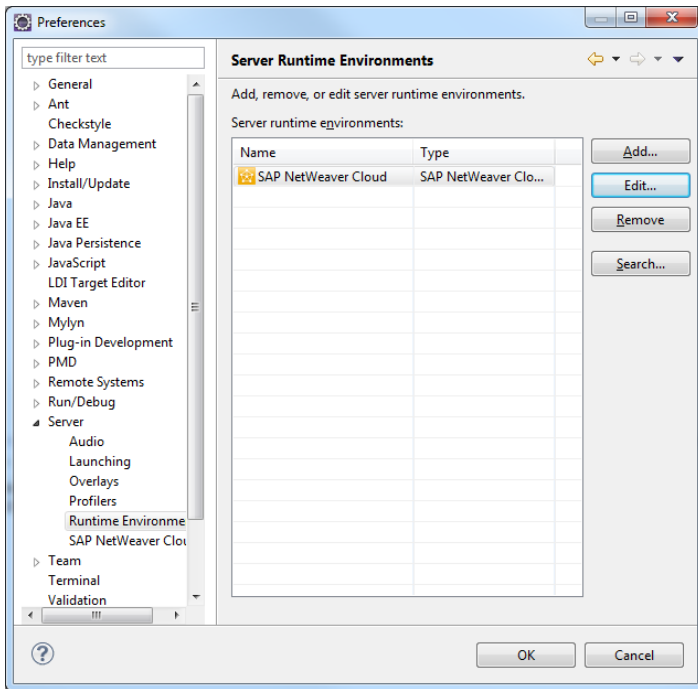
10.*.*



1.1.4 Verify Target Runtime Settings

In the **Preferences** window, open the **Server** tab.

Click on **Runtime Environments**.



Verify that there is a **SAP NetWeaver Cloud** entry. Click on it and then click on the **Edit** button.

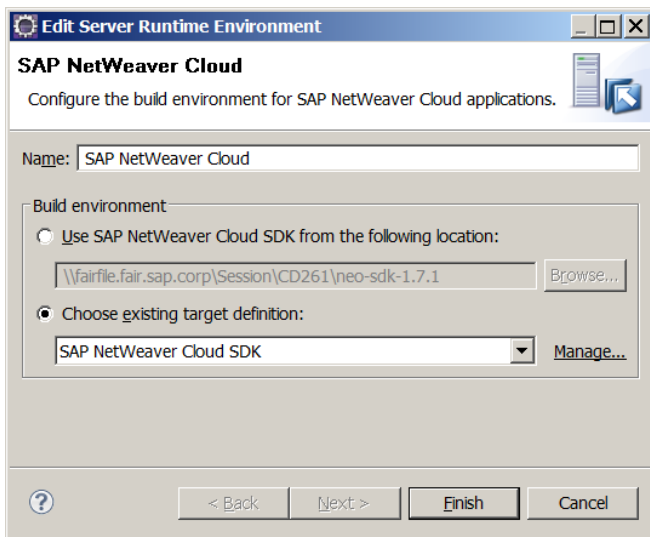
Verify that the popup window contains the following information:

Name: **SAP NetWeaver Cloud**

Build environment:

Choose existing target definition: **SAP NetWeaver Cloud SDK**.

Click on the **Cancel** button.



1.1.5 Verify NW Cloud Server Settings

In the **Preferences** window, still in the **Server** tab, click on **SAP NetWeaver Cloud**.

Verify the following fields:

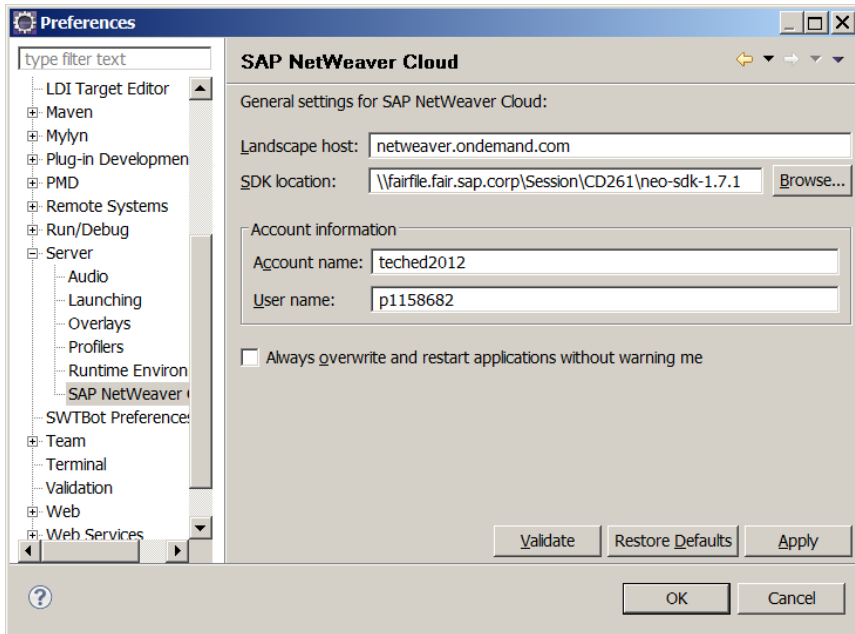
Landscape host: **netweaver.ondemand.com**

SDK location: **[\\fairfile.fair.sap.corp\Session\CD261\neo-sdk-1.7.1](#)**

Update the following fields:

Account name: **teched2012**

User name: **<your assigned username>**



Click on the **Validate** button and verify a positive validation result in a popup window. Click on **OK**.
Click on the **Apply** button.

1.1.6 Verify Maven Settings

In the **Preferences** window, open the **Maven** tab.

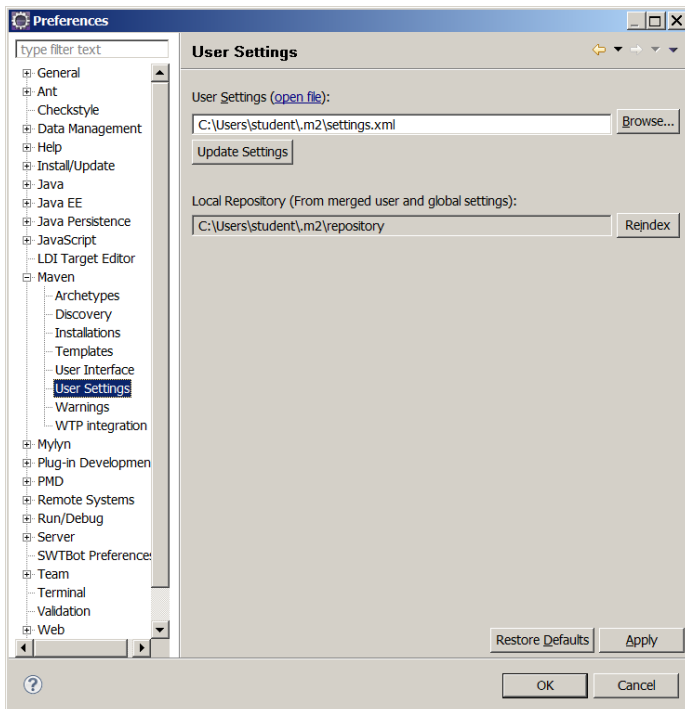
Click on **User Settings**.

Verify the following settings:

User Settings: **C:\Users\student\m2\settings.xml**.

Local Repository: **C:\Users\student\m2\repository**.

Click on **Update Settings**, then on **Apply**, then on **OK**.



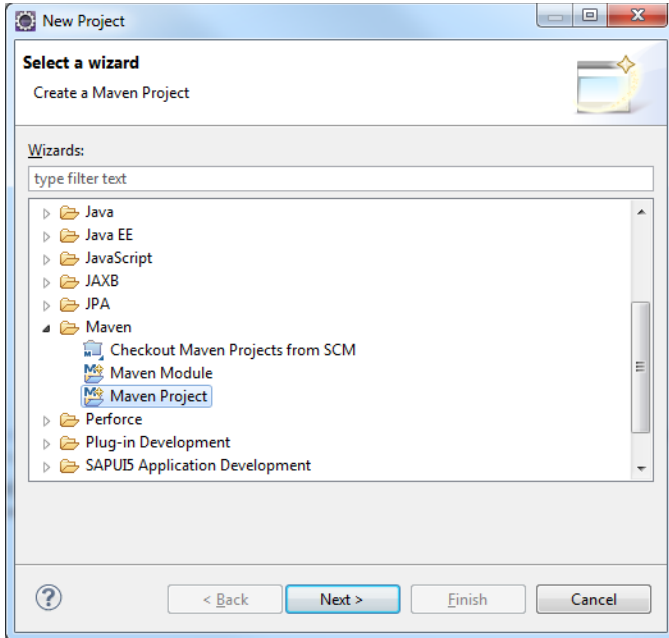
1.2 Create Maven Project

1.2.1 Create Project

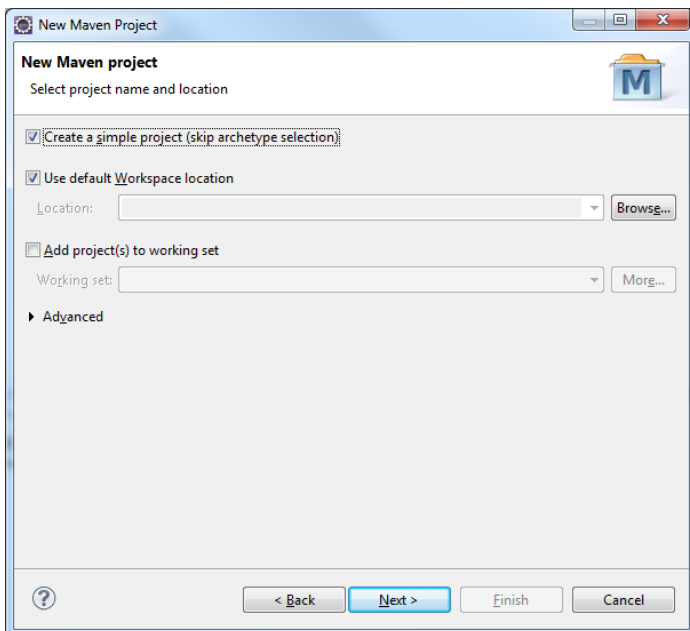
In Eclipse, click on **File**→**New**→**Project**.

In the resulting popup wizard, scroll down and open the **Maven** tab.

Select **Maven Project** and click on the **Next** button.



On the next page of the wizard, check the **Create a simple project (skip archetype selection)** box. Leave the other defaults and click on **Next**.



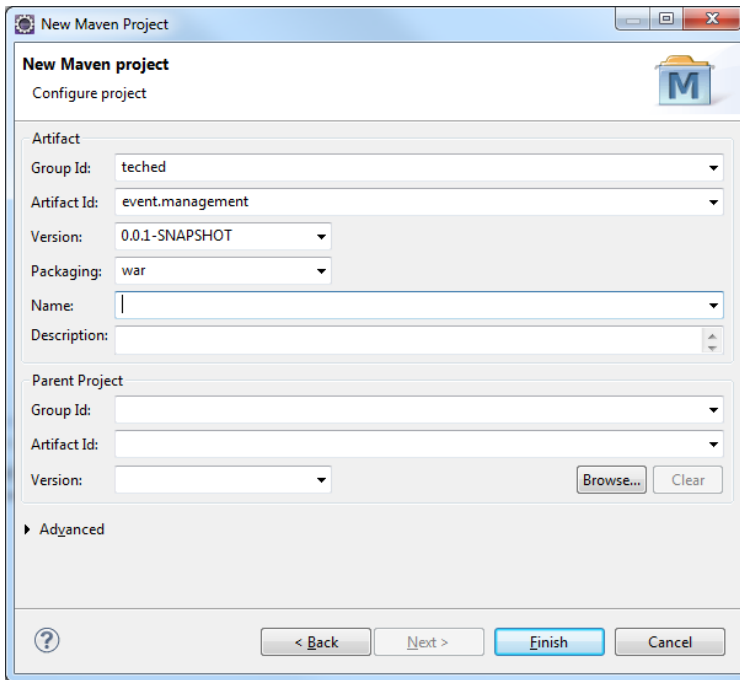
On the following page of the wizard, enter the following values:

Group Id: *teched*

Artifact Id: *event.management*

Packaging: select *war*

Leave everything else as is. Click on **Finish**.



1.2.2 Update pom.xml File

In Eclipse, in the Project Explorer, open the tree for your newly created *event.management* project.

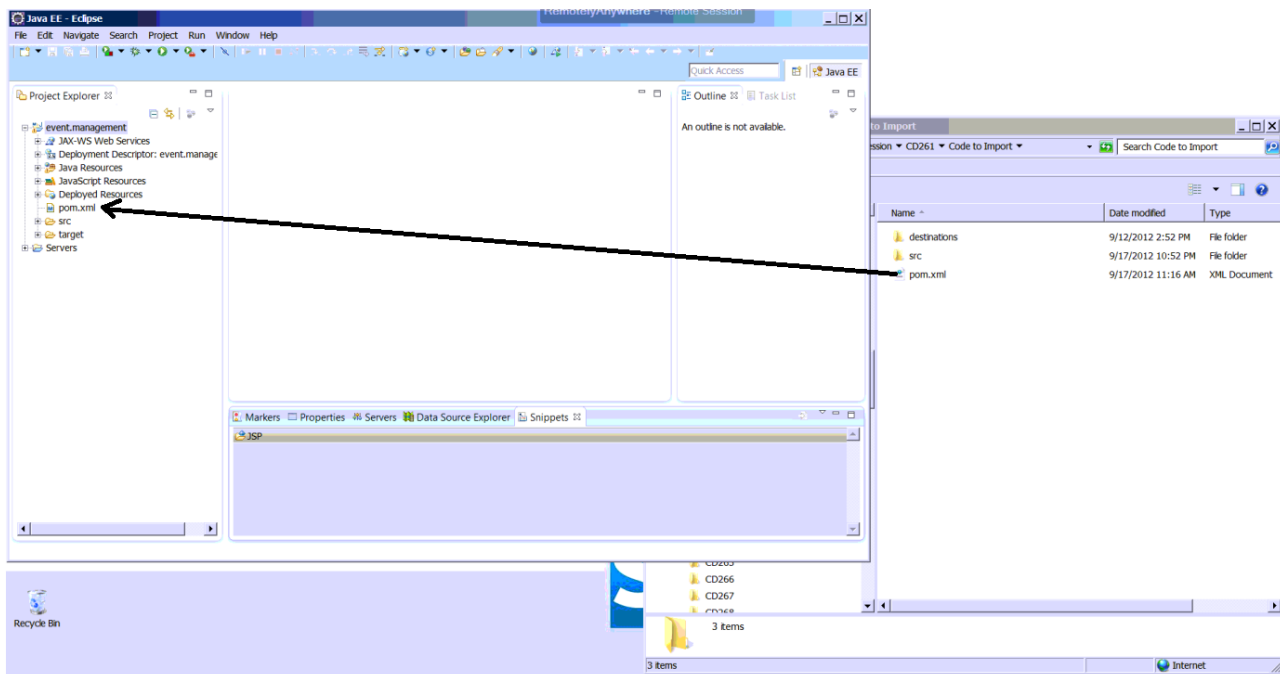
On your desktop, double-click on the *Session (TechEd FileServer)* folder.

In the Explorer window that opens up, double-click on the *CD261* folder.

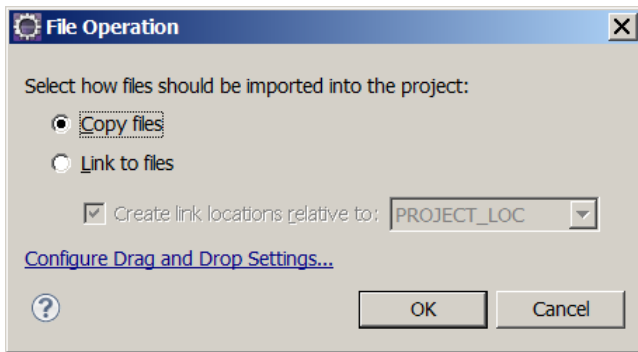
Double-click on the *Code to Import* folder.

Select the *pom.xml* file and drag it over to Eclipse.

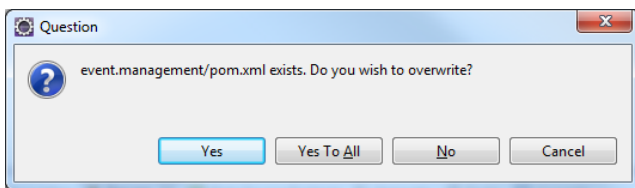
Drop it over the existing *pom.xml* entry in the Project Explorer.



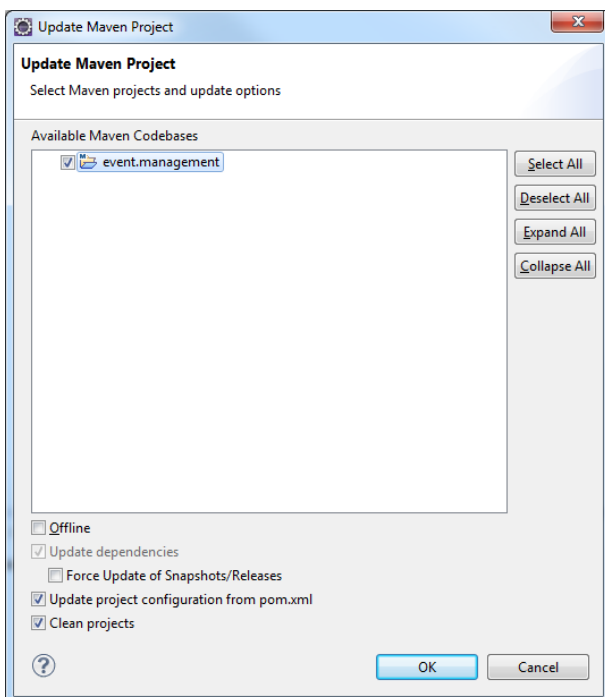
Click on **OK** in the *File Operation* window that pops up.



Select **Yes** in the *Question window that pops up*.



Right-click on the **event.management** title at the top level of the project tree in the Project Explorer. Click on the **Maven** entry, and then click on **Update Project**. Leave the defaults in the window that pops up and click on **OK**.

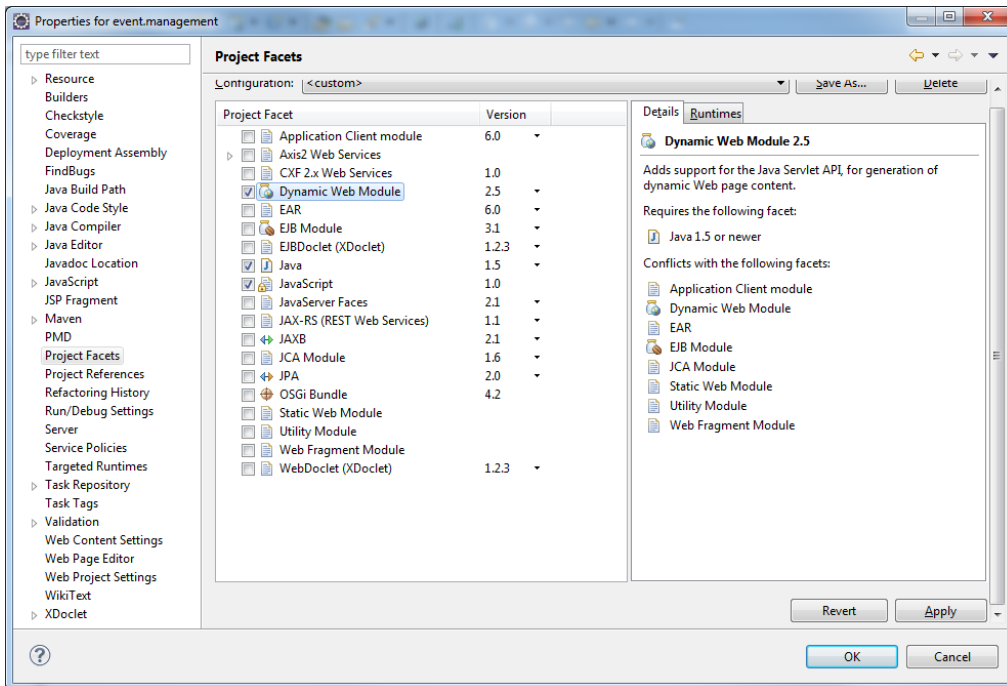


1.2.3 Update Project Properties

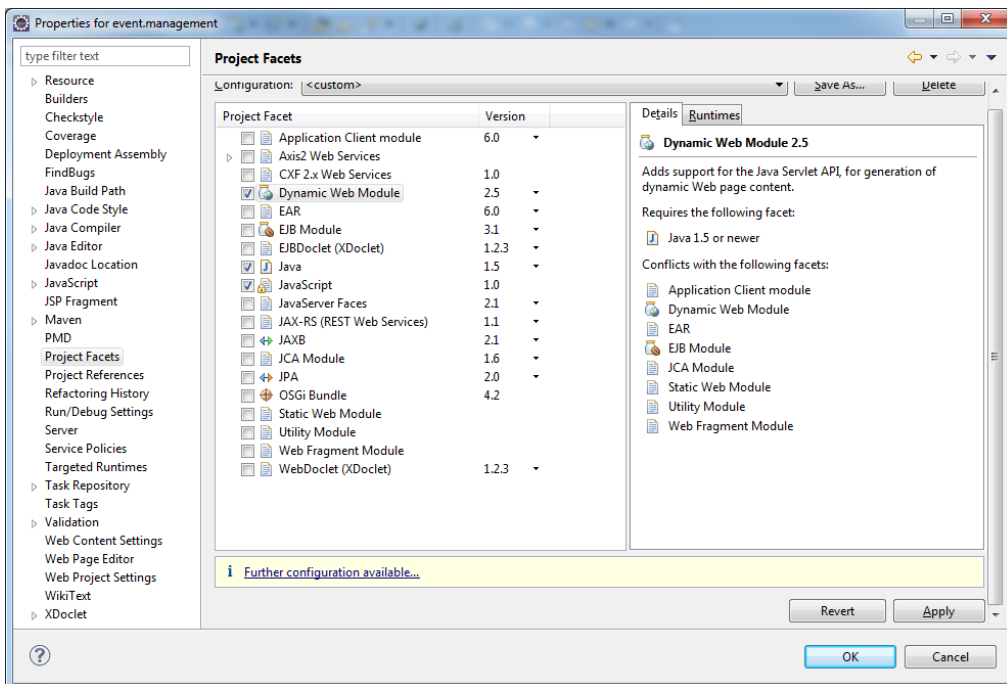
In Eclipse, in the Project Explorer, right-click on the **event.management** title at the top level of the project tree in the Project Explorer.

Click on the **Properties** entry.

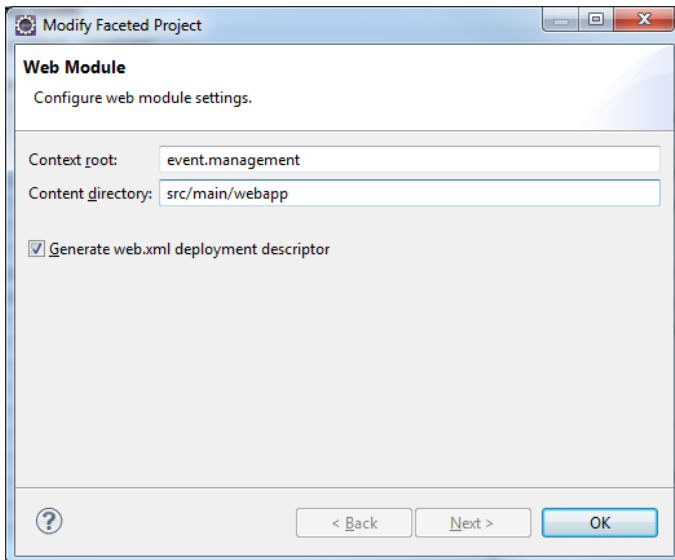
In the *Properties* window, click on **Project Facets**.



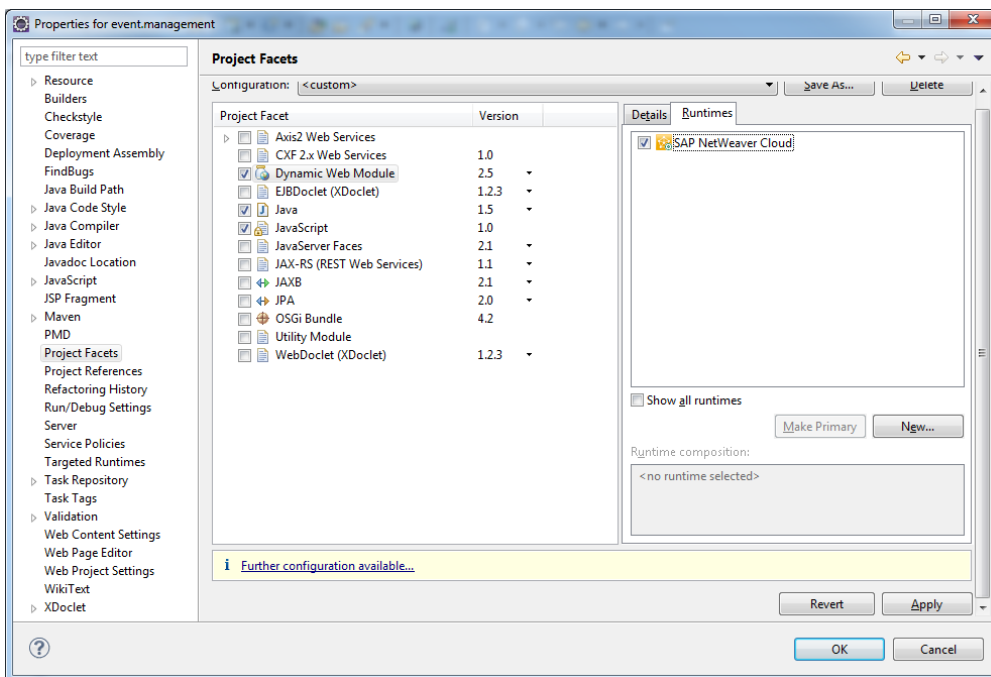
Select the **Dynamic Web Module** entry and uncheck it, and then click on the **Apply** button. Reselect the **Dynamic Web Module** entry and check it. Update the version to **2.5**. Then click on the **Further configuration available** link that appears at the bottom of the window.



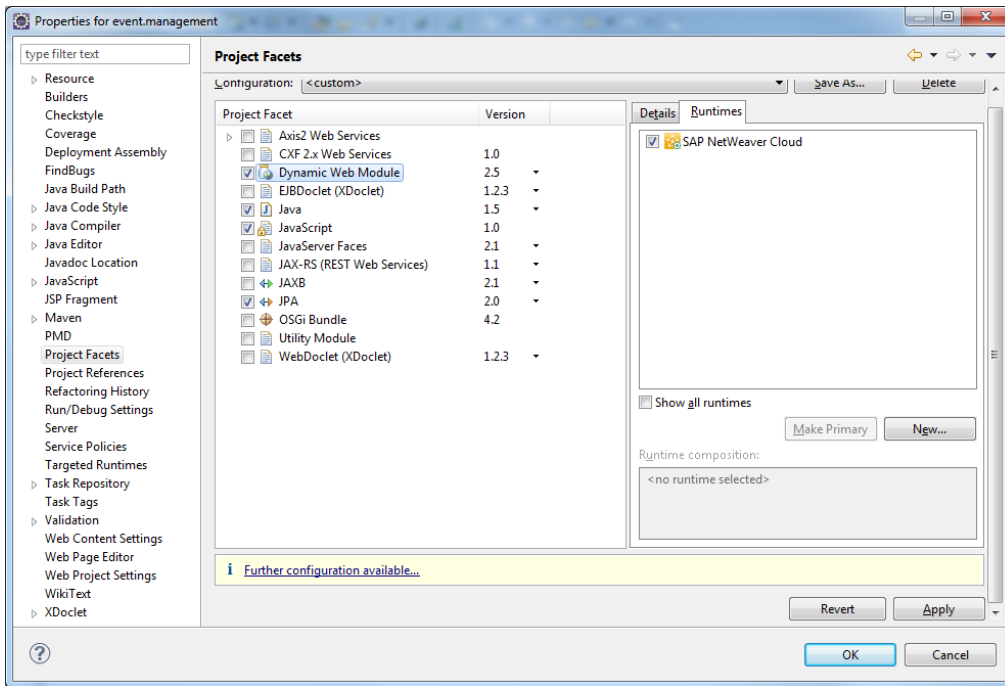
Update the **Content directory** entry to **src/main/webapp** and click on **OK**.



Click on the **Runtimes** tab in the upper right part of the window. Check the checkbox for the **SAP NetWeaver Cloud** entry and click on **Apply**.



Next, select the **JPA** entry and check its checkbox. Then click on the **Further configuration required** link that appears at the bottom of the window.

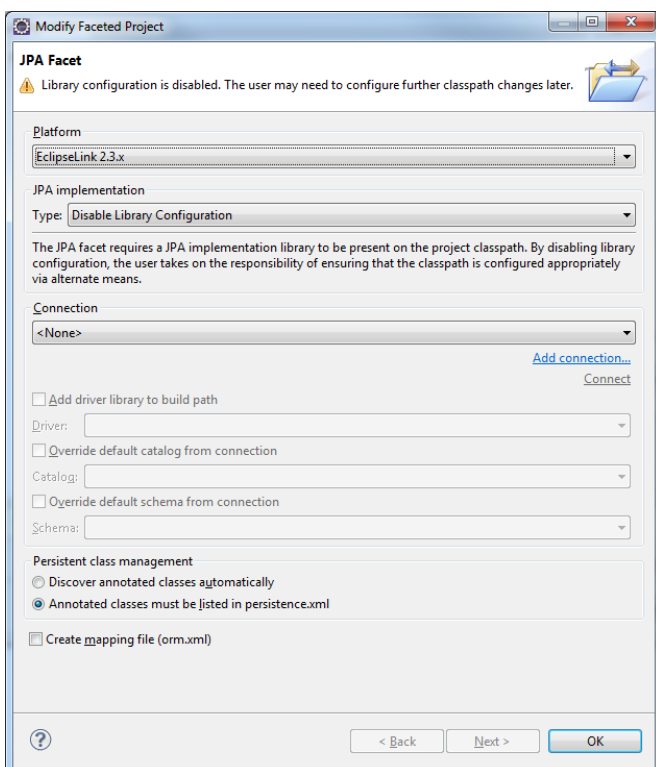


In the **JPA Facet** window that appears, update the following fields:

Platform: EclipseLink 2.3.x

Type: Disable Library Configuration

Click on **OK**, then on **Apply**, then again on **OK**.



If you see a Maven error appear, then repeat the project update:

Right-click on the **event.management** title at the top level of the project tree in the Project Explorer.

Click on the **Maven** entry, and then click on **Update Project**.

Leave the defaults in the window that pops up and click on **OK**.

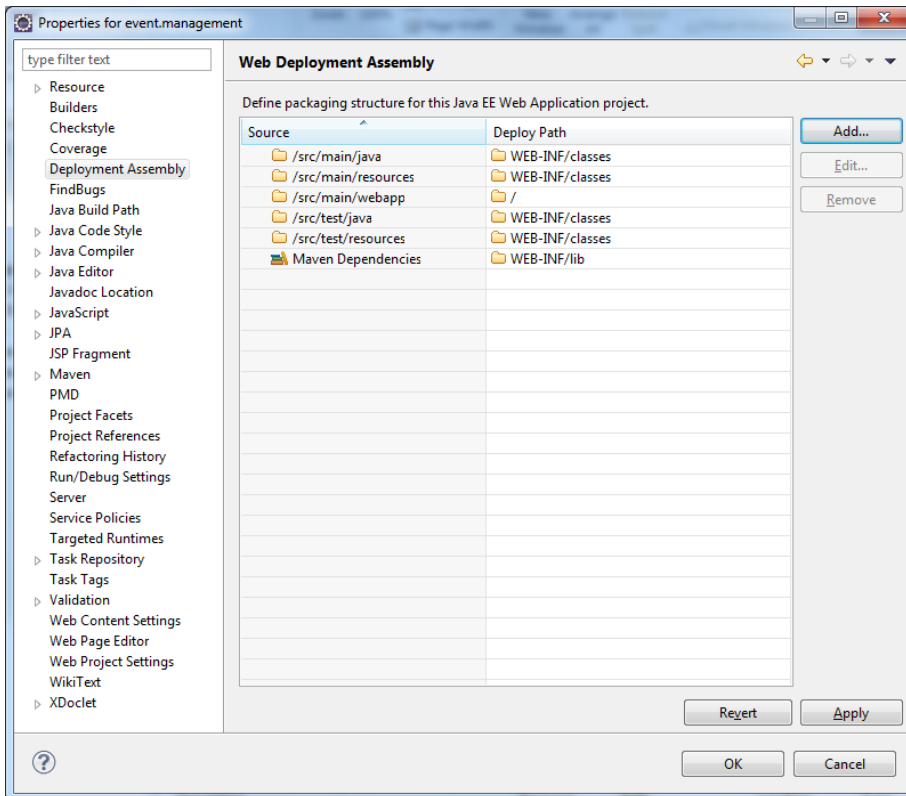
1.2.4 Verify Java Build Path Entries

In Eclipse, in the Project Explorer, right-click on the *event.management* title at the top level of the project tree in the Project Explorer.

Click on the **Properties** entry.

In the *Properties* window, click on **Deployment Assembly**.

Verify that there is an entry called **Maven Dependencies** in the *Web Deployment Assembly* window, and that it is set to **WEB-INF/lib**.



Exercise 2 – Import Code Files

In this exercise, you will import all the files you need for your application project and update a few additional settings.

2.1 Import Destinations

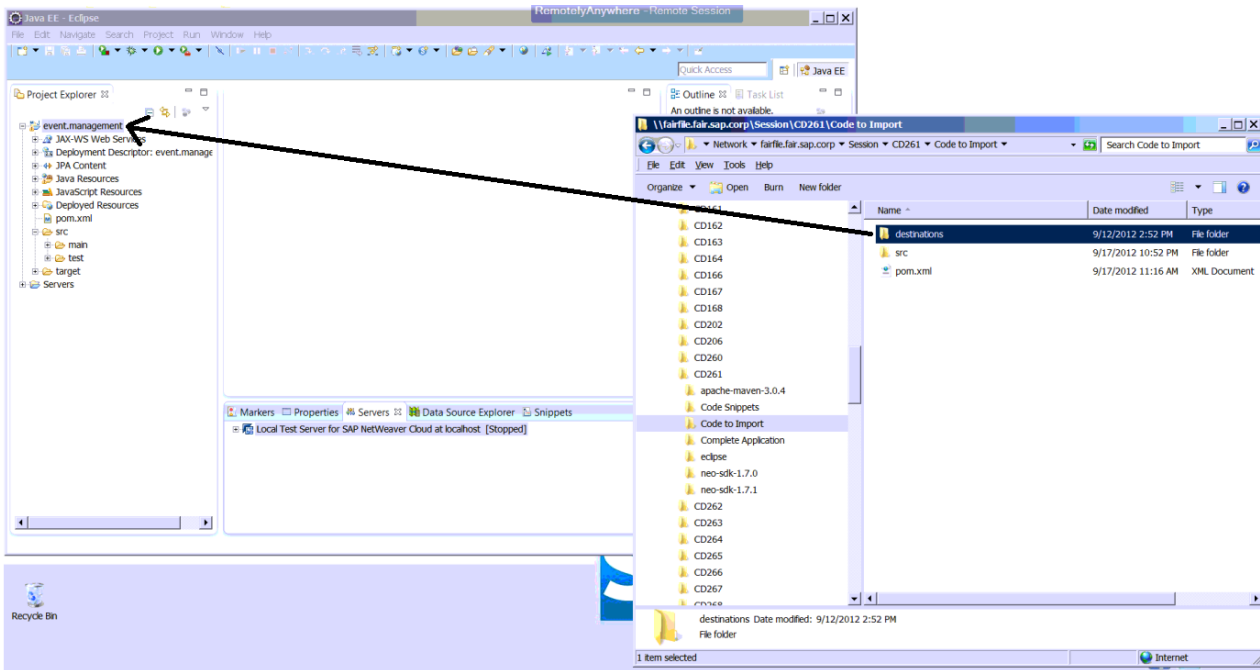
On your desktop, double-click on the **Session (TechEd FileServer)** folder.

In the Explorer window that opens up, double-click on the **CD261** folder.

Double-click on the **Code to Import** folder.

Select the **destinations** folder and drag it over to Eclipse.

Drop it over the existing **event.management** title at the top level of the project tree in the Project Explorer.



Click on **OK** if any *File Operation* window pops up.

Select **Yes** if any *Question* window pops up.

2.2 Import Data

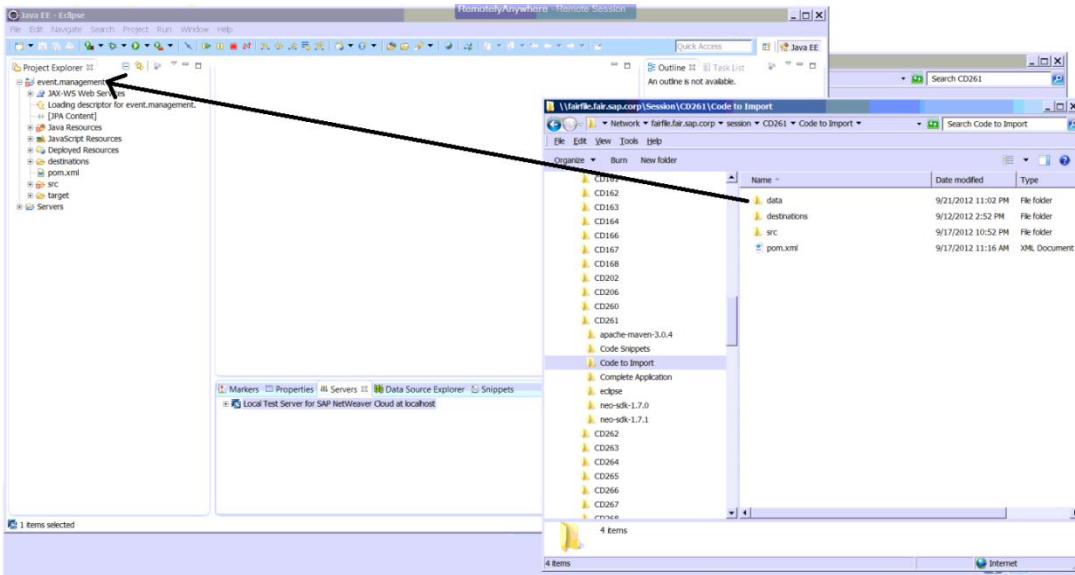
On your desktop, double-click on the **Session (TechEd FileServer)** folder.

In the Explorer window that opens up, double-click on the **CD261** folder.

Double-click on the **Code to Import** folder.

Select the **data** folder and drag it over to Eclipse.

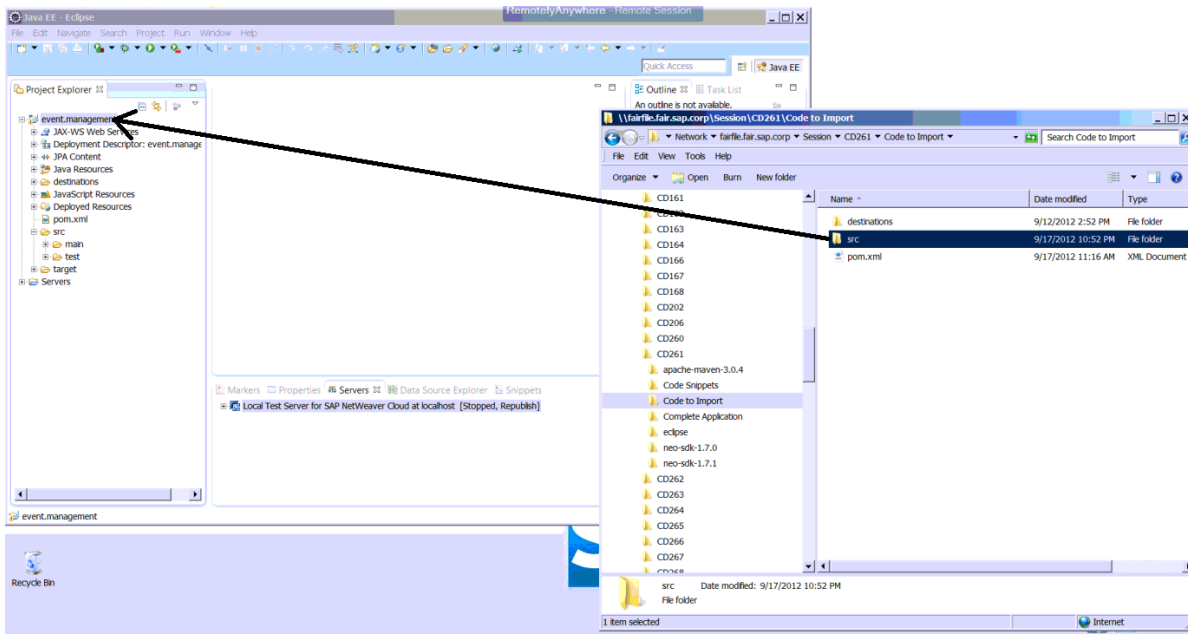
Drop it over the existing **event.management** title at the top level of the project tree in the Project Explorer.



Click on **OK** if any *File Operation* window pops up.
Select **Yes** if any *Question* window pops up.

2.3 Import Source Code

On your desktop, double-click on the **Session (TechEd FileServer)** folder.
In the Explorer window that opens up, double-click on the **CD261** folder.
Double-click on the **Code to Import** folder.
Select the **src** folder and drag it over to Eclipse.
Drop it over the existing **event.management** title at the top level of the project tree in the Project Explorer.



Click on **OK** in the *File Operation* window that pops up.
Select **Yes** in the *Question* window that pops up.

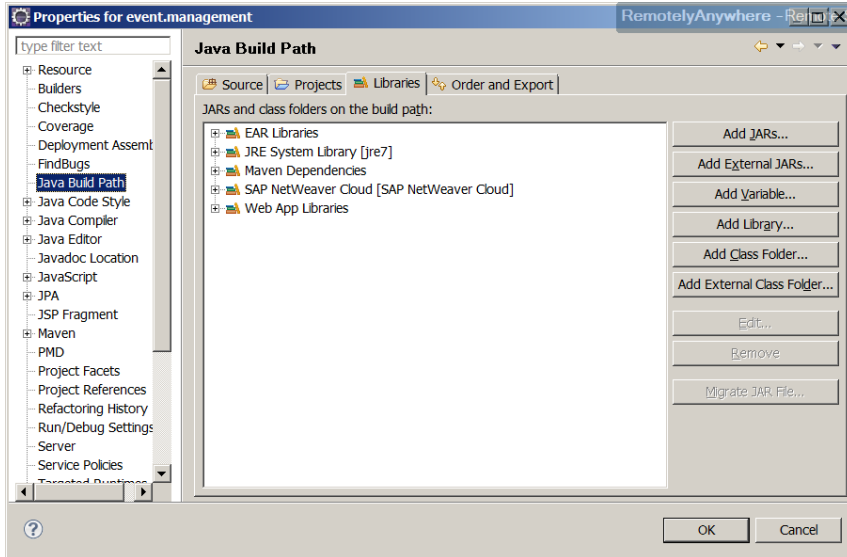
2.4 Add JARs to Build Path

In Eclipse, in the Project Explorer, right-click on the *event.management* title at the top level of the project tree in the Project Explorer.

Click on the *Properties* entry.

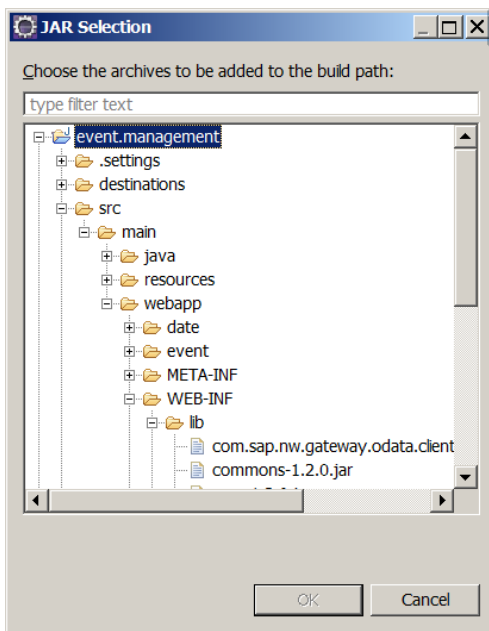
In the *Properties* window, click on *Java Build Path*.

Click on the *Libraries* tab.

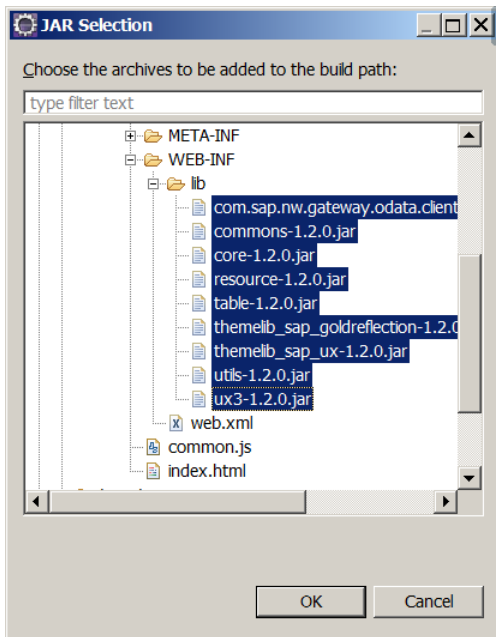


Click on the *Add JARs* button.

In the *JAR Selection* window that appears, open the *event.management* entry, then further open down to *src* → *main* → *webapp* → *WEB-INF* → *lib*.



Select all the jar files in that location and click on **OK**.



Click on **OK** in the *Properties window*.

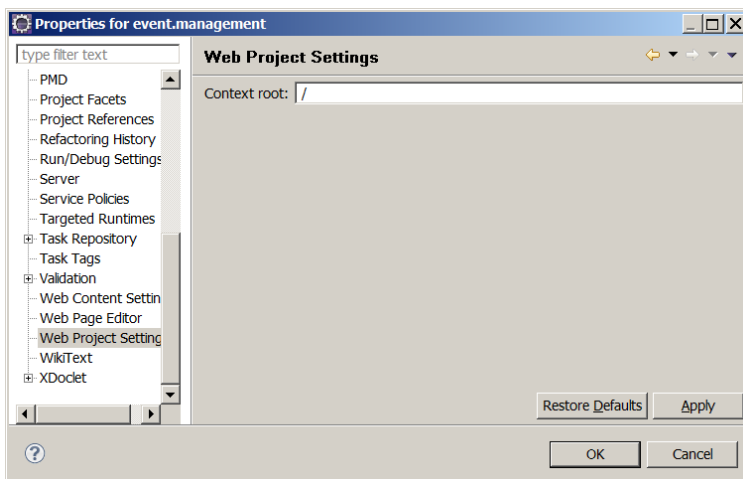
2.5 Update Web Project Settings

In Eclipse, in the Project Explorer, right-click on the **event.management** title at the top level of the project tree in the Project Explorer.

Click on the **Properties** entry.

In the *Properties* window, click on **Web Project Settings**.

Update the *Context root* field by replacing the word **events** with **/**.



Click on **Apply**, then on **OK** in the *Properties* window. Click **OK** on any other confirmation window that may appear, and then **OK** again in the *Properties* window.

Exercise 3 – Develop Persistence Code

In this exercise, you will complete the code needed for the persistence layer of the application.

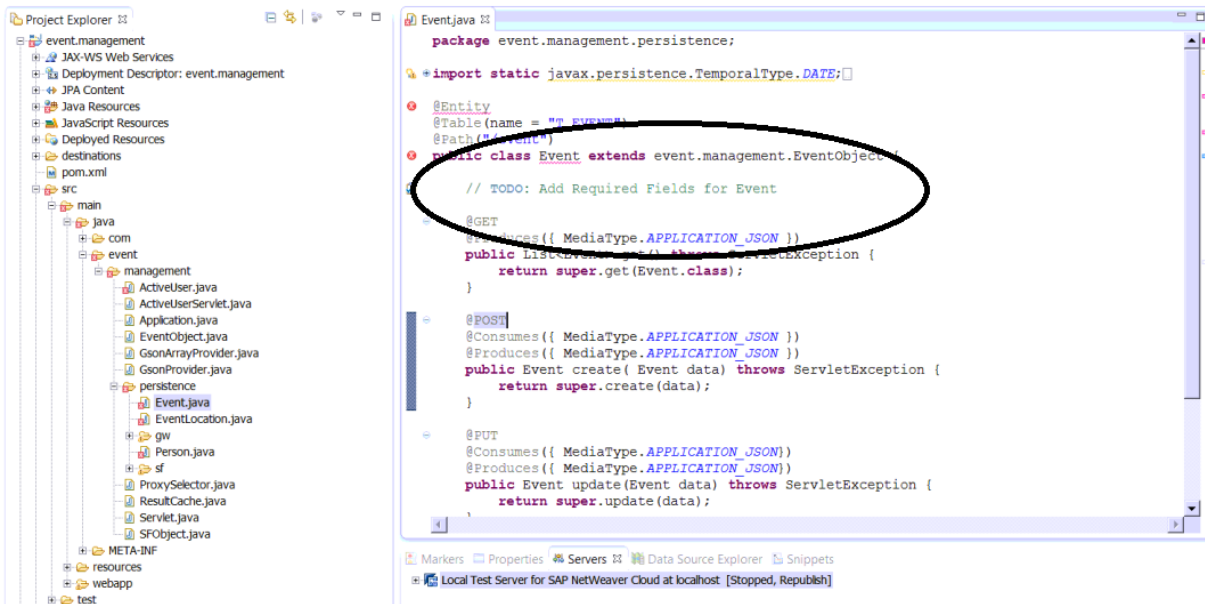
3.1 Complete Event.java

In Eclipse, in the Project Explorer, navigate to `src`→`main`→`java`→`event`→`management`→`persistence` under the `event.management` project.

Double-click on the `Event.java` file.

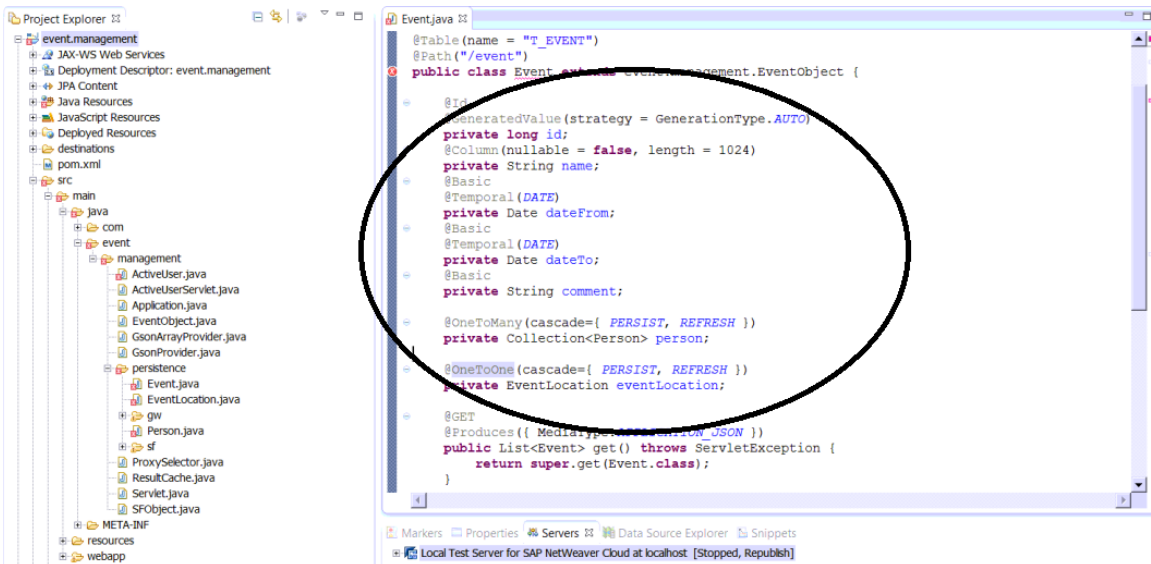
In the file, scroll down to the location with the following comment:

```
// TODO: Add Required Fields for Event
```



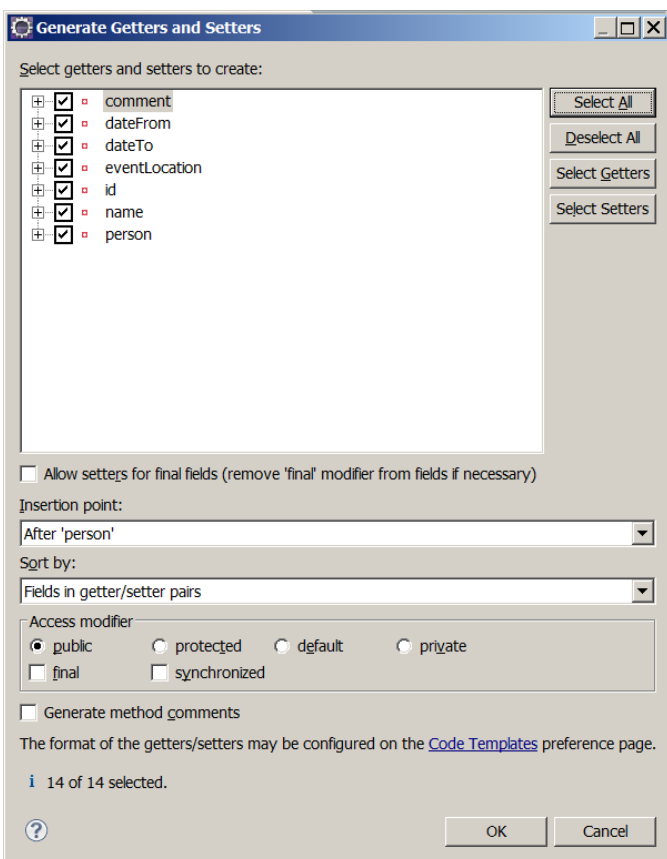
Replace that line with the following code snippet:

```
@Id
@GeneratedValue(strategy = GenerationType.AUTO)
private long id;
@Column(nullable = false, length = 1024)
private String name;
@Basic
@Temporal(DATE)
private Date dateFrom;
@Basic
@Temporal(DATE)
private Date dateTo;
@Basic
private String comment;
@OneToMany(cascade={ PERSIST, REFRESH })
private Collection<Person> person;
@OneToOne(cascade={ PERSIST, REFRESH })
private EventLocation eventLocation;
```



Alternatively, copy and paste the code directly from the file **Network**→**fairfile.fair.sap.corp**→**session**→**CD261**→**Code Snippets**→**Event.java**.

From the top level menus in Eclipse, click on **Source**→**Generate Getters and Setters**. Click on **Select All** in the window that appears, and click on **OK**.



Save the file in Eclipse.

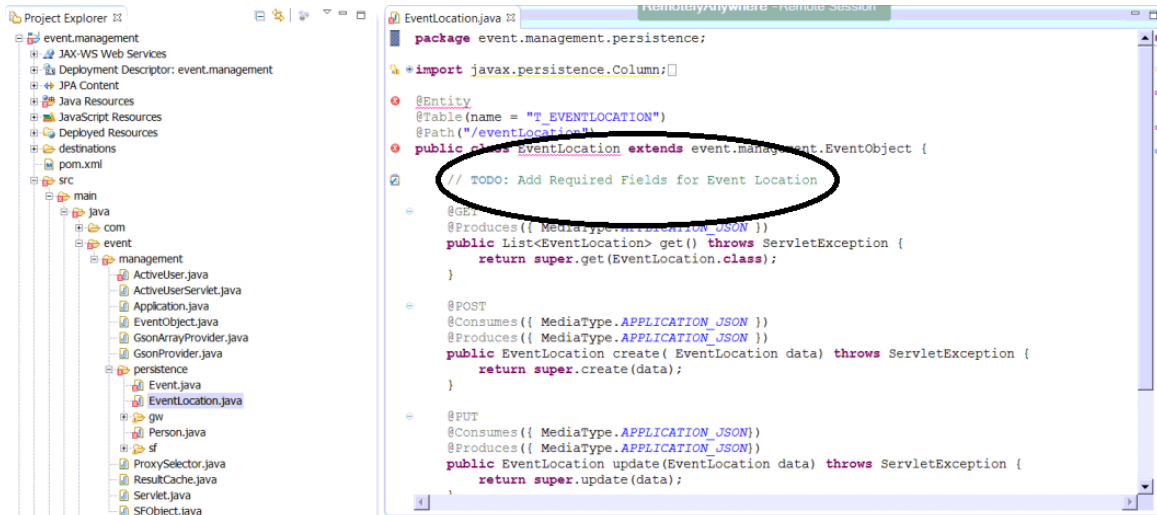
3.2 Complete EventLocation.java

In Eclipse, in the Project Explorer, navigate to **src**→**main**→**java**→**event**→**management**→**persistence** under the **event.management** project.

Double-click on the **EventLocation.java** file.

In the file, scroll down to the location with the following comment:

```
// TODO: Add Required Fields for Event Location
```



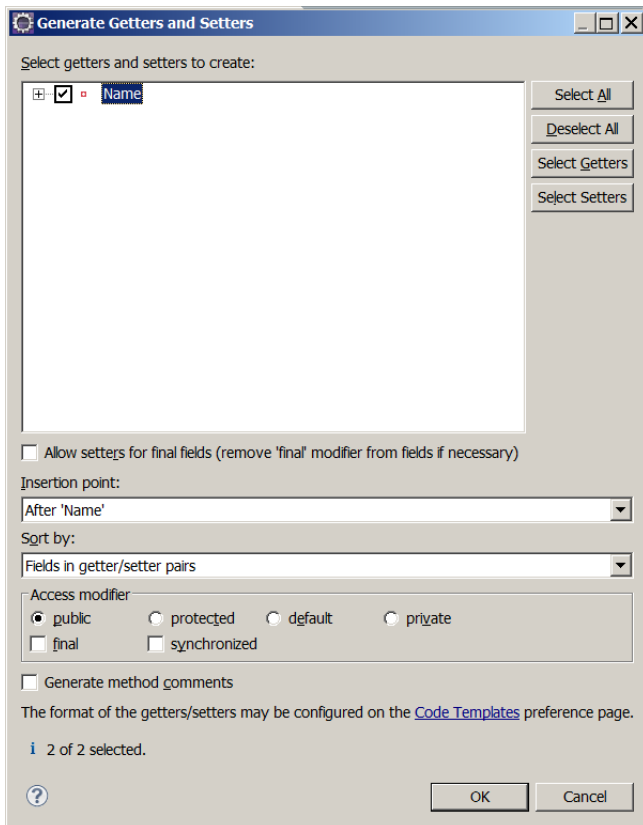
Replace that line with the following code snippet:

```
@Id
@Column(nullable = false, length = 128)
private String Name;
```



Alternatively, copy and paste the code directly from the file **Network**→**fairfile.fair.sap.corp**→**session**→**CD261**→**Code Snippets**→**EventLocation.java**.

From the top level menus in Eclipse, click on **Source**→**Generate Getters and Setters**. Click on **Select All** in the window that appears, and click on **OK**.



Save the file in Eclipse.

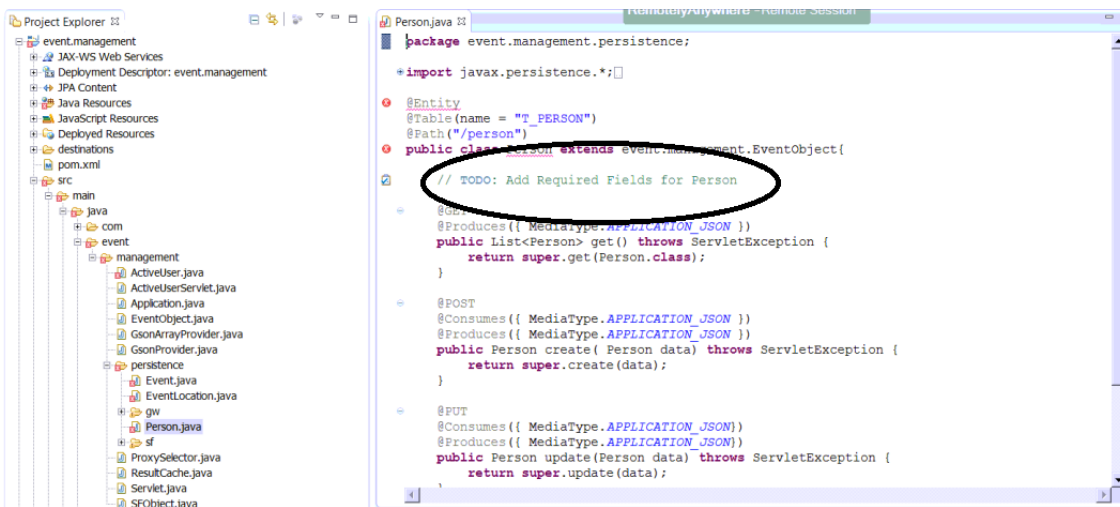
3.3 Complete Person.java

In Eclipse, in the Project Explorer, navigate to *src*→*main*→*java*→*event*→*management*→*persistence* under the *event.management* project.

Double-click on the *Person.java* file.

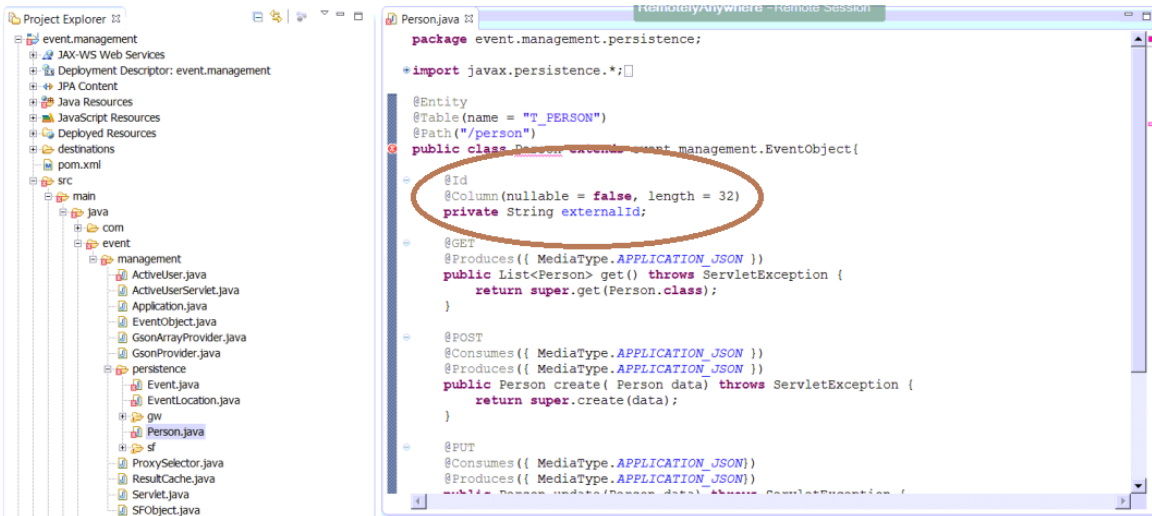
In the file, scroll down to the location with the following comment:

```
// TODO: Add Required Fields for Person
```



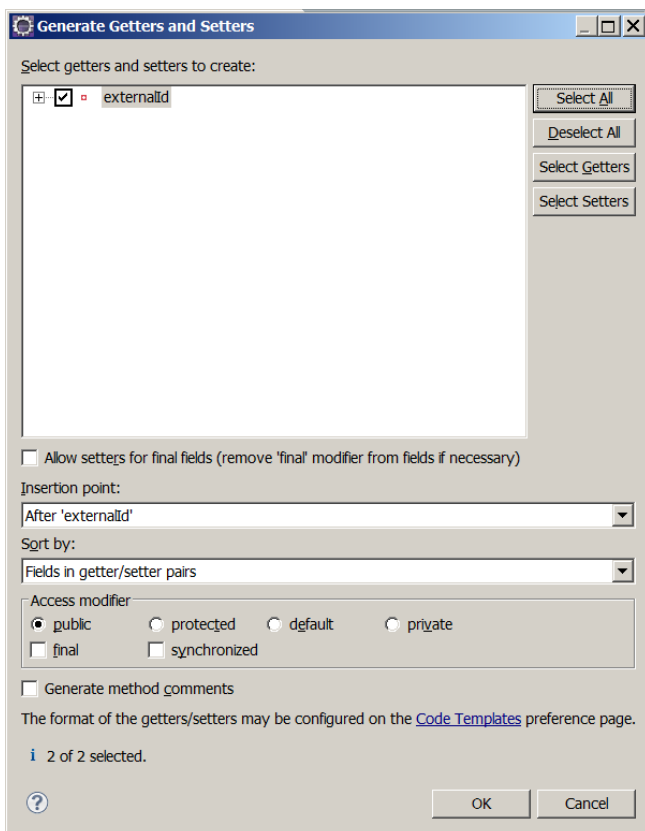
Replace that line with the following code snippet:

```
@Id
@Column(nullable = false, length = 32)
private String externalId;
```



Alternatively, copy and paste the code directly from the file **Network**→**fairfile.fair.sap.corp**→**session**→**CD261**→**Code Snippets**→**Person.java**.

From the top level menus in Eclipse, click on **Source**→**Generate Getters and Setters**. Click on **Select All** in the window that appears, and click on **OK**.



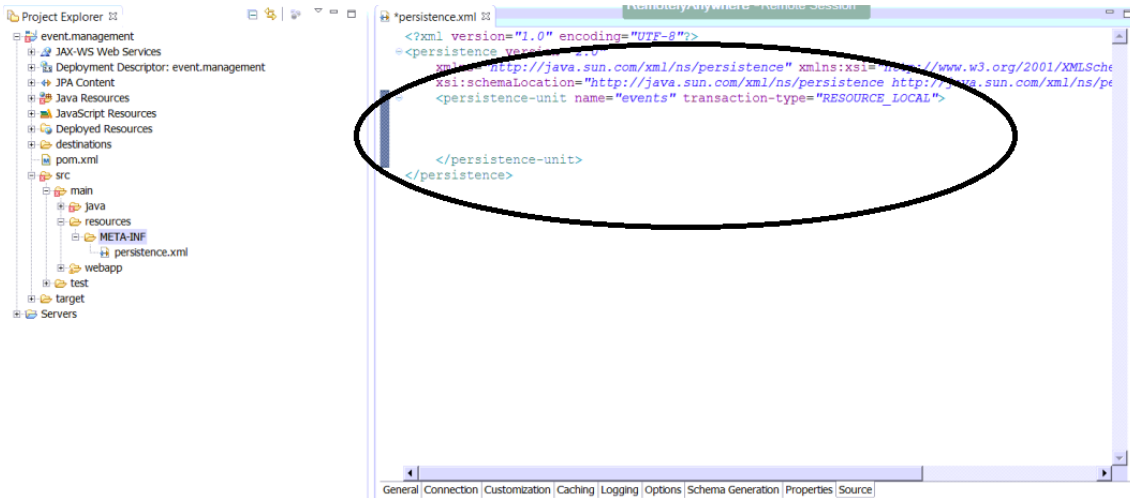
Save the file in Eclipse.

3.4 Complete persistence.xml

In Eclipse, in the Project Explorer, navigate to **src**→**main**→**resources**→**META-INF**→**persistence.xml** under the **event.management** project.

Double-click on the **persistence.xml** file.

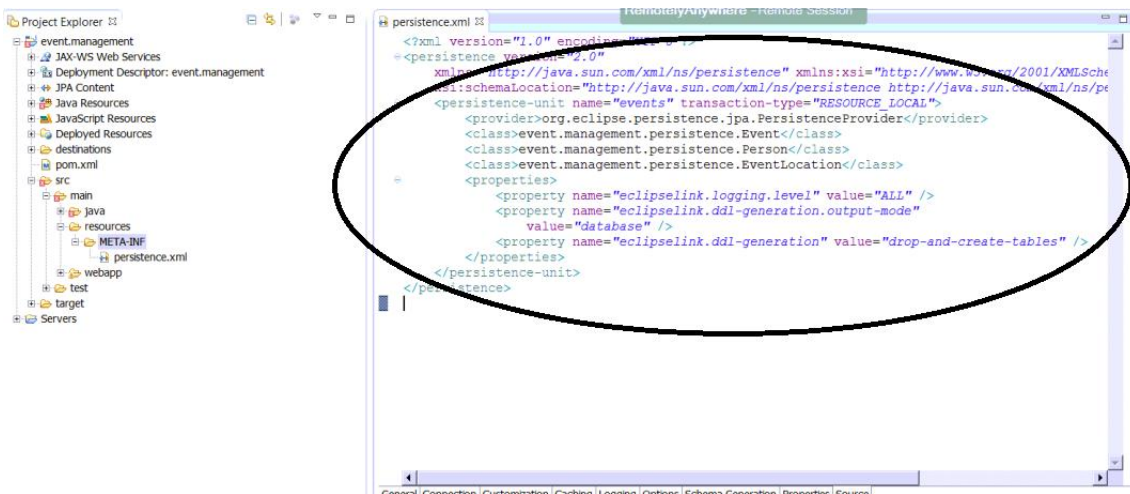
Click on the **Source** tab.



In between the **<persistence-unit>** beginning and end tags, add the following code snippet:

```
<provider>org.eclipse.persistence.jpa.PersistenceProvider</provider>
<class>event.management.persistence.Event</class>
<class>event.management.persistence.Person</class>
<class>event.management.persistence.EventLocation</class>
<properties>
  <property name="eclipselink.logging.level" value="ALL" />
  <property name="eclipselink.ddl-generation.output-mode" value="database" />
  <property name="eclipselink.ddl-generation" value="drop-and-create-tables" />
</properties>
```

Click on **Source**→**Format** to correctly format the file.



Alternatively, copy and paste the code directly from the file **Network**→**fairfile.fair.sap.corp**→**session**→**CD261**→**Code Snippets**→**persistence.xml**. NOTE: remember to open the snippet file in an editor such as Wordpad, not a web browser.

Save the file in Eclipse.

Click on the **General** tab for the *persistence.xml* file.

Add the following entries:

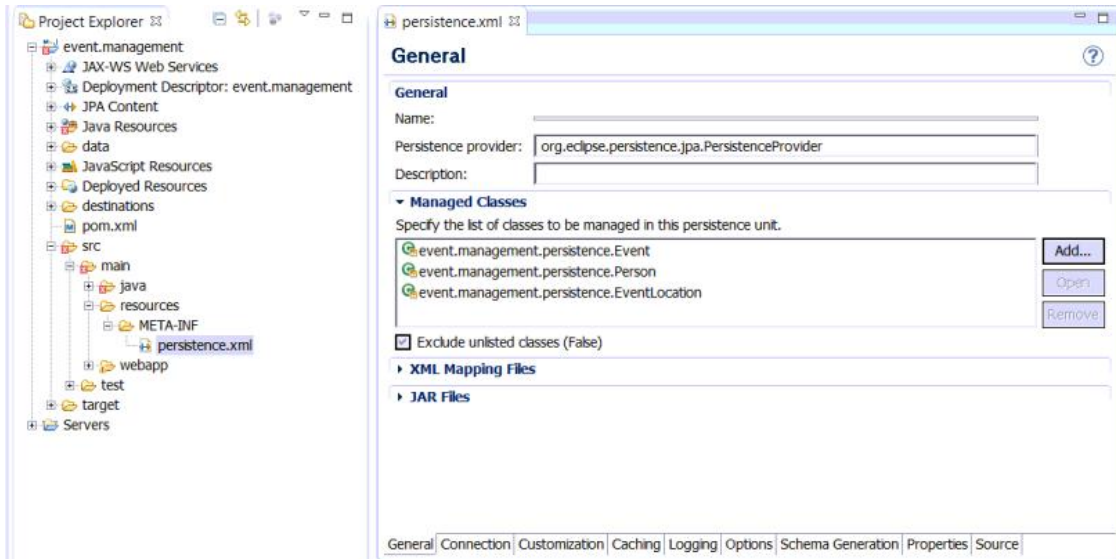
Persistence provider: *org.eclipse.persistence.jpa.PersistenceProvider*

Managed Classes:

event.management.persistence.Event

event.management.persistence.Person

event.management.persistence.EventLocation



Close the xml file.

3.5 Review Imported Code

Open up and review the following completed persistence related files:

src → *main* → *java* → *event* → *management* → *persistence* → *gw* → *Location.java*

src → *main* → *java* → *event* → *management* → *persistence* → *sf* → *User.java*

src → *main* → *java* → *com* → *sap* → *nw* → *gateway* → *oDataObject.java*

src → *main* → *java* → *com* → *sap* → *nw* → *gateway* → *ZTM_EventsOnDemand.java*

Exercise 4 – Develop Business Logic Code

In this exercise, you will complete the code needed for the business logic layer of the application.

4.1 Complete ActiveUser.java

In Eclipse, in the Project Explorer, navigate to `src` → `main` → `java` → `event` → `management` under the `event.management` project.

Double-click on the `ActiveUser.java` file.

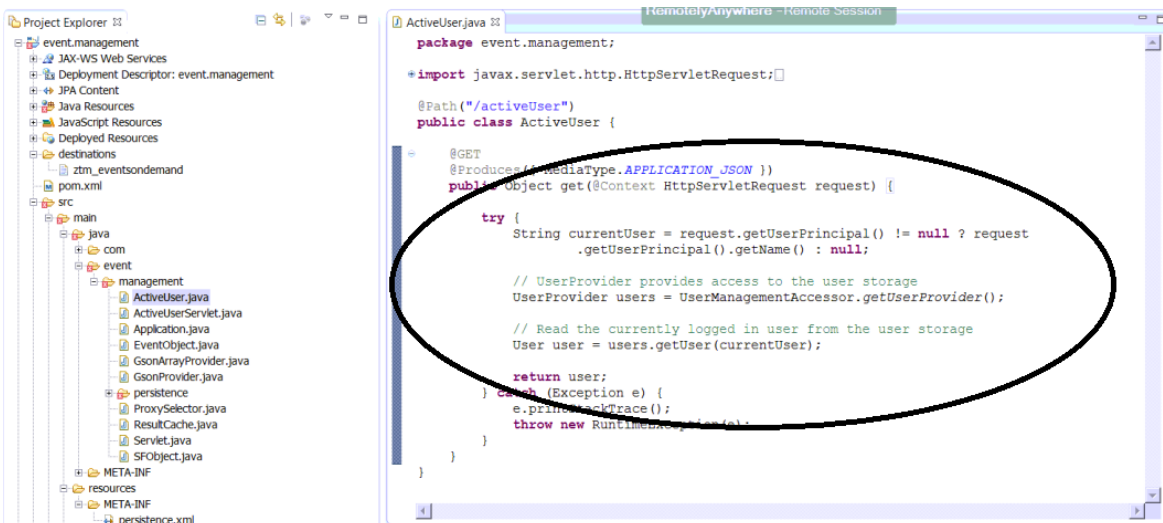
In the file, scroll down to the location with the following comment:

```
// TODO: Get current logged-in user
```



Replace that line with the following code snippet:

```
String currentUser = request.getUserPrincipal() != null ?
request.getUserPrincipal().getName() : null;
// UserProvider provides access to the user storage
UserProvider users = UserManagementAccessor.getUserProvider();
// Read the currently logged in user from the user storage
User user = users.getUser(currentUser);
return user;
```



Alternatively, copy and paste the code directly from the file **Network→fairfile.fair.sap.corp→session→CD261→Code Snippets→ActiveUser.java**.
Save the file in Eclipse.

4.2 Review Imported Code

Open up and review the following completed business logic related files:

src→main→java→event→management

Application.java

EventObject.java

GsonArrayProvider.java

GsonProvider.java

ProxySelector.java

ResultCache.java

Servlet.java

SFObject.java

Exercise 5 – Develop UI

In this exercise, you will complete the code needed for the UI layer of the application.

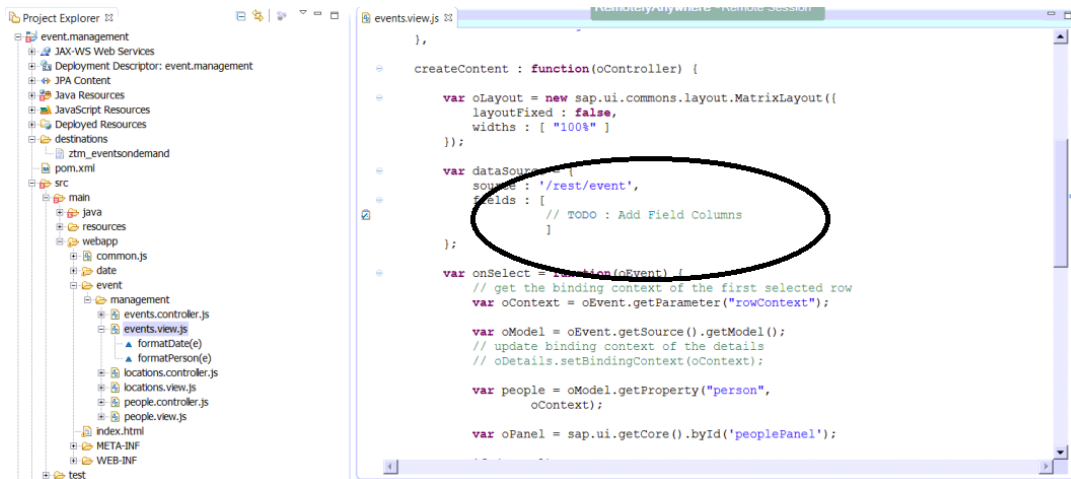
5.1 Complete events.view.js

In Eclipse, in the Project Explorer, navigate to *src*→*main*→*webapp*→*event*→*management* under the *event.management* project.

Double-click on the *events.view.js* file.

In the file, scroll down to the location with the following comment:

```
// TODO: Add Field Columns
```



Replace that line with the following code snippet:

```
{
    label : "Name",
    binding : "name",
    width : "100px"
}, {
    label : "Venue",
    binding : "eventLocation/Name",
    width : "250px",
    dropbox : {
        source : "/rest/gw/Location",
        binding : "Name",
        formatter : function(e) { return e; }
    }
}, {
    label : "Start Date",
    binding : "dateFrom",
    width : "200px",
    date : { format : "yyyy-MM-dd" },
    formatter : formatDate
}, {
    label : "End Date",
    binding : "dateTo",
    width : "200px",
    date : { format : "yyyy-MM-dd" },
    formatter : formatDate
},
{
    label : "Comment",
    binding : "comment",
    width : "250px"
}, {
    label : "People",
    binding : "person",
```

```

    visible : false,
    listbox : {
        source : "/rest/sf/ec/user",
        binding : "externalId",
        formatter : formatPerson
    }
}

```

Click on **Source** → **Format** to correctly format the file.

Alternatively, copy and paste the code directly from the file **Network** → **fairfile.fair.sap.corp** → **session** → **CD261** → **Code Snippets** → **events.view.js**. NOTE: remember to open the snippet file in an editor such as Wordpad, not a web browser.

Save the file in Eclipse.

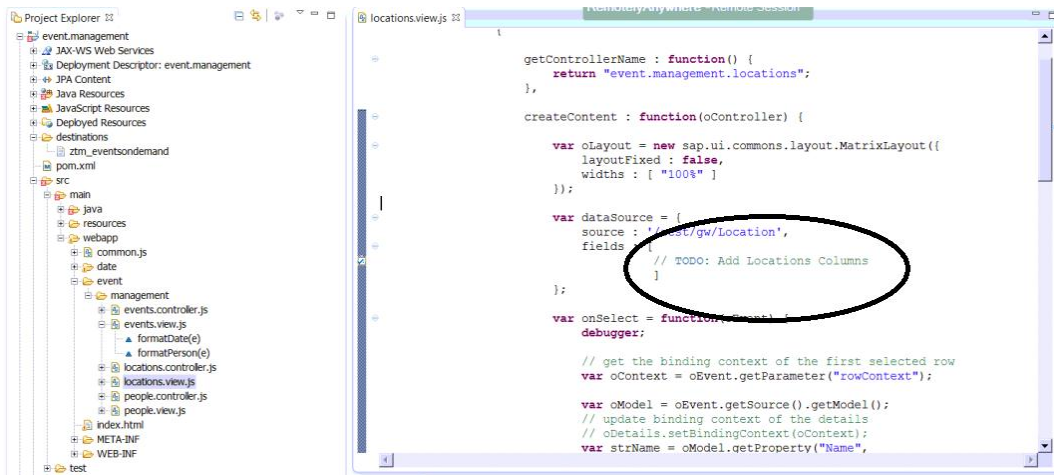
5.2 Complete locations.view.js

In Eclipse, in the Project Explorer, navigate to **src** → **main** → **webapp** → **event** → **management** under the **event.management** project.

Double-click on the **locations.view.js** file.

In the file, scroll down to the location with the following comment:

```
// TODO: Add Locations Columns
```



Replace that line with the following code snippet:

```

{
    label : "Name",
    binding : "Name",
    primary : true,
    width : "220px"
}, {
    label : "Street",
    binding : "Street",
    width : "380px"
}, {
    label : "City",
    binding : "City",
    width : "400px"
}, {
    label : "State",
    binding : "State",
    width : "50px"
}, {
    label : "Zip Code",
    binding : "Zipcode",
    width : "75px"
}

```

Click on **Source**→**Format** to correctly format the file.

Alternatively, copy and paste the code directly from the file **Network**→**fairfile.fair.sap.corp**→**session**→**CD261**→**Code Snippets**→**locations.view.js**. NOTE: remember to open the snippet file in an editor such as Wordpad, not a web browser.

Save the file in Eclipse.

5.3 Complete people.view.js

In Eclipse, in the Project Explorer, navigate to **src**→**main**→**webapp**→**event**→**management** under the **event.management** project.

Double-click on the **people.view.js** file.

In the file, scroll down to the location with the following comment:

```
// TODO: Add People Columns
```



Replace that line with the following code snippet:

```
{
    label : "First Name",
    binding : "firstName",
    width : "200px"
}, {
    label : "Last Name",
    binding : "lastName",
    width : "200px"
}, {
    label : "Title",
    binding : "title",
    width : "300px"
}, {
    label : "Department",
    binding : "department",
    width : "300px"
}
}
```

Click on **Source**→**Format** to correctly format the file.

Alternatively, copy and paste the code directly from the file **Network**→**fairfile.fair.sap.corp**→**session**→**CD261**→**Code Snippets**→**people.view.js**. NOTE: remember to open the snippet file in an editor such as Wordpad, not a web browser.

Save the file in Eclipse.

5.4 Review Imported Code

Open up and review the following completed UI related file:

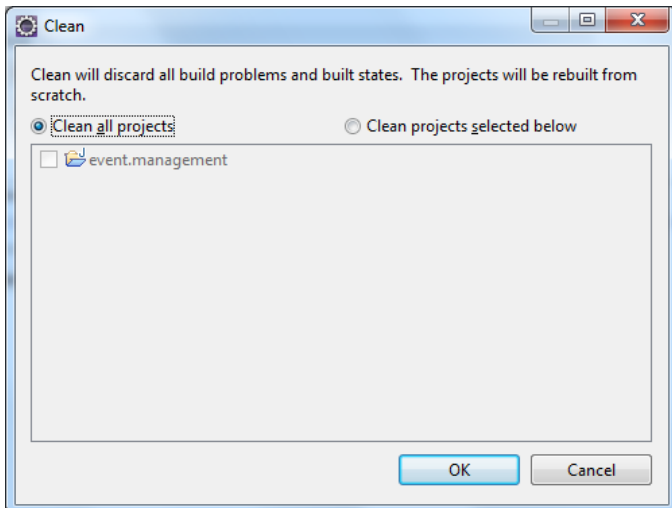
src→**main**→**webapp**→**event**→**index.html**

EXERCISE 6 – RUN APPLICATION LOCALLY

In this exercise, you will deploy the completed application to the local NW Cloud server on your laptop that you created in Exercise 1.

6.1 Build Project

In Eclipse, click on the **Project** menu from the top level menu bar, and then click on **Clean**. Make sure the **Clean all projects** option is selected in the window that pops up, and then click **OK**.



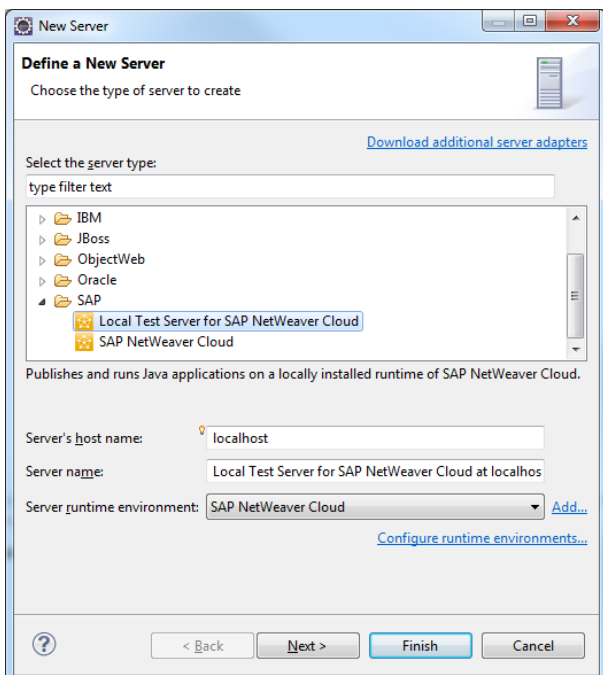
6.2 Add Local NW Cloud Server

6.2.1 Create Server

In Eclipse, go to the lower right section of the window and click on the **Servers** tab.

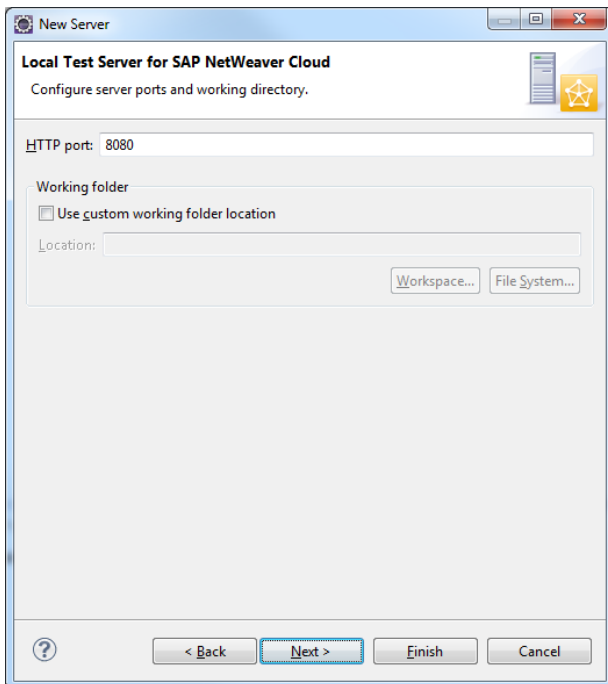
Right-click anywhere inside the tab area, and click on **New** in the window that pops up, then on **Server**.

In the wizard that pops up, select **Local Test Server for SAP NetWeaver Cloud** as the server type, and click on **Next**.

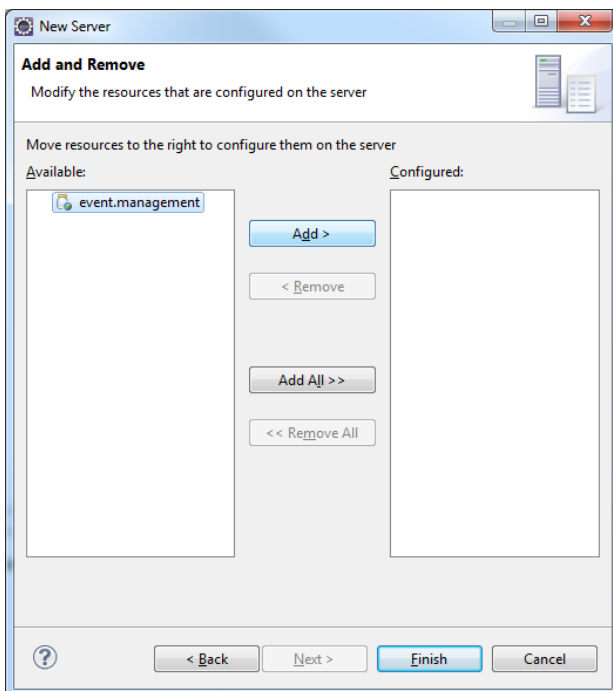


Leave the defaults on the next page of the wizard, and click on **Next**.

Make a note of the HTTP port number on the following page of the wizard, especially if it isn't the default 8080. You will need it in section 6.4. Click on **Next** again.



On the next page, select your project on the *Available* list and click on **Add**. Then hit **Finish**.



6.2.2 Add On-Premise Connection and Users

Double-click on the newly created server entry titled **Local Test Server for SAP NetWeaver Cloud at localhost** in the **Servers** tab.

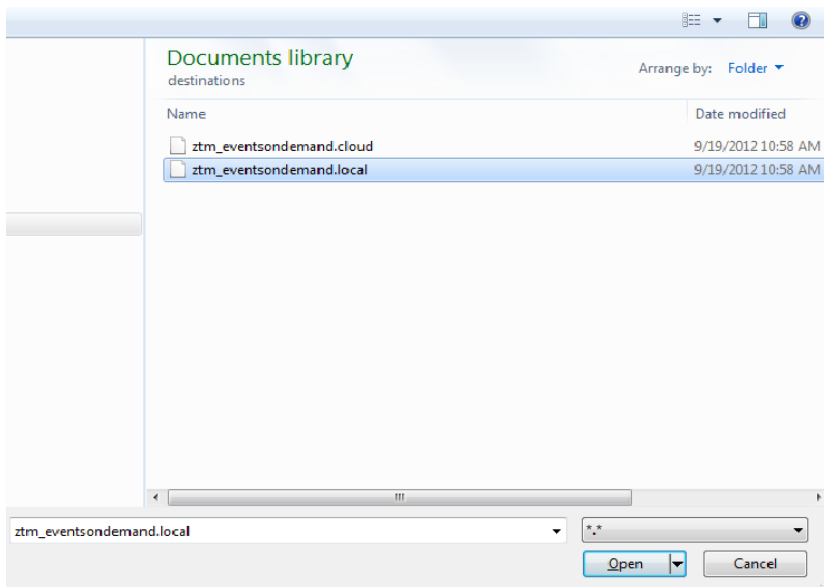
Click on the **Connectivity** tab under the list of tabs that appears in the upper section of the Eclipse window.



Click on the **Import** icon.

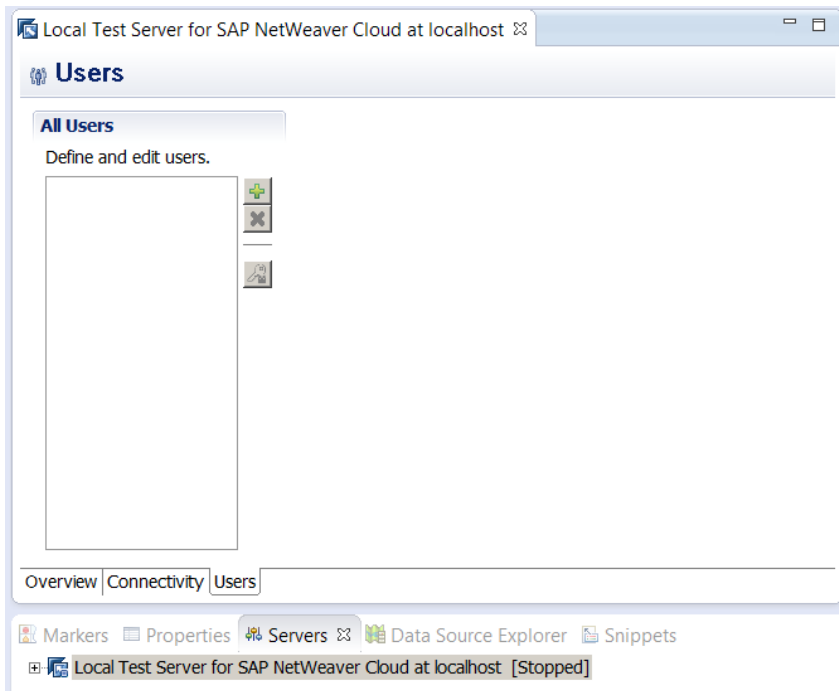
In the Windows File Explorer that appears, navigate to **Network**→**fairfile.fair.sap.corp**→**session**→**CD261**→**Code to Import**→**destinations**.

Double-click on the **ztm_eventsondemand.local** file.



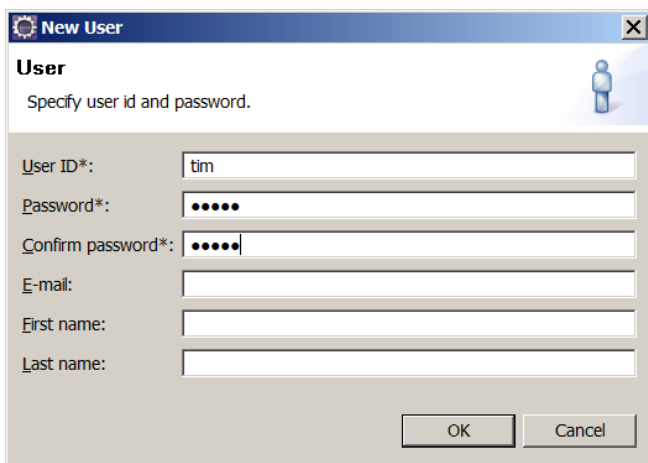
Save the entry.

Click on the **Users** tab under the list of tabs that appears in the upper section of the Eclipse window.



Click on the **Add** icon.

In the *New User* window that appears, enter a **User ID** and **Password** of your choice, and then do a **Confirm password**. The other fields can remain blank. Click on **OK**. Make a note of the user id and password you entered – you will need them in section 6.4.



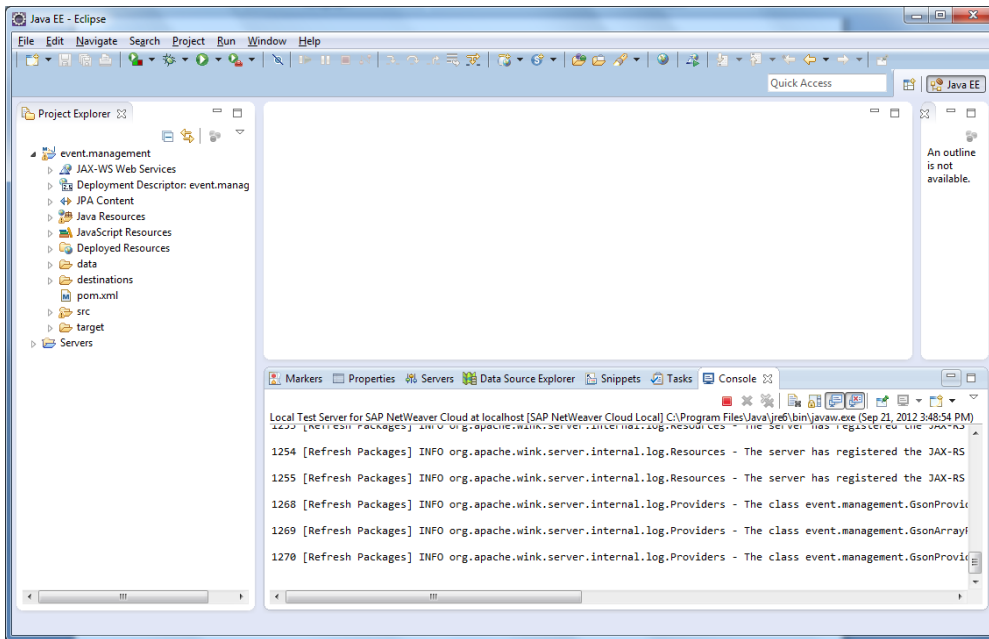
Save the entry.

6.3 Start Server

In Eclipse, make sure you are in the **Servers** tab in the lower right section of the window. Also make sure the **Local Test Server for SAP NetWeaver Cloud at localhost** is listed in the tab.

Right-click on the entry and click on **Start**.

Make sure the server has started without any errors. You can observe any messages and overall progress in the **Console** tab.

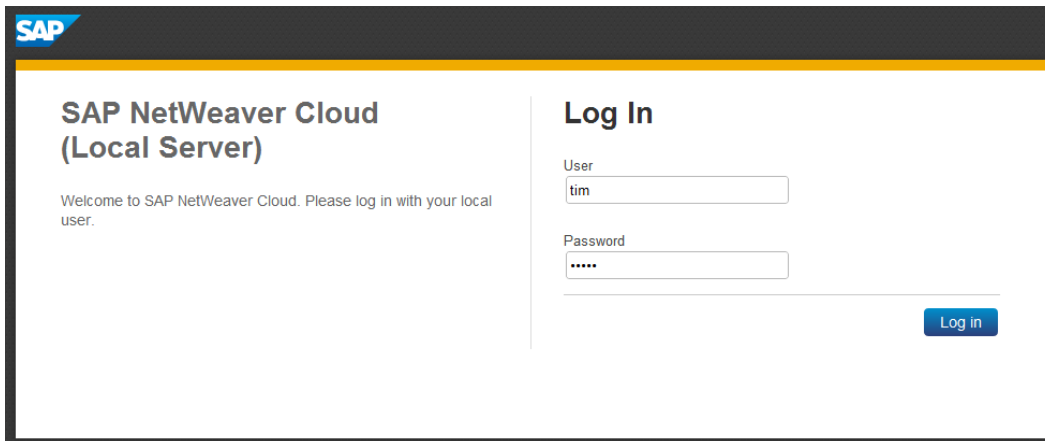


6.4 Test Application

Open up a Google Chrome browser window.

Enter the URL: <http://localhost:<port number>>, where **<port number>** is the port number you noted in Exercise 1, Section 1.3.1 when you created the server. NOTE: the default port number is 8080 – eg. enter <http://localhost:8080>.

The login page for the application should come up. Enter the username and password you defined in Section 6.2.2, and click on the **Log in** button.



You should see the **Events** page appear. It should be empty with no data.

SAP Events On-Demand Welcome

Events Event Venues My Team

Create Modify Delete Reload

Name	Venue	Start Date	End Date	Comment

Click on the **Event Venues** tab. You should see some location data that was pulled directly from the on-premise ERP system.

SAP Events On-Demand Welcome

Events **Event Venues** My Team

Create Modify Delete Reload

Name	Street	City
Orange County Convention Center	9800 International Drive	Orlando
San Jose Convention Center	50 West San Carlos Street	San Jose

Click on the **My Team** tab. You should see some people data that was pulled directly from a SuccessFactors Employee Central application in the cloud.

SAP Events On-Demand Welcome

Events Event Venues **My Team**

Reload

First Name	Last Name	Title	Department
Carla	Grant	VP, Sales	Sales (SALES)
Jennifer	Lo	Administrative Assistant	Sales (SALES)
Marcus	Hoff	Sales Director, NE	Marketing (MKTG)
Richard	Maxx	Sales Director, NC	Product Management (PM)
Sid	Mormony	Sales Director, SC	Sales (SALES)
Vic	Stokes	Sales Director, SW	Sales (SALES)
Wilma	Sown	Sales Director, NW Sales	Sales (SALES)

Now go back to the **Events** tab, and try adding one or two event records by clicking on the **Create** button. Make sure the records save successfully.

New Entry

Name: TechEd

Venue: San Jose Convention Center

Start Date: 22/10/2012

End Date: 27/09/2012

Comment:

People: Grant, Carla; Hoff, Marcus; Lo, Jennifer; Max, Richard; Mormony, Sid

Cancel Save

Events

Name	Venue	Start Date	End Date	Comment
TechEd	San Jose Convention Center	Monday, October 22, 2012	Thursday, September 27, 2012	

Congratulations! You have successfully built and deployed a working application on your local instance of NetWeaver Cloud.

Exercise 7 – Run Application in the Cloud

In this exercise, you will deploy the same application to the cloud itself.

7.1 Update ProxySelector.java

In Eclipse, in the Project Explorer, navigate to *src*→*main*→*java*→*event*→*management* under the *event.management* project.

Double-click on the *ProxySelector.java* file.

In the file, scroll down to the location with the following code:

```
l.add(Proxy.NO_PROXY);
```

Comment out this line, and un-comment the line immediately below:

```
l.add(new Proxy(Proxy.Type.HTTP, new InetSocketAddress("proxy", 8080)));
```



Save the file.

Clean the project by clicking on *Project*→*Clean*.

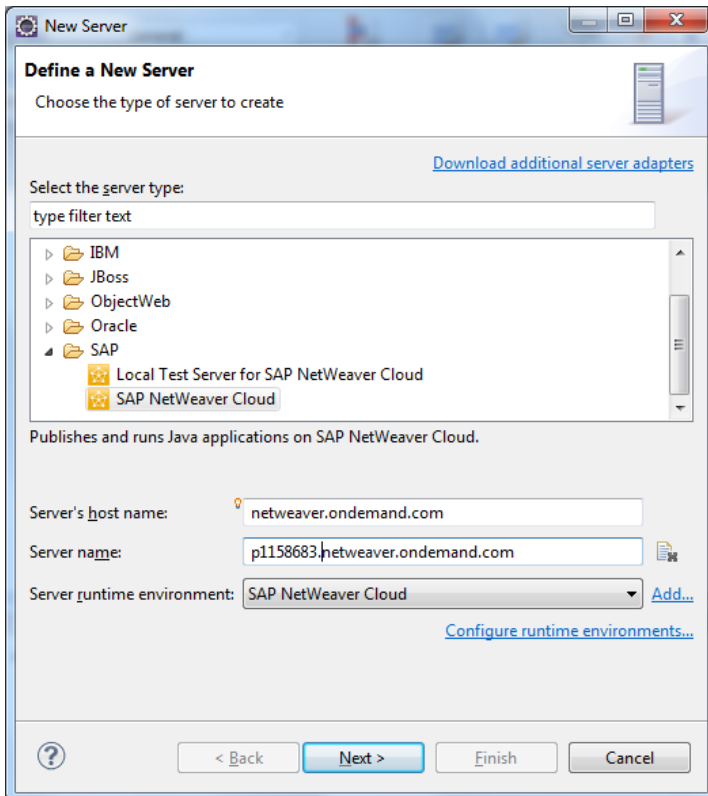
7.2 Deploy Application to Server

In Eclipse, make sure you are in the **Servers** tab in the lower right section of the window. Right-click in that space, and then select **New**→**Server**.

Select **SAP NetWeaver Cloud** as the server type in the popup wizard that appears (titled **Define a New Server**).

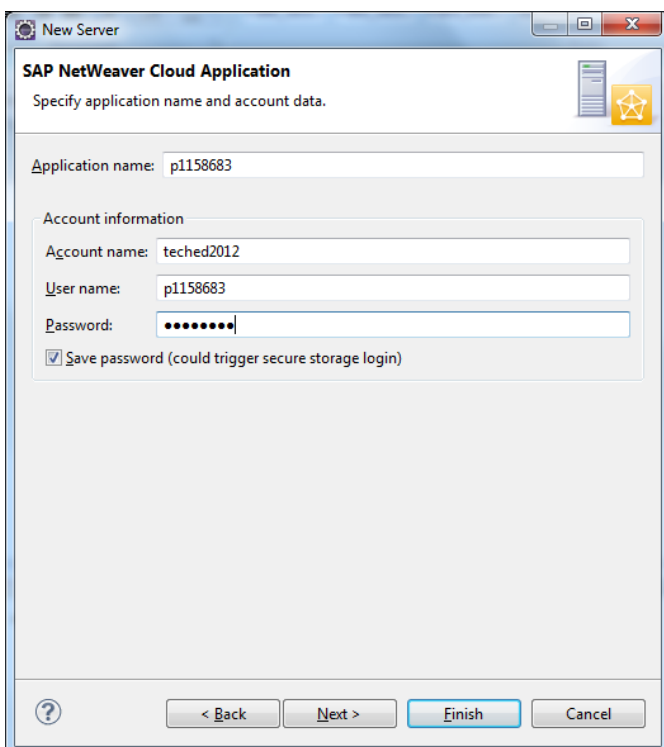
Update the **Server name** field to **<your-assigned-user>.netweaver.ondemand.com**, where **<your-assigned-user>** is the individual user id given to you by the workshop instructor. NOTE: do not include the '<' and '>' characters in your entry.

Leave everything else as-is, and then click on **Next**.

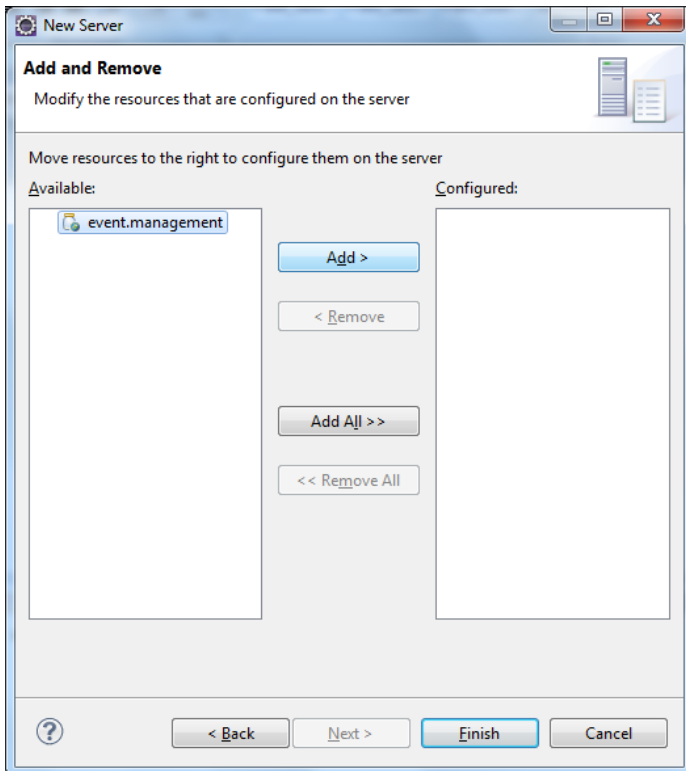


In the next page of the wizard (titled **SAP NetWeaver Cloud Application**), enter the following fields:

- *Application name:* **<your-assigned-user>** (NOTE: do not include the '<' and '>' characters in your entry)
- *Account name:* **teched2012**
- *User name:* **<your-assigned-user>** (NOTE: do not include the '<' and '>' characters in your entry)
- *Password:* **Abcd1234**
- Click on **Next**.



In the next page of the wizard (titled **Add and Remove**), select the **event.management** entry under the *Available* box on the left, and click on the **Add** button. Click on **Finish**.

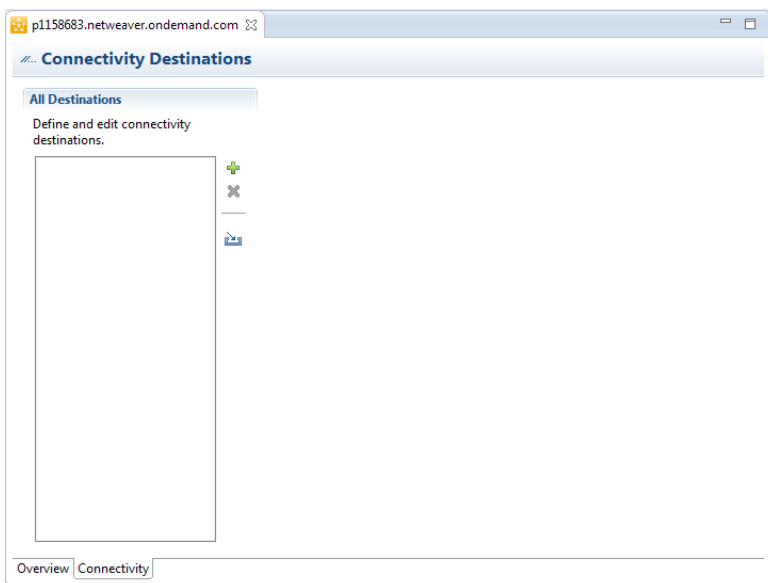


The server should get created.

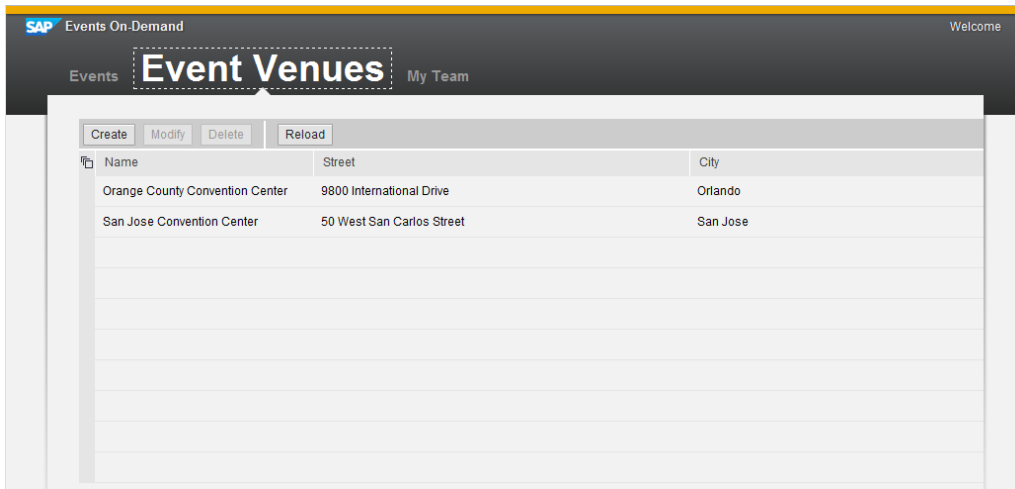
7.3 Add On-Premise Connection

Double-click on the newly created server entry titled **<your-assigned-user>.netweaver.ondemand.com** in the **Servers** tab.

Click on the **Connectivity** tab under the list of tabs that appears in the upper section of the Eclipse window.



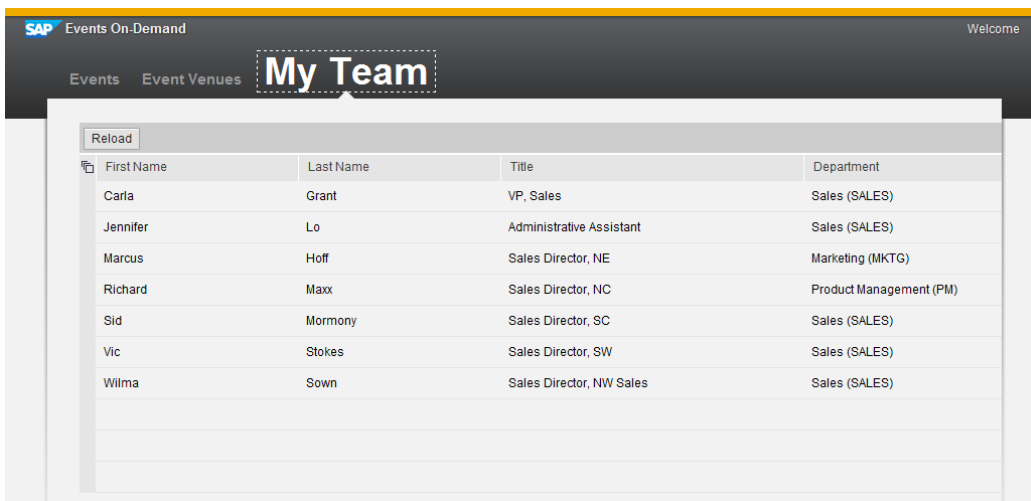
Click on the **Event Venues** tab. You should see some location data that was pulled directly from the on-premise ERP system.



SAP Events On-Demand interface showing the **Event Venues** tab. The table displays two venue records:

Name	Street	City
Orange County Convention Center	9800 International Drive	Orlando
San Jose Convention Center	50 West San Carlos Street	San Jose

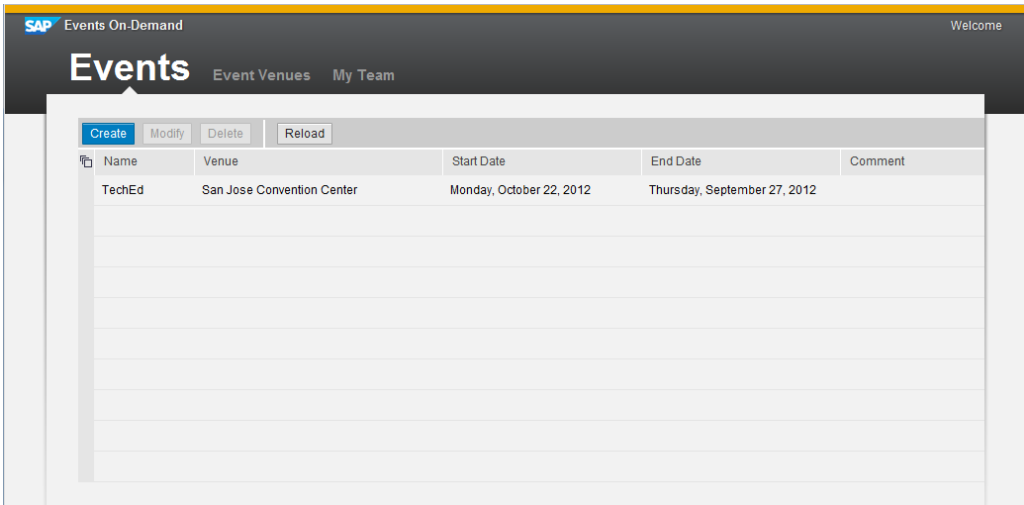
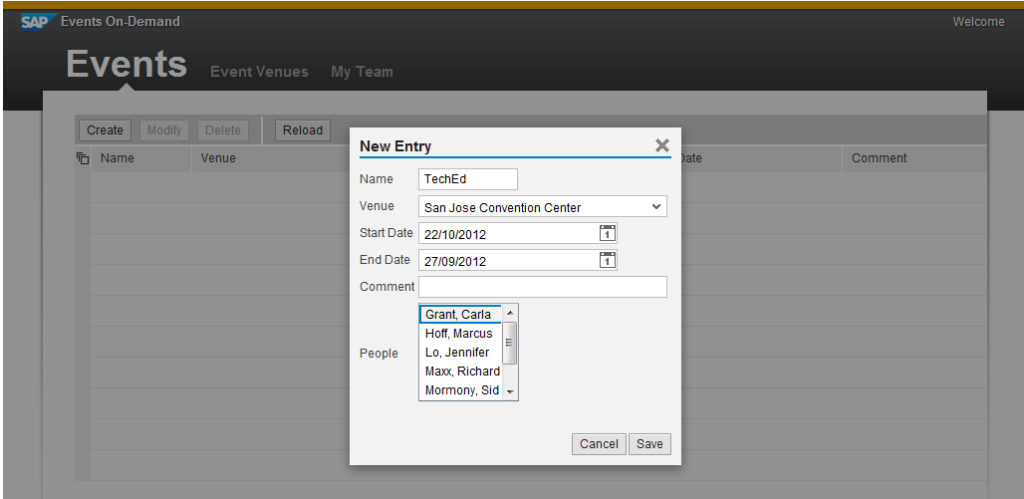
Click on the **My Team** tab. You should see some people data that was pulled directly from a SuccessFactors Employee Central application in the cloud.



SAP Events On-Demand interface showing the **My Team** tab. The table displays seven employee records:

First Name	Last Name	Title	Department
Carla	Grant	VP, Sales	Sales (SALES)
Jennifer	Lo	Administrative Assistant	Sales (SALES)
Marcus	Hoff	Sales Director, NE	Marketing (MKTG)
Richard	Maxx	Sales Director, NC	Product Management (PM)
Sid	Mormony	Sales Director, SC	Sales (SALES)
Vic	Stokes	Sales Director, SW	Sales (SALES)
Wilma	Sown	Sales Director, NW Sales	Sales (SALES)

Now go back to the **Events** tab, and try adding one or two event records by clicking on the **Create** button. Make sure the records save successfully.



Congratulations! You have successfully deployed your application to SAP NetWeaver Cloud.

Exercise 8 – Monitor Application

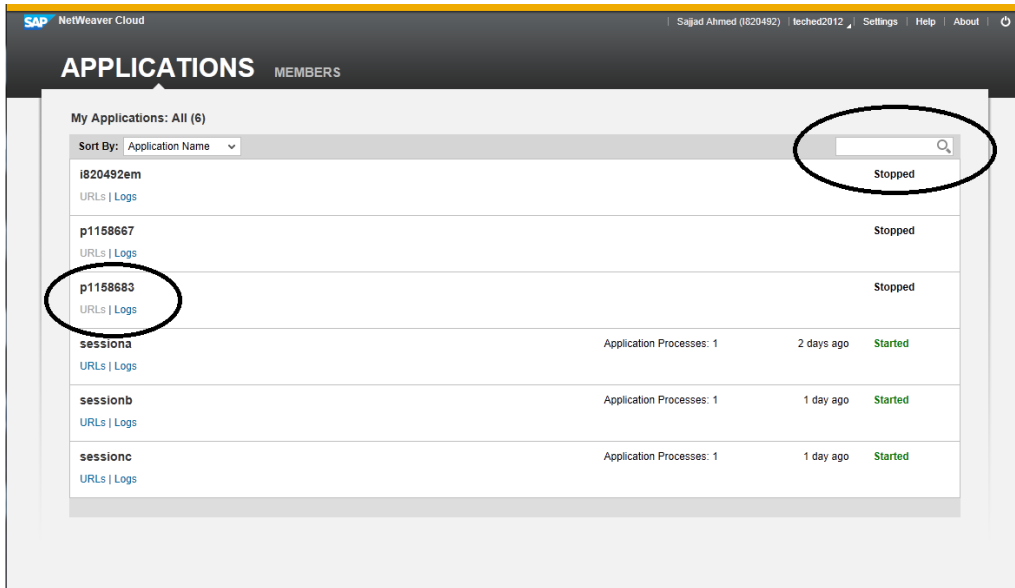
In this exercise, you will check to make sure your application is actually running without any issues in the cloud.

8.1 Locate Your Application

In a web browser, enter the following URL:

<https://account.netweaver.ondemand.com/>

Once the NetWeaver Cloud accounts page appears, type your allocated user id in the search field and hit the magnifying glass icon.



Once you have found your server, verify that it is in a **started** state. If you click on the Logs link, you will be taken to a separate page where you can view the detailed log files.

This concludes the workshop. It was designed to give you hands-on experience with the NetWeaver Cloud platform, so that you can feel comfortable using the tools and services to develop and deploy your own applications in the cloud. Hopefully you have found it useful.

© 2012 by SAP AG. All rights reserved.

SAP and the SAP logo are registered trademarks of SAP AG in Germany and other countries. Business Objects and the Business Objects logo are trademarks or registered trademarks of Business Objects Software Ltd. Business Objects is an SAP company. Sybase and the Sybase logo are registered trademarks of Sybase Inc. Sybase is an SAP company. Crossgate is a registered trademark of Crossgate AG in Germany and other countries. Crossgate is an SAP company.



The Best-Run Businesses Run SAP™