

DB2 Technologies: Software installation guide

Contents

1 Software requirements.....	3
2 Installation	4
2.1 Installing the latest JRE and JDK	4
2.1.1 Windows.....	4
2.1.2 OS X.....	6
2.1.3 Linux.....	6
2.1.4 Check if everything went fine.....	7
2.2 Installing Eclipse	7
2.3 Installing TomEE.....	8
Windows:	8
OS X	9
Linux	10
2.4 Installing MySQL Server & Workbench	10
OS X	11
Linux	12
2.4.1 Starting MySQL Server	13
3 Configuration	14
3.1 Setting up the latest JDK with Eclipse	14
3.2 Adding TomEE to Eclipse	14
3.3 Configuring TomEE.....	17
3.4 Starting TomEE.....	19
4 Publishing content.....	22
4.1 Creating a dynamic web project	22
4.2 Writing the Web component.....	23
4.3 Testing your application	26
5 Connecting with JDBC.....	28
5.1 Add mysql connector to the project	28
5.2 Installing Connector/J in Tomee.....	29
5.3 Create a new servlet	32

5.4 Create a new database.....	33
5.5 Run the project.....	34
6 Stopping Tomee.....	34
7 Troubleshooting	34
7.1 Windows MySQL TimeZone Error	34
8 Running the Book examples:.....	38
Download the Book examples	38
Running Chapter 2 – employeeService example.....	38
Running Chapter 3 examples (1-15):	41
EJB project.	41
Web Project	45
Running Chapter 3 example 16 and onwards.....	47
9 Importing the “Expense Management” Example.....	55
Create the data base.....	55
Import the projects to eclipse.....	56
References.....	58

1 Software requirements

Download the following software packages:

Windows & OS X & Linux:

- Eclipse IDE for Java Enterprise developers: <https://www.eclipse.org/downloads/packages/>
- Apache TomEE 8.0.3 (plume): <https://tomee.apache.org/download-archive.html>
- MySQL connector: <https://dev.mysql.com/downloads/connector/j/> “Platform Independent (Architecture Independent), ZIP or TAR Archive”.

Windows & OS X only:

- Oracle JDK and JRE, last version:
<https://www.oracle.com/technetwork/java/javase/downloads/index.html>
- MySQL Community server 8.0:
<http://dev.mysql.com/downloads/mysql/>
Note: MySQL Installer is 32 bit but will install both 32-bit and 64-bit binaries.
- MySQL Workbench:
<http://dev.mysql.com/downloads/workbench/>

2 Installation

2.1 Installing the latest JRE and JDK

The Java installation guide for all platforms can be found here:
<https://docs.oracle.com/en/java/javase/index.html>

2.1.1 Windows

Go to the Java SE download page, download the installer and run it as in Figure 1.

<https://www.oracle.com/java/technologies/javase-jdk13-downloads.html>



Figure 1: JDK installation wizard

Set the windows environment variables, as explained e.g., here:

<https://stackoverflow.com/questions/1672281/environment-variables-for-java-installation>

When using Java SE 13 you don't need to set the JRE_HOME variable.

An example of environment variables is:

JAVA_HOME: C:\Program Files\Java\jdk-13.0.2

JDK_HOME: %JAVA_HOME%

CLASSPATH: .;%JAVA_HOME%\lib

Modify the PATH variable by adding the %JAVA_HOME%\bin to the already existing ones in your PC

PATH: **your-unique-entries;%JAVA_HOME%\bin**

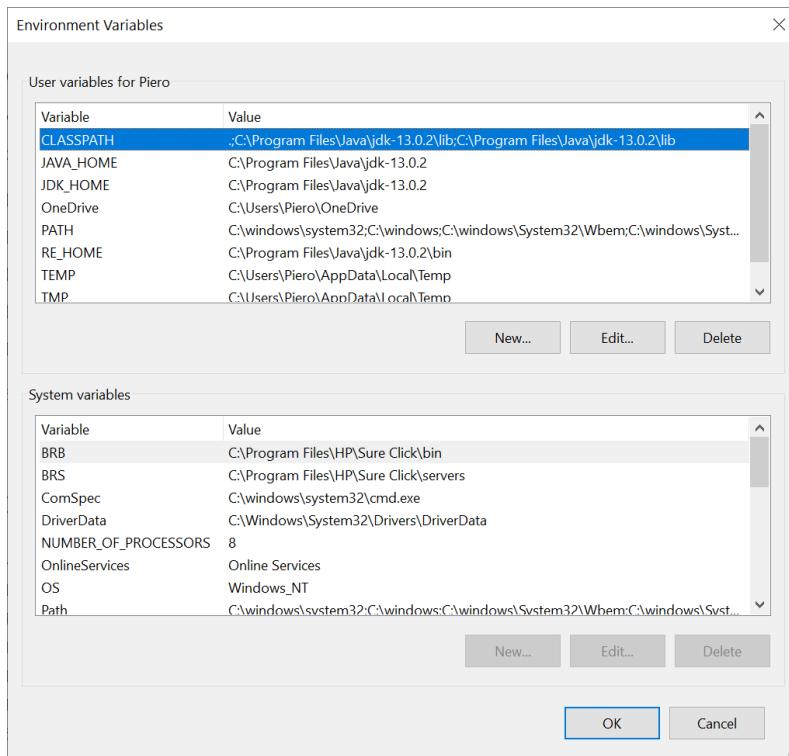


Figure 2: setting Windows environment variables

2.1.2 OS X

Download package and install as in Figure 3.

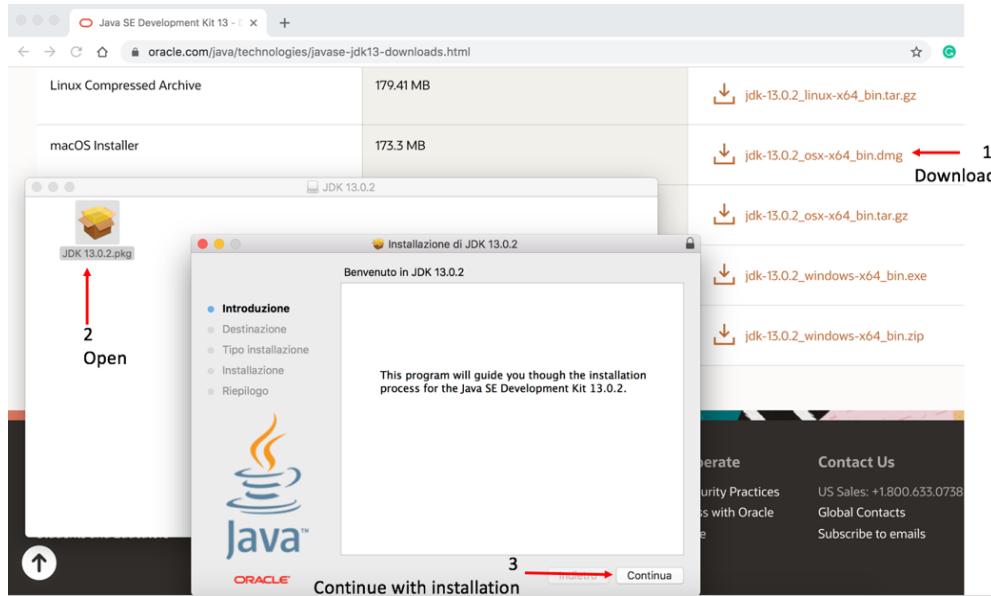


Figure 3: Installing the Oracle JDK in OS X

2.1.3 Linux

In the remainder of this guide, we will indicate with Linux any Debian-based distribution. Please notice that the provided instructions have been tested on the latest Ubuntu Linux LTS release.

You need to add the `linuxuprising/java` third-party repository. Open a terminal and execute the following commands:

```
sudo add-apt-repository ppa:linuxuprising/java  
sudo apt update
```

Then, to Install the Oracle JDK, execute:

```
sudo apt install oracle-java13-installer
```

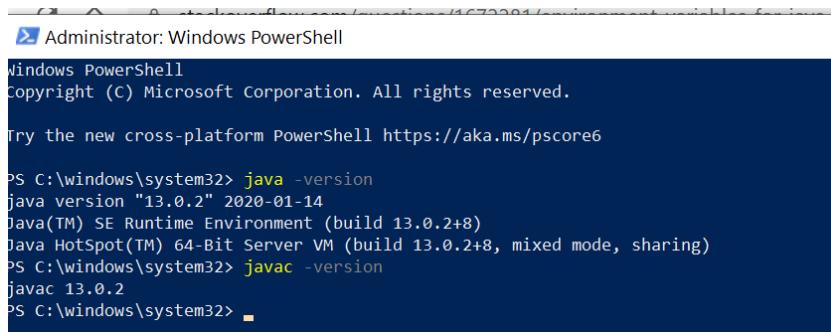
You will be prompt a message to confirm the Oracle license terms & agreement, accept them (use the tab button to navigate the options), and continue.

2.1.4 Check if everything went fine

At the end of the Java installation process, open a terminal (Linux, OS X) or a command prompt (Windows) and type:

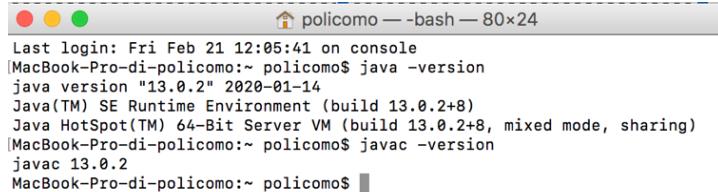
```
java -version  
javac -version
```

If everything went fine, the reported JRE and JDK versions should match the ones you just installed (or had already installed).



```
Administrator: Windows PowerShell  
Windows PowerShell  
copyright (C) Microsoft corporation. All rights reserved.  
Try the new cross-platform PowerShell https://aka.ms/pscore6  
PS C:\windows\system32> java -version  
java version "13.0.2" 2020-01-14  
Java(TM) SE Runtime Environment (build 13.0.2+8)  
Java HotSpot(TM) 64-Bit Server VM (build 13.0.2+8, mixed mode, sharing)  
PS C:\windows\system32> javac -version  
javac 13.0.2  
PS C:\windows\system32>
```

Figure 4: checking the version of Java in Windows



```
policomo — bash — 80x24  
Last login: Fri Feb 21 12:05:41 on console  
[MacBook-Pro-di-policomo:~ policomo$ java -version  
java version "13.0.2" 2020-01-14  
Java(TM) SE Runtime Environment (build 13.0.2+8)  
Java HotSpot(TM) 64-Bit Server VM (build 13.0.2+8, mixed mode, sharing)  
[MacBook-Pro-di-policomo:~ policomo$ javac -version  
javac 13.0.2  
MacBook-Pro-di-policomo:~ policomo$
```

Figure 5: checking the version of Java in Mac OS

2.2 Installing Eclipse

For all platforms:

- Download the .zip or .tar.gz file. At the end of the download process, extract the content of file you downloaded. You will end up with an eclipse folder that can be placed wherever you like on your computer. To start Eclipse, just double-click on the Eclipse executable.

For Windows:

- Download the installation wizard and launch it (requires internet connection).

For Mac:

- Open the .dmg and in the appearing window move Eclipse to the Applications folder

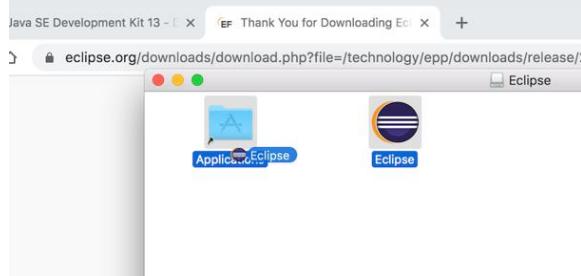


Figure 6: Move Eclipse to Apps folder

For Linux:

- Extract the content of the .tar.gz using the command:
tar -zxf yourEclipsefile.tar.gz
- Navigate to the extracted eclipse folder, execute the eclipse application.

2.3 Installing TomEE

Windows:

- After unzipping, copy the TomEE folder into your favorite drive (e.g., C: or D:)
- Launch the TomEE Java Web server by executing the “TomEE.exe” file in the extracted TomEE folder, e.g: C:\Program Files\Apache Software Foundation\apache-tomee-plume-8.0.3\bin
- Alternatively, you can execute the file “startup.bat” on the same folder using the command line

Commented [RNT1]: Tomcat o Tomee?

```
[C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps\sample.war]
06-Feb-2020 09:31:30.468 WARNING [main] org.apache.catalina.util.SessionGeneratorBase.createSecureRandom Creation of SecureRandom instance using SHA1PRNG with seed length 207 milliseconds.
06-Feb-2020 09:31:30.527 INFO [main] org.apache.catalina.startup.HostConfig.deployWAR Deployment of web application archive [C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps\sample.war] has finished in [586] ms
06-Feb-2020 09:31:30.528 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deploying web application directory [C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps\docs]
06-Feb-2020 09:31:30.577 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application in directory [C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps\docs] has finished in [45] ms
06-Feb-2020 09:31:30.577 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deploying web application directory [C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps\examples]
06-Feb-2020 09:31:30.995 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application in directory [C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps\examples] has finished in [418] ms
06-Feb-2020 09:31:30.998 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deploying web application directory [C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps\host-manager]
06-Feb-2020 09:31:31.059 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application in directory [C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps\host-manager] has finished in [51] ms
06-Feb-2020 09:31:31.059 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deploying web application directory [C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps\manager]
06-Feb-2020 09:31:31.108 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application in directory [C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps\manager] has finished in [57] ms
06-Feb-2020 09:31:31.188 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deploying web application directory [C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps\ROOT]
06-Feb-2020 09:31:31.145 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application in directory [C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps\ROOT] has finished in [37] ms
06-Feb-2020 09:31:31.156 INFO [main] org.apache.coyote.AbstractProtocol.start Starting ProtocolHandler ["http-nio-8080"]
06-Feb-2020 09:31:31.178 INFO [main] org.apache.coyote.AbstractProtocol.start Starting ProtocolHandler ["ajp-nio-8009"]
06-Feb-2020 09:31:31.187 INFO [main] org.apache.catalina.startup.Catalina.start Server startup in [1,333] milliseconds
```

Figure 7 the console of the Tomee Java web server

- Open a browser and type the address: <http://localhost:8080>

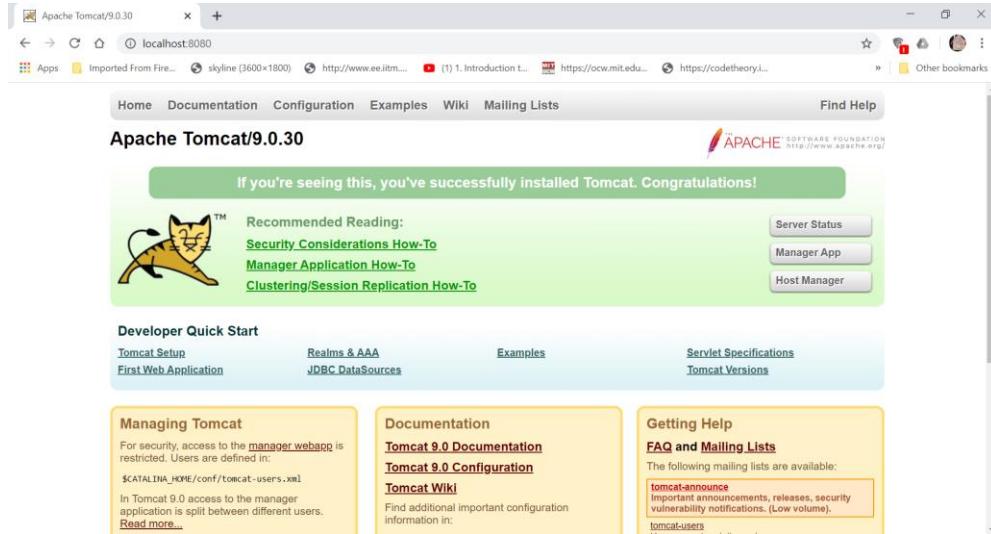


Figure 8: the welcome page of the Tomee Java Web server

OS X

- After unzipping, rename the folder to Tomee and copy the folder into the /Applications folder
- Move to Applications/Tomee/bin and run the command ./startup.sh

```
/Applications/Tomee/bin
MacBook-Pro-de-Nahime:bin nahimetrores$ clear
MacBook-Pro-de-Nahime:bin nahimetrores$ pwd
/Applications/Tomee/bin
MacBook-Pro-de-Nahime:bin nahimetrores$ ./startup.sh
Using CATALINA_BASE:   /Applications/Tomee
Using CATALINA_HOME:   /Applications/Tomee
Using CATALINA_TMPDIR: /Applications/Tomee/temp
Using JRE_HOME:        /Library/Java/JavaVirtualMachines/jdk-13.0.2.jdk/Contents/Home
Using CLASSPATH:       /Applications/Tomee/bin/bootstrap.jar:/Applications/Tomee/bin/tomcat-juli.jar
Tomcat started.
MacBook-Pro-de-Nahime:bin nahimetrores$
```

Figure 9: the console of the Tomee Java web server

- Open a browser and type the address: <http://localhost:8080>. Check Figure 8 above.
- To stop the service, execute ./shutdown.sh

Note: It is possible that some of the scripts you need to execute do not have the execution permissions. If that is the case, you need to execute from the console:

```
sudo chmod +x scriptname.sh
```

Or to all scripts in the folder: sudo chmod +x Tomee/bin/*.sh

Linux

- Extract the content of the downloaded tomee file (e.g., tar xzf apache-tomee-plume-8.0.3.tar.gz) by opening a Terminal and typing:

```
tar xzf apache-tomee-plume-8.0.3.tar.gz
```

- Navigate to the extracted folder:

```
cd apache-tomee-plume-8.0.3
```

- Start the tomee server:

```
./startup.sh
```

- Open a browser and type the address: <http://localhost:8080>. Check Figure 8 above.
- To stop the service, execute ./shutdown.sh

2.4 Installing MySQL Server & Workbench

Windows

Install MySQL Community and MySQL Workbench, accepting all the default configurations and executing the steps necessary for installing possibly missing prerequisite packages. Be sure to store the root password in a safe and protected place. By default, MySQL is launched as a service at startup.

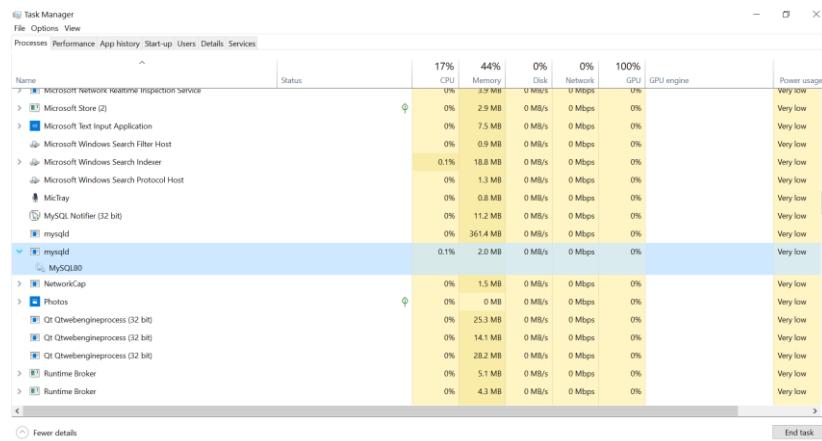


Figure 10: checking that the MySQL service is running in the Control Panel -> Administrative Tools -> Services

Open the workbench and check the status of the server.

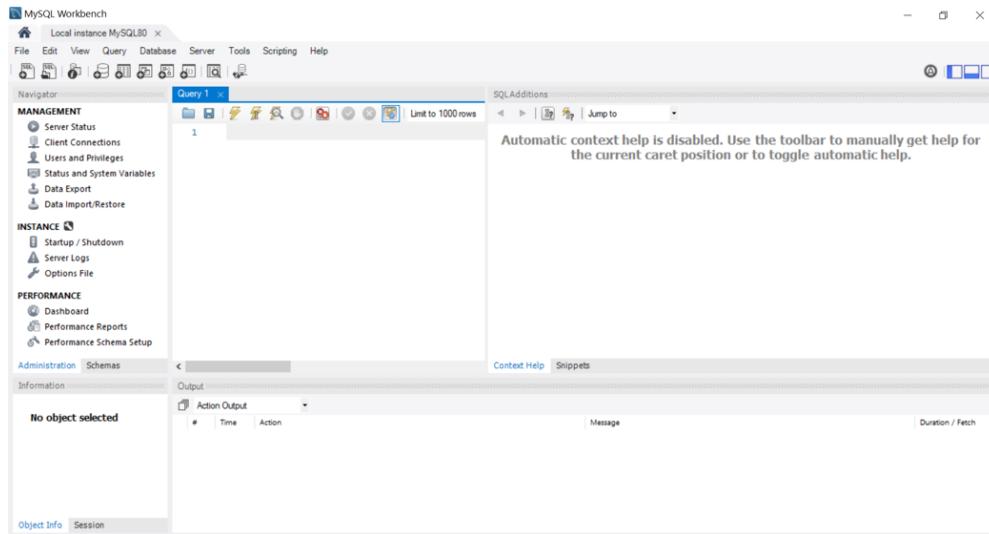


Figure 11: checking the server status in MySQL Workbench

OS X

- Install MySQL Community with all default configurations

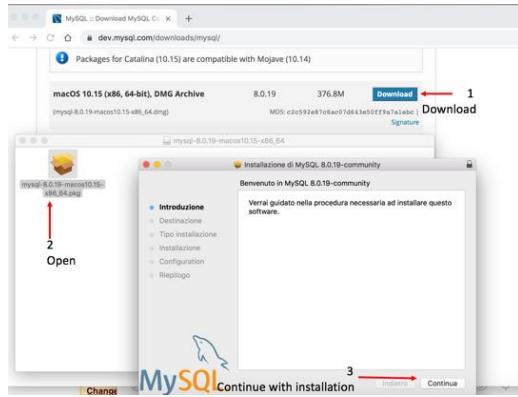


Figure 12 Installing MySQL in OS X

- When password is required insert one. Be sure to store the root password in a safe and protected place.



Figure 13 Installing MySQL in OS X

- MySQL Workbench open the .dmg and move the MySQL workbench to the Applications Folder (depending on your OS X version you might need to install and older version).

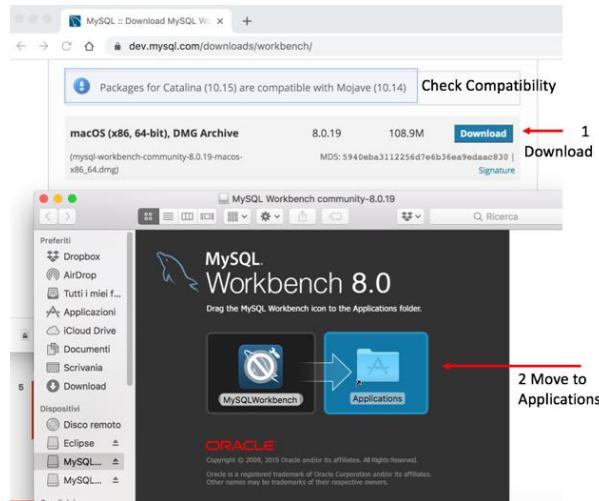


Figure 14 Installing MySQL Workbench in OS X

- Install MySQL Community and MySQL Workbench. After installing MySQL Workbench, open a terminal and type:

```
sudo ln -s /Applications/MySQLWorkbench.app/Contents/MacOS/mysqldump
/usr/local/bin/mysqldump
```

Linux

Open a terminal and type:

```
sudo apt-get install mysql-server mysql-workbench
```

During the installation you will be prompted to provide a password for the root user of the sql server, remember the password and store it in a safe and protected place.

Once installation is completed, the service will start automatically, and the MySQL Workbench application will be available in the app drawer. To verify the status of the service, execute the following:

```
sudo systemctl status mysql
```

2.4.1 Starting MySQL Server

Depending on the operating system, it is possible to start/stop the database as follows:

Windows

Configure at installation MySQL to start as a service at system start-up and use the workbench to check the server status and work with the server.

OS X

Open System Preferences and select MySQL. Clicking on the Start MySQL Server button will cause the MySQL server to start. (If it doesn't appear automatically after MySQL installation try restarting the computer).

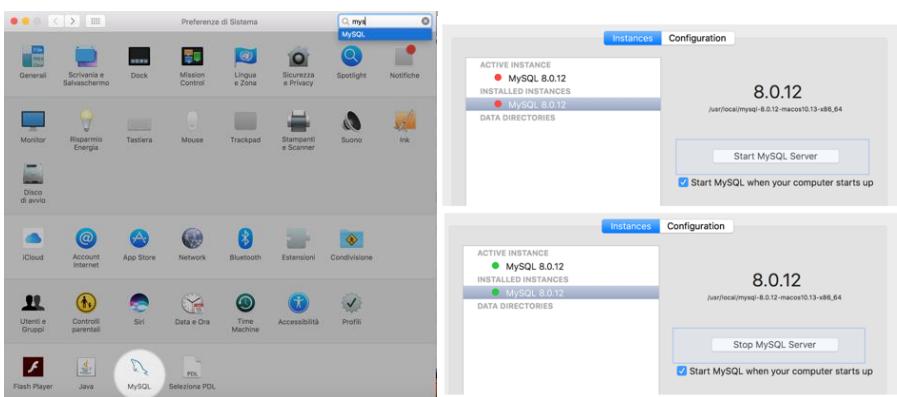


Figure 15 Start/Stop MySQL in MacOS X

Linux:

By default the server starts at boot time. If you wish to remove mysql-server from the startup, open a terminal and type

```
sudo update.rc-d mysql disable.
```

You can start and stop the server by using, respectively:

```
sudo service mysql start  
sudo service mysql stop
```

3 Configuration

3.1 Setting up the latest JDK with Eclipse

Open Eclipse. Go to the menu Preferences:

Windows: Window > Preferences

OS X: Eclipse > Preferences

Linux: Window > Preferences

Select the entry Java > Installed JREs

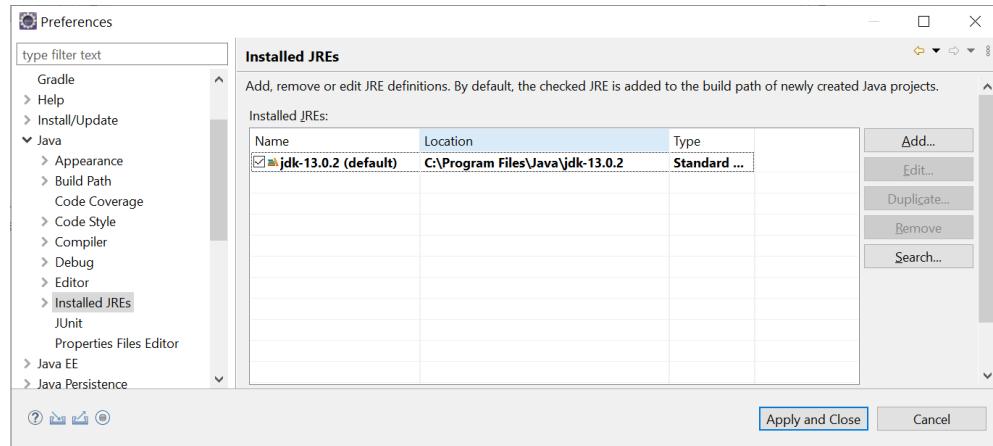


Figure 16 Checking the installed JREs in Eclipse

If Eclipse failed to automatically detect the presence of the JRE proceeds as follows.

Press the button Add. Then, select MacOS X VM (Mac) or Standard VM (Windows, Linux). Finally, specify the location of the JRE installed in your PC.

Example path for JRE in MacOS X:

/Library/Java/JavaVirtualMachines/jdk-13.0.2.jdk/Contents/Home

3.2 Adding TomEE to Eclipse

Open Eclipse. Go to the menu Preferences:

Windows: Window > Preferences

OS X: Eclipse > Preferences

Linux: Window > Preferences

Select the entry Server > Runtime Environments. Press the button Add.

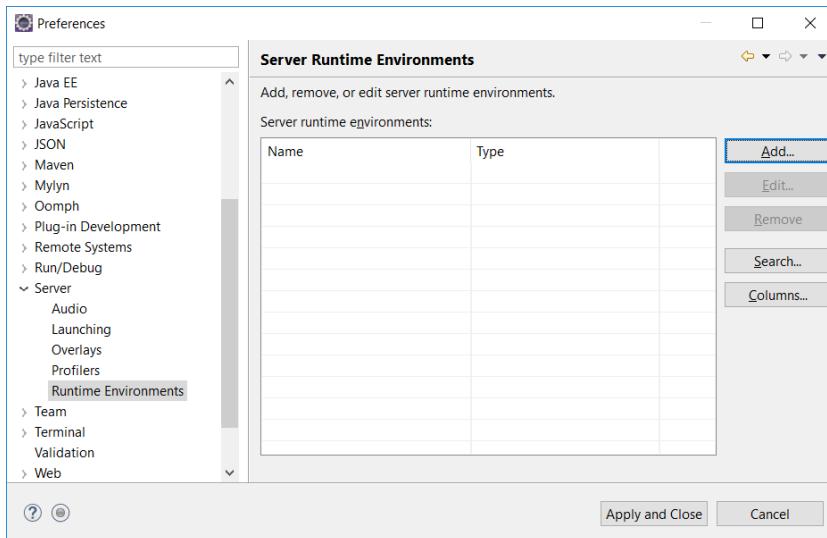


Figure 17 adding a runtime environment for Tomee

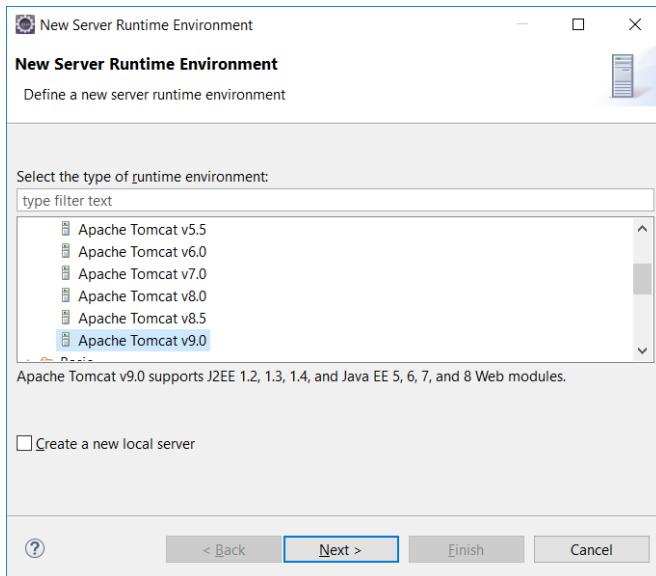


Figure 18 selecting the installed version of Tomee

Select the path of the Tomee folder on your system and press Finish.

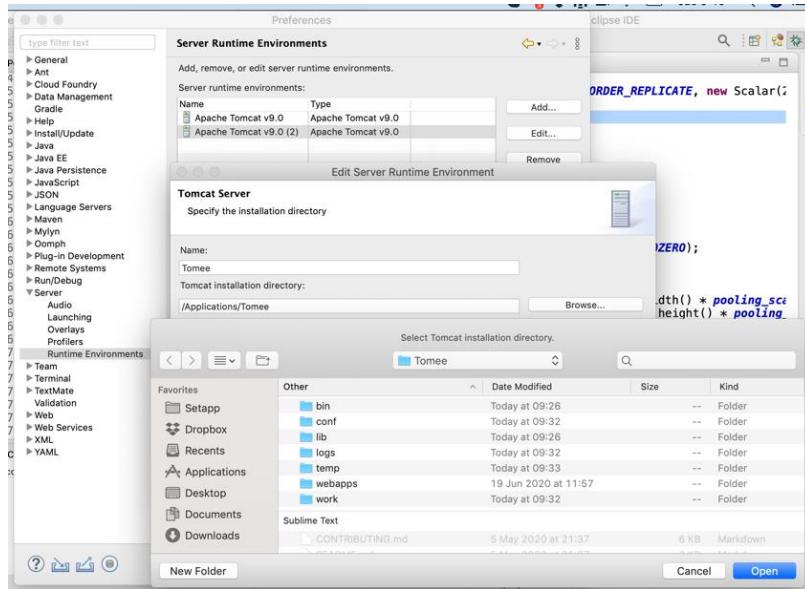


Figure 19 setting the TomEE folder

Select the tab Servers. (If not visible go to Windows→Select View→Other.. → Servers)

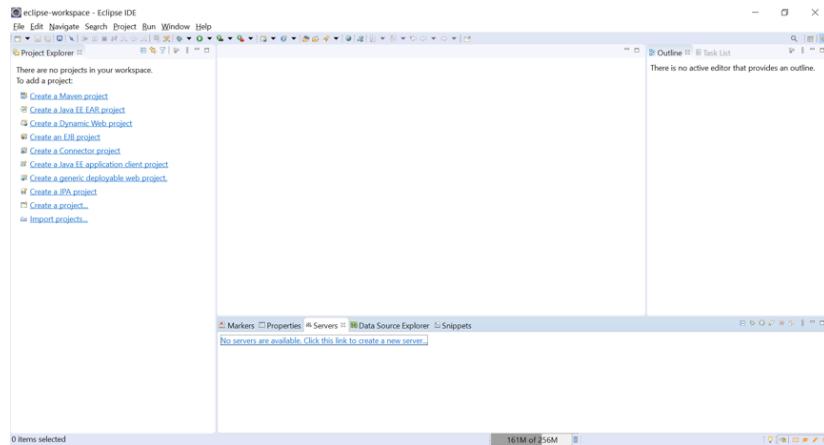


Figure 20 the Servers tab

Use the command to add a new server

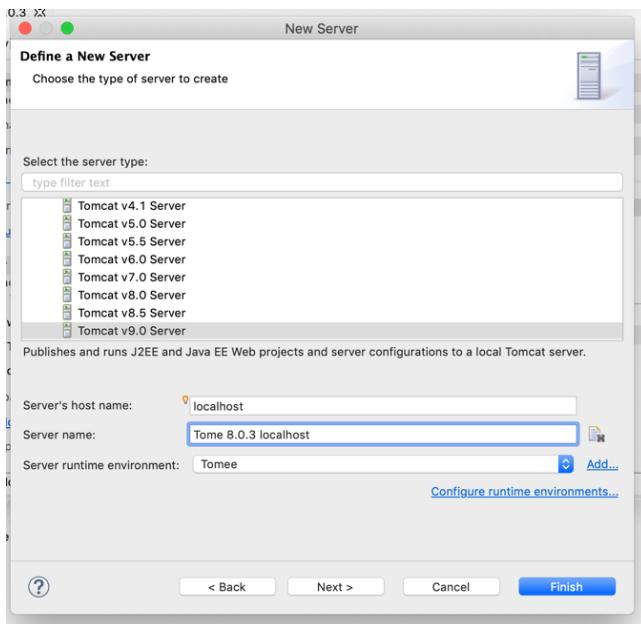


Figure 21 add TomEE as a server

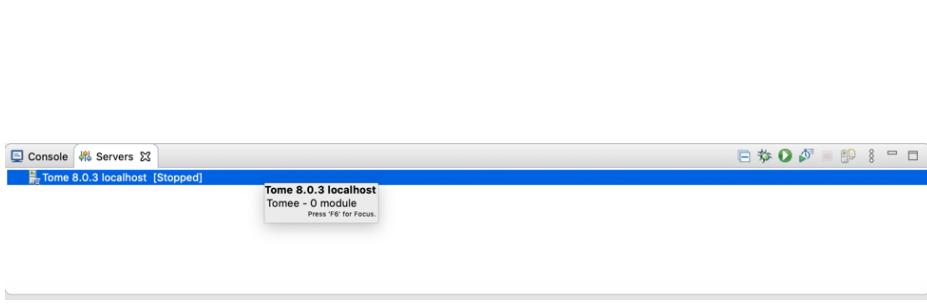
Select the server runtime environment that you created in the previous step and click on Finish. At this point, the Tomee server is connected to Eclipse and one can interact with it through the Eclipse interface.



Figure 22 TomEE appears under the Servers node in the Project Explorer

3.3 Configuring TomEE

In the Servers tab, double click on the Tomee server.



Note: If you don't see this tab, go to Window→Show View → other and search for Servers.

The server properties will open. Under the Server Locations section, select the option “*Use Tomcat installation (takes control of Tomcat installation)*”.

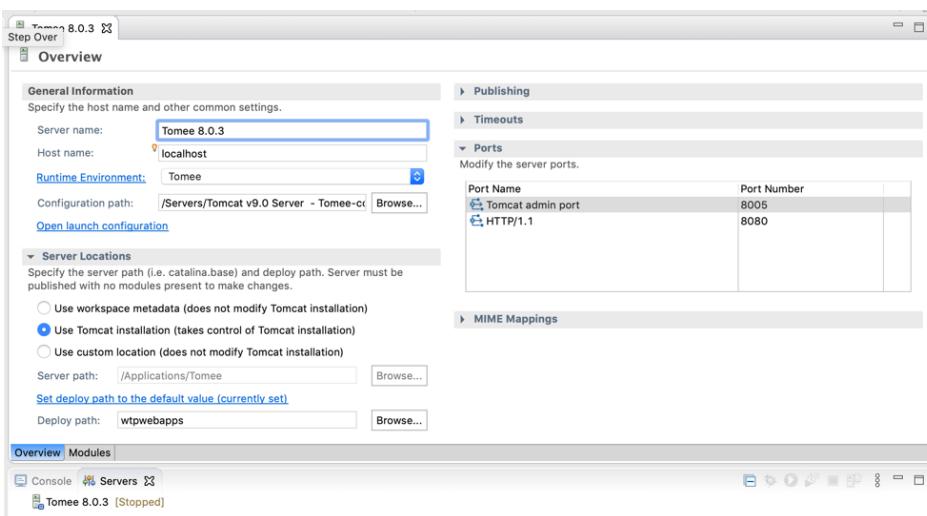


Figure 23 configuring TomEE

In the Server Locations path, make you're the selected one is “*Use Tomcat Installation*”

Set the admin port to a distinct value as shown.

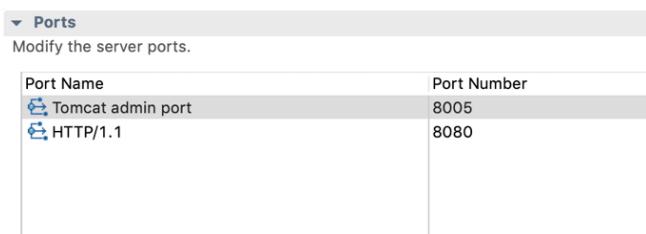
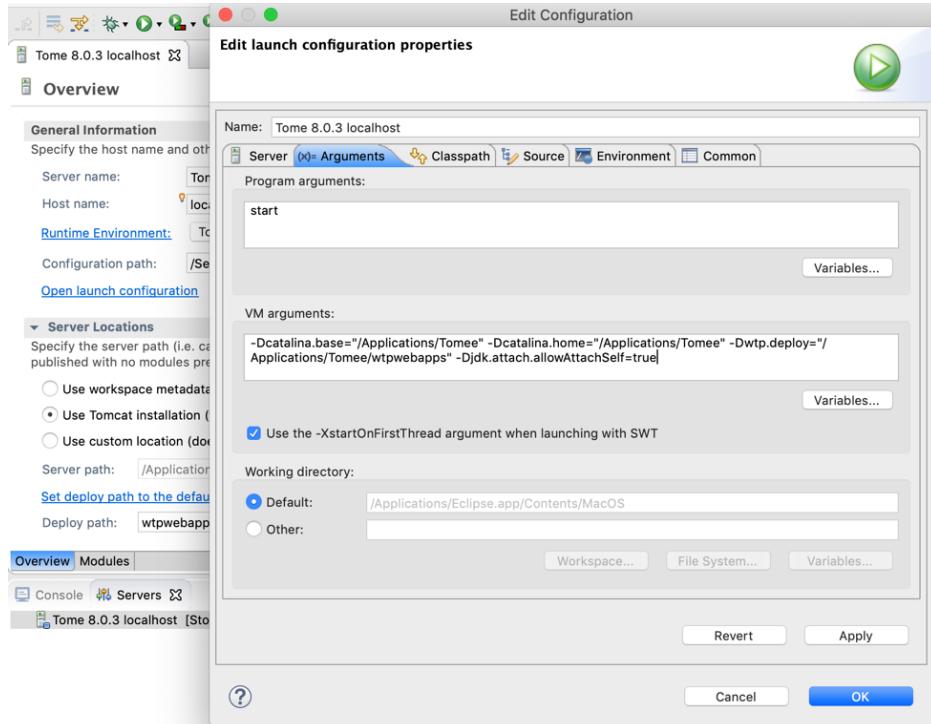


Figure 24: setting the admin port of TomEE in Eclipse

Now press on “Open launch configuration” and in the new window select the “Arguments” tab. Here you have to add “**-Djdk.attach.allowAttachSelf=true**” after all the VM arguments as shown in the figure.



3.4 Starting TomEE

In the tab Servers, right click on the Tomee server and click on Start. The console shows the server status. When the message “*INFO: Server startup in XXX ms*” is shown, the server is up and running.

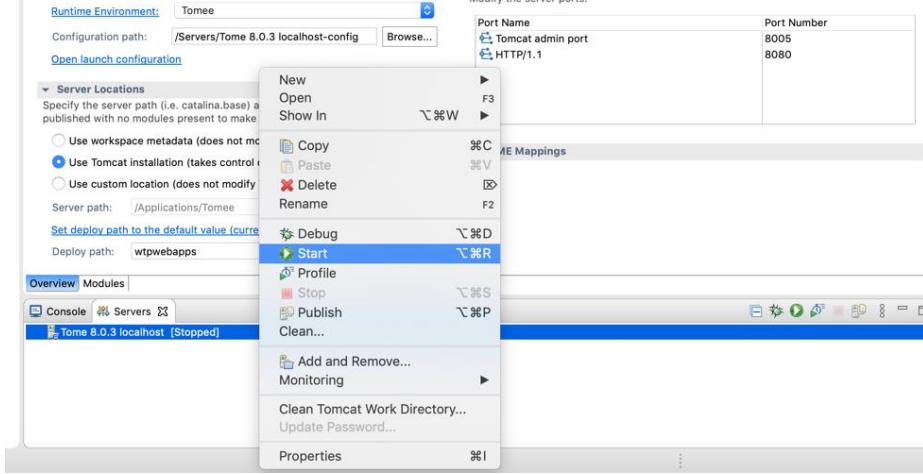


Figure 25: starting Tomee from the servers tab

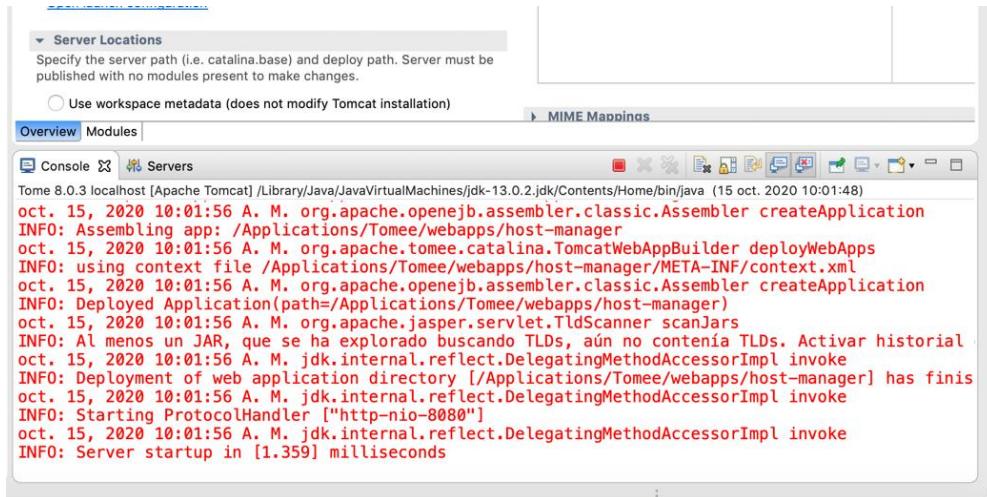


Figure 26: console message showing that Tomee is running

To check that Tomee is running open the internal web browser:

Windows/Mac OS/Linux

Go to the menu Window → Show View → Internal Web Browser

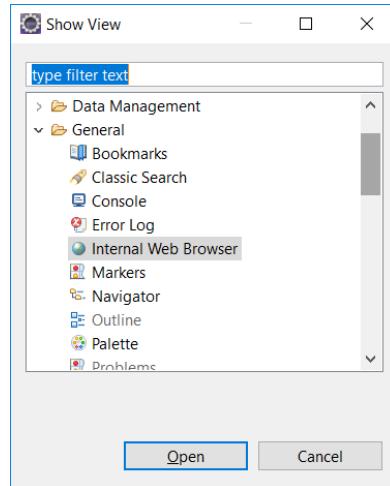


Figure 27: showing the internal web browser

Type `http://localhost:8080` in the location area of the internal browser.

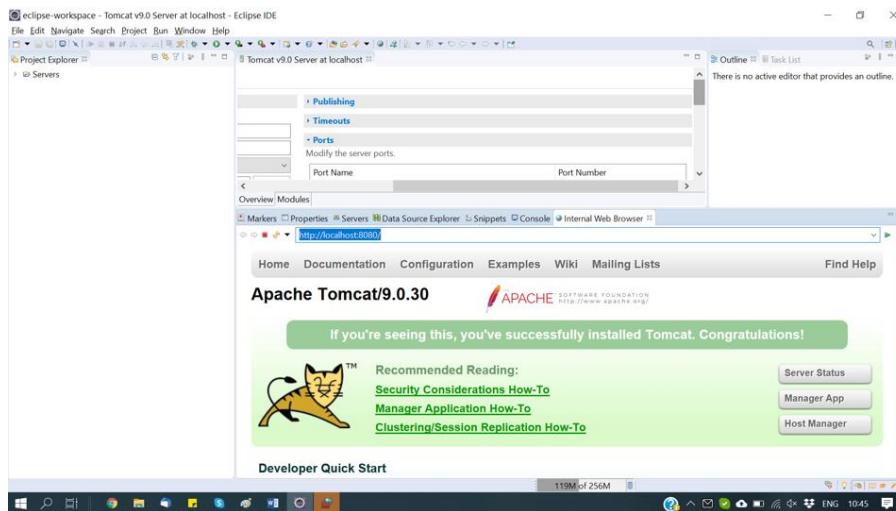


Figure 28 opening the home page of tomee in the internal browser

4 Publishing content

4.1 Creating a dynamic web project

In the Project Explorer section, right click and select New -> Project -> Web -> Dynamic Web Project

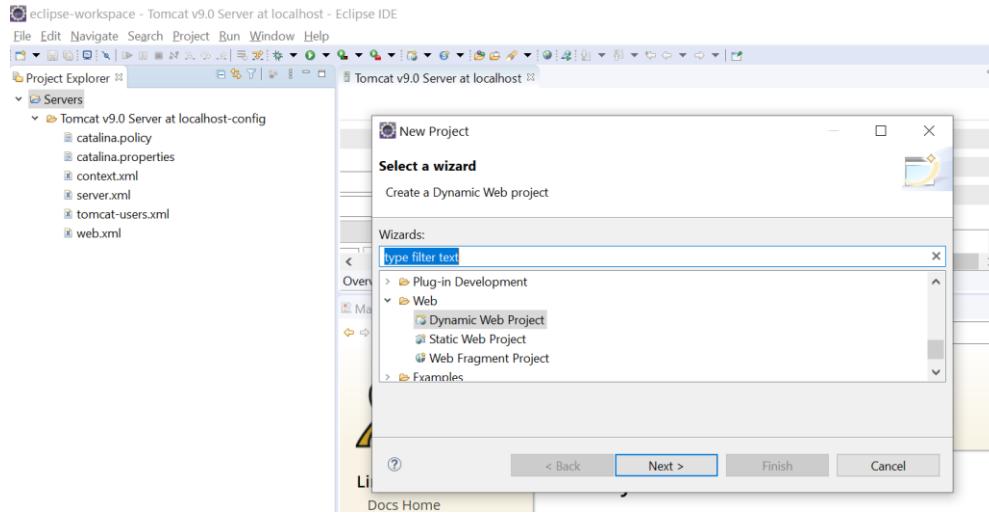


Figure 29: creating a dynamic web project

In the project creation wizard, name the project TestProject, select Tomee as target runtime and accept all the creation default values.

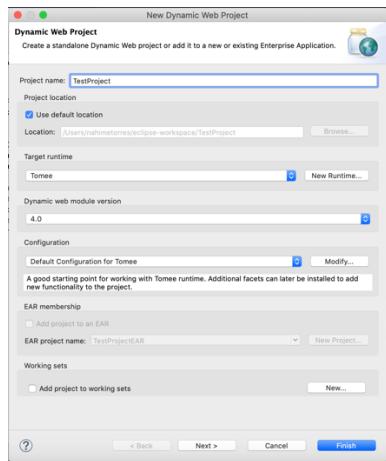


Figure 30: naming and creating a dynamic web project

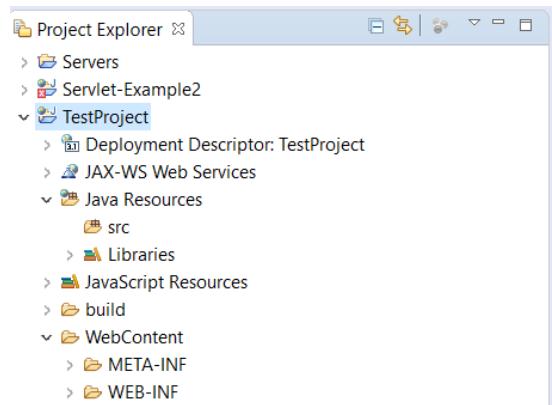


Figure 31: dynamic project structure created in the project explorer

4.2 Writing the Web component

Right click on the project source folder, i.e., Java Resources > src inside the created application, then select New > Servlet.

Give the name TestServlet to the servlet and the name *it.polimi.tiw.test* to the package. In the servlet creation wizard, accept all defaults and finish.

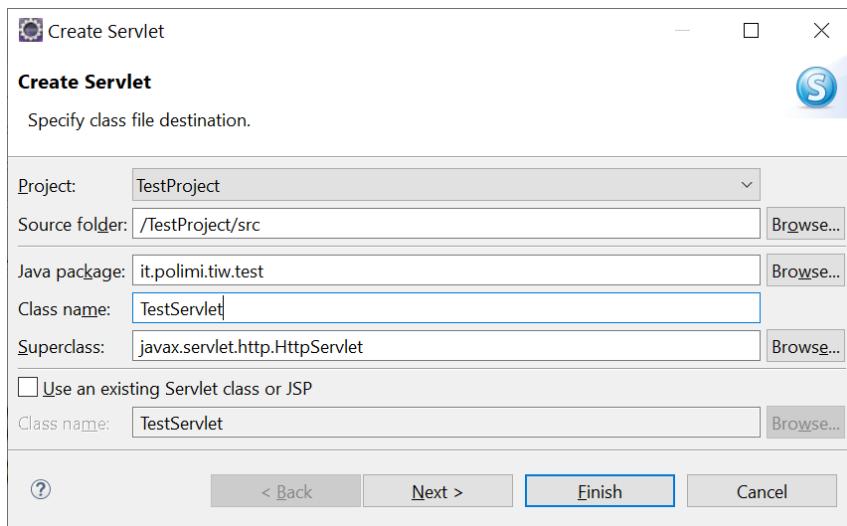


Figure 32: name the servlet class and the enclosing package

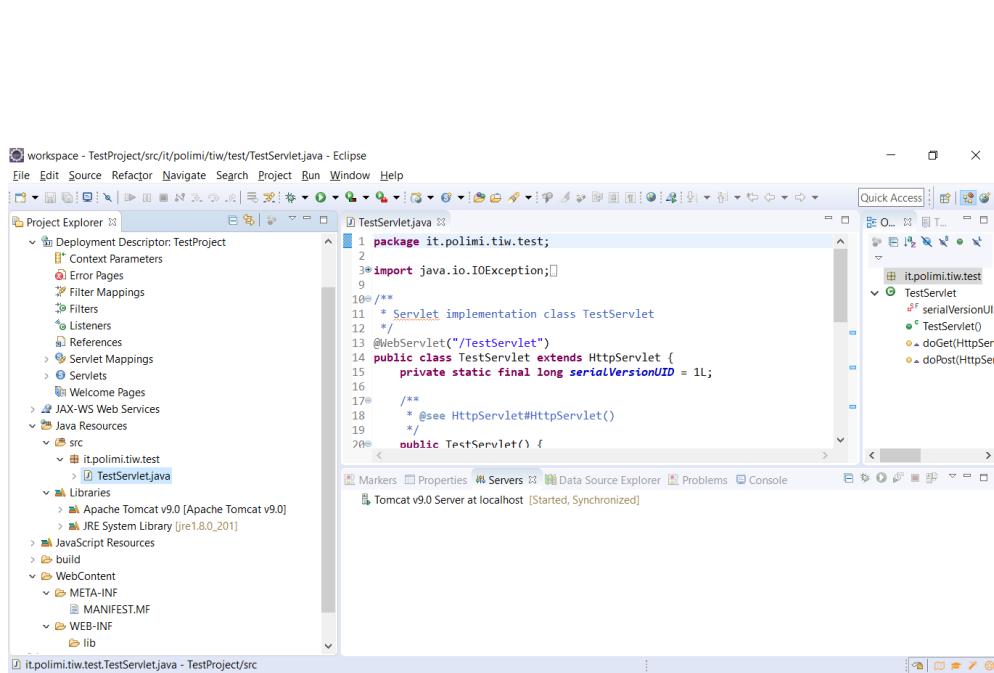


Figure 33: a newly created servlet

Remove entirely the auto-generated code, paste the following, and save the project files.

```
package it.polimi.tiw.test;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class TestServlet extends HttpServlet {

    private static final long serialVersionUID = 1L;

    protected void doGet(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException {
        response.setContentType("text/plain");

        PrintWriter out = response.getWriter();

        out.println("Hello this is a test");

        out.close();
    }
}
```

In the WebContent/WEB-INF folder, create a file named `web.xml`. To do this right click on the project folder → Java EE Tools → Generate Deployment Descriptor Stub.

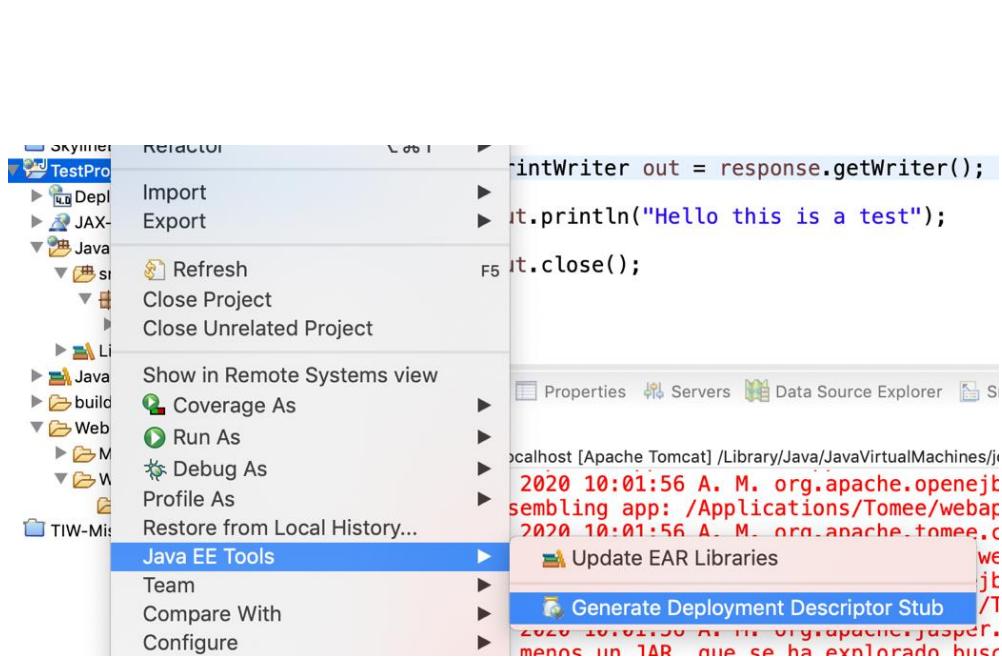


Figure 3433: a newly created servlet

Right click on it to edit it with a standard text editor and put this content in it:

```

<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
    http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
  version="3.1">
  <servlet>
    <servlet-name>TestServlet</servlet-name>
    <servlet-class>it.polimi.tiw.test.TestServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>TestServlet</servlet-name>
    <url-pattern>/</url-pattern>
  </servlet-mapping>
</web-app>

```

You can use **CRTL-SHIFT-F** to indent the code automatically.

To add the created project to the Tomee server, right click on the server (in the Servers tab), then select Add and Remove. Select TestProject from the left-hand side of the menu, press the Add button and finally click on Finish.

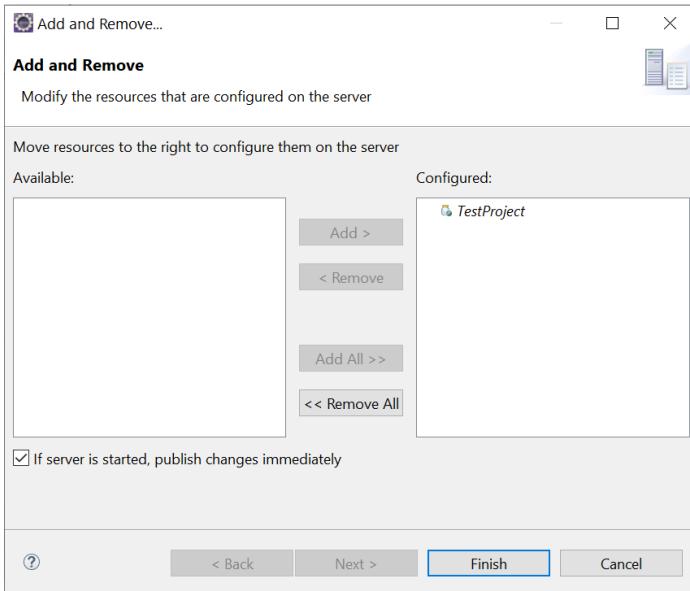


Figure 3534: adding the project to the Tomee server

```

Markers Properties Servers Data Source Explorer Snippets Console Internal Web Browser
Tomcat v9.0 Server at localhost [Apache Tomcat] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (6 Feb 2020, 10:42:30)
Feb 06, 2020 11:01:54 AM org.apache.catalina.startup.HostConfig deployDescriptor
INFO: Server startup in [1,537] milliseconds
Feb 06, 2020 11:01:54 AM org.apache.catalina.startup.HostConfig deployDescriptor
INFO: Deploying deployment descriptor [C:\Program Files\Apache Software Foundation\Tomcat 9.0\conf\Catalina\localhost\TestProject.xml]
Feb 06, 2020 11:01:54 AM org.apache.catalina.startup.HostConfig deployDescriptor
WARNING: The path attribute with value [/TestProject] in deployment descriptor [C:\Program Files\Apache Software Foundation\Tomcat 9.0\conf\Catalina\localhost\TestProject.xml] inside the host appBase has been specified
Feb 06, 2020 11:01:54 AM org.apache.catalina.startup.HostConfig deployDescriptor
WARNING: A docBase [C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps\TestProject] inside the host appBase has been specified
Feb 06, 2020 11:01:54 AM org.apache.catalina.startup.HostConfig deployDescriptor
INFO: Deployment of deployment descriptor [C:\Program Files\Apache Software Foundation\Tomcat 9.0\conf\Catalina\localhost\TestProject.xml]

```

Figure 3635: deployment of the project in Tomee

4.3 Testing your application

Select the TestProject project in the Project Explorer, right click on it, choose the RunAs command, then select RunOnServer.

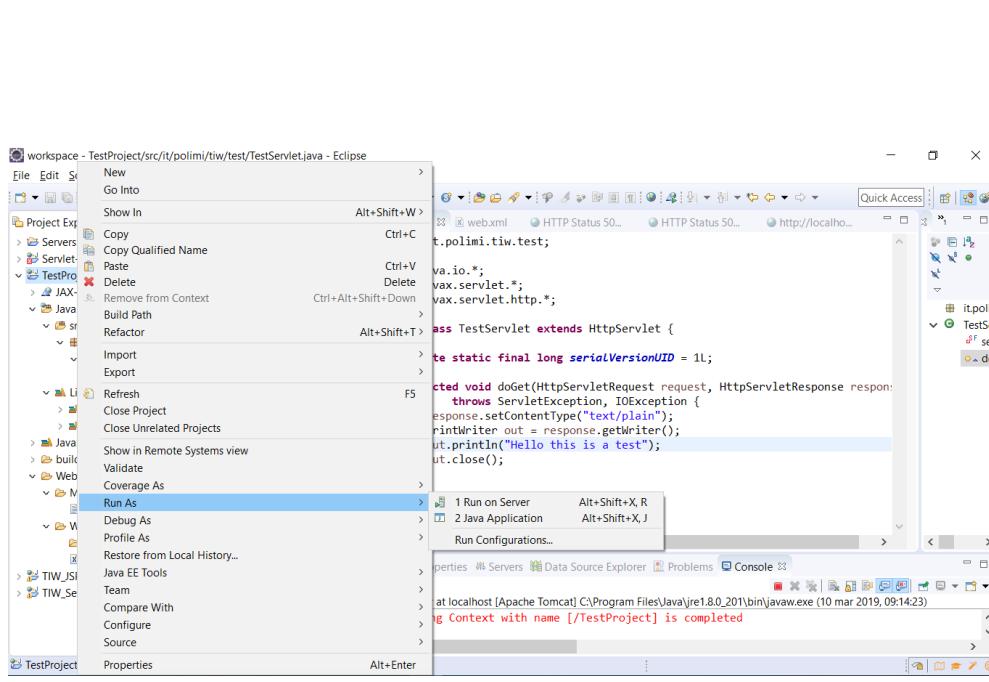


Figure 3736: running the project on a server

Select the Tomee server and finish.

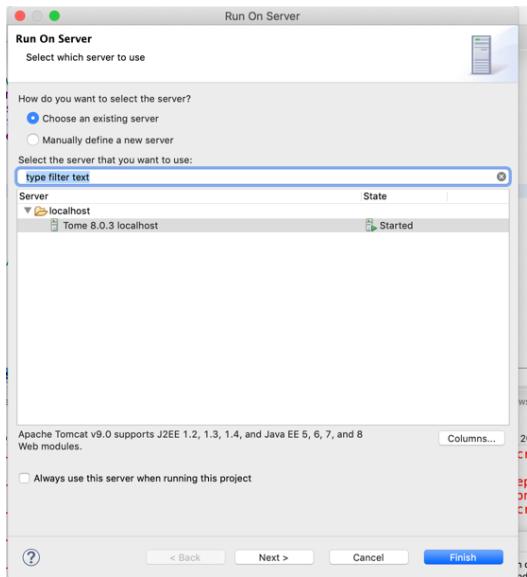


Figure 3837: deploying the project on Tomee

Restart the server and continue.

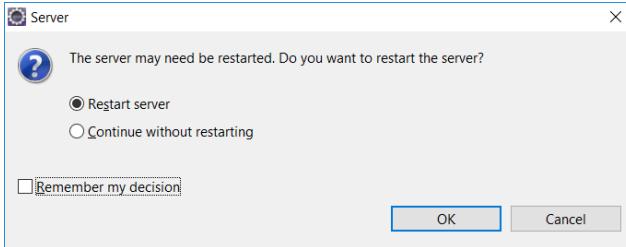


Figure 3938: restarting the server (optional)

Now the internal browser window opens on the location of the home page of the project.

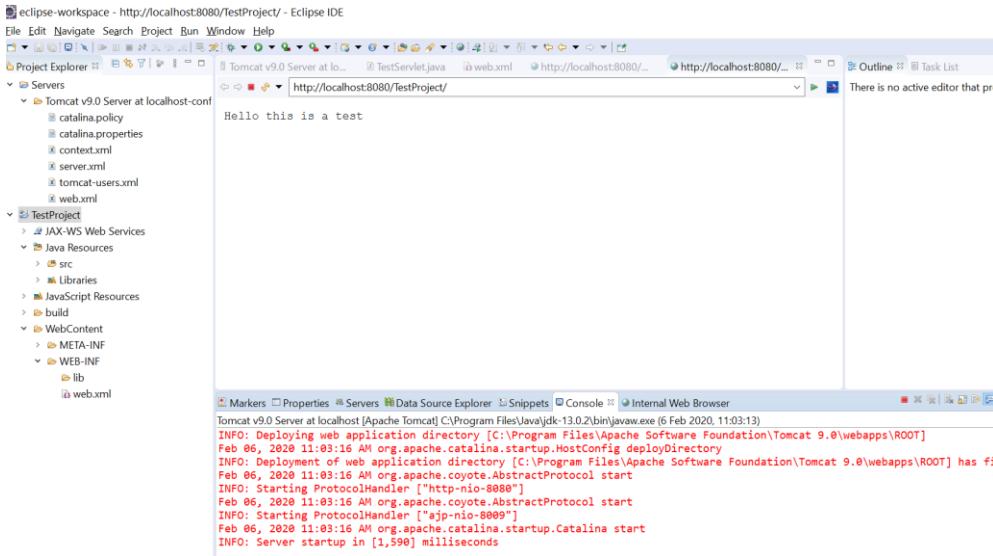


Figure 4039: the internal browser opens on the location of the servlet

5 Connecting with JDBC

5.1 Add mysql connector to the project

Obtain the Connect/J jar file

WINDOWS

The file is already part of the MySQL installation. It can be found, for example, at the location

```
C:\Program Files (x86)\MySQL\Connector J 8.0\mysql-connector-java-8.0.19.jar
```

LINUX and OS X

Download MySQL Connector/J, the official JDBC driver for MySQL. (check it is for the same version of your mysql version)

<https://dev.mysql.com/downloads/connector/j/>

Choose the platform independent version.

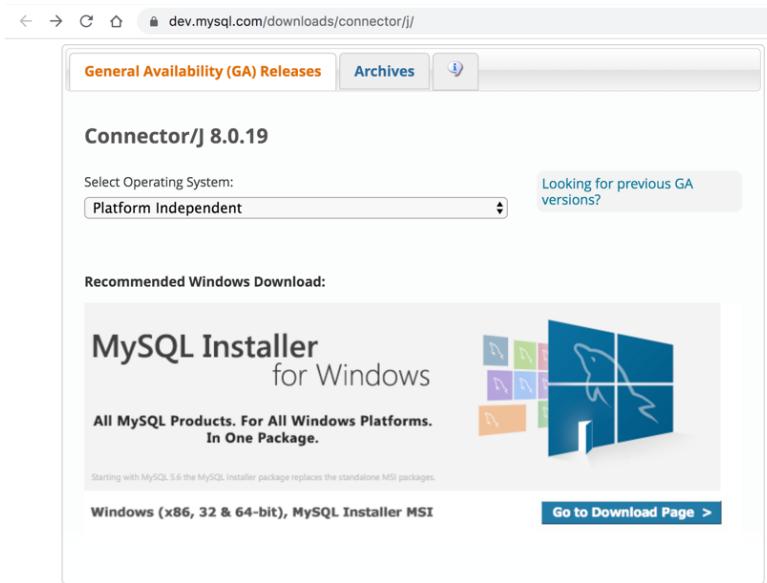


Figure 4140: downloading the proper version of the connector library

Extract to a local folder the file mysql-connector-java-{version}.zip (or tar.gz).

5.2 Installing Connector/J in Tomee

5.2.1 First copy the mysql-connector-java-{version}.jar into tomee by adding it to {tomEE installation path}/lib folder.

5.2 Then to add it to the project copy the file the mysql-connector-java-{version}.jar and paste it in Eclipse into the to the project directory WebContent/WEB-INF/lib/

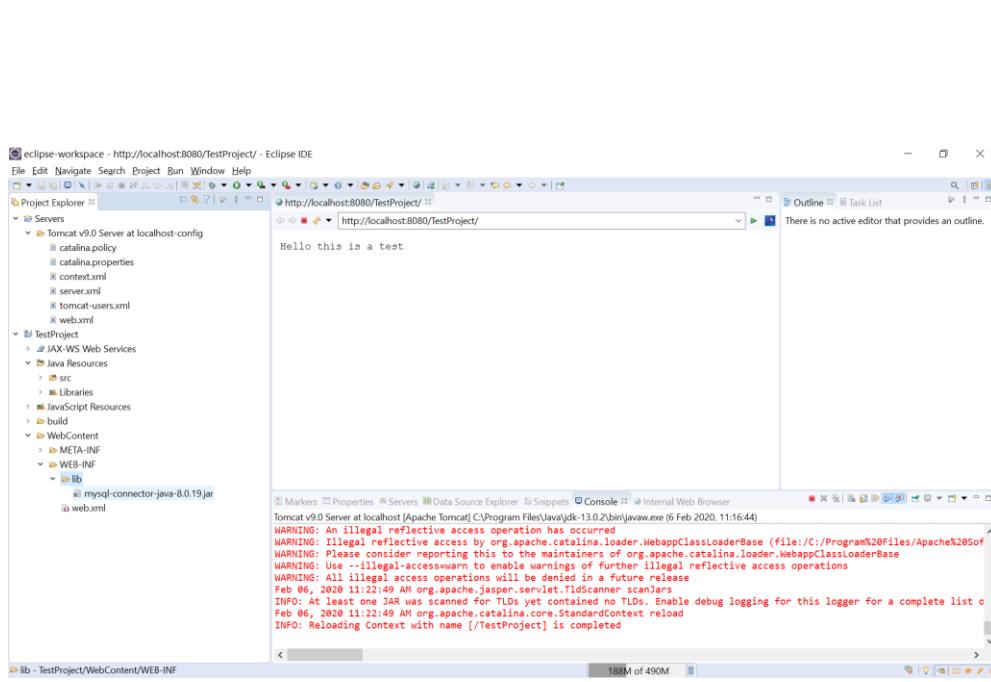


Figure 4241: adding the ConnectorJ jar file to a specific project

5.2.3 In the Eclipse Project Explorer, right click on the project, and select Build Path and add the mysql-connector-java-{version}.jar library to the project.

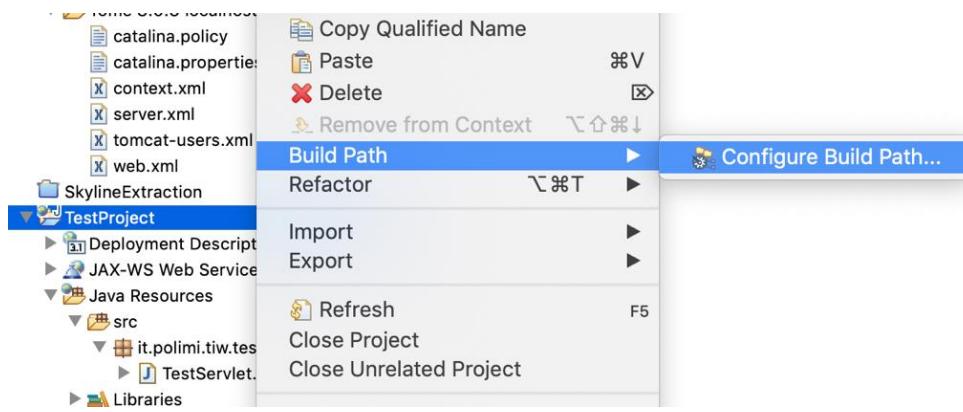


Figure 4342: configuring the build path of the project

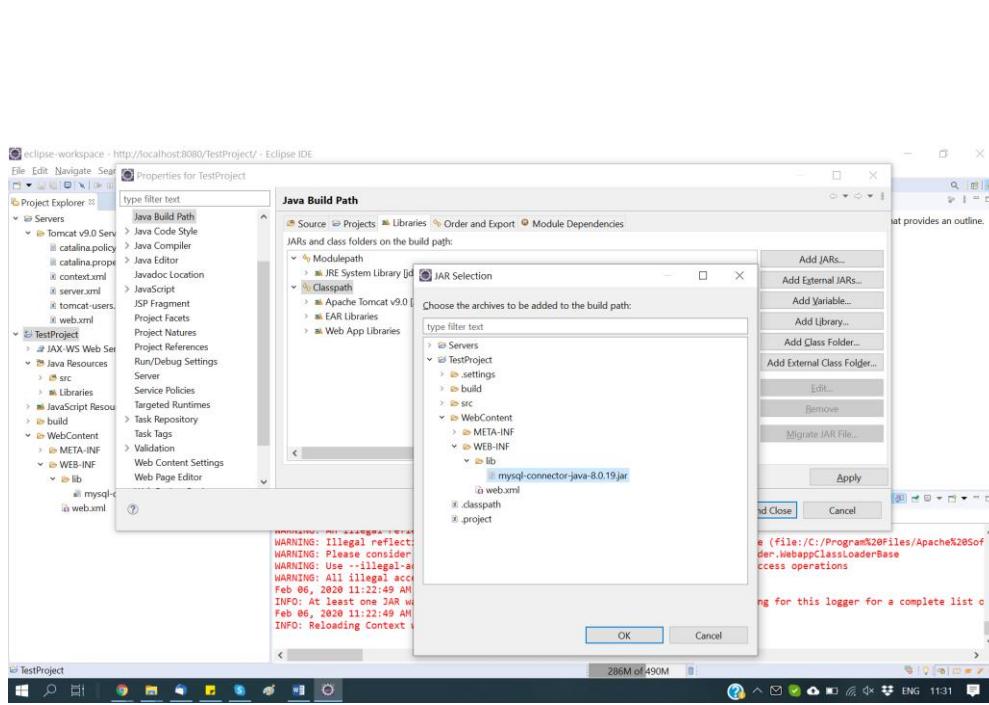


Figure 4443: adding the JAR to the project build path (to the classpath)

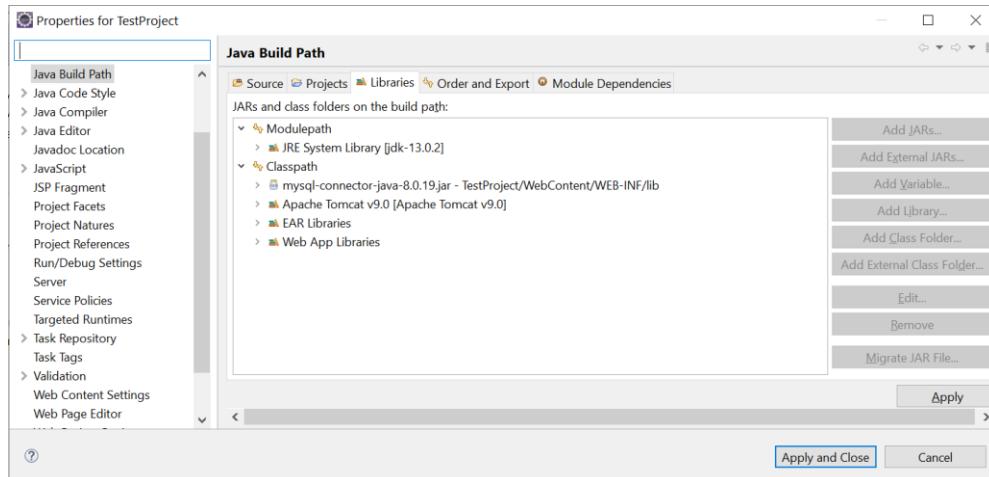


Figure 4544: checking the build path for the presence of Connector/J

5.3 Create a new servlet

Create a new dynamic web project and add to it a Servlet.

Edit the servlet source and paste this code (be sure to modify your database USER and PASSWORD to set them to a database user you have created and associated with the dbtest database schema)

```
package it.polimi.tiw.jdbcTest;

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.DriverManager;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/ConnectionTester")
public class ConnectionTester extends HttpServlet {
    private static final long serialVersionUID = 1L;

    protected void doGet(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException {

        final String DB_URL = "jdbc:mysql://localhost:3306/dbtest";
        final String USER = "piero";
        final String PASS = "fraternali";
        String result = "Connection worked";
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            DriverManager.getConnection(DB_URL, USER, PASS);
        } catch (Exception e) {
            result = "Connection failed";
            e.printStackTrace();
        }

        response.setContentType("text/plain");

        PrintWriter out = response.getWriter();
        out.println(result);
        out.close();
    }
}
```

Remember to create the web.xml file and to declare the servlet mapping in the web.xml file as done for the previous servlet.

5.4 Create a new database

To test the connection you need a SQL database. Create a new database, named dbtest, with the MySQL Workbench.

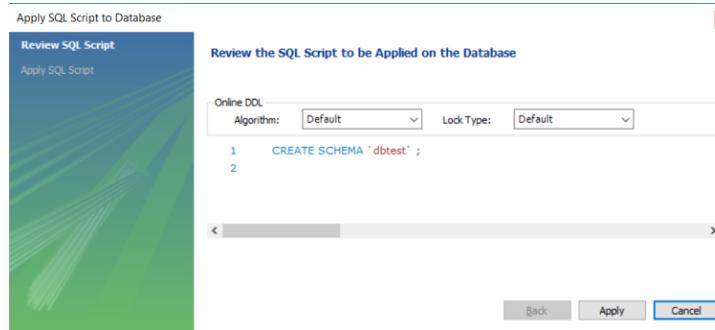
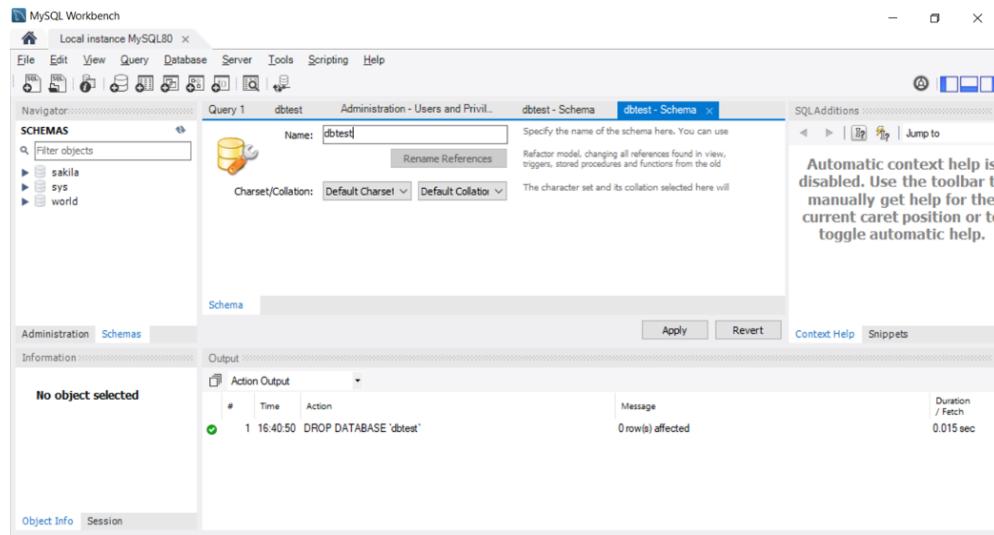


Figure 4645: creating a test database

5.5 Run the project

Run your application (Run → Run on server...) and access the new servlet:

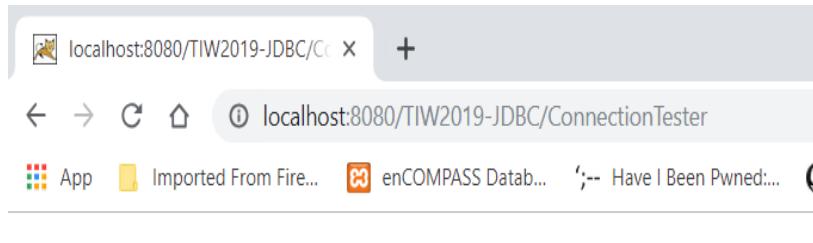


Figure 4746: testing the servlet and the database connection

6 Stopping Tomee

When you are not using Tomee anymore, you can stop the server by right clicking on the server name and then selecting Stop.

7 Troubleshooting

7.1 Windows MySQL TimeZone Error

LOCAL SOLUTION (SUGGESTED FOR MacOS)

To every database connection append the parameter "?serverTimezone=UTC".

An example is "jdbc:mysql://localhost:3306/dbtest?serverTimezone=UTC" in the web.xml file or in your servlet class.

This will overwrite the default time zone set by Windows.

GLOBAL SOLUTION (SUGGESTED FOR WINDOWS)

This solution modifies the configuration file of mysql server so it will be applied to every database connection without the need to set the above parameter. **This is the suggested solution.**

- Open MySQL Workbench and connect to a database (if you don't have any database just create one and connect to it)
- In the new window, you must click on the **Administration** tab and then the "**Options File**" option
- Now you must stay in the **general** tab and scroll to the bottom until you find the "**default-time-zone**" checkbox
- Select the checkbox and in the text field write '+01:00' (**remember to write also the two '**)
- In the bottom-right corner check that "**mysqld**" is selected and the press apply

For the change to take effect, you need to restart the MySQL server service:

- Press WIN + R and type “**services.msc**” then press enter
- In the services window you must find the service called **MySQL80** (if you are using MySQL 8.0) and restart it

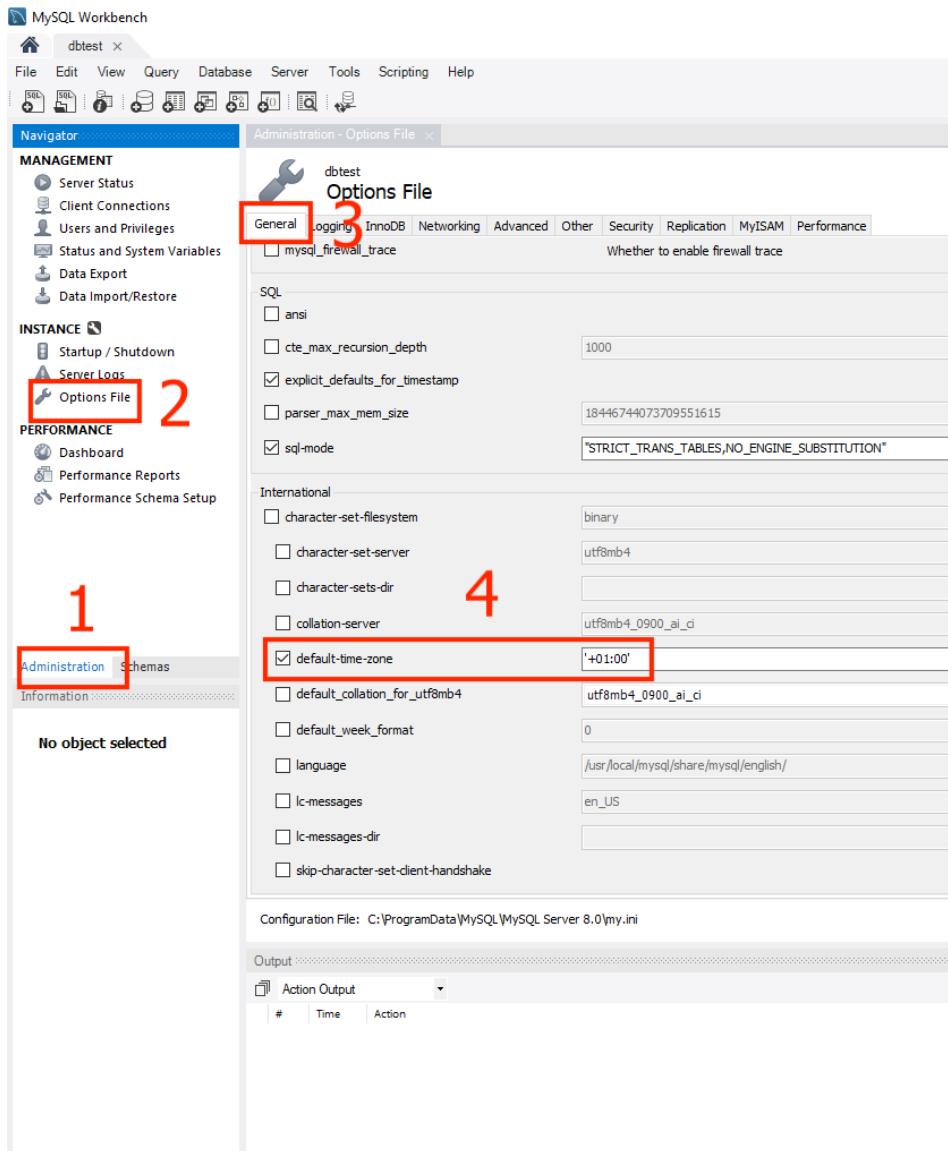


Figure 47

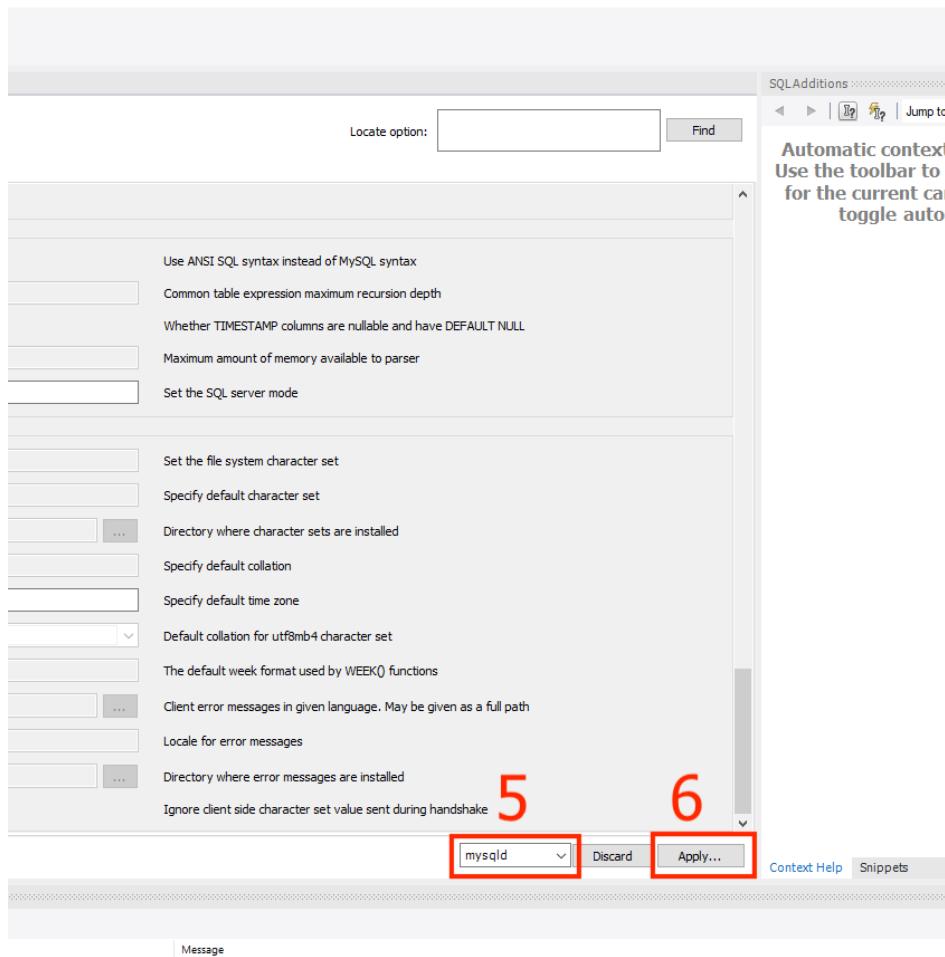


Figure 48

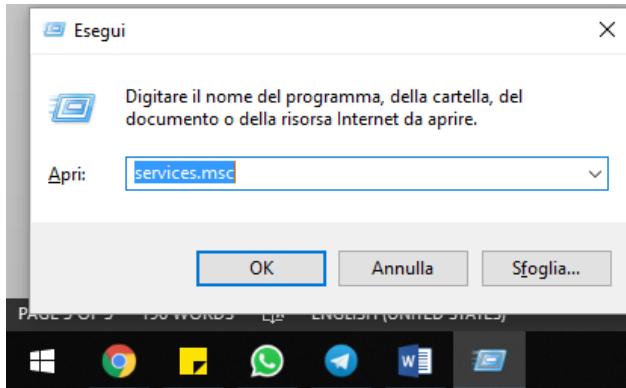


Figure 49

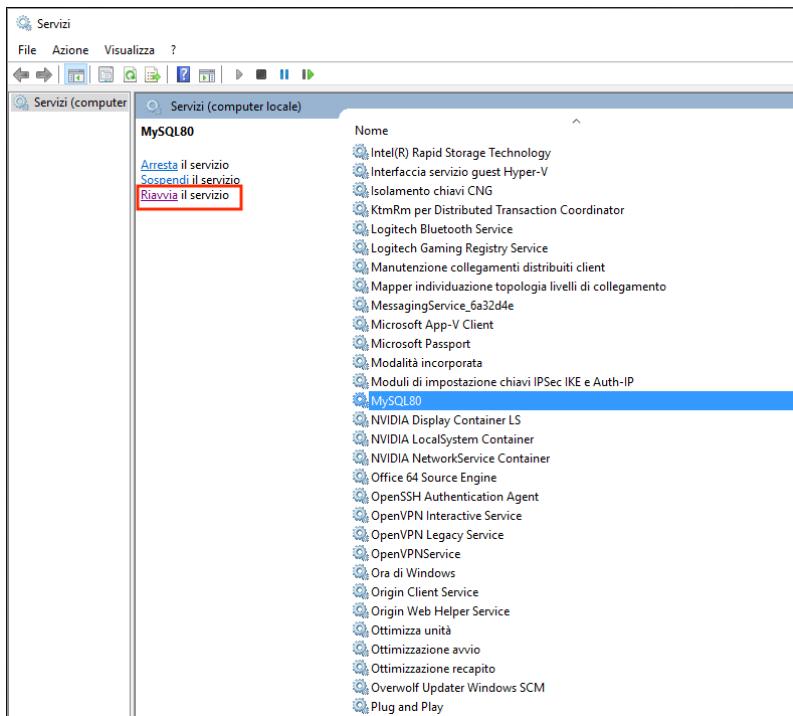


Figure 50

8 Running the Book examples:

Download the Book examples

The book provides a set of examples that can be download here:

<https://github.com/Apress/pro-jpa-2-in-java-ee-8>

Examples can be download as ZIP file or by cloning the git repository.

The book example can be executed with a set of tools that include java JDK 8, Glassfish server, Derby database and apache Ant scripts, some of those tools are outdated and require configuration.

In the following section of the guide We will provide the necessary steps to execute the examples with Java JDK 13, Eclipse, TomEE and MySQL.

In the guide We refer to the relative paths of the content of the zip file or repository.

Running Chapter 2 – employeeService example

1.-Create a new schema (*EmpServDB*) as explain in “[5.4 Create a new database](#)”, using the workbench run the \examples\Chapter2\employeeService\etc\sql\db.sql script to create the employee table, if you face an error use quote on the table name as follows:

```
DROP TABLE IF EXISTS `EMPLOYEE`;
```

2.-On eclipse create a new JPA project:

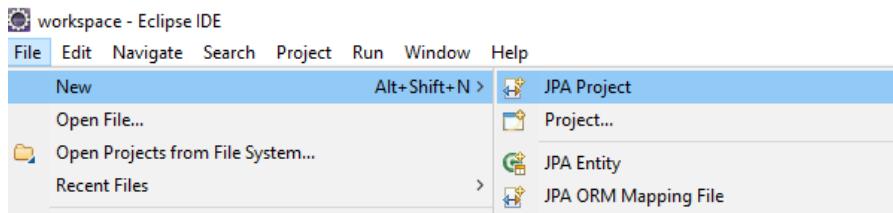


Figure 51

3.-Give it a name, select Tomee as target runtime and leave the default values:

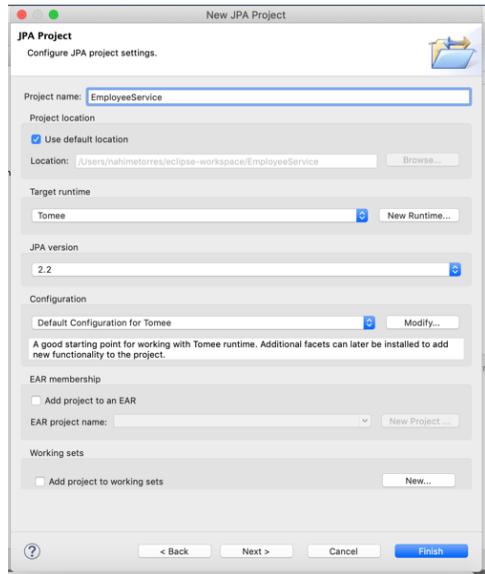


Figure 52

4.-Create two new packages examples.client and examples.model

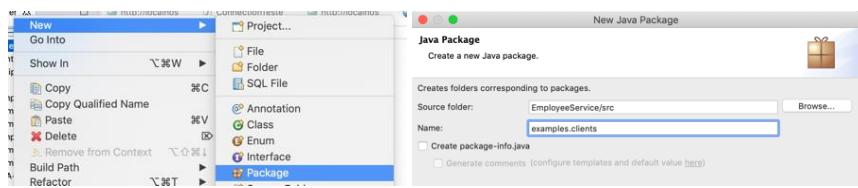


Figure 53

On the new project copy the source files from:

- \examples\Chapter2\employeeService\src\client
- \examples\Chapter2\employeeService\src\model

to the src folder. The project should look as follows:

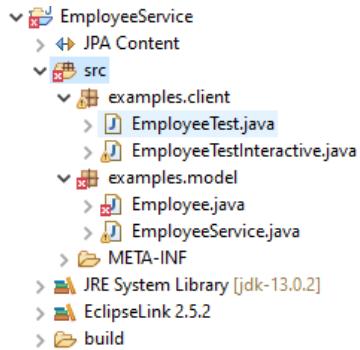


Figure 54

5.-Open the \examples\Chapter2\employeeService\etc\persistence\META-INF\persistence.xml and copy the content into the project persistent.xml

Modify the connection properties for your environment. The driver and URL should correspond to mySQL, and the user and pass should be for your local DB.

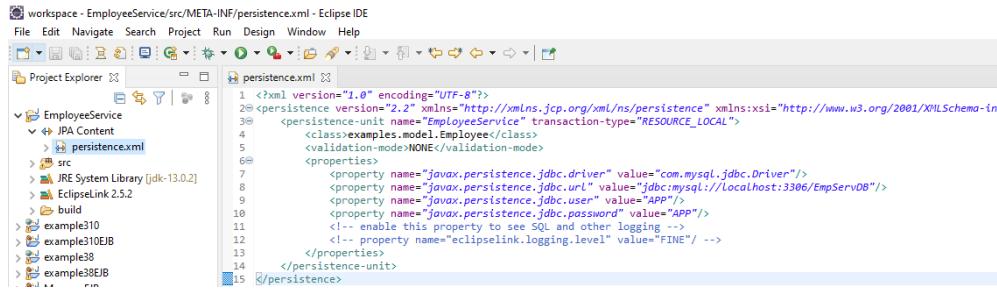


Figure 55

```

<property name="javax.persistence.jdbc.driver" value="com.mysql.jdbc.Driver"/>
<property name="javax.persistence.jdbc.url" value="jdbc:mysql://localhost:3306/EmpServDB "/>
<property name="javax.persistence.jdbc.user" value="YOUR_USER"/>
<property name="javax.persistence.jdbc.password" value="YOUR_APP"/>

```

6.-Add the MySQL JDBC connector to the project as explained in section "[5.2 Installing Connector/J in Tomee](#)
[5.2 Installing Connector/J in Tomcat](#)" (if not already in the Tomee/libs folder)

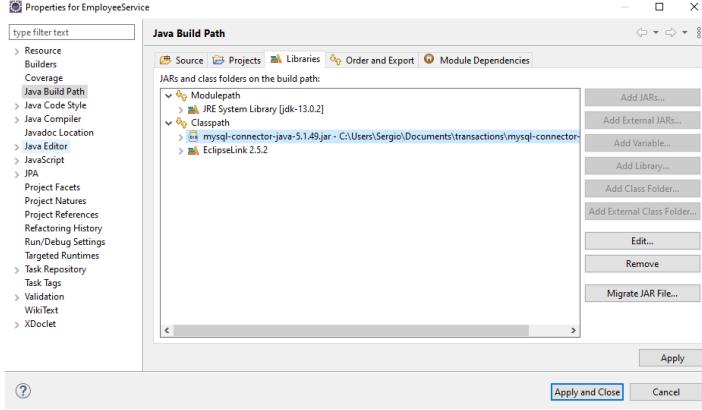


Figure 56

7.-To test this go to the [EmployeeTest.java](#) and do right-click on it and selecting “run as...”-> Java Application. You should see an output as follows:

```
Problems | Console 
<terminated> EmployeeTest (1) [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (Sep 25, 2020, 4:54:53 PM - 4:54:55 PM)
[EL Info]: 2020-09-25 16:54:54.361--ServerSession(671467883)--EclipseLink, version: Eclipse Persistence Services - 2.5.2.v20140319-9ad6abd
Fri Sep 25 16:54:54 CEST 2020 WARN: Establishing SSL connection without server's identity verification is not recommended. According to MySQL 5.5
Fri Sep 25 16:54:54 CEST 2020 WARN: Establishing SSL connection without server's identity verification is not recommended. According to MySQL 5.5
[EL Info]: connection: 2020-09-25 16:54:54.837--ServerSession(671467883)--file:/C:/Users/Sergio/Documents/transactions/workspace/EmployeeService/
Persisted Employee id: 158 name: John Doe salary: 45000
Found Employee: Employee id: 158 name: John Doe salary: 45000
Found Employee: Employee id: 1 name: TEST salary: 10
Found Employee: Employee id: 158 name: John Doe salary: 45000
Updated Employee id: 158 name: John Doe salary: 46000
Removed Employee 158
[EL Info]: connection: 2020-09-25 16:54:55.094--ServerSession(671467883)--file:/C:/Users/Sergio/Documents/transactions/workspace/EmployeeService/
```

Figure 57

Running Chapter 3 examples (1-15):

Examples in chapter 3 need to be tested with 2 projects, a Web project and an EJB project. All of them follow the same structure. Here is the guide to run the first one “01-slsbExample” but it applies to the others as well. You can test them as follows:

EJB project.

1.- Create a new EJB project

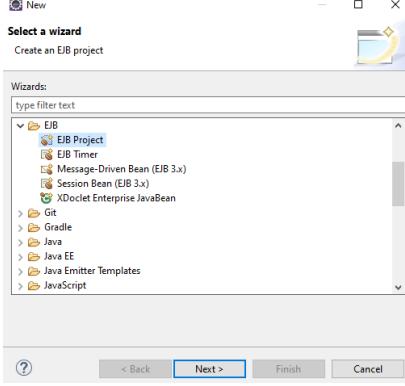


Figure 58

Give it a name with suffix EJB to make it easy to identify and click finish.

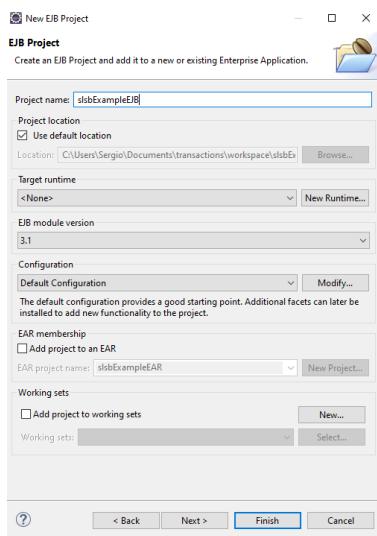


Figure 59

2.- Copy the content of \examples\Chapter3\<example name>\src\model to EJB project into the ejbModule folder. Make sure the package name matches the one in the .java, in this example this is "examples.stateless". The class HelloServiceBean will still have an error but it will be resolved following the steps.

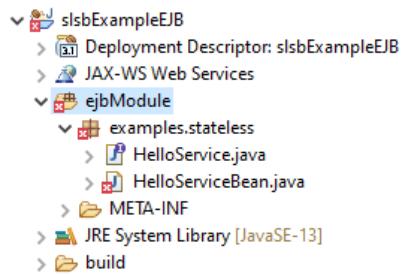


Figure 60

3.- Add the Target Runtime, right click on the EJB project and select “properties”, on properties window select “Targeted Runtimes” and click new. (If you already have create the Tomee server in the previous steps of the guide, you can skip to step 4).

On the new window expand “apache”, and select the latest version, click next and make sure that the installation directory is pointing to your TomEE installation. Click on “Show all runtimes” to see it appears ok. Click finish and the apply.

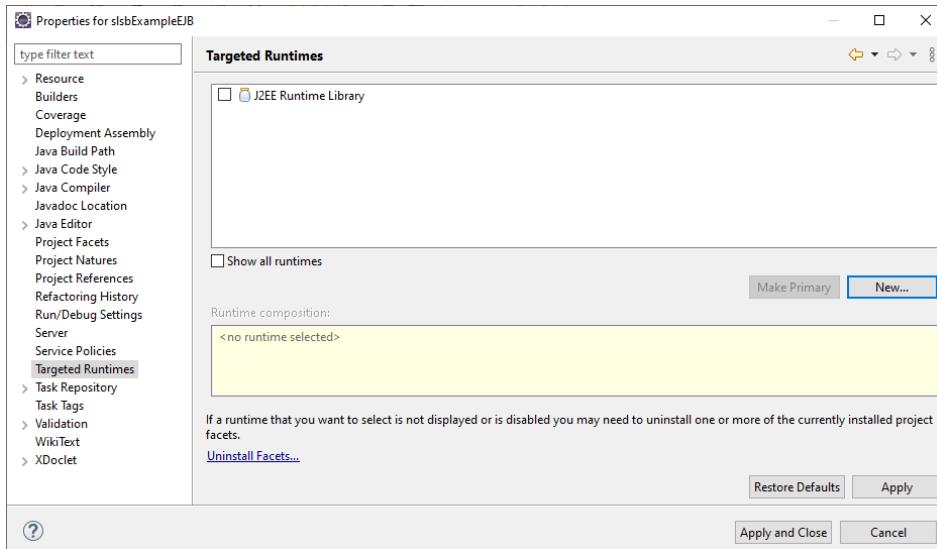


Figure 61

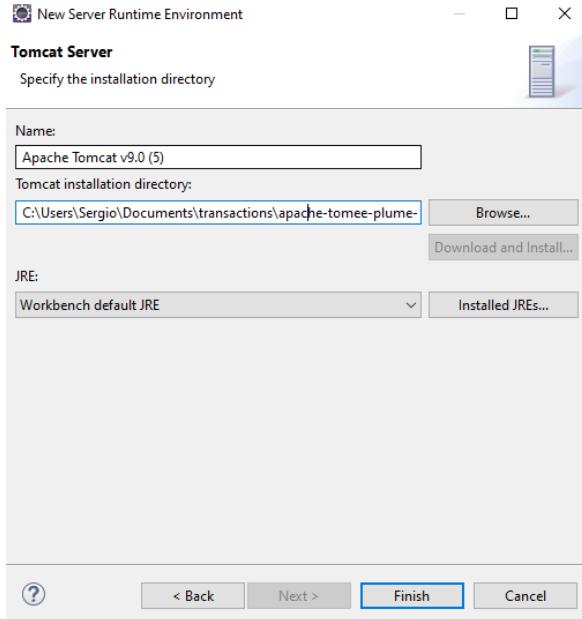


Figure 62

4.-On the properties windows go to “build path”, go to the “Libraries” tab, select “Classpath”, and click on “Add Library...”, on the new window select “Server Runtime”, click next, and select your TomEE server from the library list. Click finish and the apply and close.

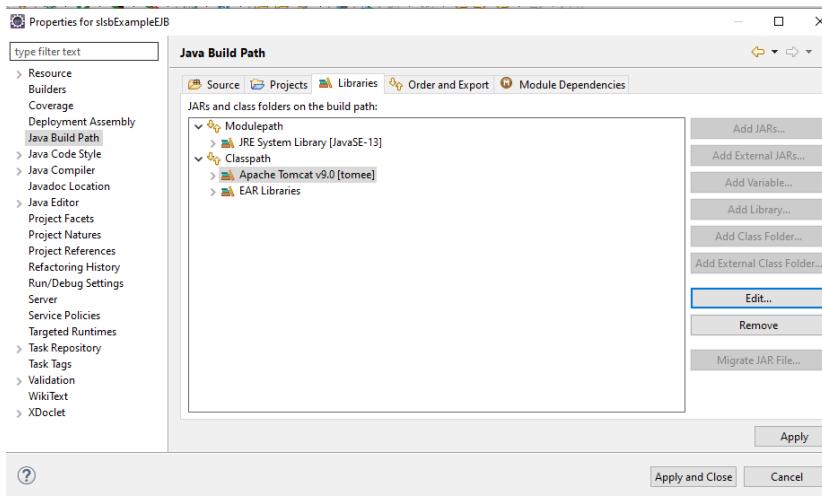


Figure 63

Web Project

1.- Create a new Dynamic Web project

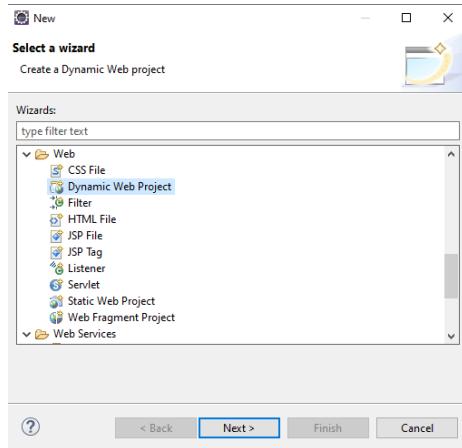


Figure 64

Give it a new with suffix WEB to make it easy to identify. Select the target runtime as your TomEE installation.

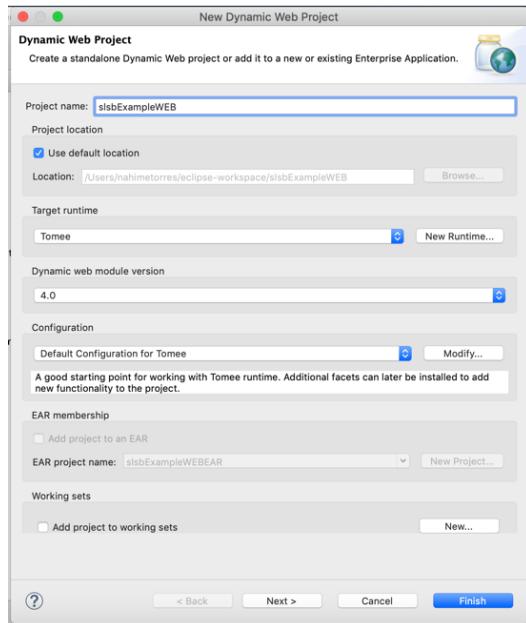


Figure 65

Click next 2 times, and on the Web module screen check the “Generate web.xml” option and click finish.

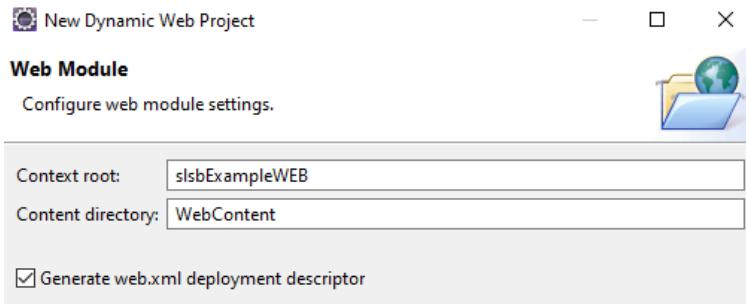


Figure 66

2.- Copy the content of \examples\Chapter3\<example name>\src\servlet to the Web project, into the src folder.

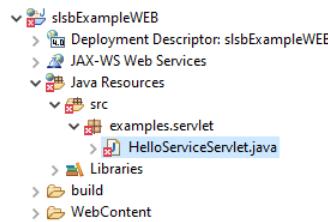


Figure 67

3.- Right click on the Web project and select “Properties”, go the “Deployment Assembly” option. Click the “Add...” button, on the selection window choose “Project”, click next and select your EJB project. Click ok, and then Apply and close.

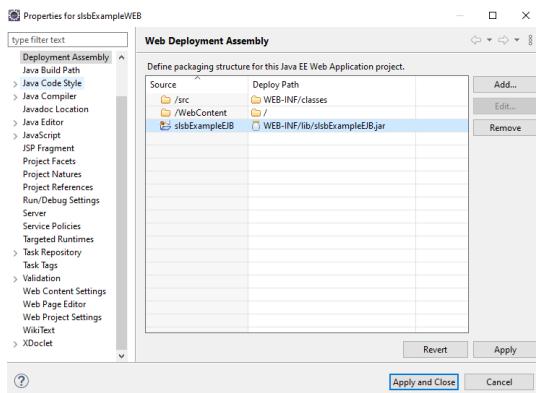
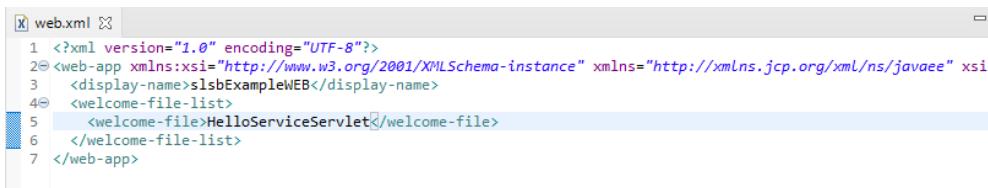


Figure 68

4.- On the web project open the web.xml file (WebContent/WEB-INF/web.xml), edit the “welcome-file” tag to content only the name of your servlet. As shown here:



```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.jcp.org/xml/ns/javaee" xsi
3   <display-name>slsbExampleWEB</display-name>
4 <welcome-file-list>
5   <welcome-file>HelloServiceServlet</welcome-file>
6 </welcome-file-list>
7 </web-app>
```

5.- Run the example by right clicking on the web project, select “Run as...”, and “Run on Server” then click OK. TomEE will start, a web browser will open with the main page of the web application.



Figure 69

Note: Consider that the Database evolves with the examples in the book, you will need to add tables as the example requires them.

Running Chapter 3 example 16 and onwards

From example 16 of chapter 3, and the rest of the example using JPA, an extra step is needed for the EJB project. The JPA Facet need to be included.

1.-Follow the steps explained in the previous section (

[EJB project](#), up to step 4.

2.-Right click on the EJB project and select “properties”. On the properties window select “Project Facets”, click on the JPA checkbox.

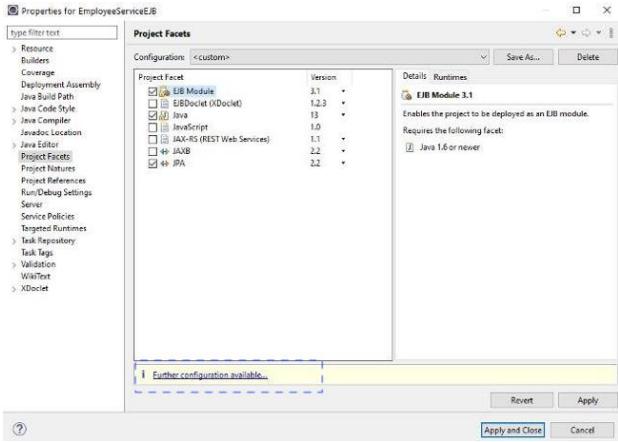


Figure 70

3.- A message indicating that further configuration is needed will appear, if it is your very first JPA project you will need to provide the JPA implementation library, else click apply and proceed to step 7.

If further configuration message does not appear:

3.1 Unselect JPA facet and click Apply and Close

3.2 Open again this section, select JPA facet, it should appear now

4.- Click on the message, new window will open, on the “JPA implementation” section click on the “download library...” button and select the latest version of EclipseLink (2.5.2). Click next, accept the terms and click finish.

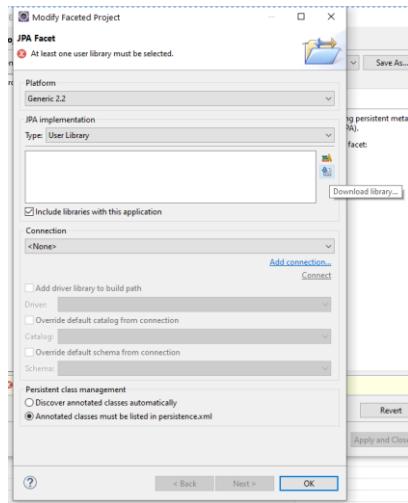


Figure 71

Note: If there's an error while downloading the libraries you will have to add it manually. Follow these steps:

4.1 Download the library manually from this link (we recommend 2.5.2)

<https://www.eclipse.org/eclipselink/downloads/>

4.2 Click in the icon to manage libraries

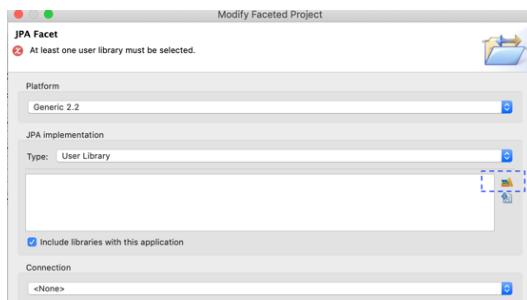


Figure 72

4.3 Click on New and name it eclipseLink (Check the System Library item)

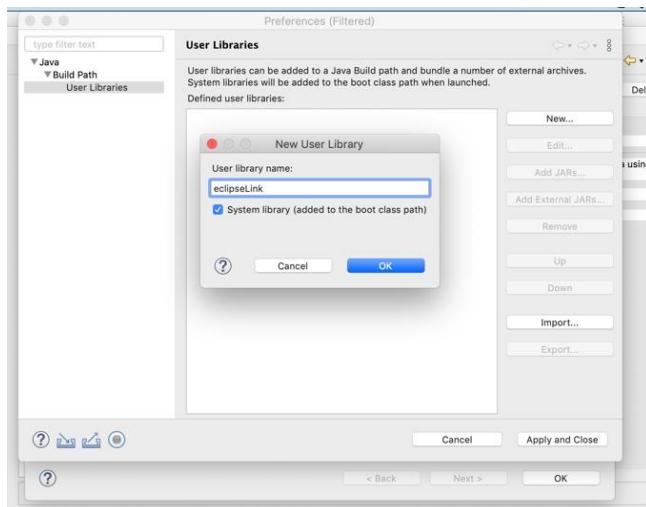


Figure 73

4.4 Select the library and click on Add External JARs

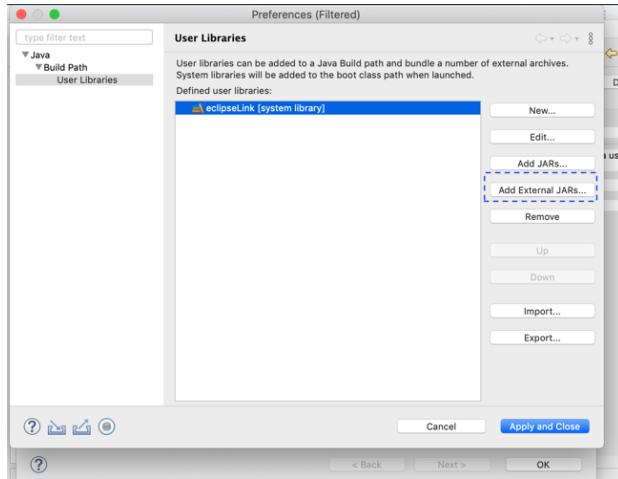


Figure 74

4.5 Import first the `eclipseLink.jar` from the downloaded folder inside `/jlib/` and then the several `.jar` inside `/jlib/jpa/*.jar`

4.6 Once the jars are added, click Apply and Close

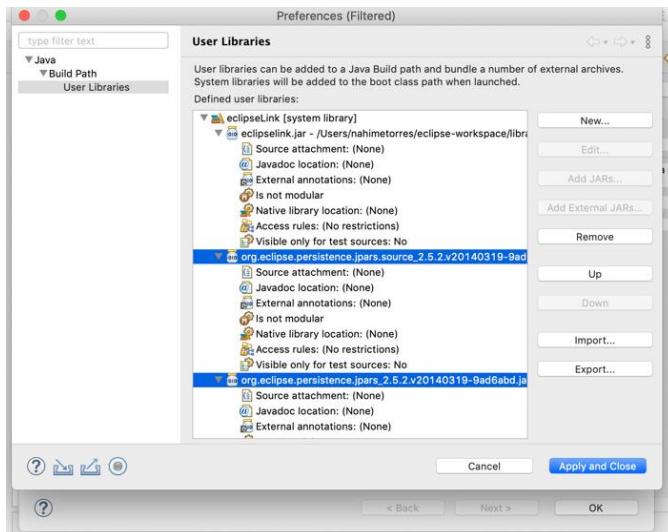


Figure 75

4.7 The library will appear now, select it

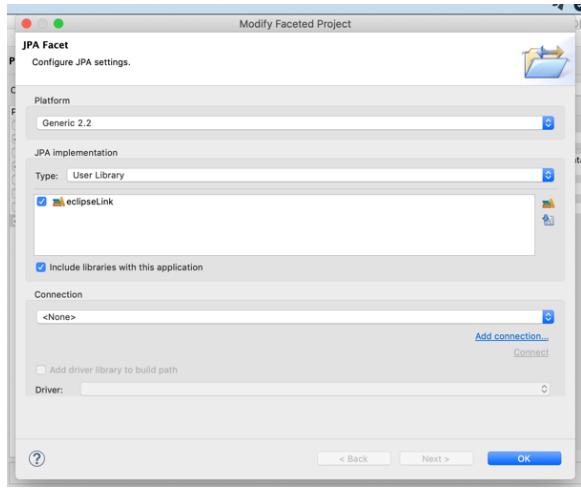


Figure 76

5. Once the library is installed you need to provide the information of the DB that you want to use (the schema must exists you can follow 5.4 Create a new database). Click on “Add Connection...”. Provide the data for your connection, and test it using the button “Test connection”. If everything is correct click finish.

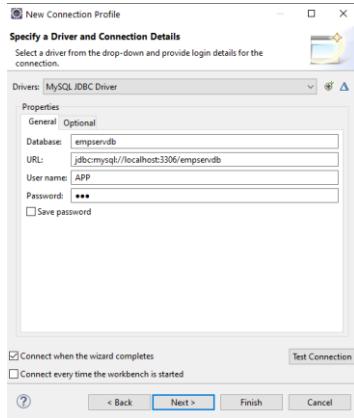


Figure 77

Possible troubleshooting:

5.1 In the drivers make sure you have one selected, otherwise click to add a new one

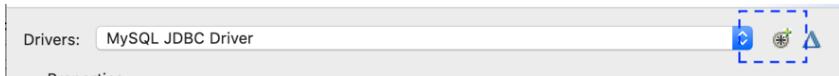


Figure 78

5.2 Choose a driver (5.1 systemversion is ok)

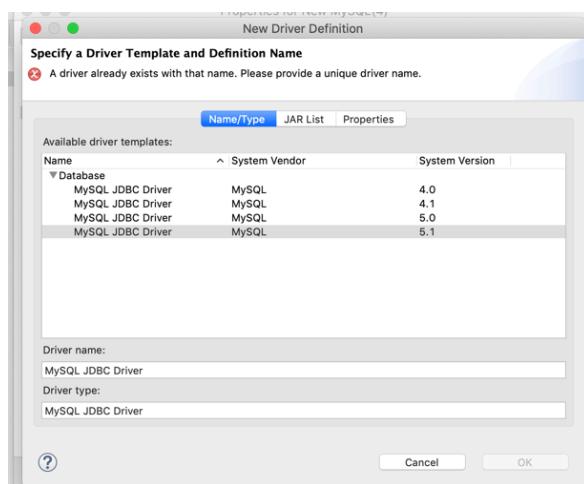


Figure 79

5.3 In the JAR list tab click on "Add JAR/Zip" and import your mysql.jar downloaded in previous steps

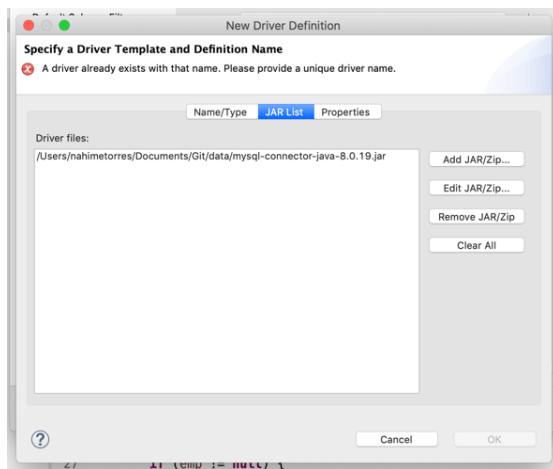


Figure 80

6. Check the “Add library to the build path” option on the screen. And click OK.

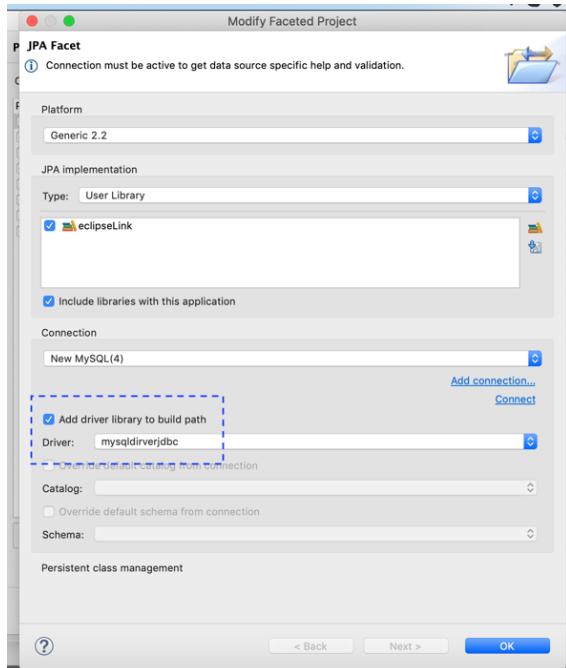


Figure 81

7. In your server Folder, open the Tomee instance created in precedent steps, look for the tomee.xml and add the Resource Configuration for the db you are using. (Be sure to have the connector/jar inside the {tomEE installation path}/lib folder)

Note: if the file does not exist, you can create it and insert this content with your modifications

```
<?xml version="1.0" encoding="UTF-8"?>
<tomee>
<Resource id="EmployeeTestResource" type="DataSource">
    <JdbcDriver com.mysql.cj.jdbc.Driver />
    <JdbcUrl jdbc:mysql://localhost:3306/EmpServ />
    <UserName piero />
    <Password fraternali />
</Resource>
</tomee>
```

Figure 82

```

<?xml version="1.0" encoding="UTF-8"?>
<tomee>
<Resource id="EmployeeTestResource" type="DataSource">
JdbcDriver com.mysql.cj.jdbc.Driver
JdbcUrl jdbc:mysql://localhost:3306/EmpServDB
UserName YOUR_USER
Password YOUR_PASSWORD
</Resource>
</tomee>

```

9. – In the EJB project a JPA Content section was created on the project, expand it and right click on the persistence.xml, and click on Open.

Select the Connection tab and choose Transaction type “Default (JTA)” and in JTA Data Source, put the name of the resource ID you created in the tome.xml file (in this case “EmployeeTestResource”)

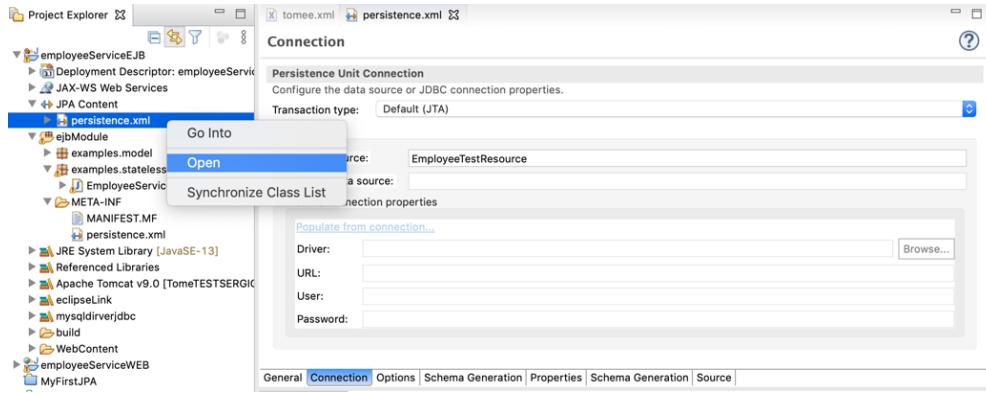


Figure 83

10. – Also in the EJB project expand the JPA Content and right click on the persistence.xml, and click on synchronize Class list. All errors should disappear after the process is finished.

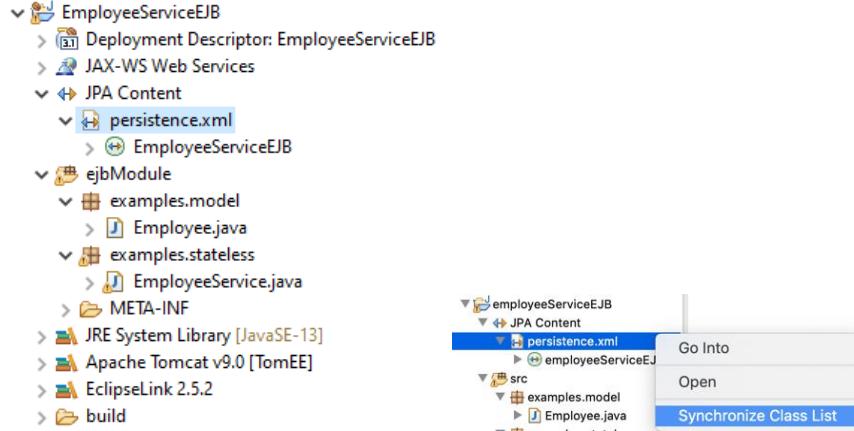


Figure 83

11.- Continue with the steps on for your web application, as explained in section "[Web Project](#)" and test your application.

Note: Consider that the Database evolves with the examples in the book, you will need to add tables as the example requires them.

9 Importing the “Expense Management” Example

You will be provided with a zip file containing an EJB project, a WEB Project, and a SQL file containing the database structure. Unzip the file to any location on your PC.

Create the data base

1. Open the data base structure (.sql file) with a text editor and copy the content.
2. Open Workbench and connect to your local instance.
3. Open a query tab, if not open yet, and paste the sql code.
4. Click on the thunder icon () to execute the script, on the lower part of the screen you will see the result of the execution
5. On the “Schemas” panel click the refresh icon.
6. The “db_expense_management” schema should appear contains the tables and data for the exercise.

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' tree view is open, showing a single schema named 'db_expense_management'. The main panel displays a SQL editor titled 'SQL File 15*' with the following content:

```

CREATE DATABASE IF NOT EXISTS `db_expense_management` /*!40100 DEFAULT CHARACTER SET
USE `db_expense_management`;
-- MySQL dump 10.13 Distrib 8.0.19, for Win64 (x86_64)
-- 
-- Host: localhost Database: db_gestione_spese

```

Figure 84

Import the projects to eclipse

Make sure you have already configured eclipse and tome according to the instructions provided in this guide.

1. Open Eclipse
2. Go to the option *File -> Import...* a window will open, navigate to *general -> existing projects into workspace*.

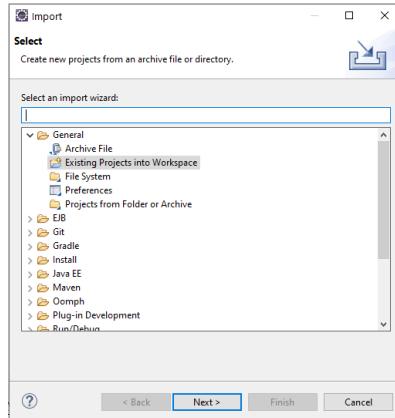


Figure 85

Click *Next*, in the new window click on *Browse...* and navigate to the folder containing the 2 extracted projects from the zip file, click *ok*, and the EJB and WEB project should appear under projects.

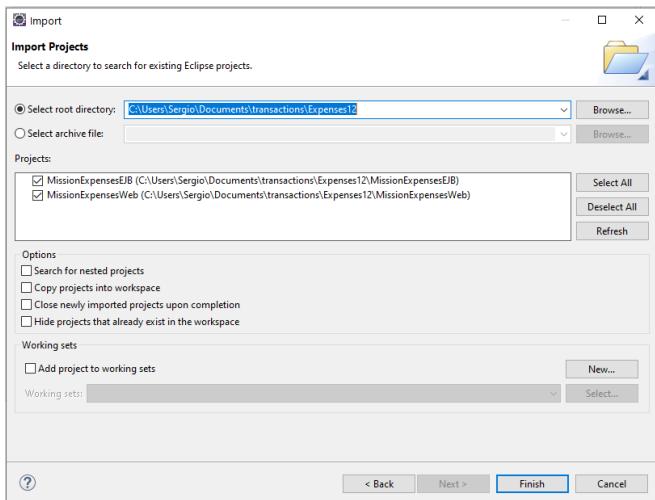


Figure 86

3. Click finish, the projects should appear normally on you project explorer.

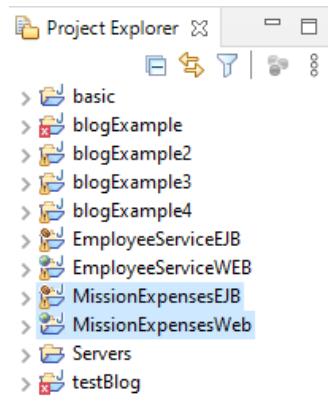


Figure 87

4. Right click on the web project and select properties. Navigate to Deployment assembly and verify that the EJB project is included. Else add it as explained in the previous section. Then click apply.

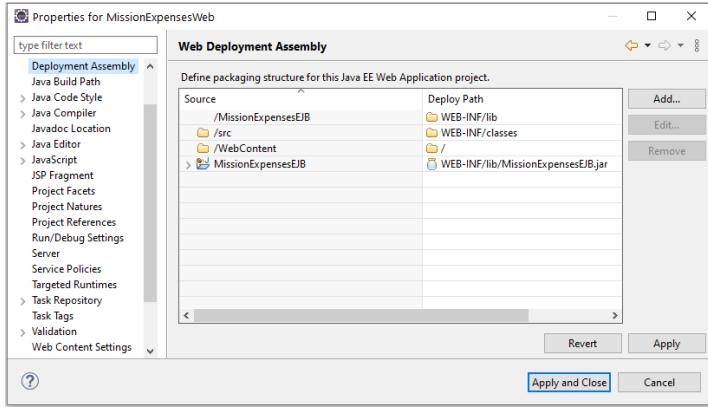


Figure 88

5. Add the resource in the TomEE server instance in the file `tomee.xml` and modify the `persistence.xml` of the EJB part to point to these new resources as in the previous examples. (Be sure to have put the connector/jar inside the `{tomEE installation path}/lib` folder)
6. For both projects, check in the Java Build Path all libraries are correctly configured (e.g. the server runtime library or the eclipse link one)
7. To execute the project, right click on the web project and select *run on server* and run it

References

- Eclipse documentation on Web Projects:
<https://help.eclipse.org/2018-12/topic/org.eclipse.wst.doc.user/topics/overview.html?cp=99>
- Eclipse Link JPA documentation:
<http://wiki.eclipse.org/EclipseLink/Examples/JPA>