

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..... | 7 |
| 2.1.4 Check if everything went fine | 7 |
| 2.2 Installing Eclipse | 8 |
| 2.3 Installing TomEE | 10 |
| Windows: | 10 |
| OS X | 11 |
| Linux..... | 12 |
| 2.4 Installing MySQL Server & Workbench..... | 12 |
| Windows..... | 12 |
| OS X | 13 |
| Linux..... | 16 |
| 2.4.1 Starting MySQL Server..... | 16 |
| 3 Configuration | 17 |
| 3.1 Setting up the latest JDK with Eclipse..... | 17 |
| 3.2 Adding TomEE to Eclipse | 19 |
| 3.3 Configuring TomEE..... | 22 |
| 3.4 Starting TomEE..... | 24 |
| OS X | 25 |
| 4 Publishing content | 26 |
| 4.1 Creating a dynamic web project..... | 26 |
| 4.2 Writing the Web component | 27 |
| 4.3 Testing your application..... | 30 |
| 5 Connecting with JDBC..... | 32 |
| 5.1 Add mysql connector to the project..... | 32 |

| | |
|---|----|
| 5.2 Installing Connector/J in Tomee | 33 |
| 5.3 Create a new servlet..... | 36 |
| 5.4 Create a new database | 37 |
| 5.5 Run the project..... | 38 |
| 6 Stopping Tomee..... | 38 |
| 7 Troubleshooting | 38 |
| 7.1 MySQL TimeZone Error..... | 38 |
| 8 Running the Book examples:..... | 38 |
| Download the Book examples..... | 38 |
| Running Chapter 2 – employeeService example..... | 39 |
| Running Chapter 3 examples (1-15): | 47 |
| EJB project..... | 47 |
| Web Project..... | 50 |
| Running Chapter 3 example 16 and onwards..... | 53 |
| 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 only:

- Oracle **JDK** and **JRE**, last version Java 16:
<https://www.oracle.com/java/technologies/javase/jdk16-archive-downloads.html>

OS X only:

- MySQL Community **server** 8.0.26:
<http://dev.mysql.com/downloads/mysql/>
- MySQL **Workbench** 8.0.26:
<http://dev.mysql.com/downloads/workbench/>

Windows only:

- MySQL **Installer** 8.0.26: <https://dev.mysql.com/downloads/installer/>
Note: MySQL Installer is 32-bit but will install both 32-bit and 64-bit binaries.

Windows & OS X & Linux:

- Eclipse IDE for Java Enterprise developers (2021-09): <https://www.eclipse.org/downloads/packages/>

Eclipse IDE for Enterprise Java and Web Developers

517 MB 107,099 DOWNLOADS

Tools for developers working with Java and Web applications, including a Java IDE, tools for JavaScript, TypeScript, JavaServer Pages and Faces, Yaml, Markdown, Web Services, JPA and Data Tools, Maven and Gradle, Git, and more.

Click [here](#) to file a bug against Eclipse Web Tools Platform.
Click [here](#) to file a bug against Eclipse Platform.
Click [here](#) to file a bug against Maven integration for web projects.
Click [here](#) to report an issue against Eclipse Wild Web Developer (incubating).

Windows x86_64
macOS x86_64
Linux x86_64 | AArch64

- Apache **TomEE** 8.0.8 (plume): <http://tomee.apache.org/download.html>

| | | | | |
|-------------------------------------|-------|-------------|-------|-----------------------------------|
| TomEE Plume TAR.GZ | 8.0.8 | 31 Aug 2021 | 73 MB | |
| TomEE Plume ZIP | 8.0.8 | 31 Aug 2021 | 73 MB | |

THE APACHE SOFTWARE FOUNDATION ESTABLISHED 1999

COMMUNITY-LED DEVELOPMENT "THE APACHE WAY"

Projects ▾ People ▾ Community ▾ License ▾ Sponsors ▾

We suggest the following mirror site for your download:
<https://dlcdn.apache.org/tomee/tomee-8.0.8/apache-tomee-8.0.8-plume.zip>

Other mirror sites are suggested below.



- **MySQL** connector 8.0.26: <https://dev.mysql.com/downloads/connector/j/> “Platform Independent (Architecture Independent), ZIP or TAR Archive”.
- **Eclipse Link** 2.7.9 (installer zip) <https://www.eclipse.org/eclipselink/downloads/>

Virtual Machine:

In case you do not want to install all the environment, you can use this linux virtual machine together with virtualbox (at least 4GB of ram, 2 core cpu and 20GB available disk space):

- https://polimi365-my.sharepoint.com/:u/g/personal/10460101_polimi_it/EfqV09-gRPFNiCm7Hz5nUDwBR4UZLnm2--jS4ifoa9knLA?e=kHYdbt
- <https://www.virtualbox.org>

2 Installation

2.1 Installing the latest JRE and JDK

Use the link of [Section 1](#) (Software requirements) to download the right version.

The full 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.

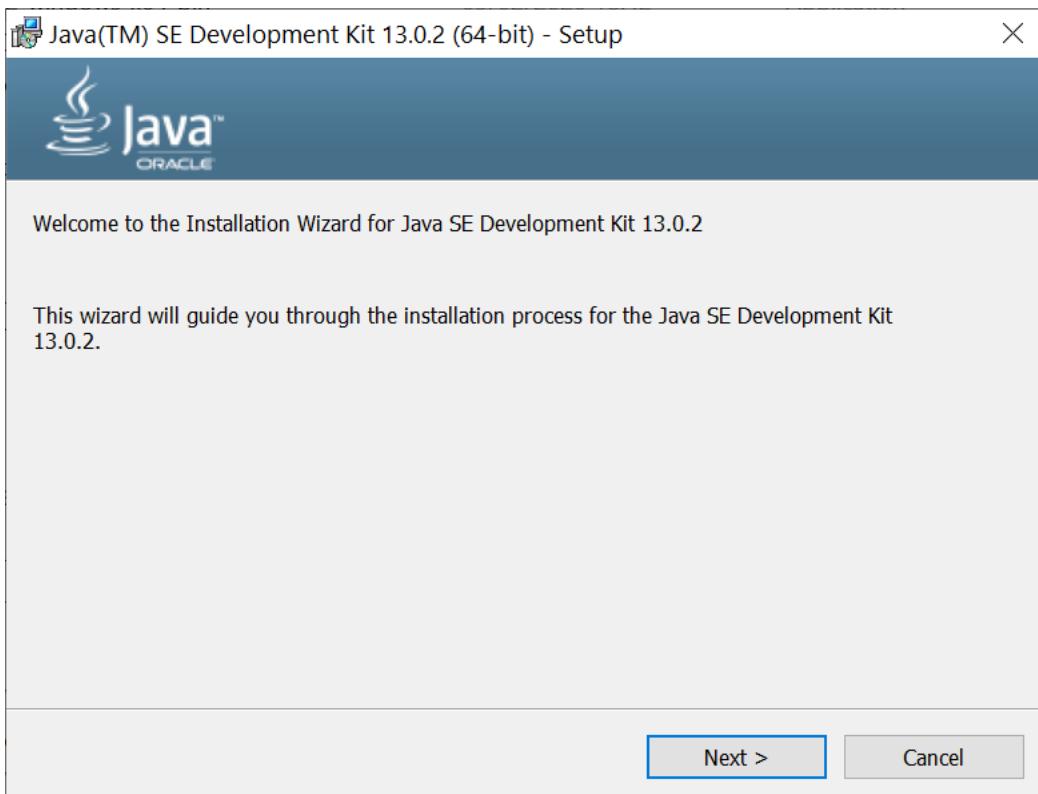


Figure 1: JDK installation wizard

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

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

When using Java SE 16 you don't need to set the JRE_HOME variable.

An example of environment variables is:

JAVA_HOME: C:\Program Files\Java\jdk-16.0.2

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**

(Your unique entries are something like "C:\windows\system32;C:\windows;etc..." that you already have in the PATH variable)

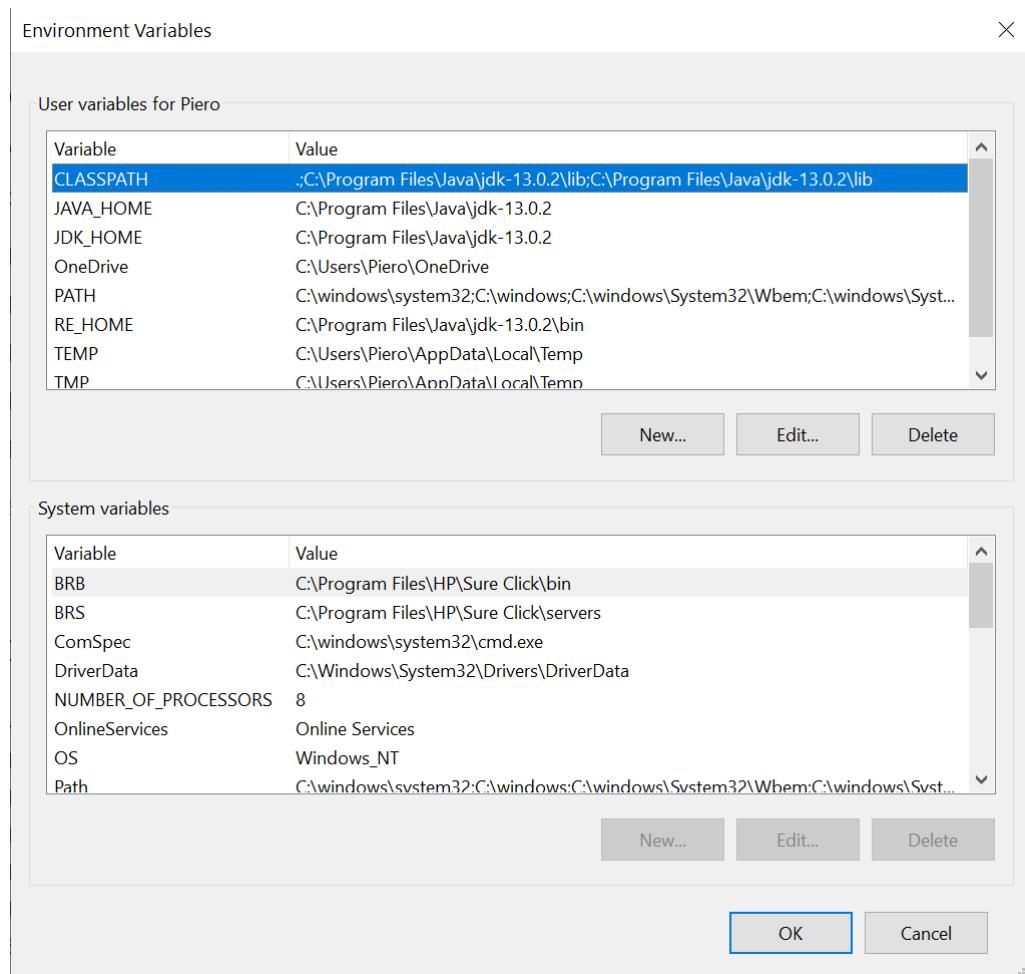


Figure 2: setting Windows environment variables

2.1.2 OSX

Download the dmg installer proceed as in Figure 3.

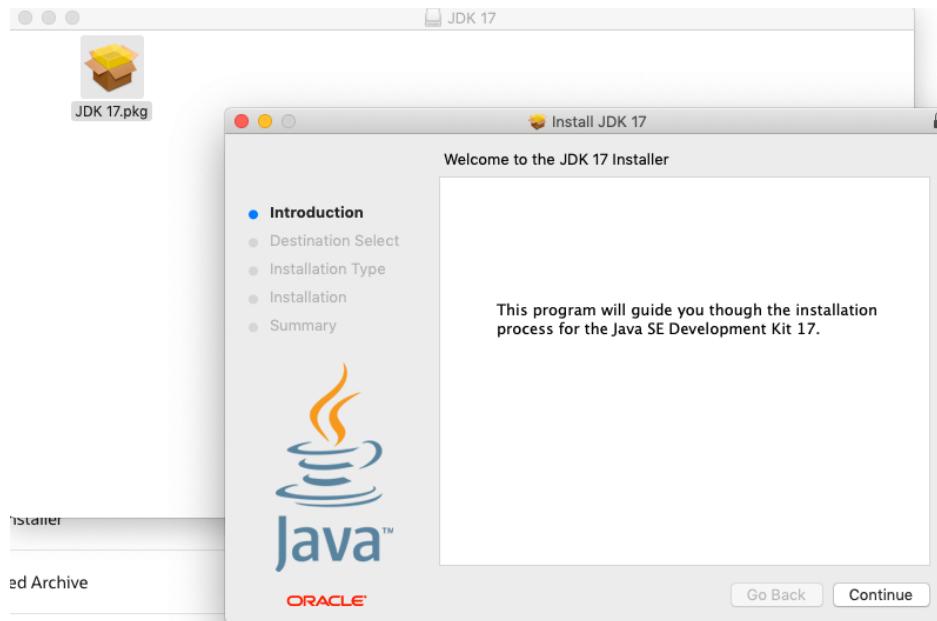


Figure 3: Installing the Oracle JDK in OS X

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-java16-installer
```

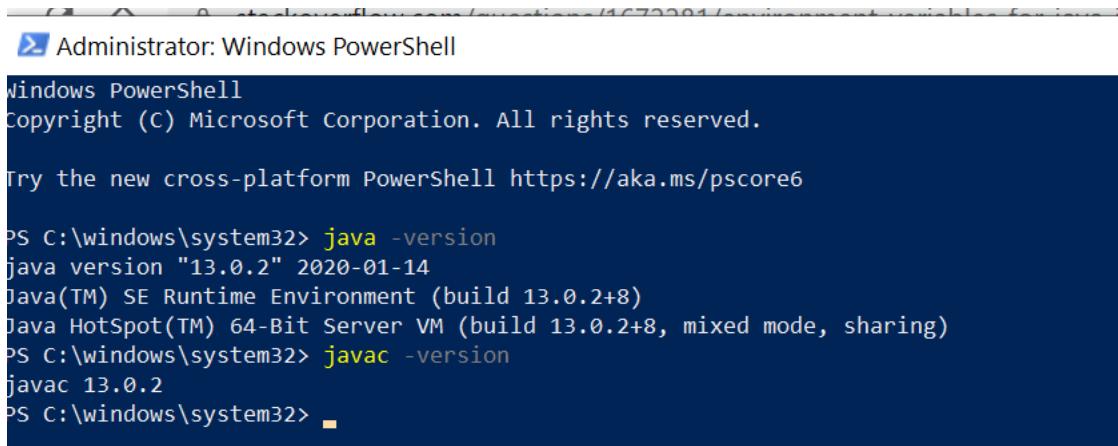
You will be prompted a message to confirm the Oracle license terms & agreement, accept them (use the tab button to navigate the options), and continue.

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



```
Last login: Mon Sep 20 10:30:01 on ttys003
[MBP-de-Nahime:~ nahimetrores$ java -version
java version "17" 2021-09-14 LTS
Java(TM) SE Runtime Environment (build 17+35-LTS-2724)
Java HotSpot(TM) 64-Bit Server VM (build 17+35-LTS-2724, mixed mode, sharing)
MBP-de-Nahime:~ nahimetrores$ ]
```

Figure 5: checking the version of Java in Mac OS

2.2 Installing Eclipse

For all platforms:

- Use the link of [Section 1](#) (Software requirements) to download the right version.
- 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

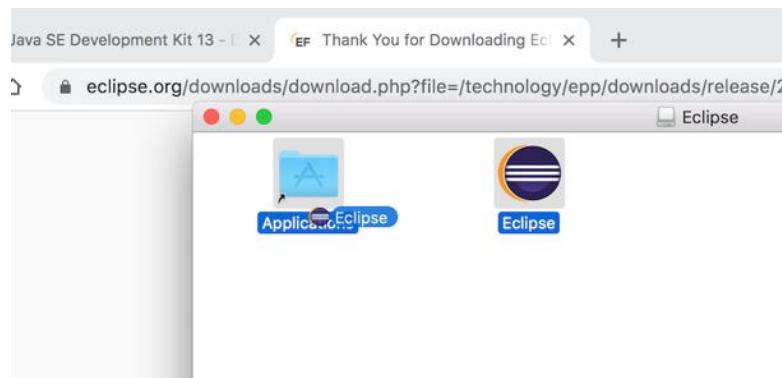


Figure 6: Move Eclipse to Apps folder

If you downloaded the installer (.dmg) instead of the zip:

- Open the .dmg and in the appearing window open the installer. Then proceed to select Eclipse IDE for Enterprise Java and Web Developers. Then if a message appears asking to install an unsigned jar in the eclipse folder, proceed to accept.

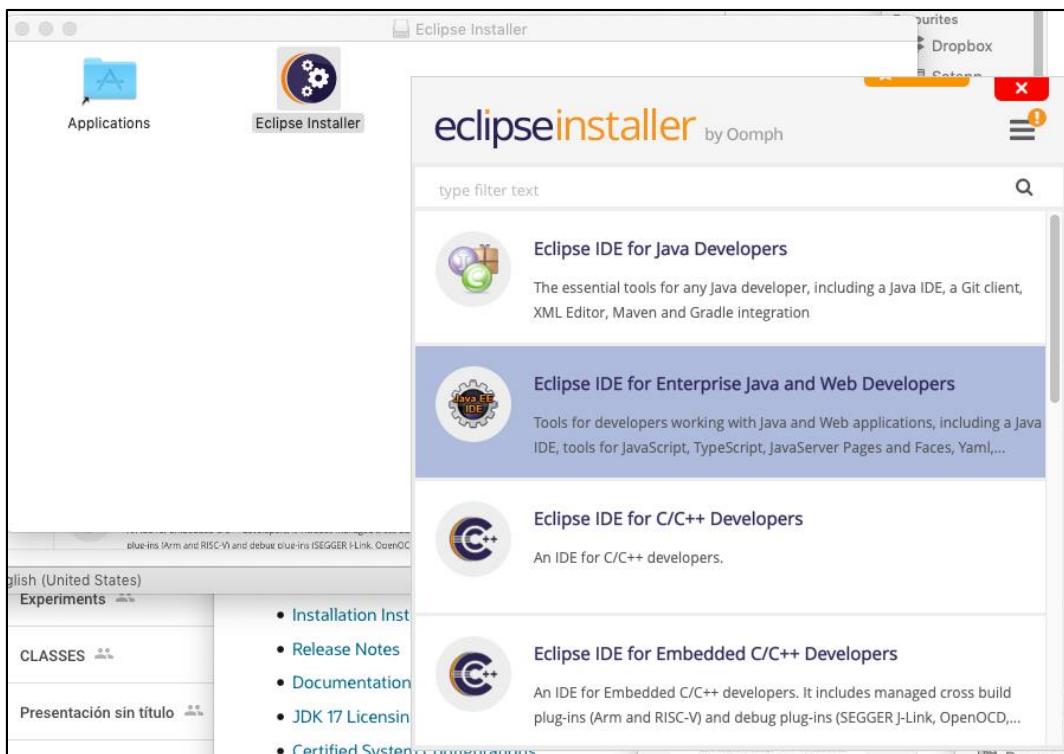




Figure 7: Select Eclipse IDE for Enterprise

For Linux:

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

2.3 Installing TomEE

Use the link of [Section 1](#) (Software requirements) to download the right version.

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-8.0.8\bin
- Alternatively, you can execute the file “startup.bat” on the same folder using the command line

```
[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.SessionIdGeneratorBase.createSecureRandom Creation of SecureRandom instance for session ID generation using [SHA1PRNG] took [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 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 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 09:31:31.050 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application directory [C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps\host-manager] has finished in [51] ms
06-Feb-2020 09:31:31.050 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 09:31:31.103 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application directory [C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps\manager] has finished in [57] ms
06-Feb-2020 09:31:31.108 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 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 09:31:31.170 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 8 the console of the Tomee Java web server

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

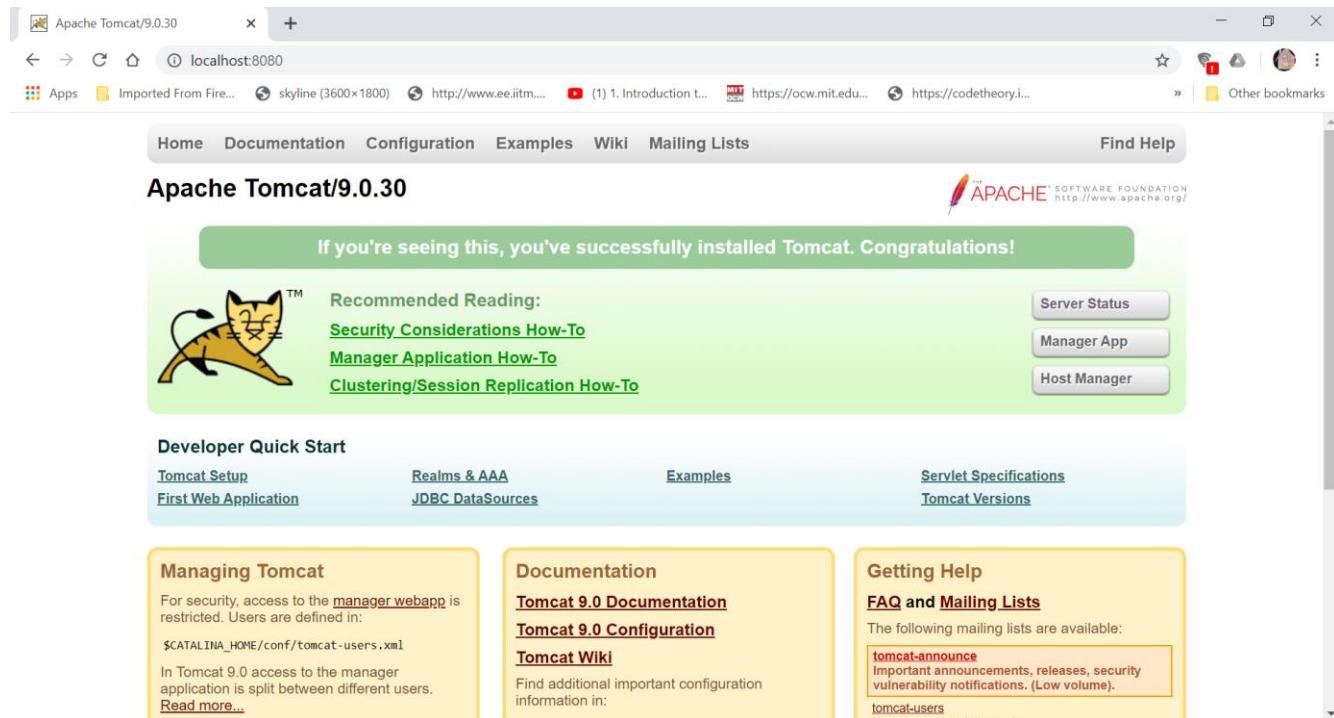


Figure 9: the welcome page of the Tomee Java Web server

OS X

- After unzipping, rename the folder to Tomee8 and copy the folder into the /Applications folder
- Move to Applications/Tomee8/bin and run the command ./startup.sh

```
[MBP-de-Nahime:bin nahimetrores$ ./startup.sh
Using CATALINA_BASE:   /Applications/Tomee8
Using CATALINA_HOME:   /Applications/Tomee8
Using CATALINA_TMPDIR: /Applications/Tomee8/temp
Using JRE_HOME:        /Library/Java/JavaVirtualMachines/jdk-17.jdk/Contents/Home
Using CLASSPATH:       /Applications/Tomee8/bin/bootstrap.jar:/Applications/Tomee8/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
```

Figure 10: 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 Tomee8/bin/*.sh
```

Linux

- Extract the content of the downloaded tomee file (e.g., tar xzf apache-tomee-plume-8.0.8.tar.gz) by opening a Terminal and typing:

```
tar xzf apache-tomee-plume-8.0.8.tar.gz
```

- Navigate to the extracted folder:

```
cd apache-tomee-plume-8.0.8/bin
```

- 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

Use the link of [Section 1](#) (Software requirements) to download the right version.

Windows

Start the MySQL Installer and install MySQL Community Server 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.

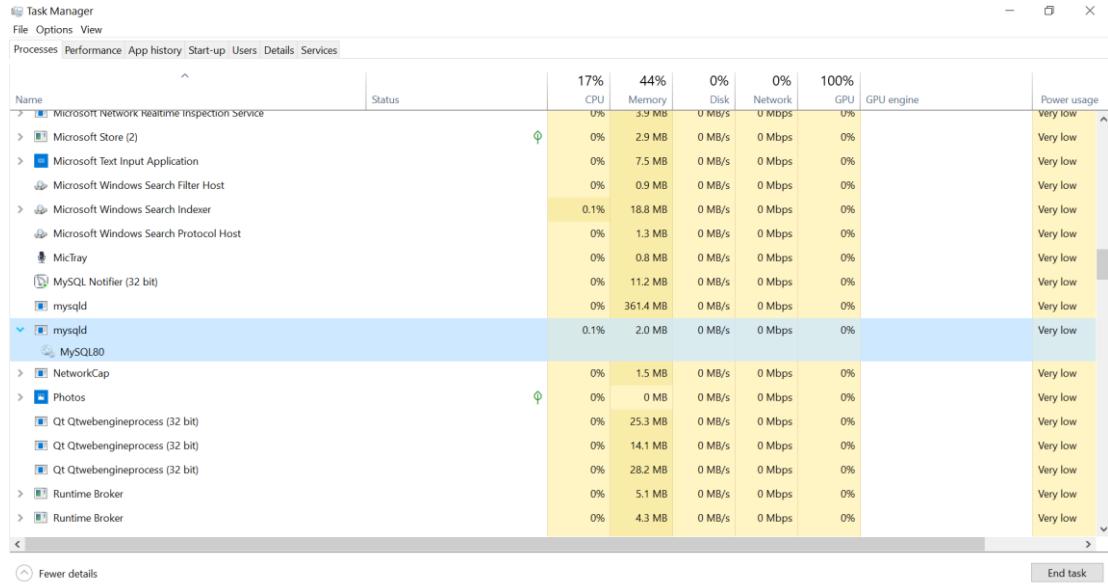


Figure 11: checking that the MYSQL service is running in the Control Panel -> Administrative Tools -> Services

Open the workbench and check the status of the server.

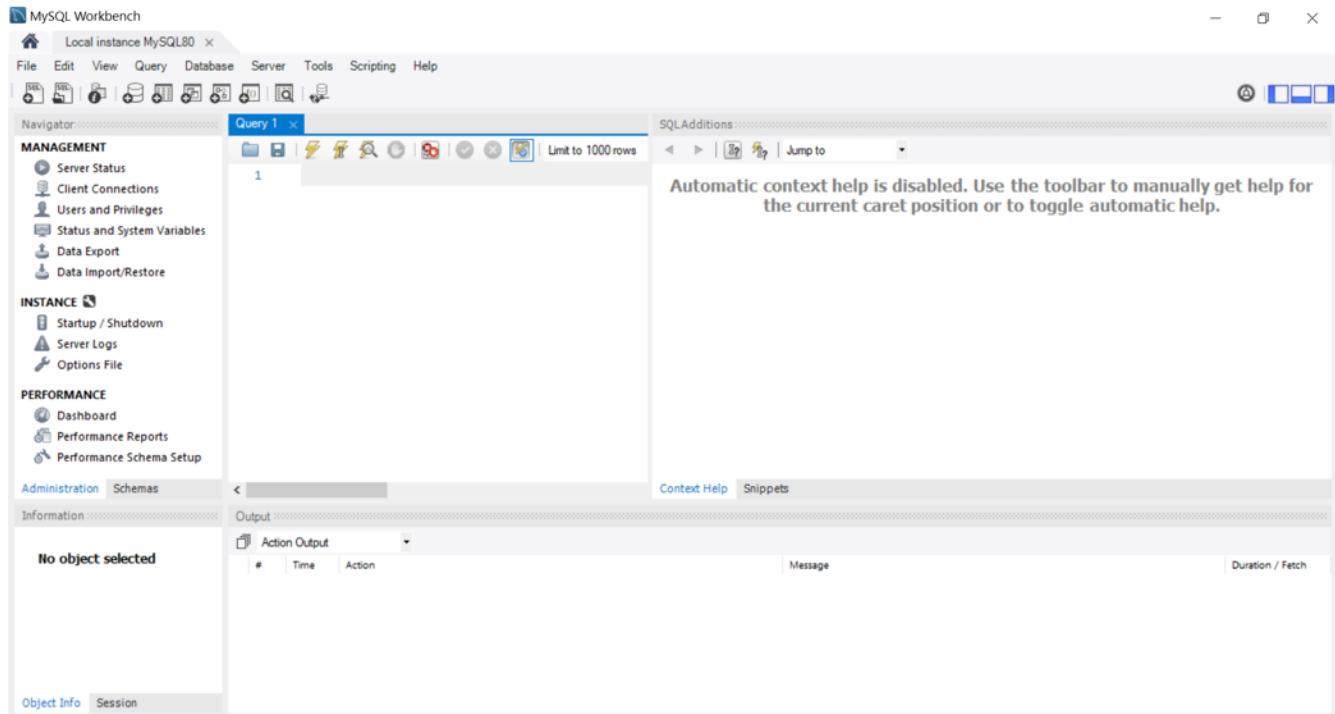


Figure 12: checking the server status in MySQL Workbench

OS X MySQL Community

- Install MySQL Community Server with all default configurations

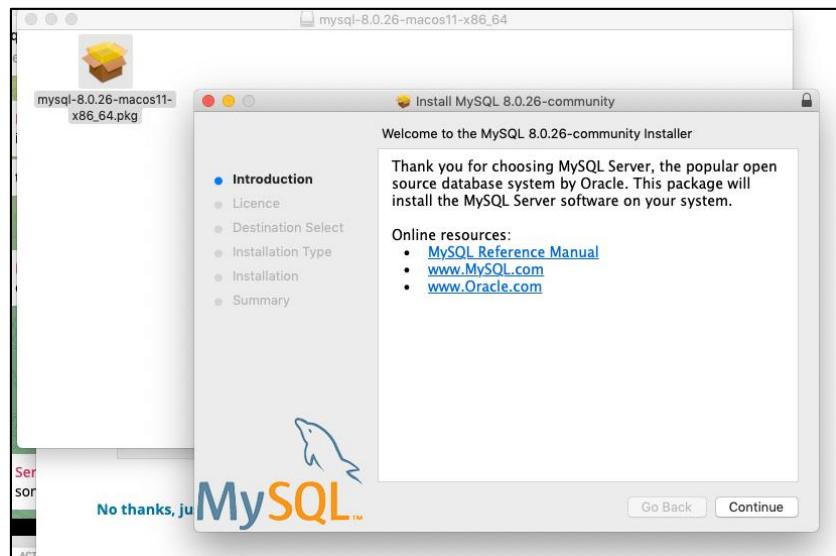


Figure 13 Installing MySQL in OS X

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

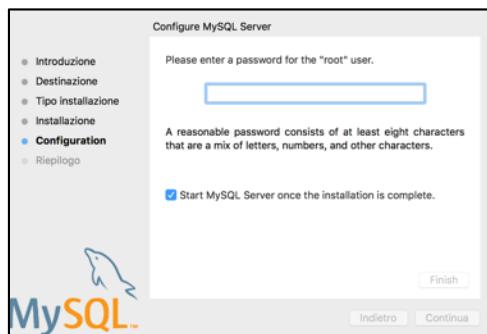


Figure 14 Installing MySQL in OS X

MySQL Workbench

- 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 an older version).

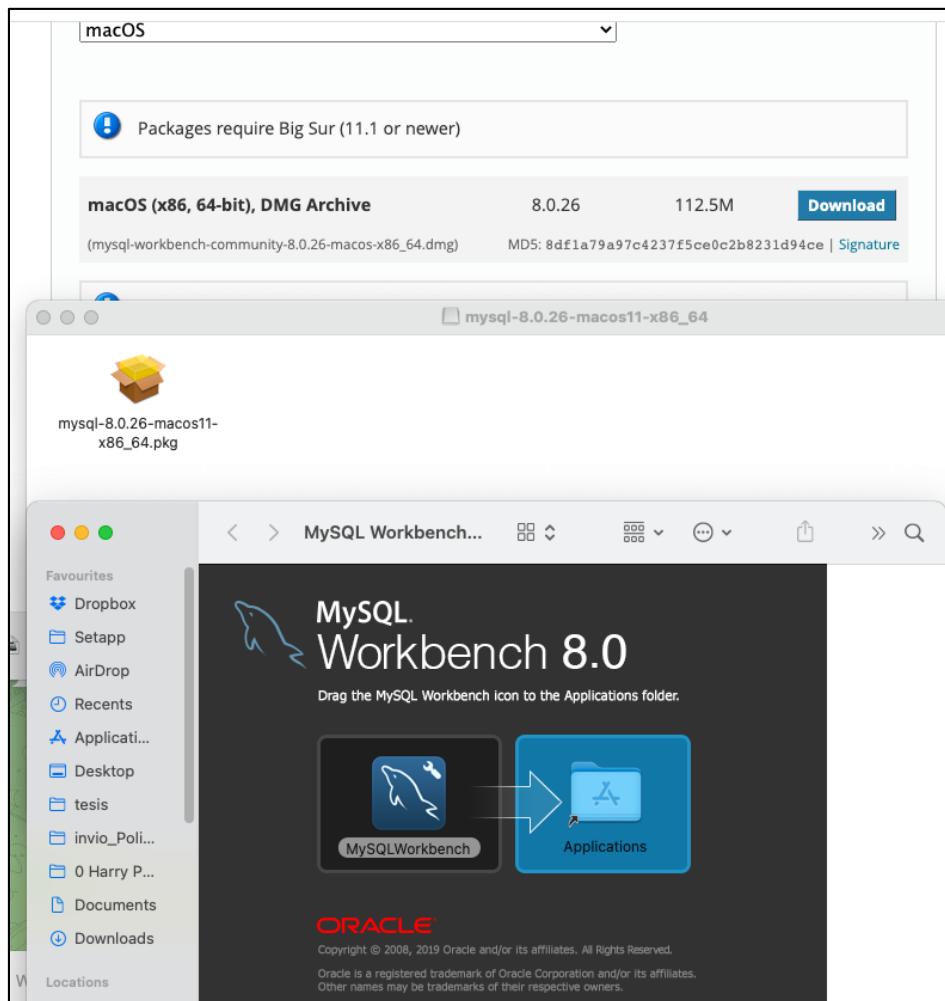


Figure 15 Installing MySQL Workbench in OS X

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

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

- If when you open MySQL workbench for the first time you encounter a “security issue”, go to Settings-> Security → Open anyway

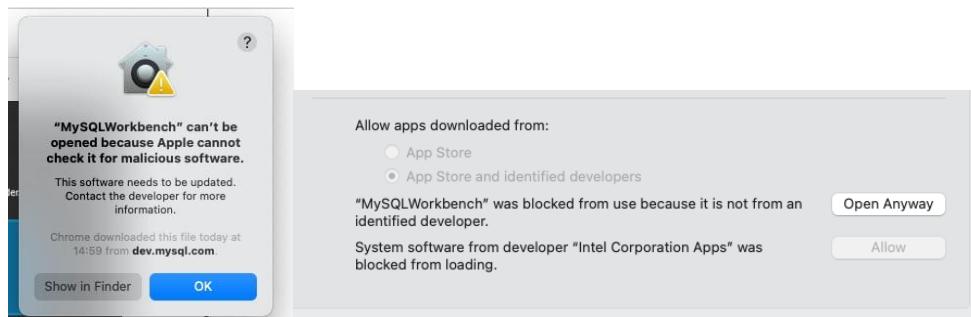


Figure 16 Security Issue MySQL Workbench in OS X

Linux

Open a terminal and type:

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

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.

If the installation does not ask you to set the root password, you can follow this guide to set it (where you have to replace *my_new_pass* with the password of your choice):

<https://askubuntu.com/a/1133325>

Once installation is completed, the service will start automatically. To verify the status of the service, execute the following:

```
sudo systemctl status mysql
```

To install mysql workbench type this in the terminal:

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

If mysql-workbench cannot be found, try following these steps:

- Download the deb file from this url: https://dev.mysql.com/get/mysql-apt-config_0.8.19-1_all.deb
- In the terminal navigate to your download directory and type:

```
sudo dpkg -i mysql-apt-config_0.5.3-1_all.deb
```

- In the new windows that appears, make sure that “mysql tools and connector” is enabled and you can disable “mysql server and cluster” by pressing enter and then selecting “none”. Next navigate to ok and press enter. Then update with:

```
sudo apt-get update
```

- Install mysql workbench with:

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

If some error occurs, follow the suggestions in the terminal. Usually another update is required or --fix-missing option must be put at the end of the last command.

MySQL Workbench application will be available in the app drawer.

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.

OSX

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 by restarting the computer).

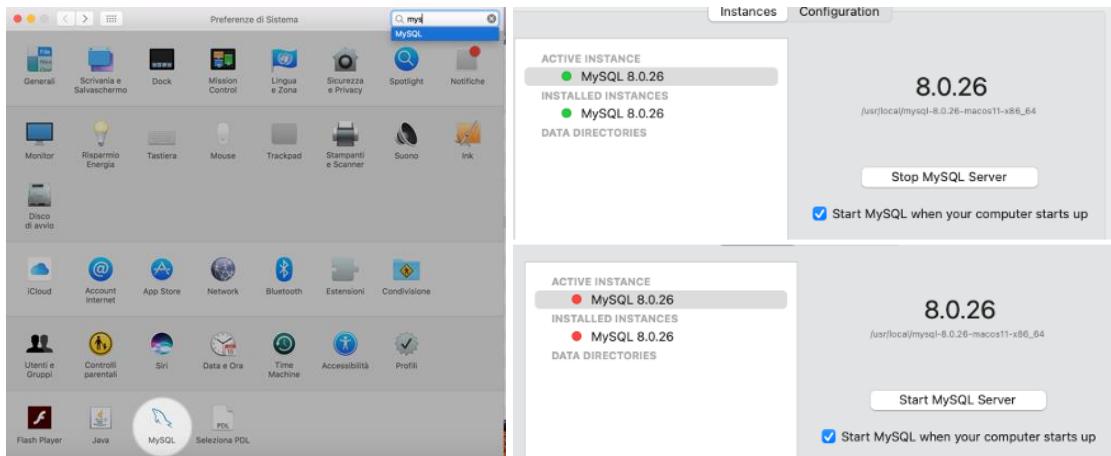


Figure 17 Start/Stop MySQL in MacOS X

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

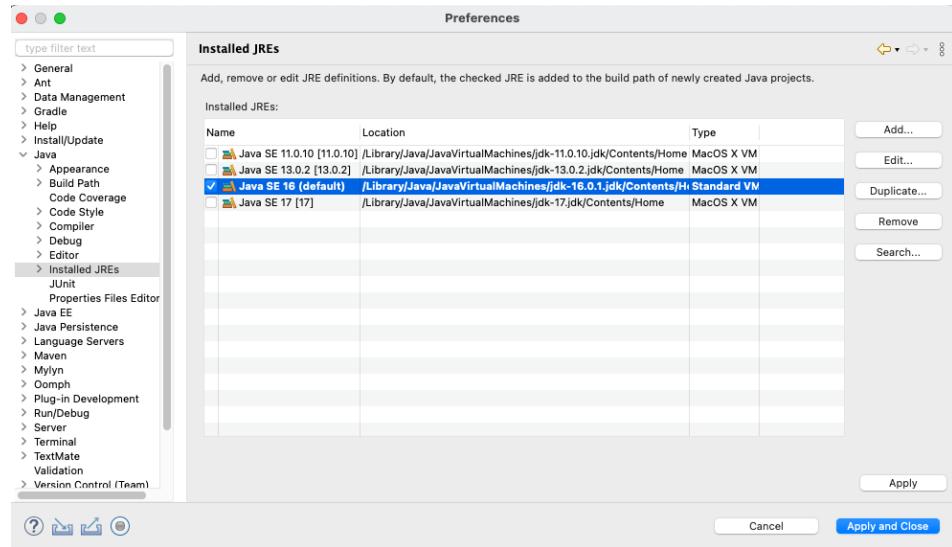


Figure 18 Checking the installed JREs in Eclipse

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-16.0.2.jdk/Contents/Home

Then select Java > Compiler and be sure that the “Compiler compliance level” value is set to 16 as in the following figure.

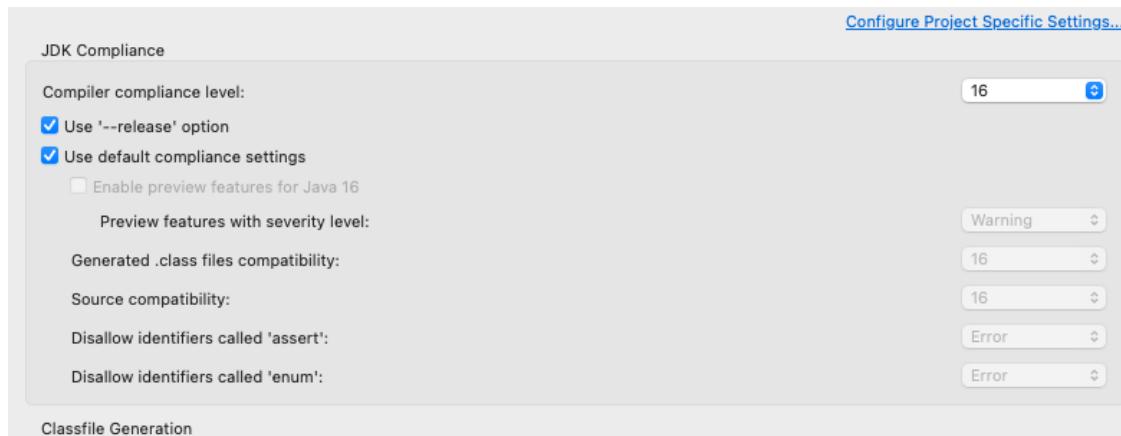


Figure 19 Setting the compiler compliance level in Eclipse

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 and select Apache -> Apache Tomcat 9.0

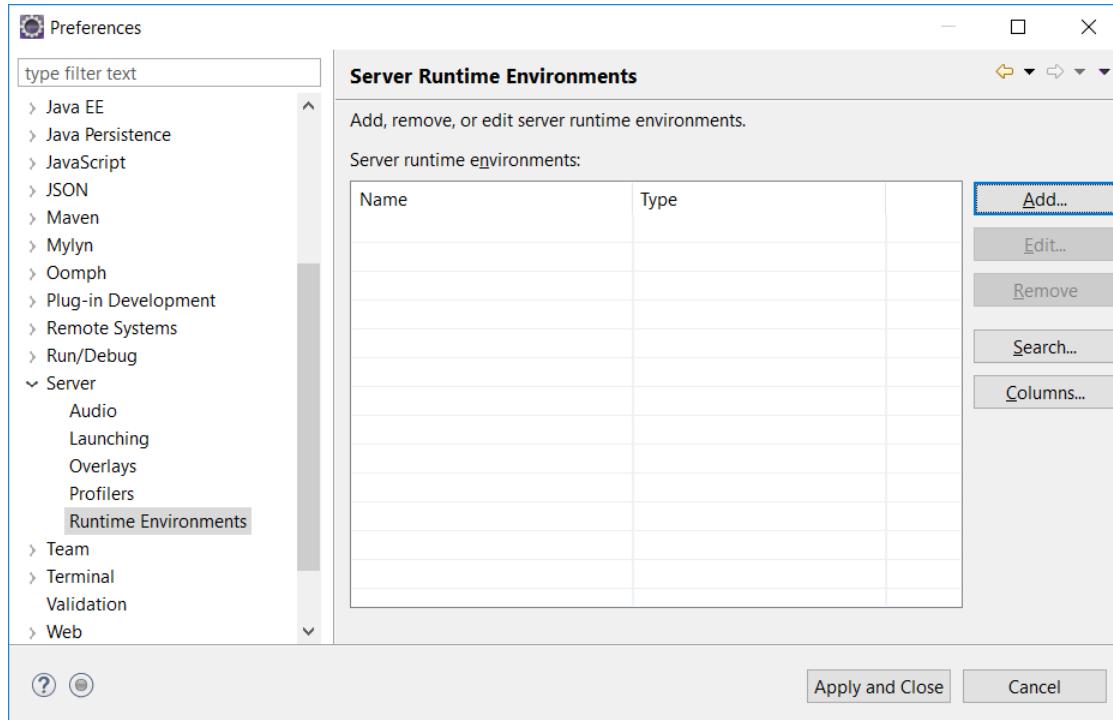


Figure 20 adding a runtime environment for Tomee

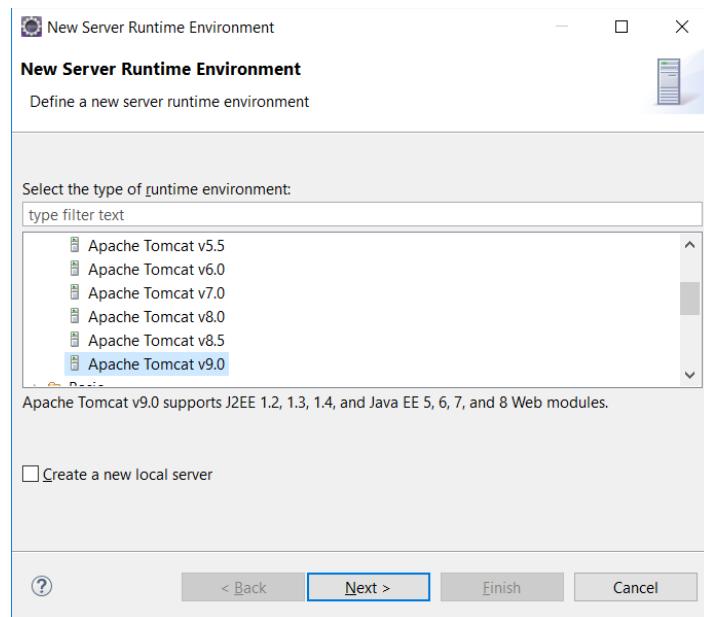


Figure 21 selecting the installed version of Tomee

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

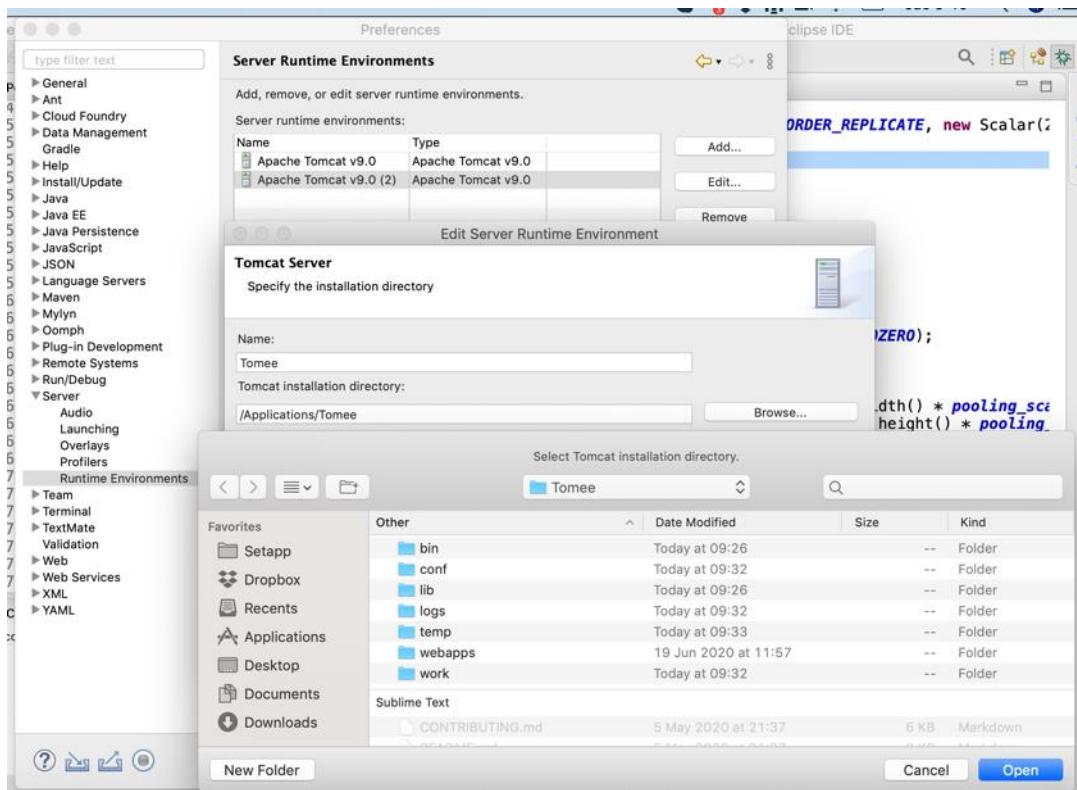


Figure 22 setting the TomEE folder

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

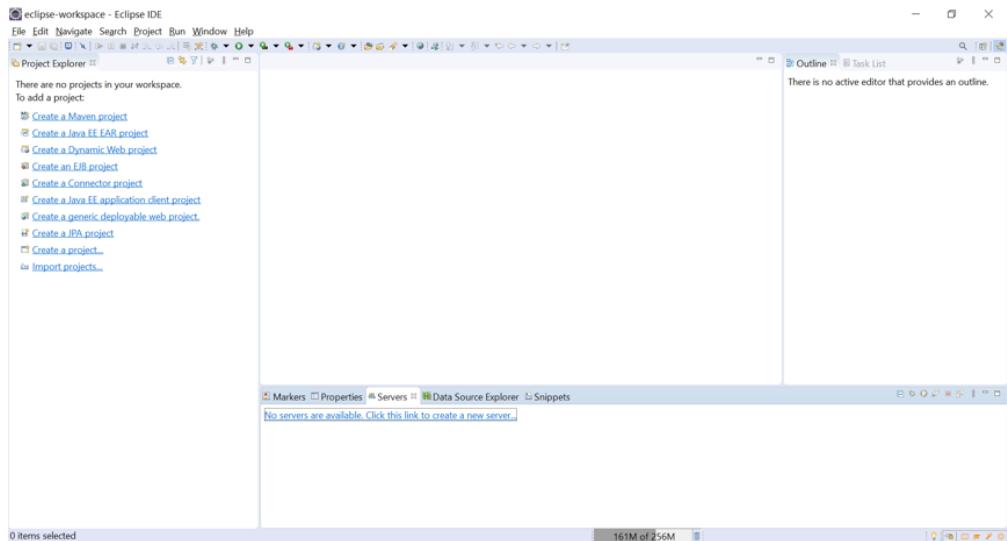


Figure 23 the Servers tab

Use the command to add a new server

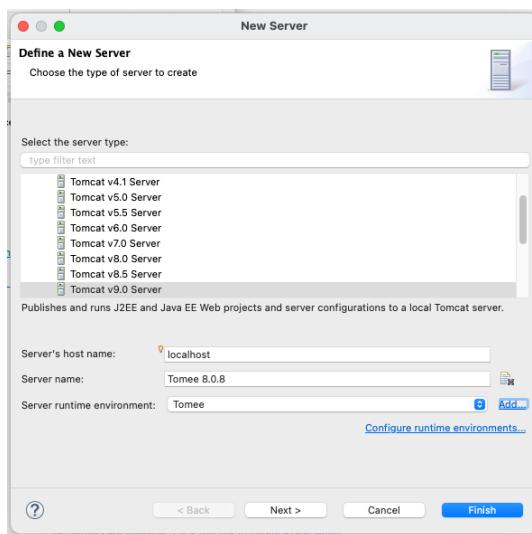


Figure 24 add Tomee as a server

Select Tomcat v9.0 and then 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.

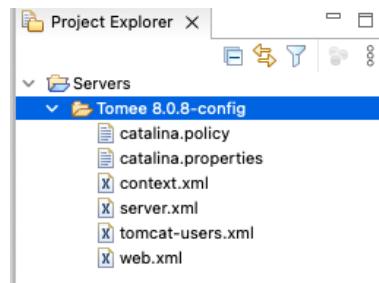
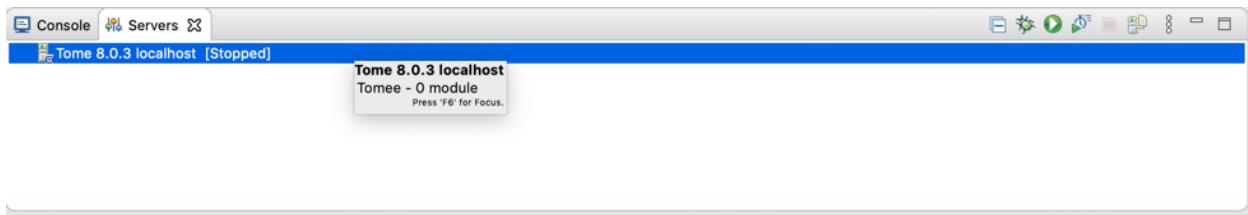


Figure 25 TomEE appears under the Servers node in the Project Explorer

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)*”.

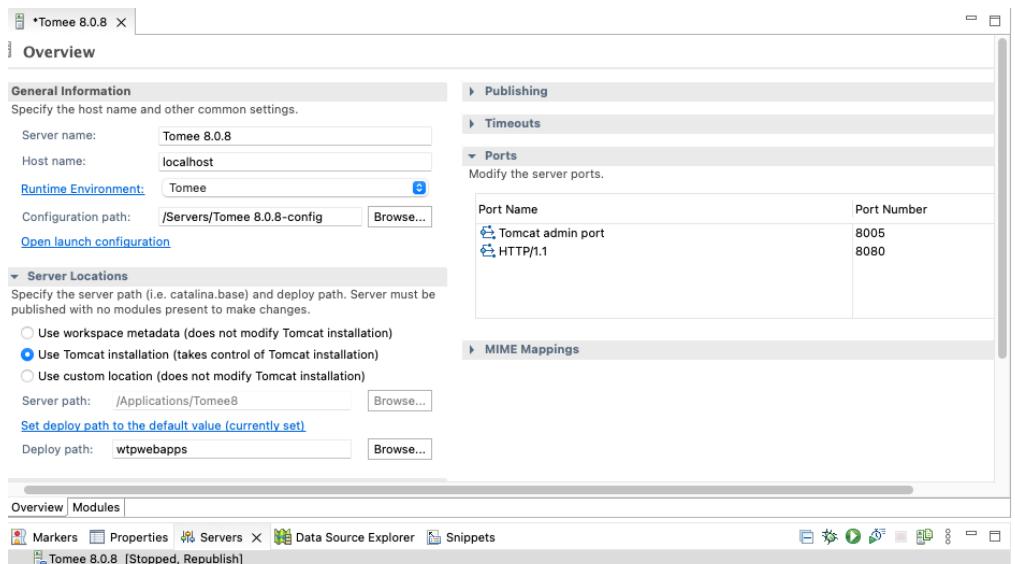


Figure 26 configuring TomEE

Set the admin port to a distinct value as shown.

| Port Name | Port Number |
|---|-------------|
|  Tomcat admin port | 8005 |
|  HTTP/1.1 | 8080 |

Figure 27: 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 other VM arguments as shown in the figure.

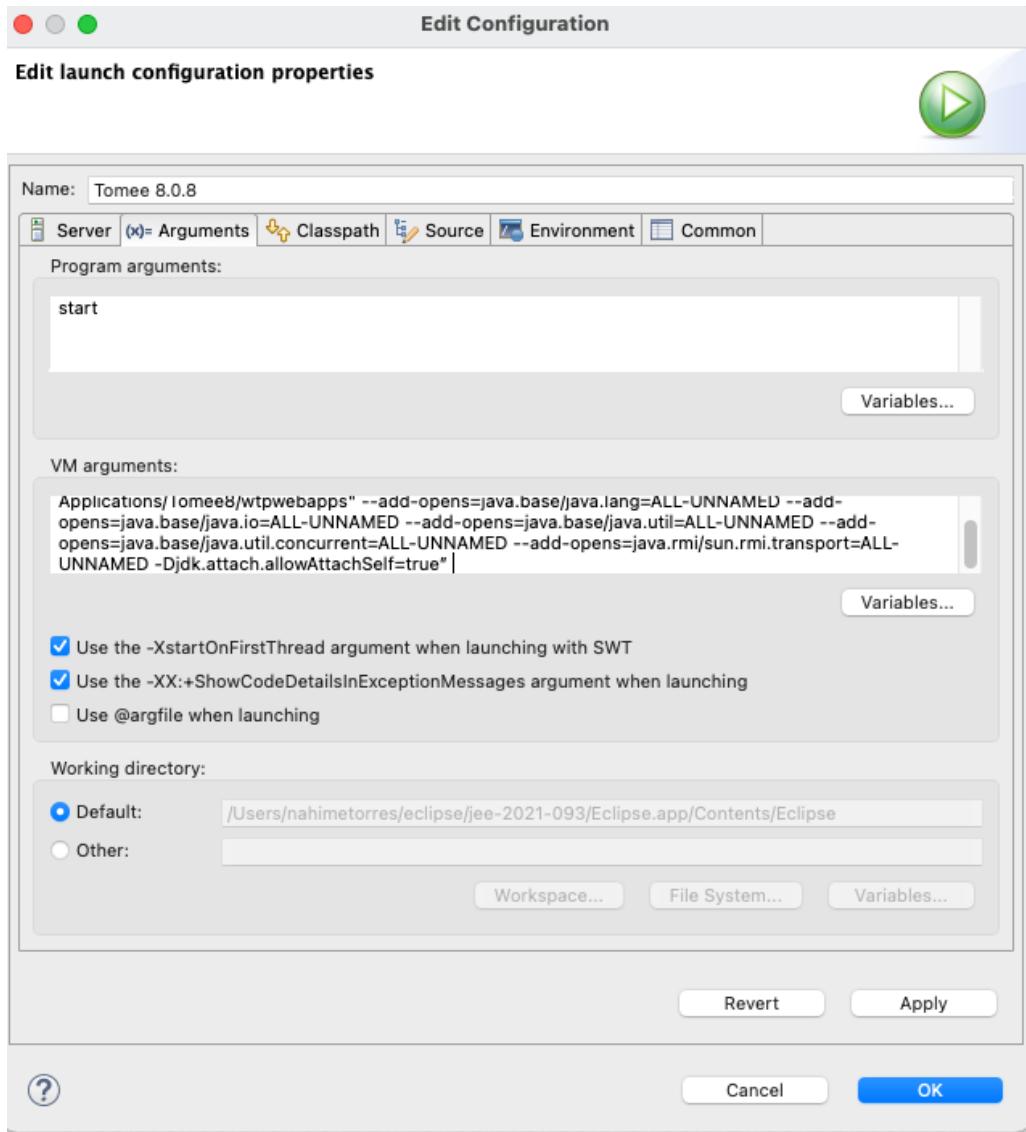


Figure 28 configuring TomEE

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.

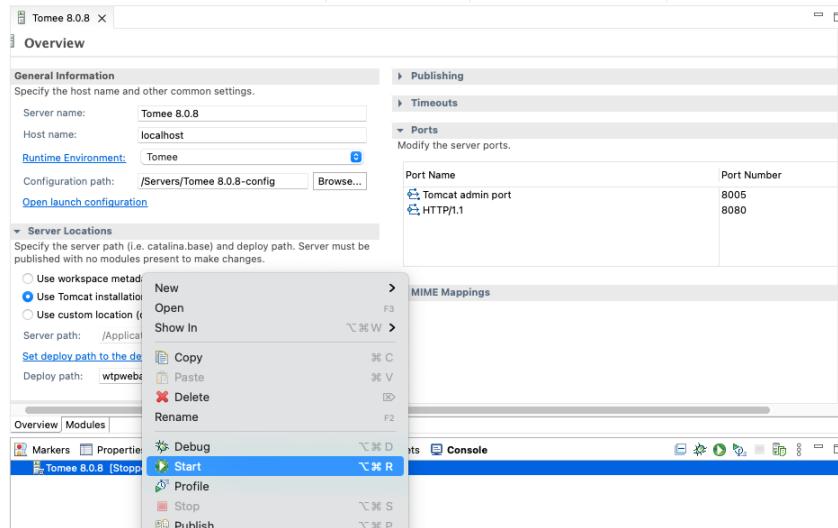


Figure 29: starting Tomee from the servers tab

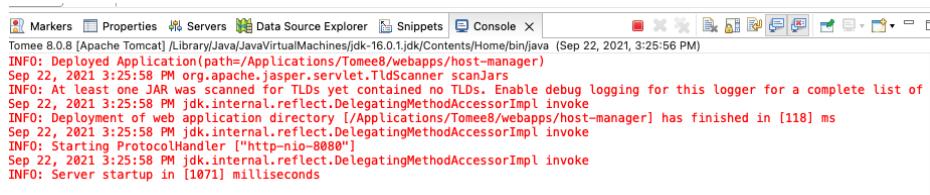


Figure 30: console message showing that Tomee is running

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

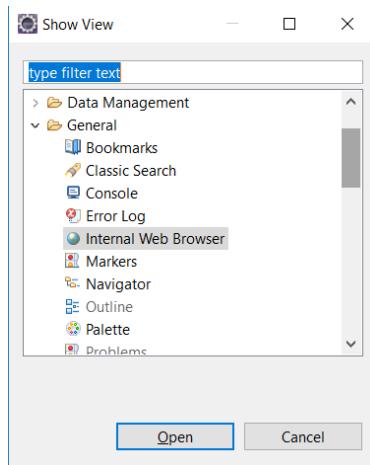


Figure 31: showing the internal web browser

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

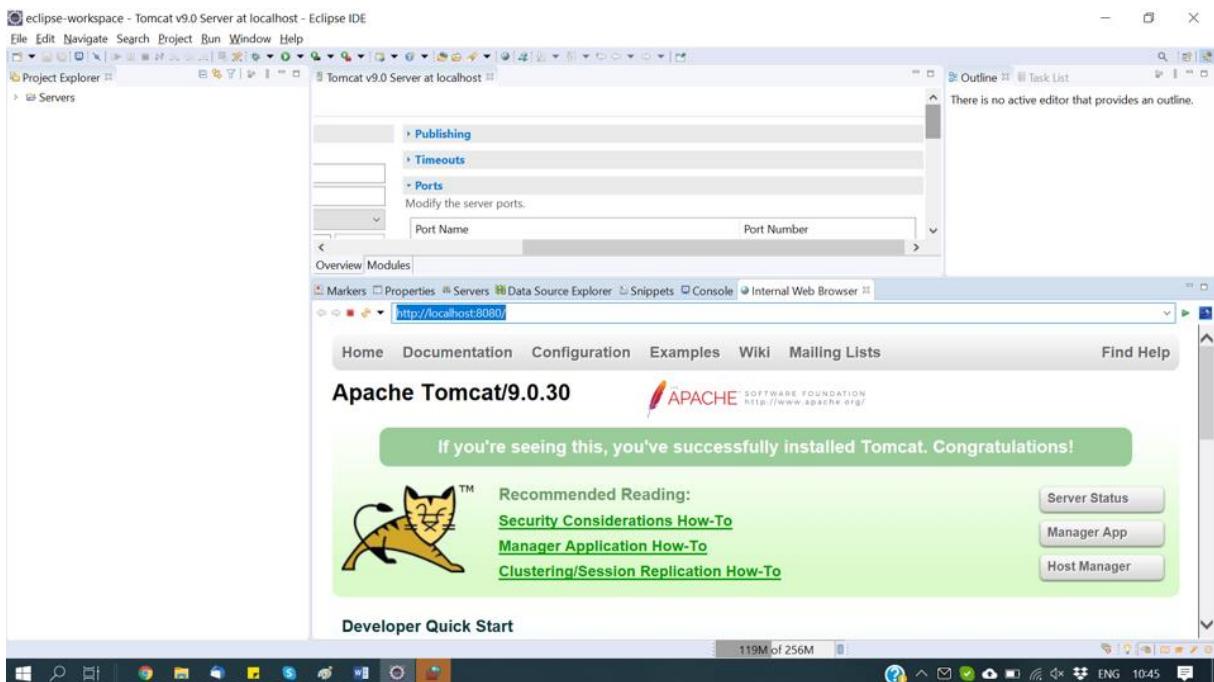


Figure 32 opening the home page of tomee in the internal browser

OS X

If the internal browser gives you an App Transport Security Error, try the `localhost:8080` on your default browser.



Figure 33 error opening internal browser

Additionally, we recommend to set your browser as default browser on Eclipse, Window->Web Browser->Default system web browser

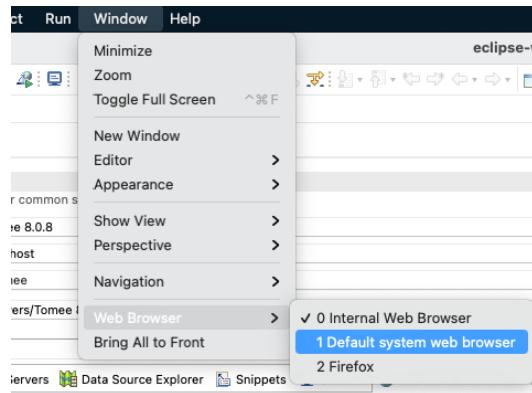


Figure 34 error opening internal browser

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

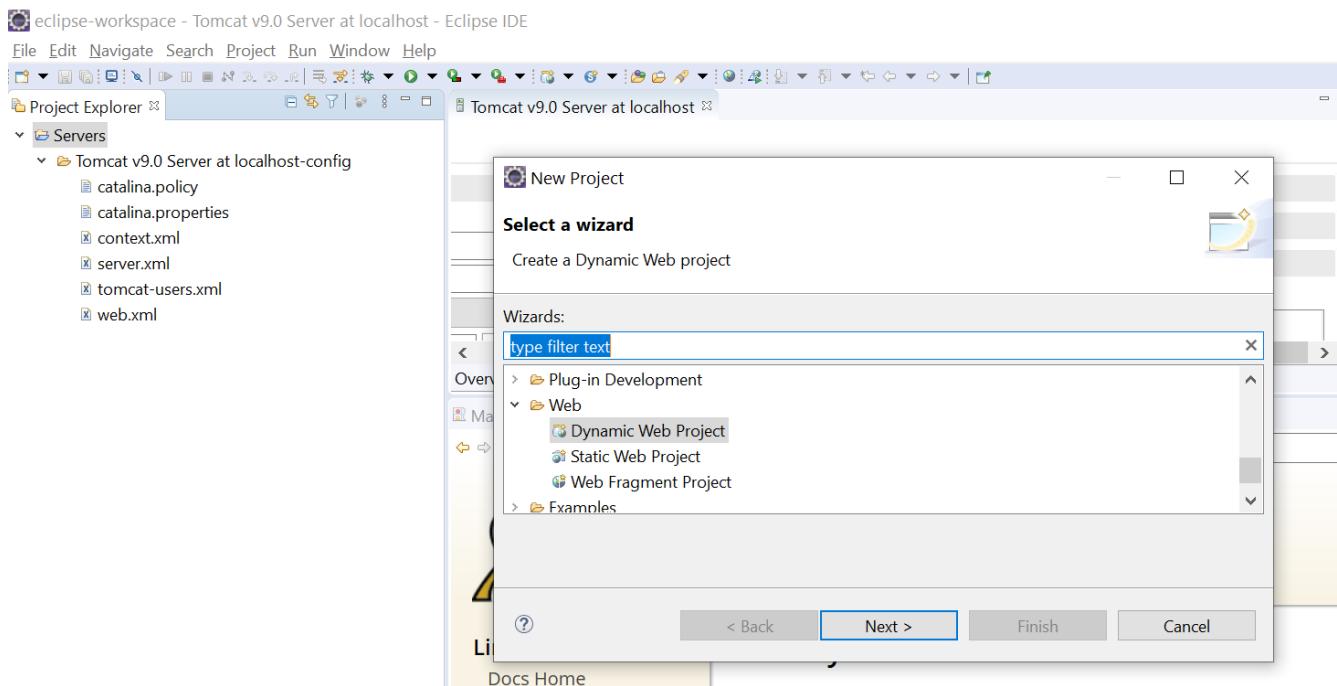


Figure 35: creating a dynamic web project

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

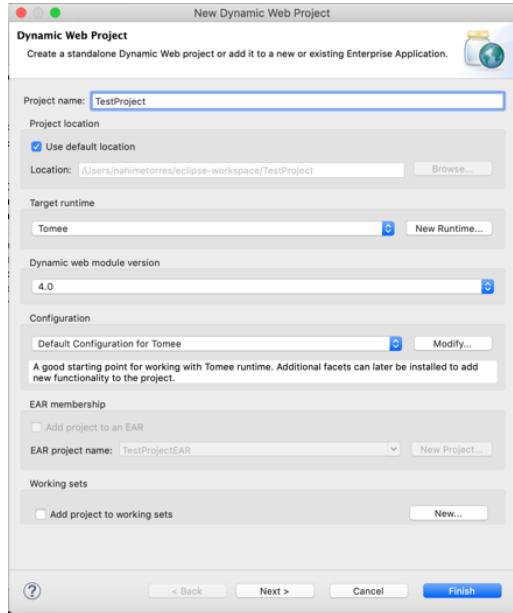


Figure 36: naming and creating a dynamic web project

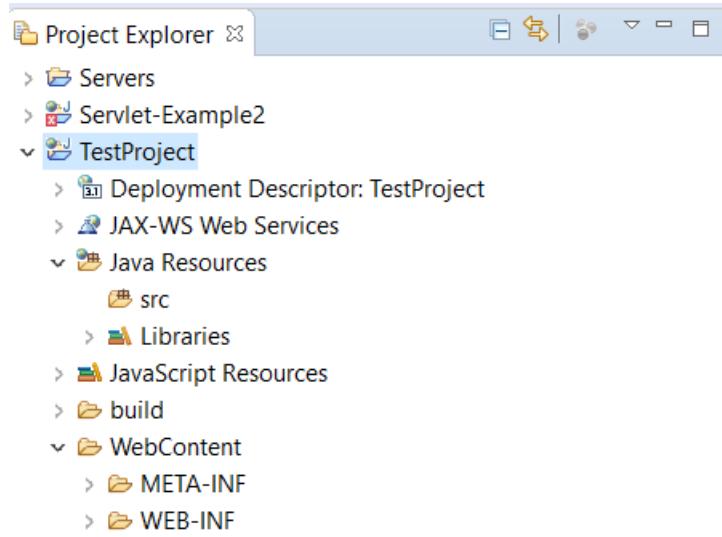


Figure 37: dynamic project structure created in the project explorer

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.db2.test* to the package. In the servlet creation wizard, accept all defaults and finish.

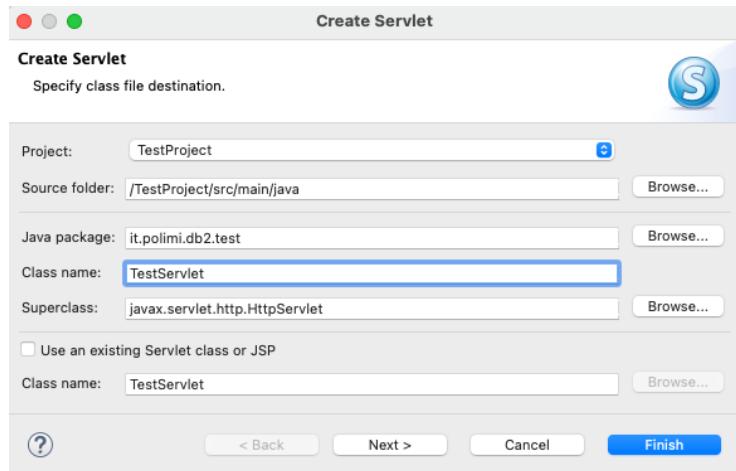


Figure 38: name the servlet class and the enclosing package

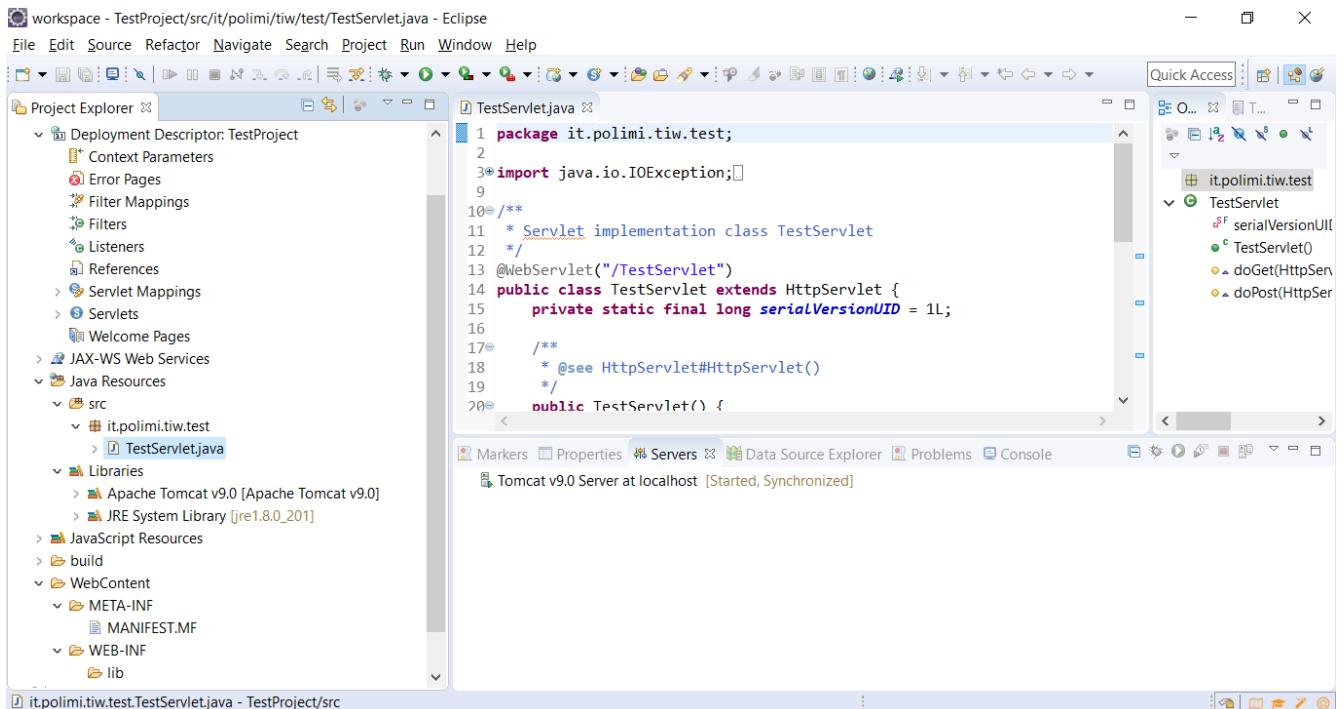


Figure 39: a newly created servlet

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

```
package it.polimi.db2.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 `src/main/webapp/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.

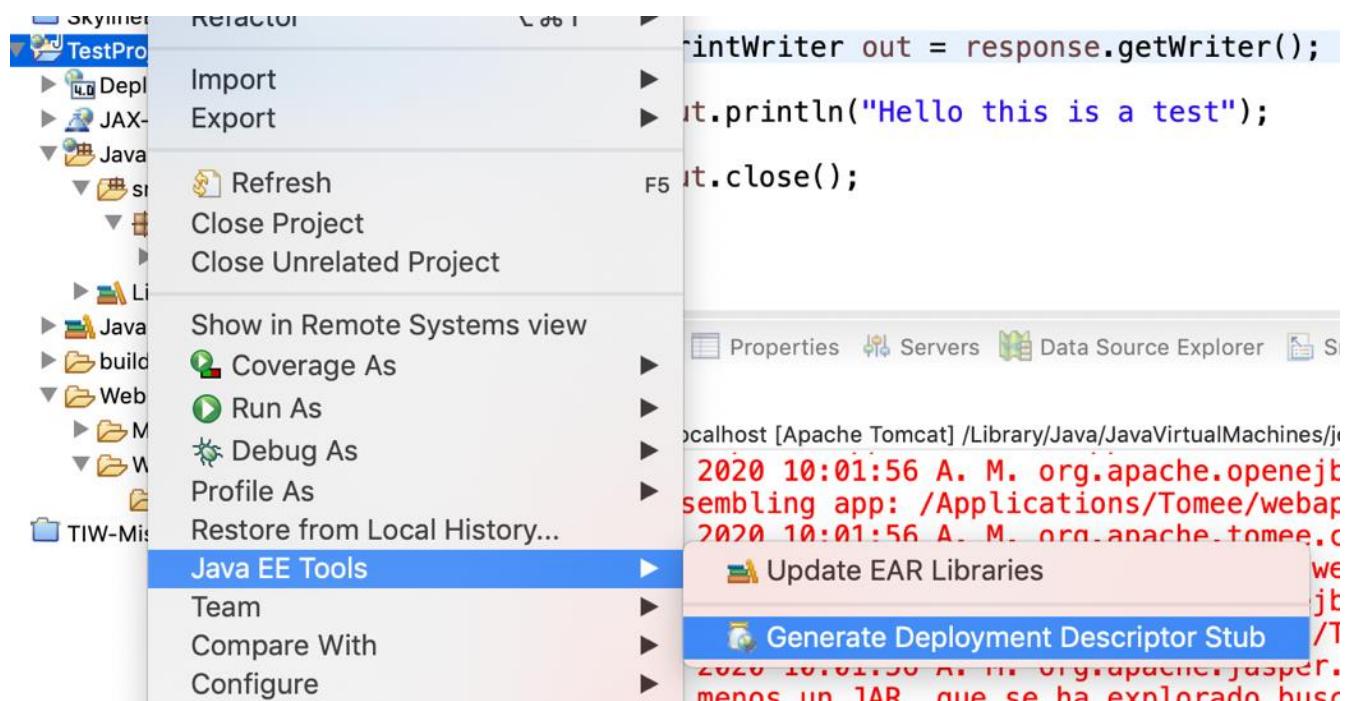


Figure 40: a newly created servlet

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_4_0.xsd"
    version="4.0">
    <servlet>
        <servlet-name>TestServlet</servlet-name>
        <servlet-class>it.polimi.db2.test.TestServlet</servlet-class>
    </servlet>
    <servlet-mapping>

```

```

<servlet-name>TestServlet</servlet-name>
<url-pattern>/</url-pattern>
</servlet-mapping>
</web-app>

```

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.

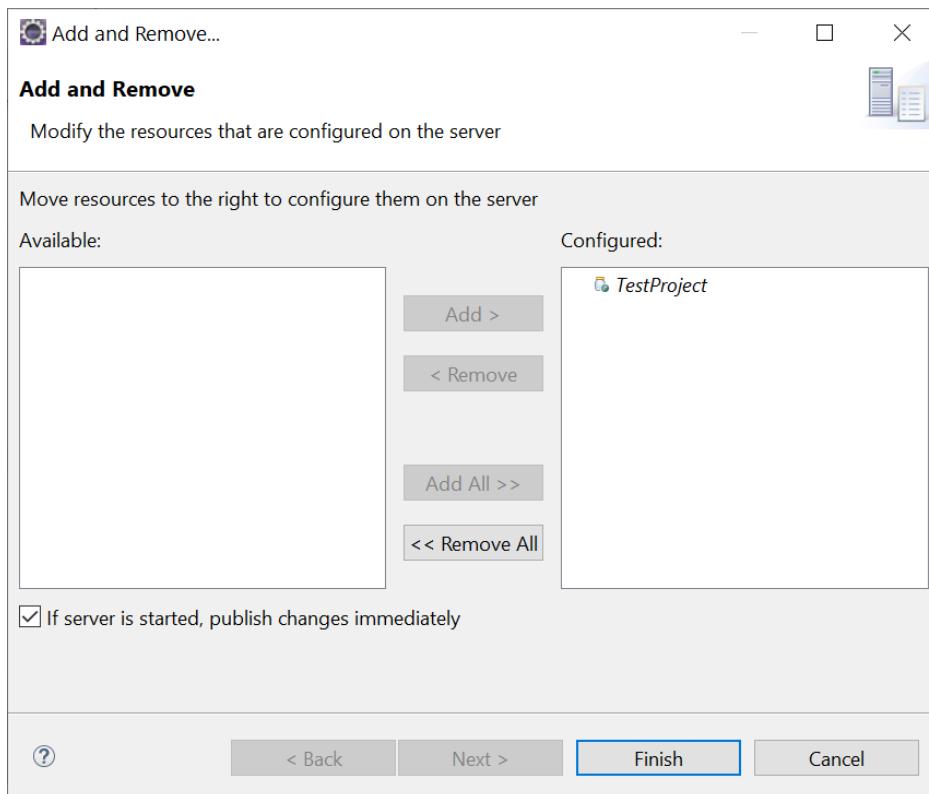


Figure 41: adding the project to the Tomee server

```

Feb 06, 2020 10:42:55 AM org.apache.catalina.startup.Catalina start
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]
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 42: 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.

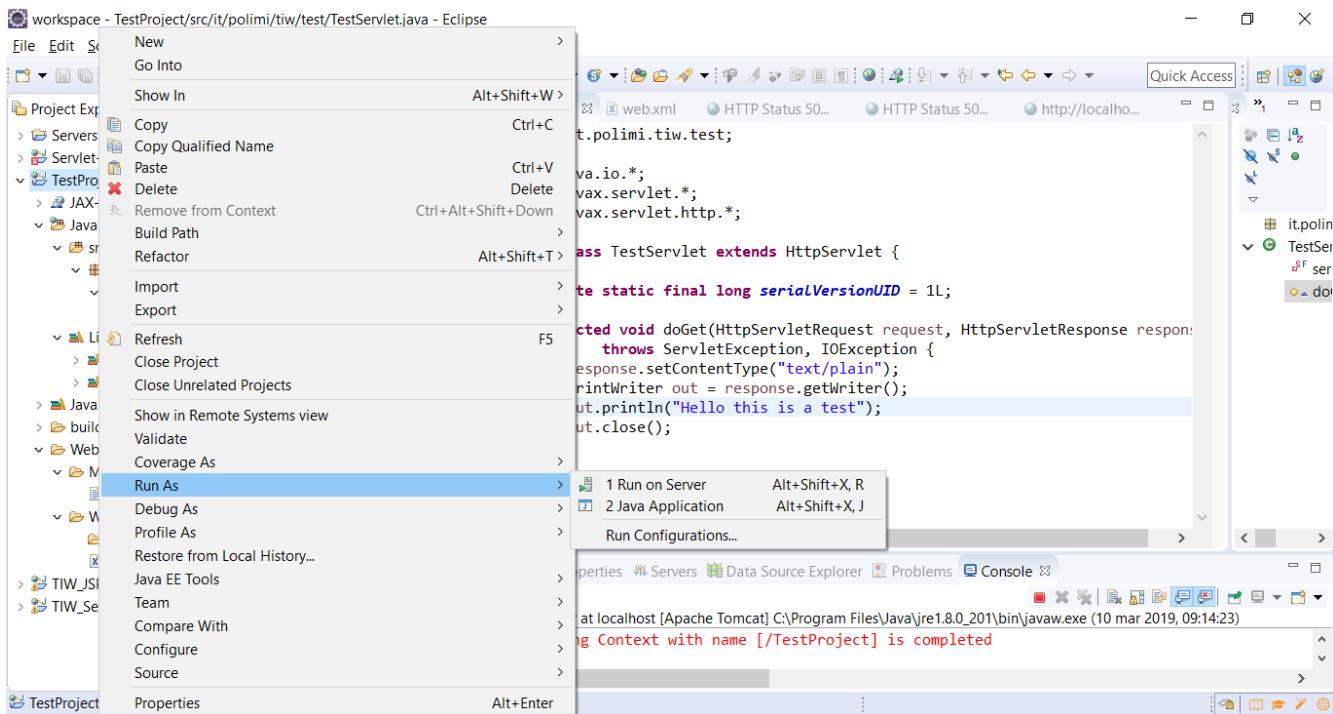


Figure 43: running the project on a server

Select the Tomee server and finish.

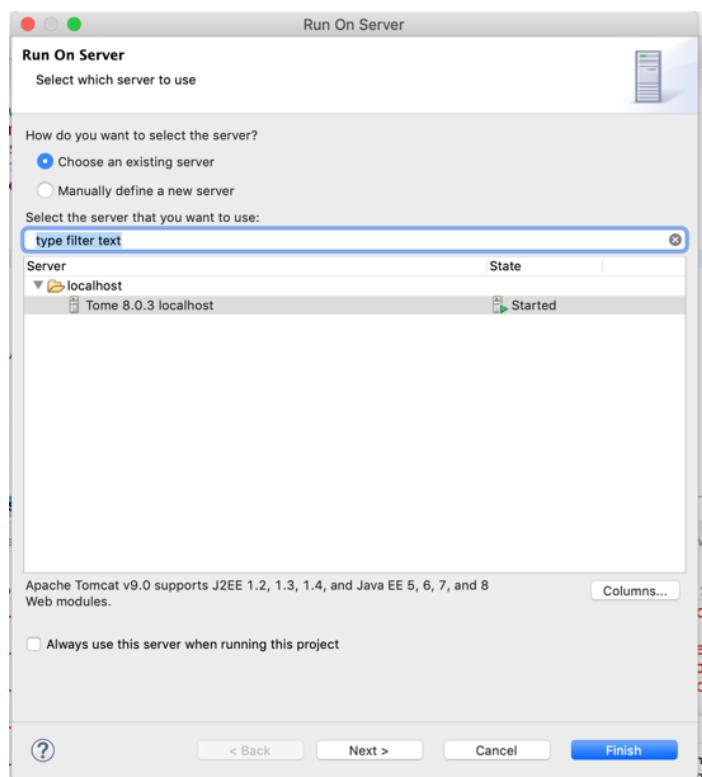


Figure 44: deploying the project on Tomee

Restart the server and continue.

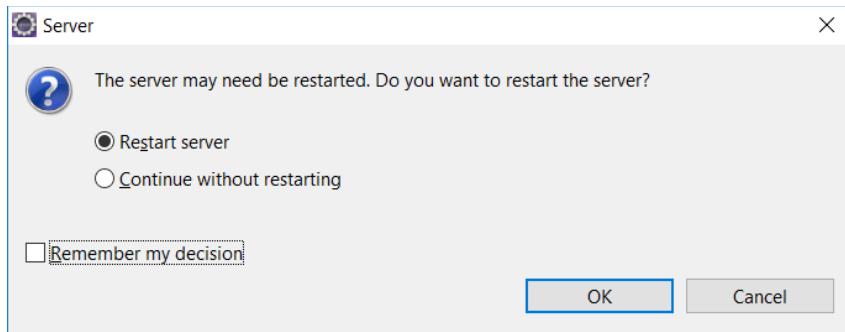


Figure 45: restarting the server (optional)

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

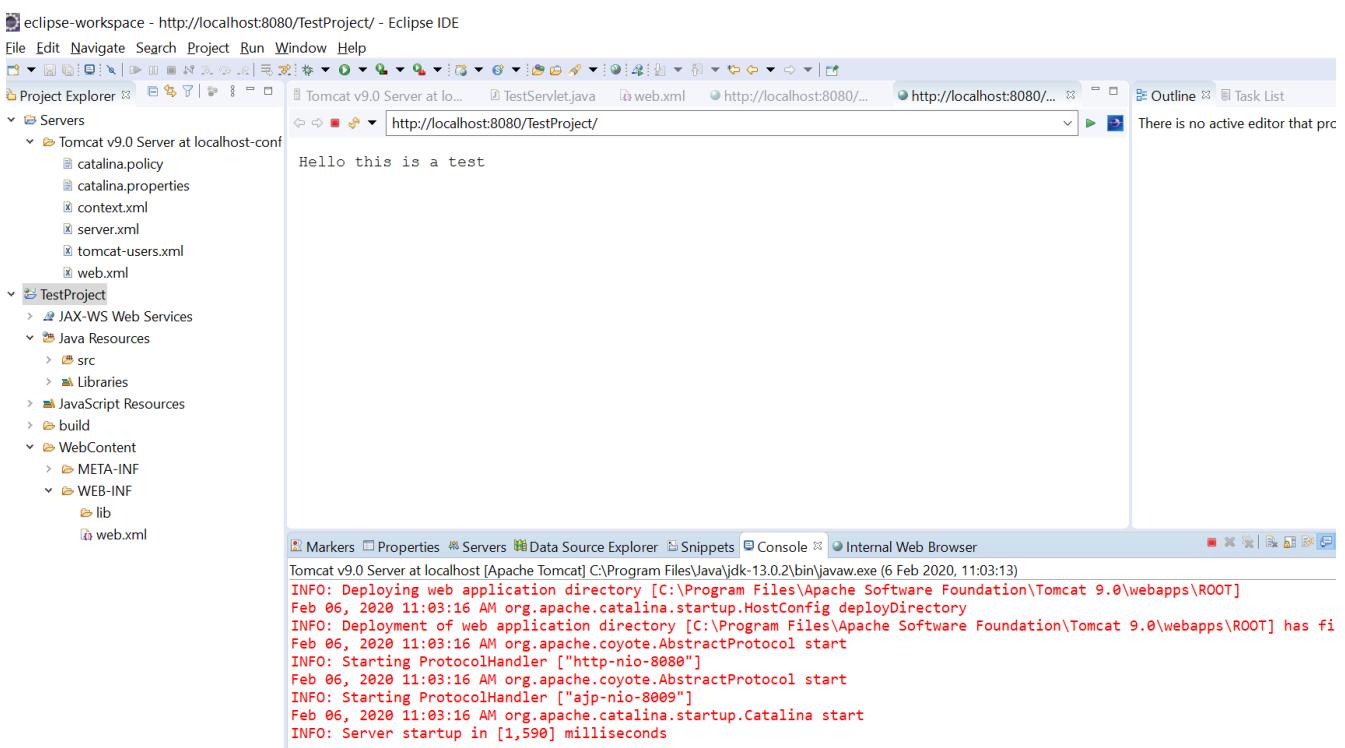


Figure 46: the internal browser opens on the location of the servlet

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 (you need to install it along with the server and workbench). It can be found, for example, at the location

C:\Program Files (x86)\MySQL\Connector J 8.0\mysql-connector-java-8.0.26.jar

If you do not have it, please follow the next steps.

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.

The screenshot shows the MySQL Connector/J download page for version 8.0.26. At the top, there are tabs for 'General Availability (GA) Releases' (which is selected), 'Archives', and a help icon. Below the tabs, the title 'Connector/J 8.0.26' is displayed. A dropdown menu labeled 'Select Operating System:' has 'Platform Independent' selected. Two download options are listed:

| File Type | Version | Size | Action |
|---|---------|------|--------------------------|
| Platform Independent (Architecture Independent), Compressed TAR Archive (mysql-connector-java-8.0.26.tar.gz) | 8.0.26 | 4.0M | Download |
| Platform Independent (Architecture Independent), ZIP Archive (mysql-connector-java-8.0.26.zip) | 8.0.26 | 4.7M | Download |

Below the download links, a note says: 'We suggest that you use the [MD5 checksums](#) and [GnuPG signatures](#) to verify the integrity of the packages you download.'

Figure 47: 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 src/main/webapp/WEB-INF/lib/

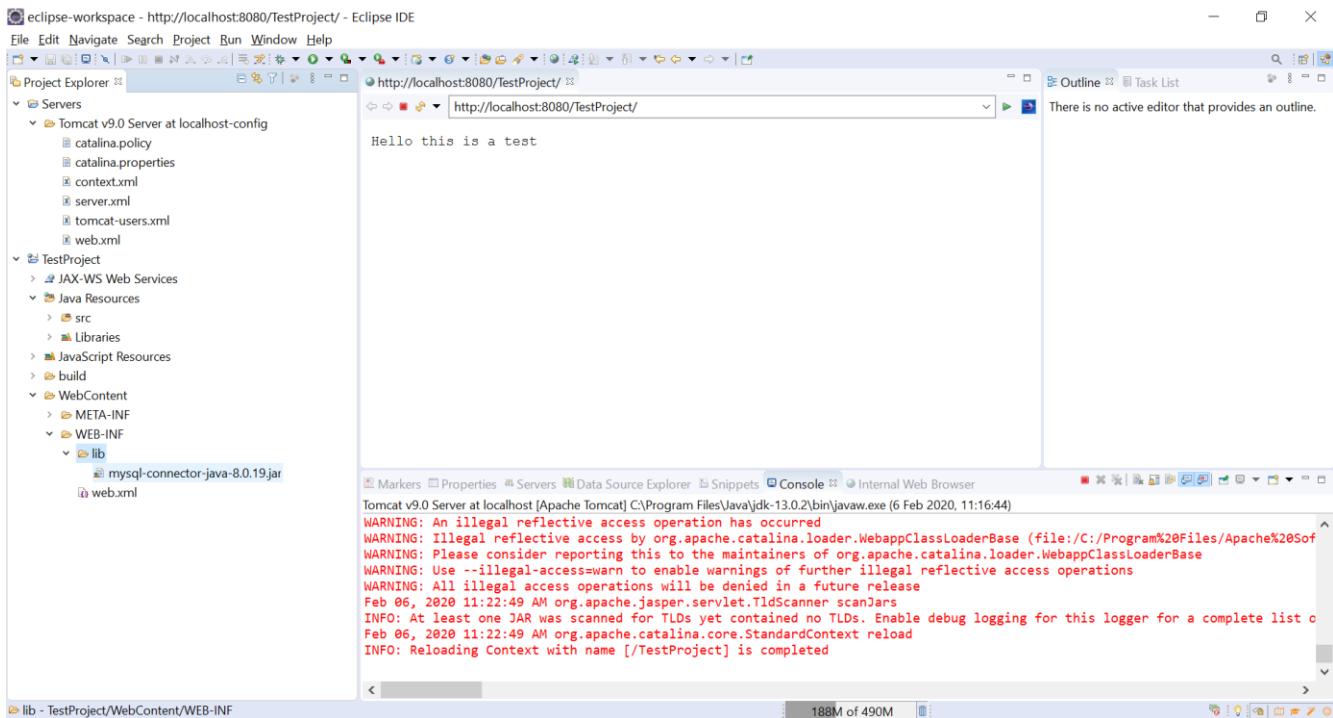


Figure 48: adding the ConnectorJ jar file to a specific project

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

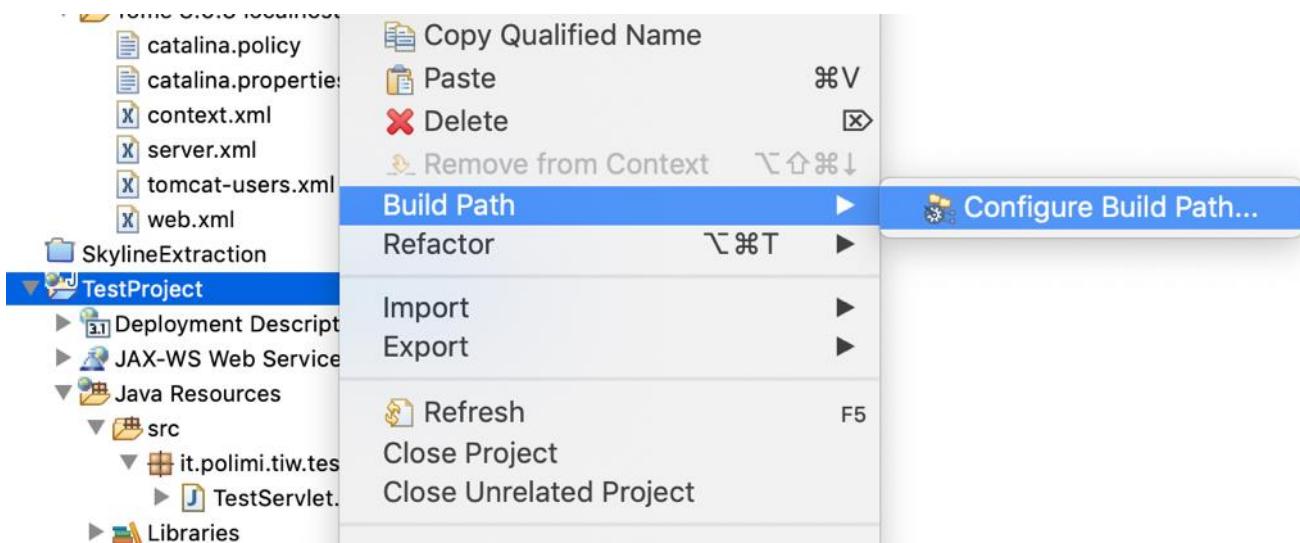


Figure 49: configuring the build path of the project

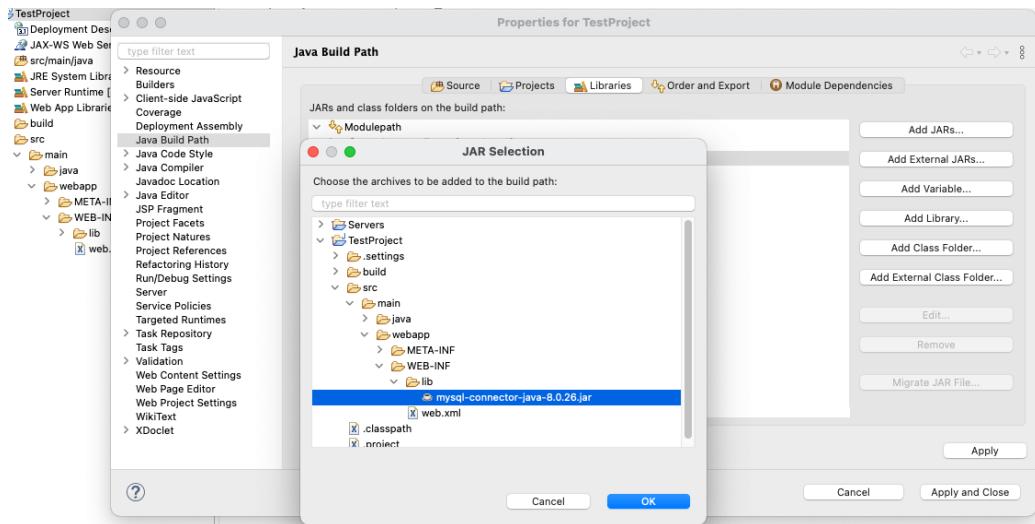


Figure 50: adding the JAR to the project build path (to the classpath)

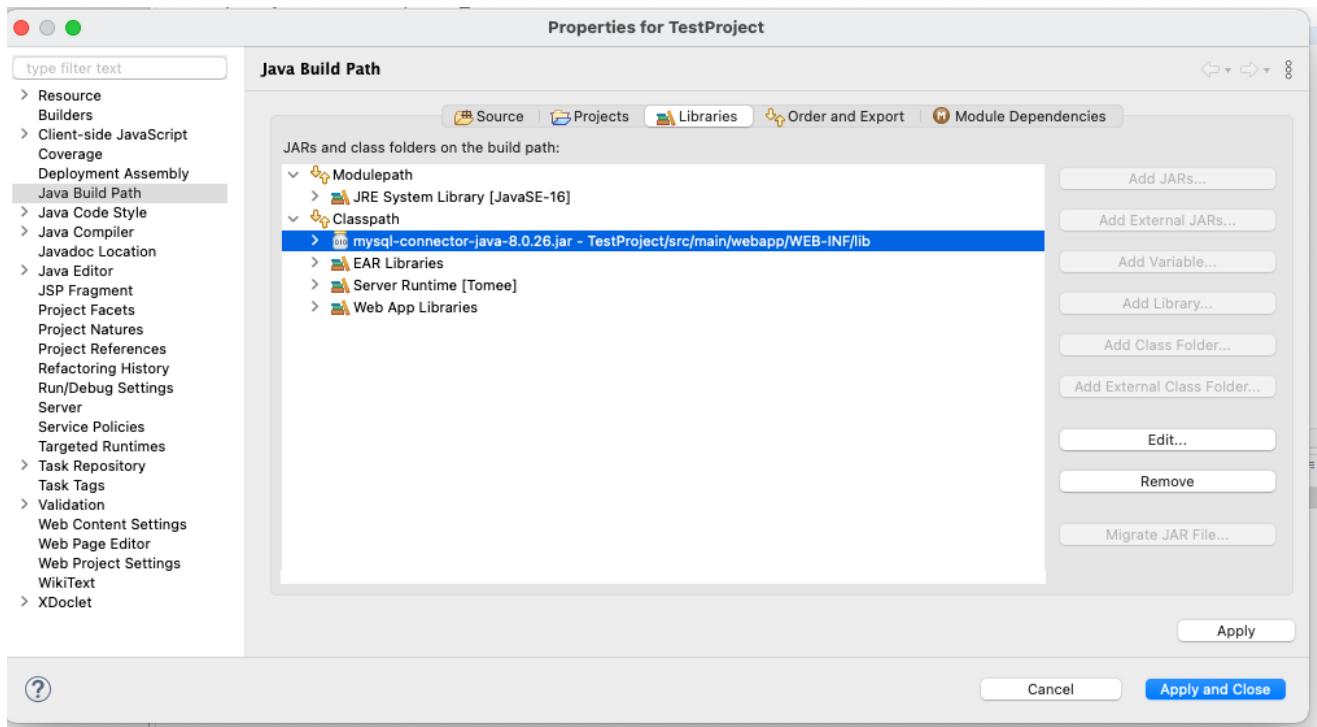


Figure 51: checking the build path for the presence of Connector/J

5.3 Create a new servlet

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

Edit the servlet source and replace the code with the following (**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.db2.jdbcdbtest;

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.

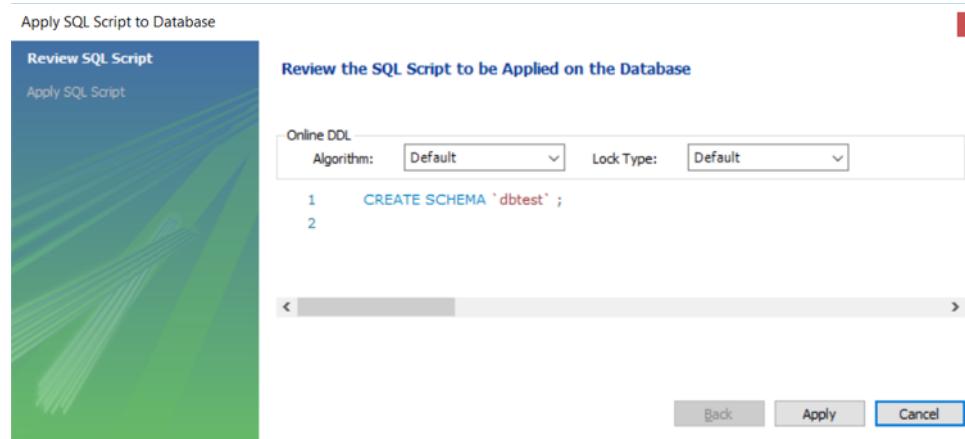
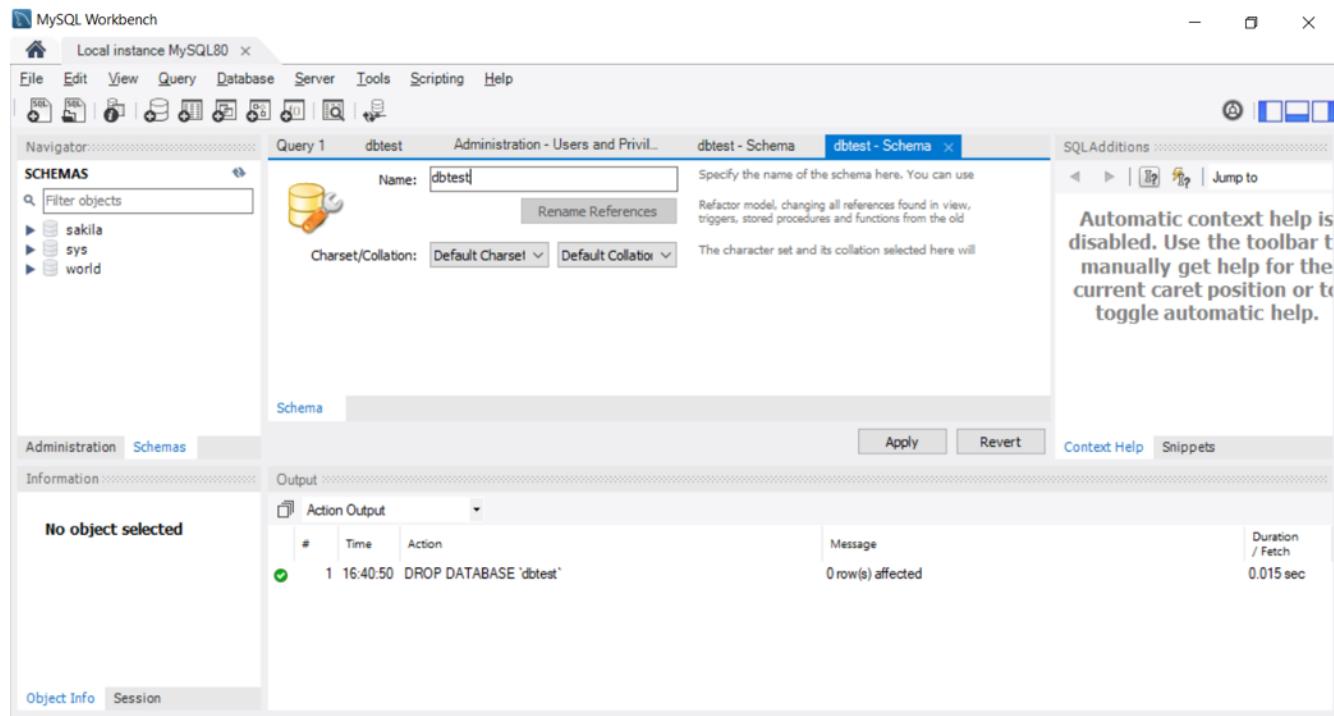


Figure 52: creating a test database

5.5 Run the project

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

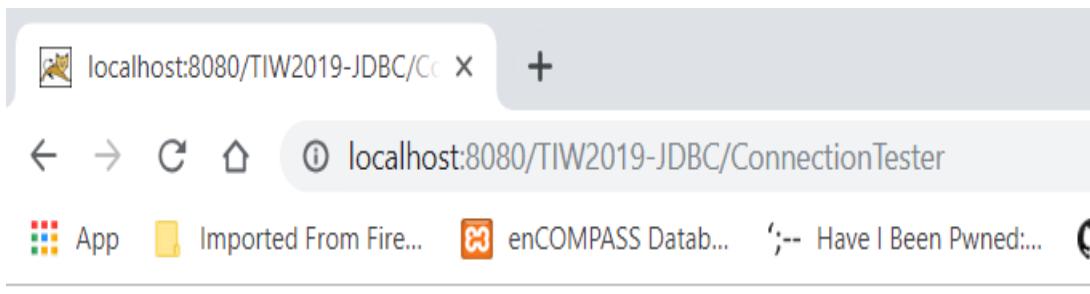


Figure 53: testing the servlet and the database connection

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 MySQL TimeZone Error

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.

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 16, 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 (copy the content of the sql file and paste it inside the workbench, then press run), 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:

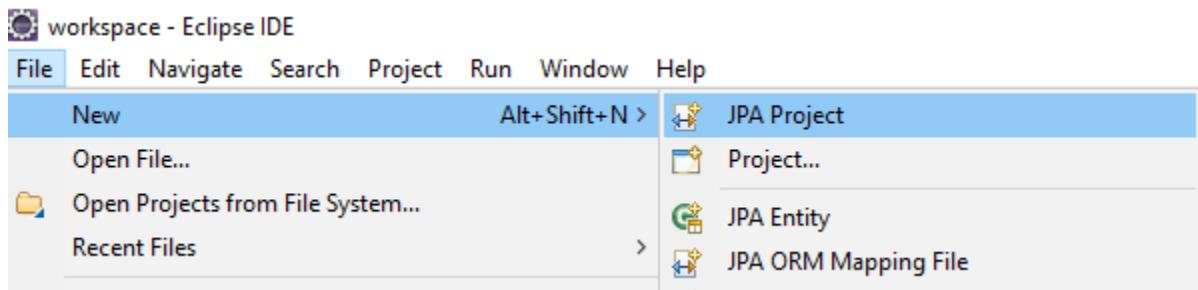


Figure 51

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

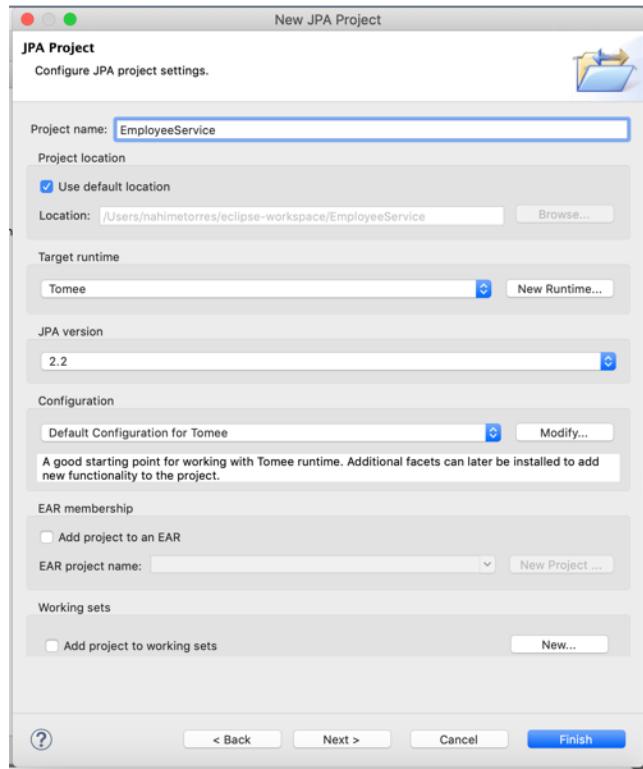


Figure 52

4 – Click next until you reach the JPA Facet part. If it is your first JPA project you will have to follow the next steps and configure the libraries and connections, otherwise go to step 6.

4.1 Download the library manually from this link (we recommend 2.7.9)

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

4.2 Click in the icon to manage libraries

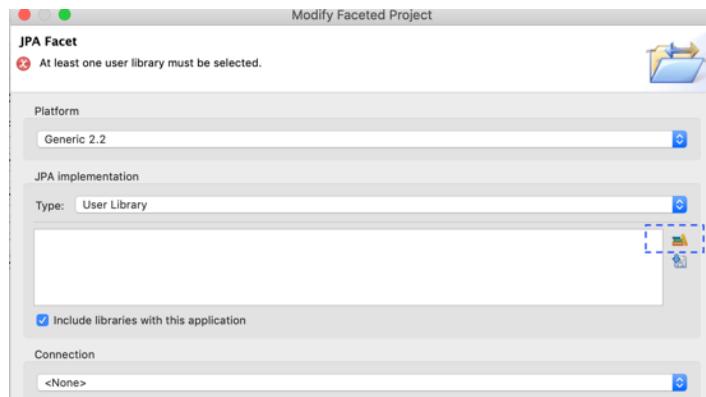


Figure 54

4.3 Click on New and name it EclipseLink 2.7.9 (DO NOT check the System Library item)

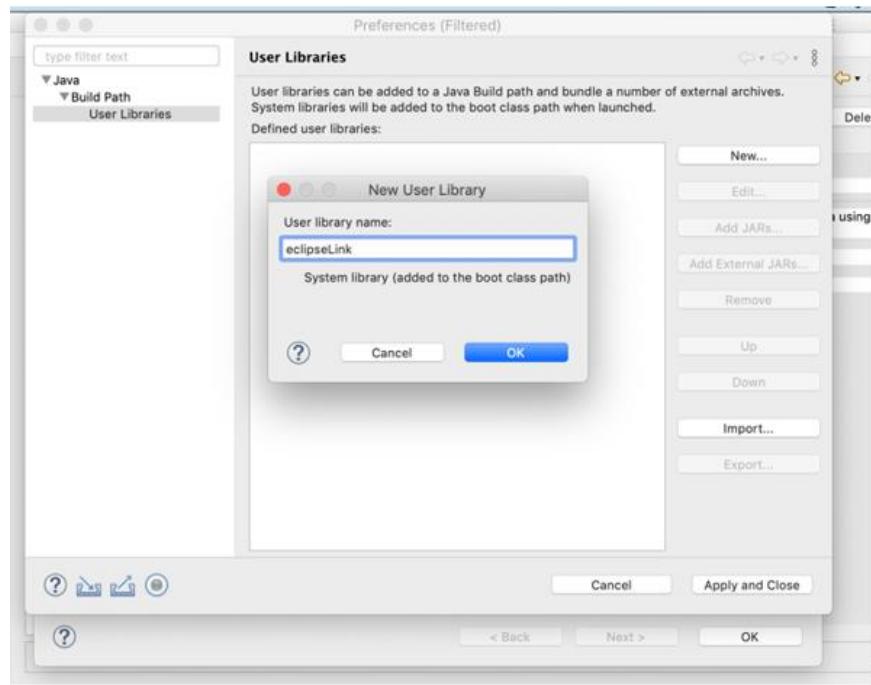


Figure 55

4.4 Select the library and click on Add External JARs

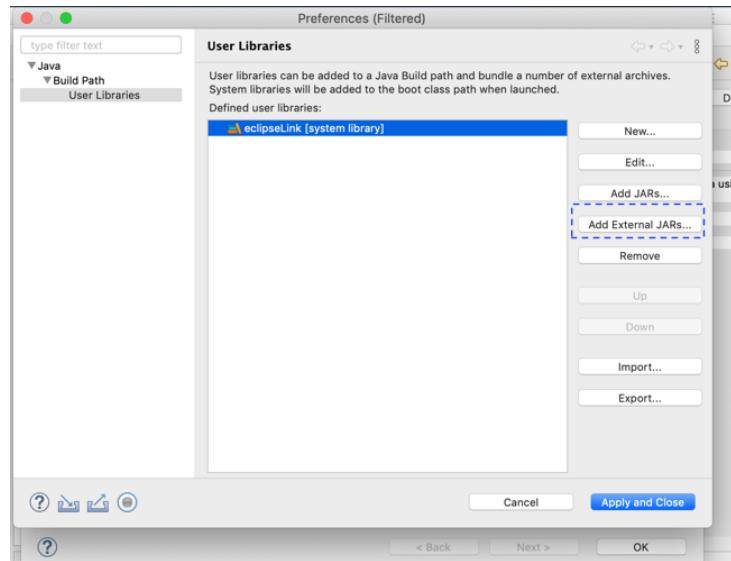


Figure 56

4.5 Import first the eclipseLink.jar from the downloaded folder inside /jlib/ and then the several .jar inside /jlib/jpa/*.jar (**/jlib/moxy/*.jar and /jlib/sdo/*.jar shoul not be needed**)

4.6 Once the jars are added, click Apply and Close

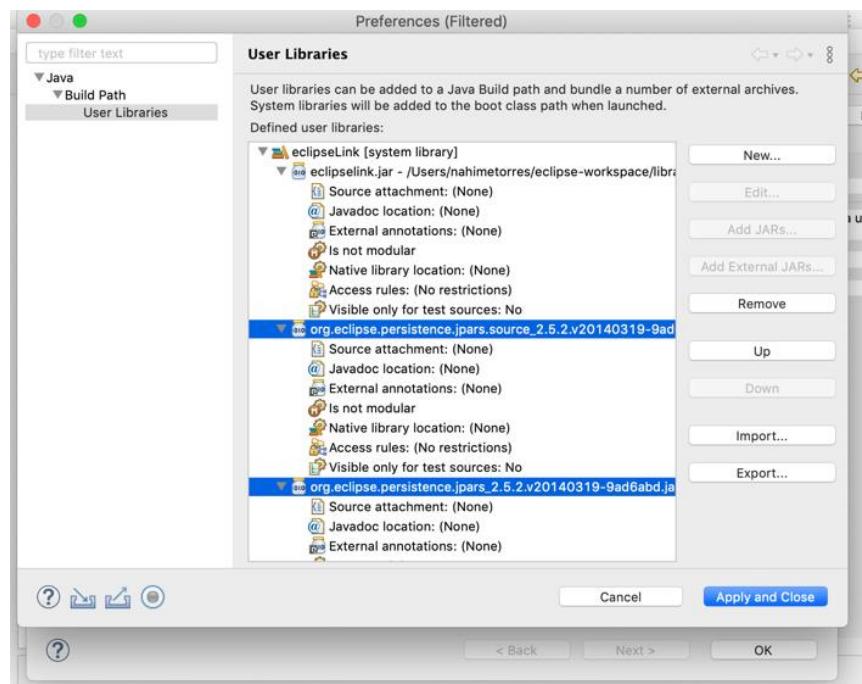


Figure 57

4.7 The library will appear now, select it. Then. if “include libraries with this application” is disabled, do not check it and continue.

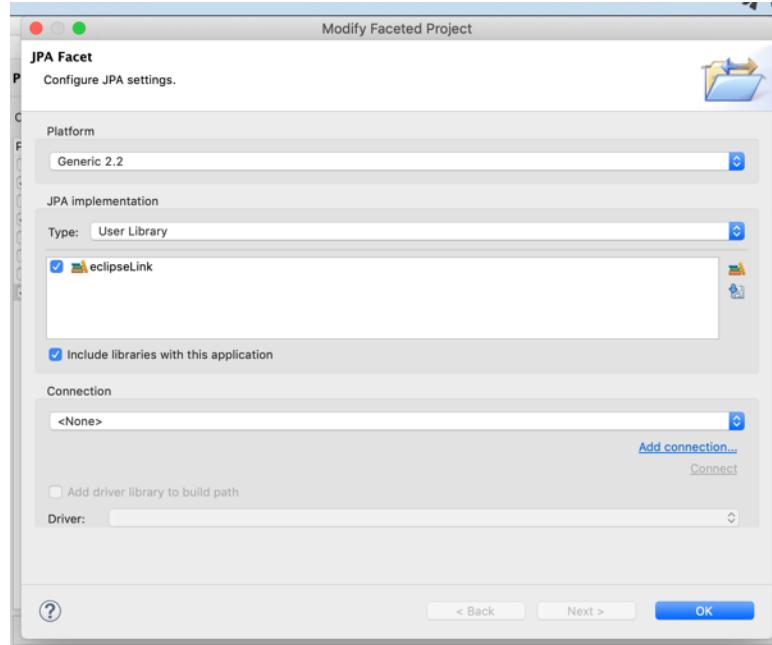


Figure 58

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...” and select mysql driver. Provide the data for your connection (the url is case sensitive and must include the schema name), and test it using the button “Test connection”. If everything is correct click finish.

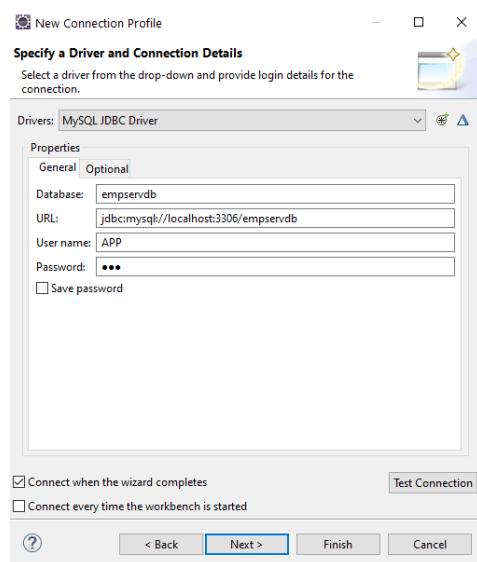


Figure 59

Possible troubleshooting:

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

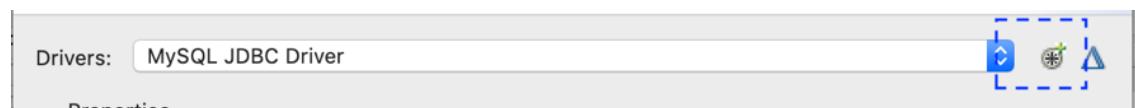


Figure 60

5.2 Choose a driver (5.1 systemversion is ok)

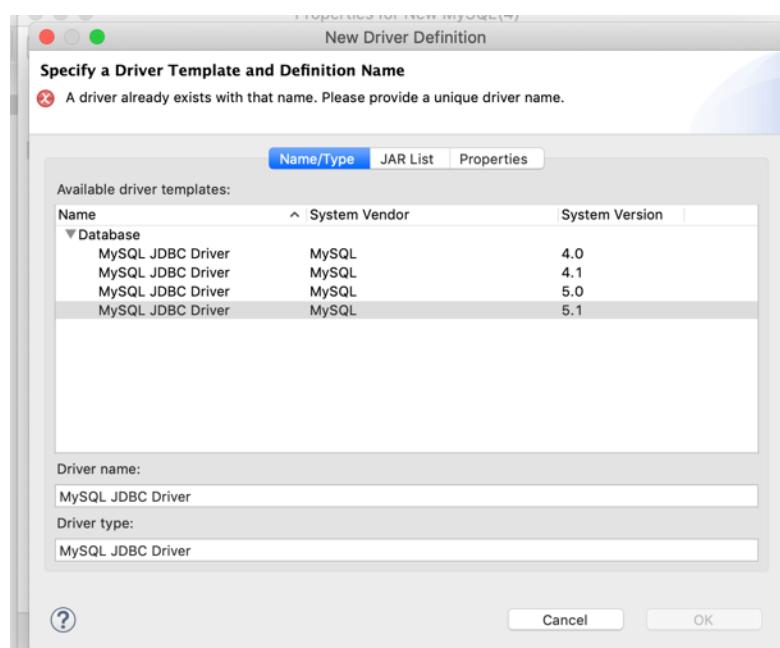


Figure 61

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

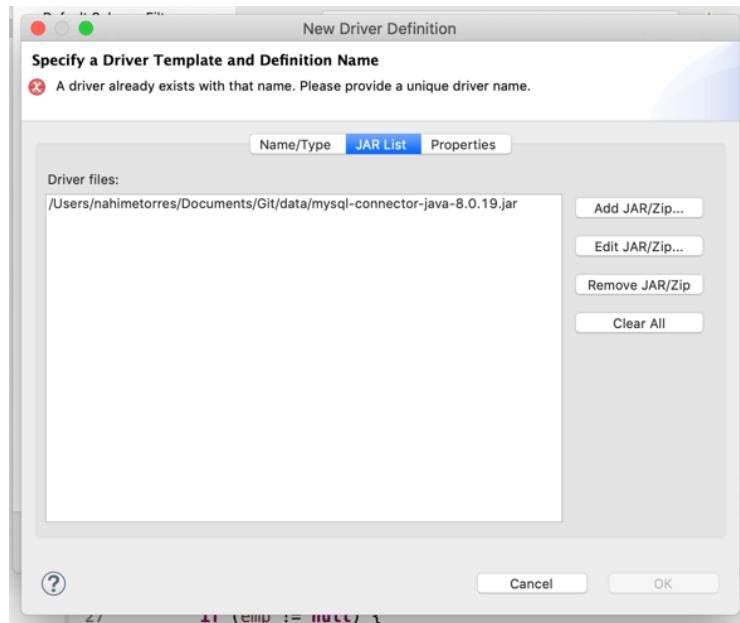


Figure 62

6. Check the “Add library to the build path” option on the screen. And click Finish. If a popup appears, do not open the JPA perspective.

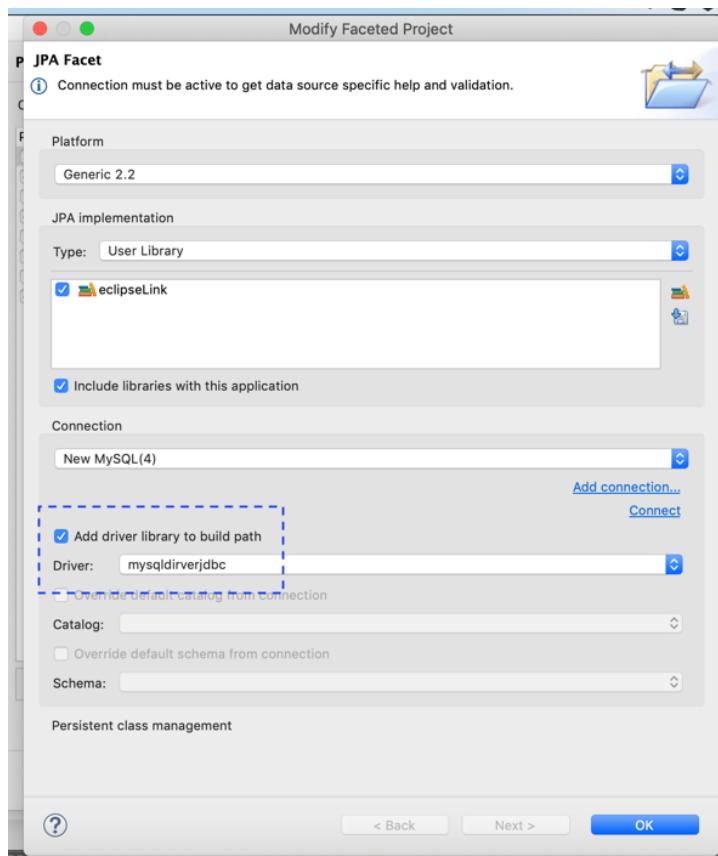


Figure 63

7.-Create two new packages **examples.client** and **examples.model**

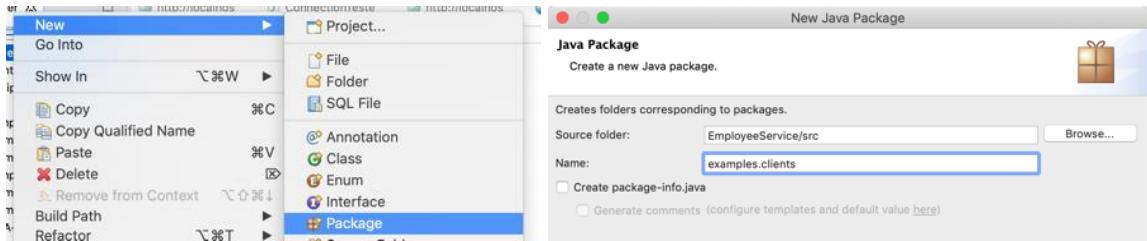


Figure 64

On the new project copy the source files from:

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

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

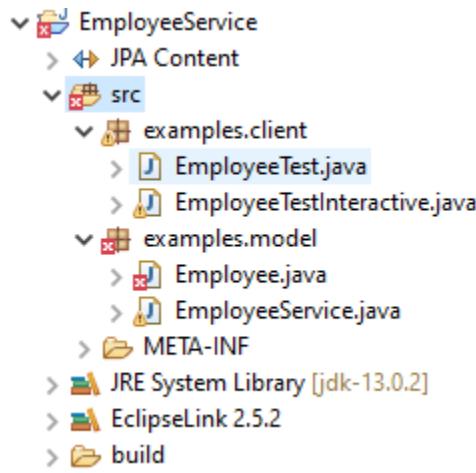


Figure 65

8.-Open the \examples\Chapter2\employeeService\etc\persistence\META-INF\persistence.xml and copy the content to replace the project persistence.xml (if some error occurs, you can copy and replace the entire persistence.xml file)

Inside eclipse, double click on the persistence.xml file and modify the connection properties for your environment (from the **source** tab). The driver and URL should correspond to MySQL, and the user and pass should be for your local DB.

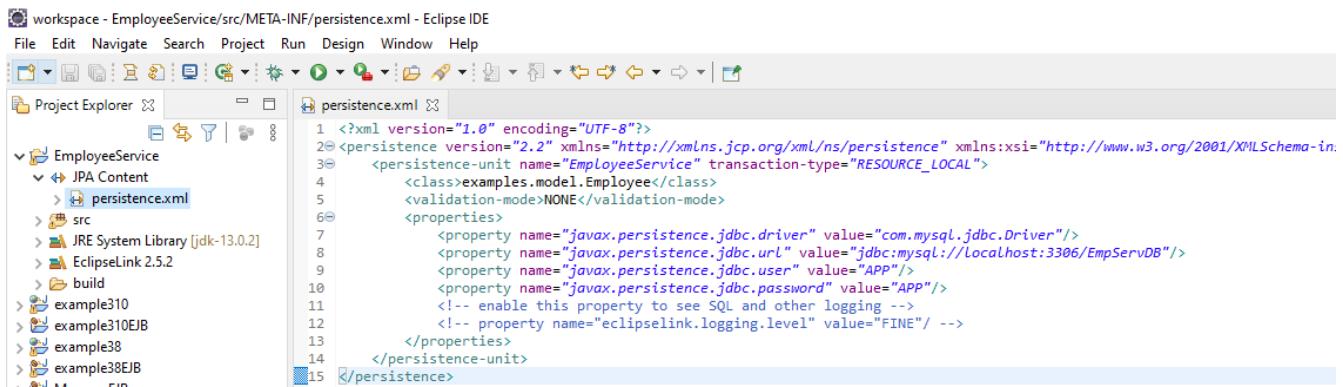


Figure 66

```

<property name="javax.persistence.jdbc.driver" value="com.mysql.cj.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_PASS"/>

```

9.-You should already have the MySQL JDBC connector int the project after the JPA facet configuration (it should also be present in the Tomee/libs folder)

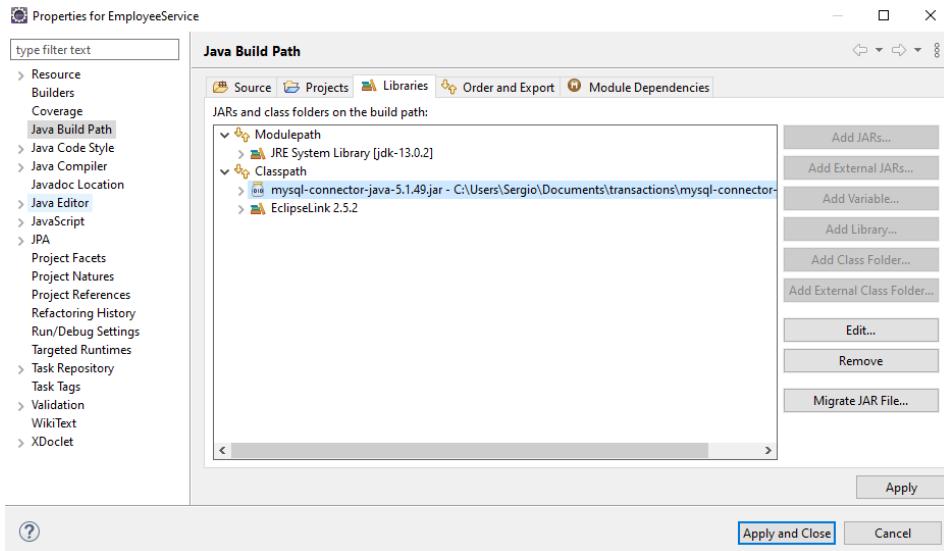
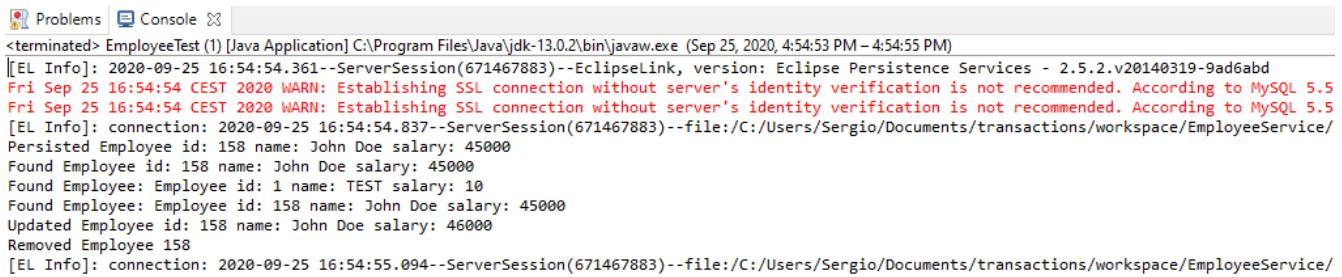


Figure 67

10.-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 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 68

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 (some changes may be required). You can test them as follows:

EJB project.

1.- Create a new EJB project

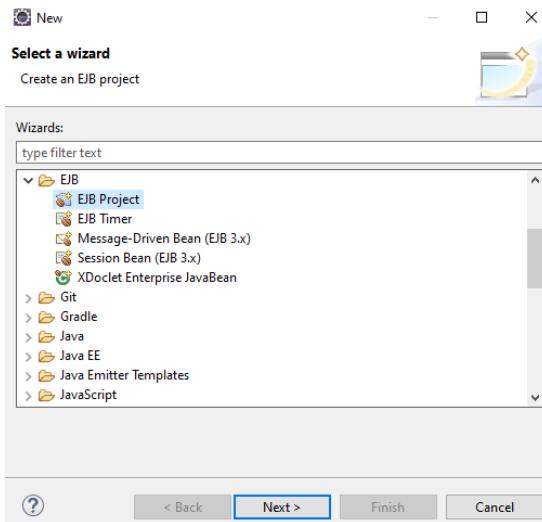


Figure 69

Give it a name with suffix EJB to make it easy to identify and click finish (if a java version error is shown, select EJB module version 3.2).

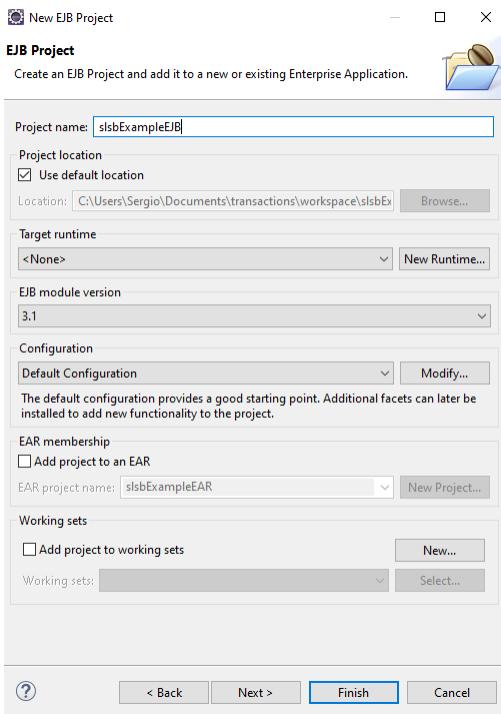


Figure 70

2.- Copy the content of \examples\Chapter3\<example name>\src\model\examples\stateless 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.

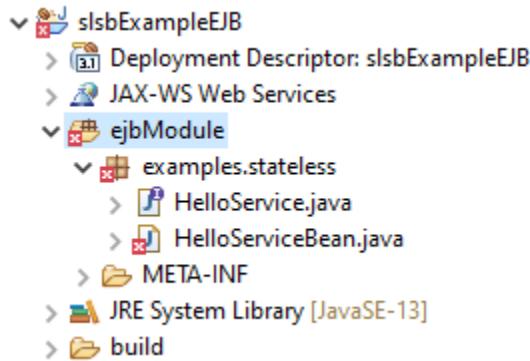


Figure 60

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 if it appears.

Also make sure that the java version in the Project Facets and in the Java Compiler are the same (java 16).

Click finish and the apply.

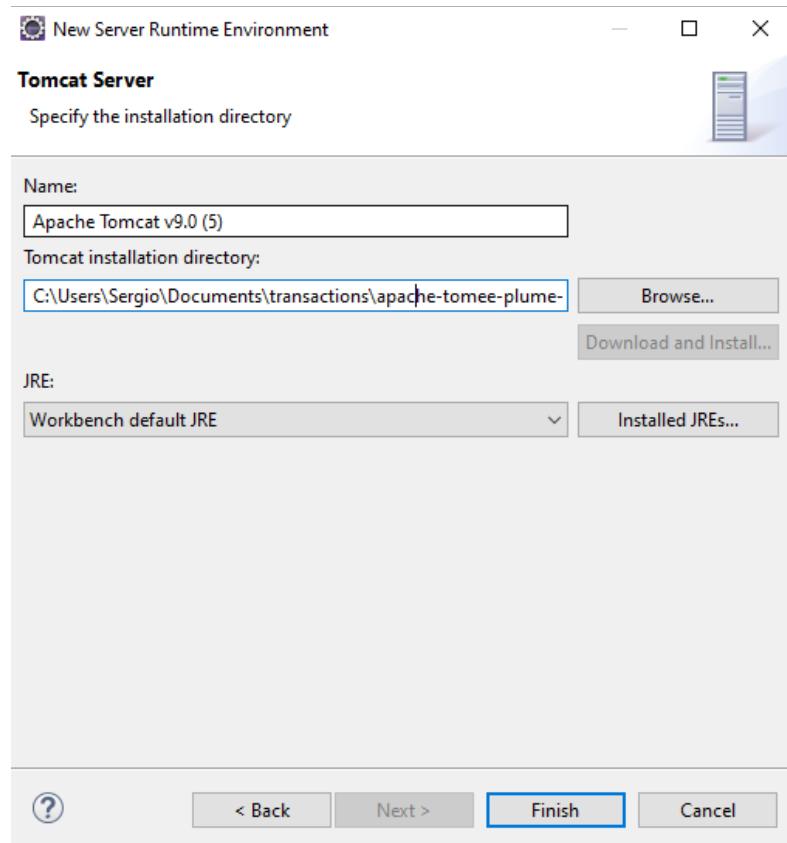


Figure 71

4.-On the properties windows go to “java 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.

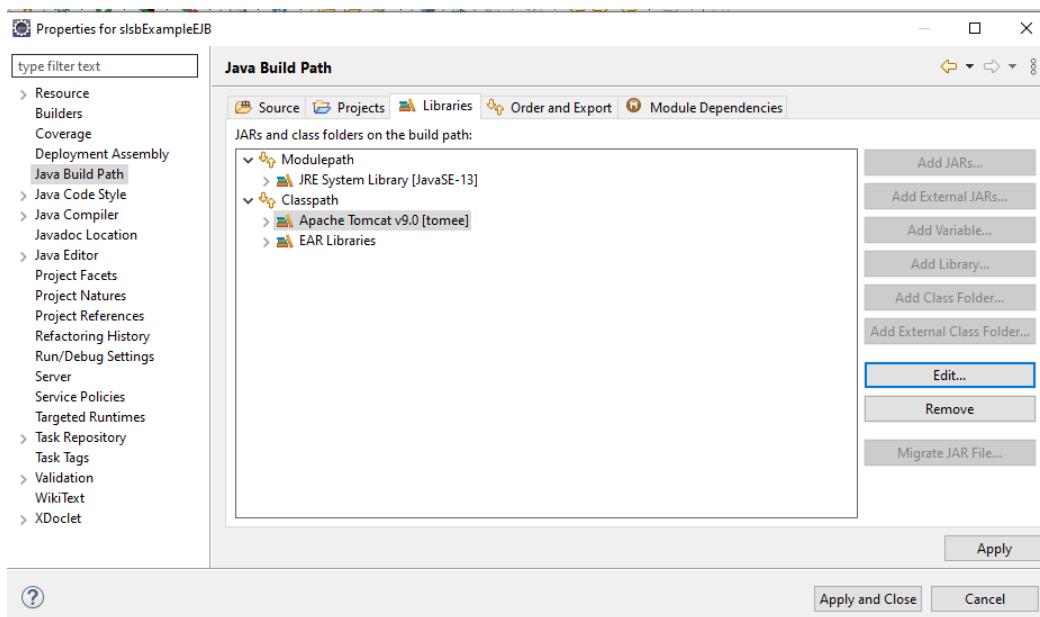


Figure 72

Web Project

1.- Create a new Dynamic Web project

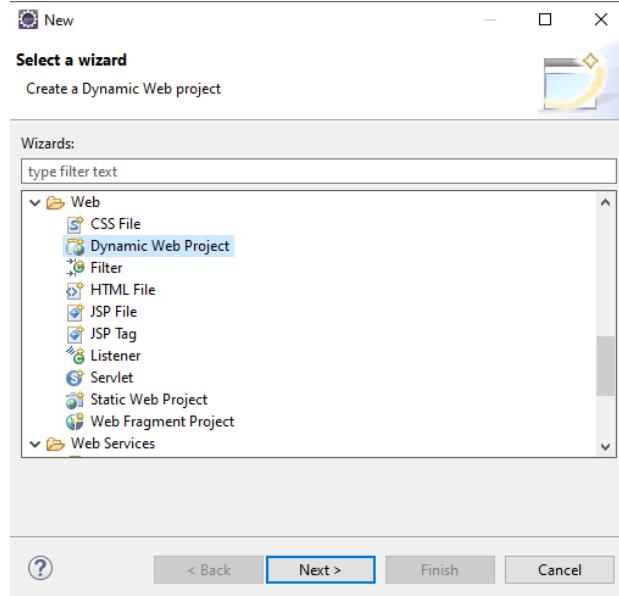


Figure 73

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

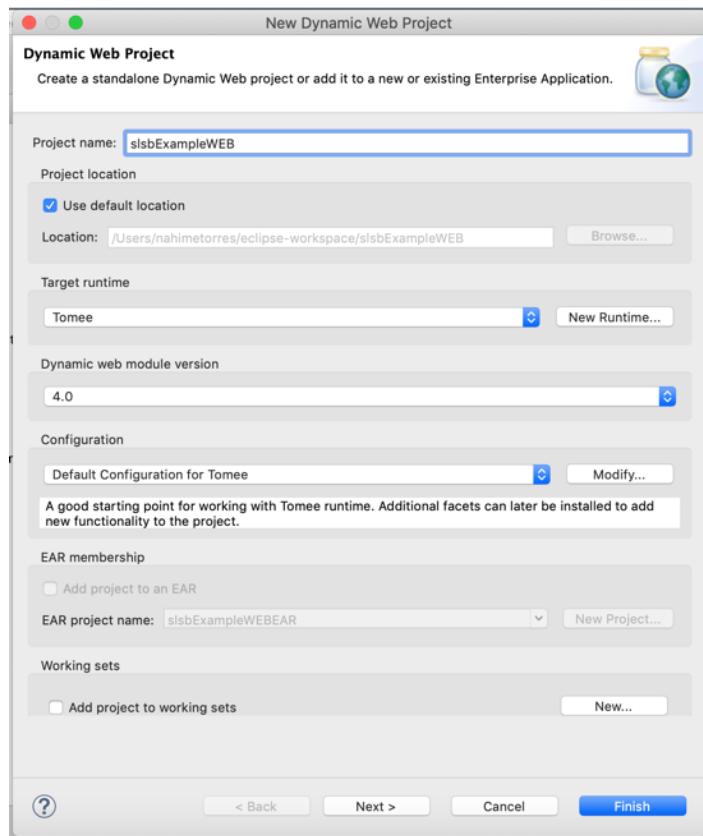


Figure 74

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

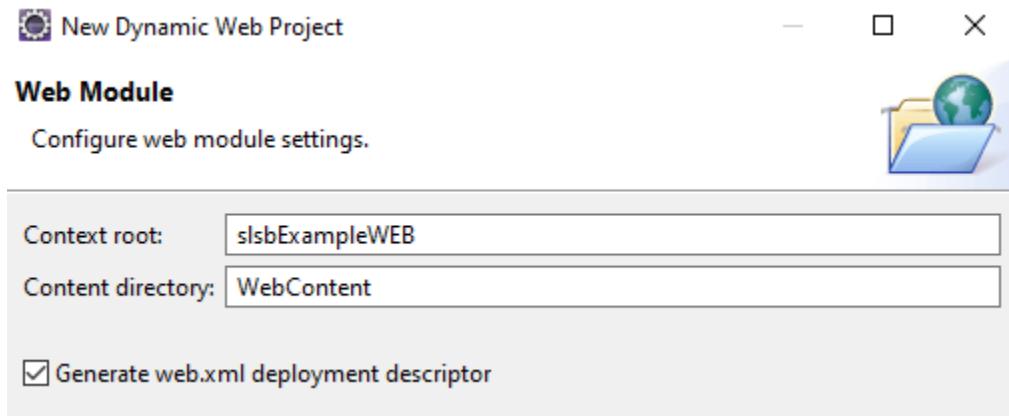


Figure 75

2.- Copy the content of \examples\Chapter3\<example name>\src\servlet\examples\servlet to the Web project, into the src folder. You also have to create the correct package with the same procedure as in the previous steps.

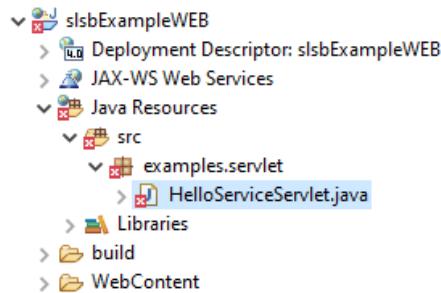


Figure 76

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.

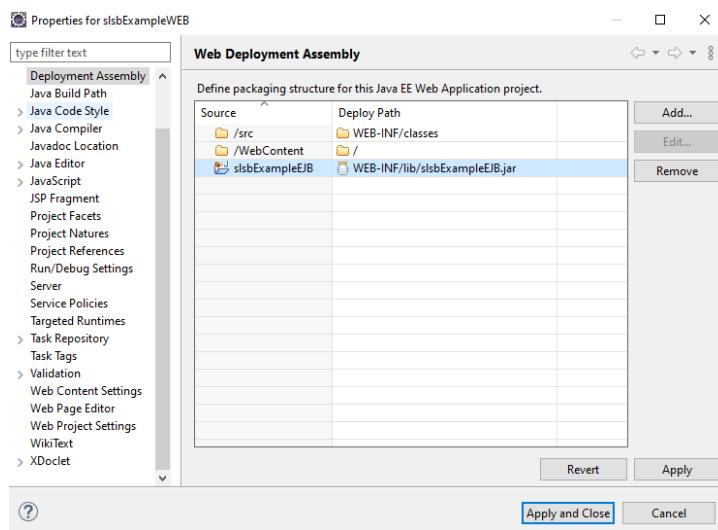


Figure 77

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

```

<?xml version="1.0" encoding="UTF-8"?>
<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>

```

Figure 78

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. **Note that only one EJB project at a time can be deployed to tomee.**

This example demonstrates the basics of defining and accessing a Stateless Session Bean.
Enter a name and click 'Go'. This will trigger a servlet client that talks to a Stateless Session Bean to create a 'hello' String that is then displayed in the browser.

Name:

Hello, Sergio

Figure 79

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.

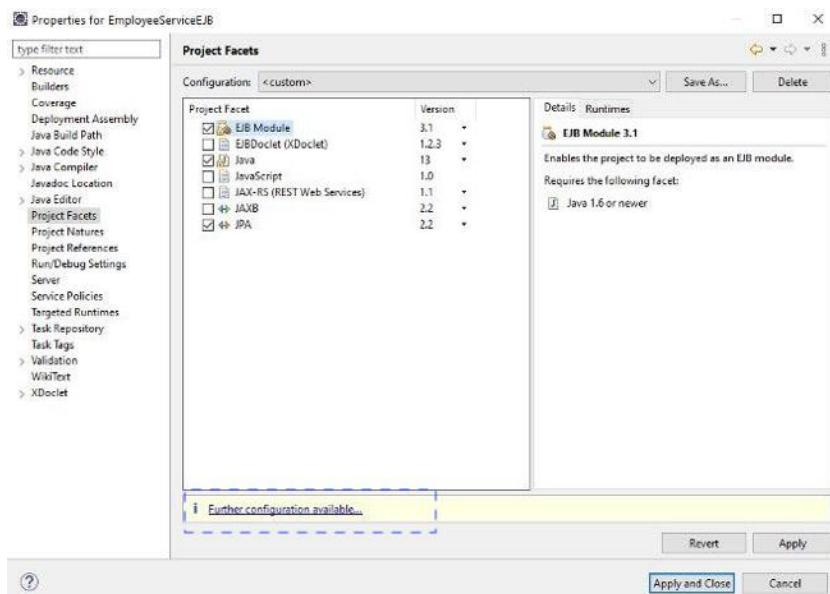


Figure 80

3.- A message indicating that further configuration is needed will appear.

4-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

5- Click on the message, new window will open, on the “JPA implementation” section. if it is your very first JPA project you will need to provide the JPA implementation library as explained in “**Running Chapter 2 – employeeService example**” step 4 to 6.

6. In your server Folder, open the Tomee instance created in precedent steps, look for the tome.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

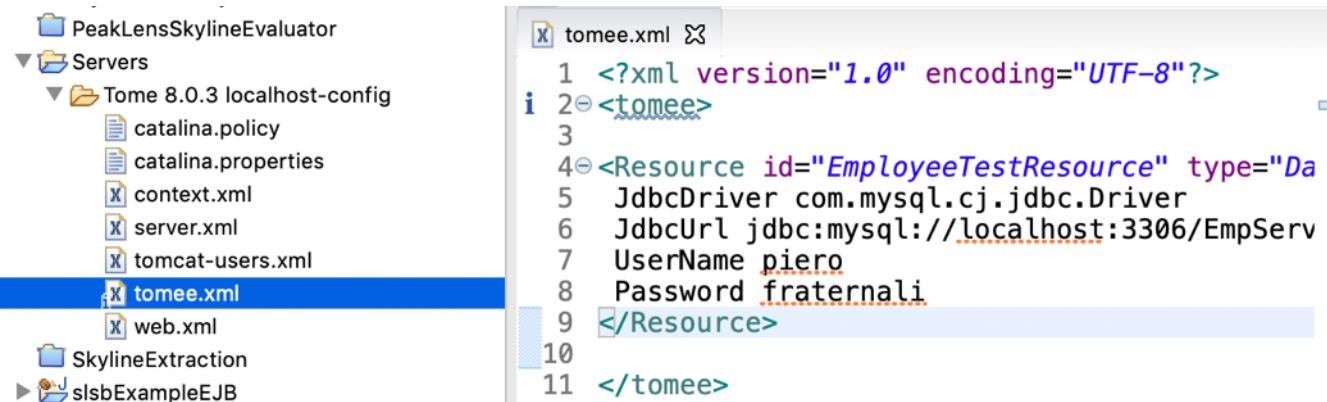


Figure 81

```
<?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.

In the General tab, change the name to be “EmployeeService”.

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”)

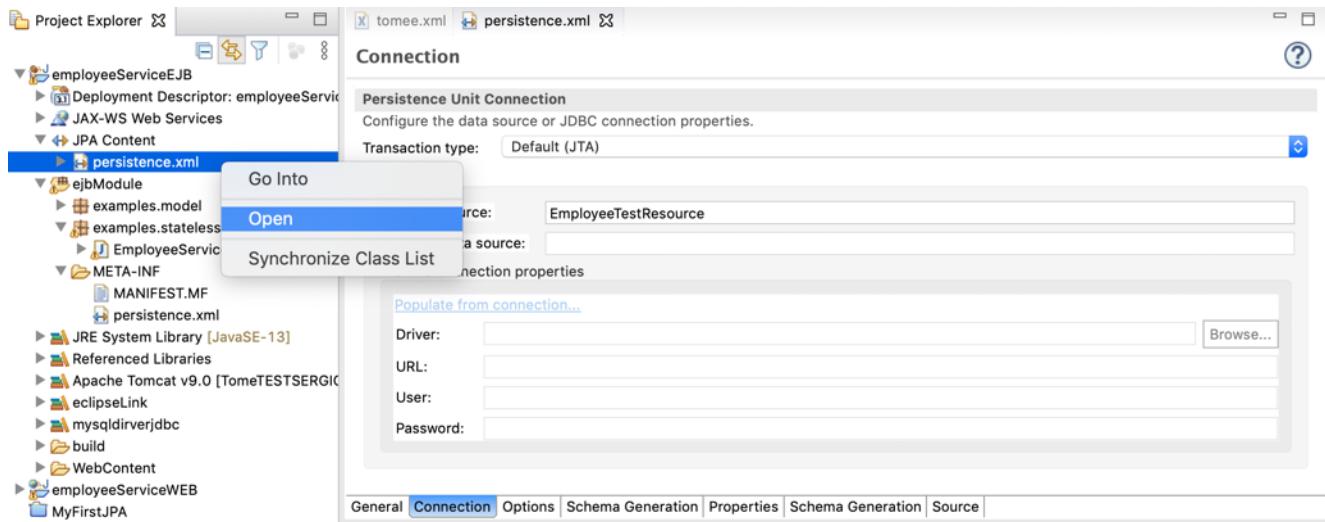


Figure 82

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.

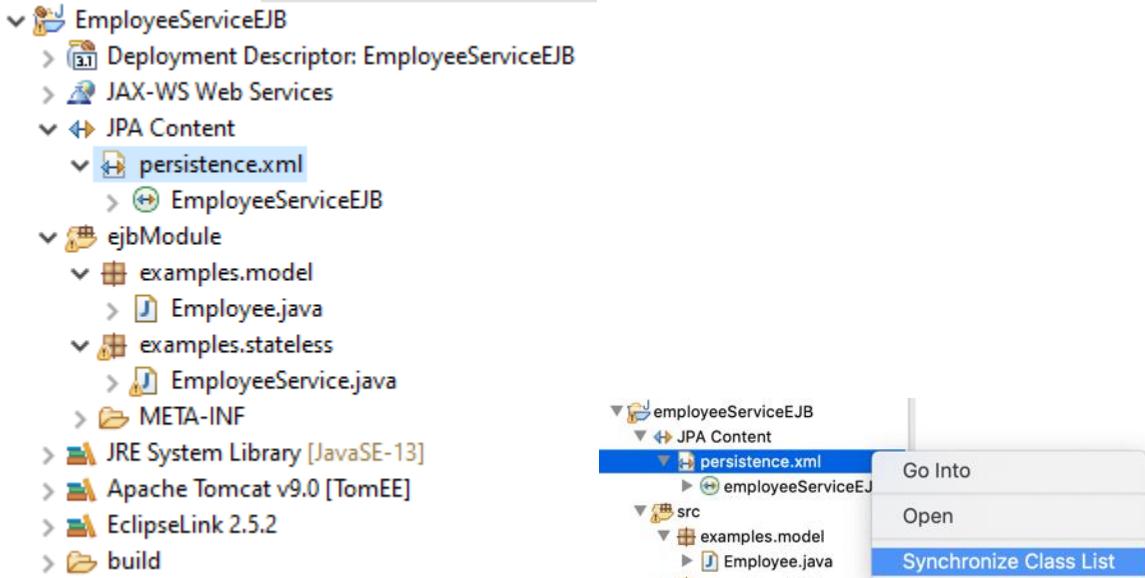


Figure 83

11.- Continue with the steps on for your web application, as explained in section “Web Project” and test your application. **Note that only one EJB project at a time can be deployed to tomee.**

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:

```

1 • CREATE DATABASE IF NOT EXISTS `db_expense_management` /*!40100 DEFAULT CHARACTER SET
2 • USE `db_expense_management`;
3 -- MySQL dump 10.13 Distrib 8.0.19, for Win64 (x86_64)
4 --
5 -- Host: localhost Database: db_gestione_spese
6

```

The status bar at the bottom indicates 'Limit to 1000 rows'.

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*.

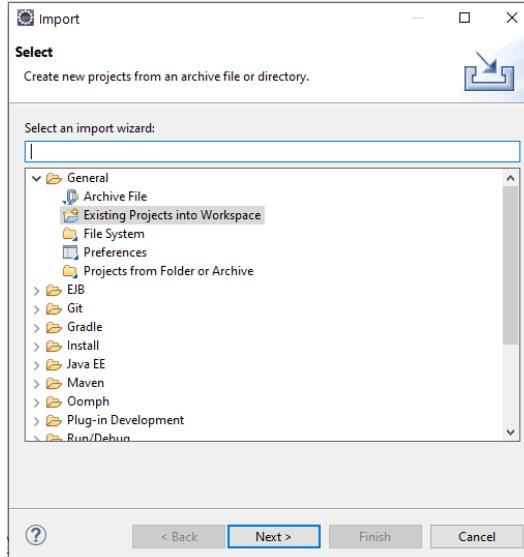


Figure 85

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.

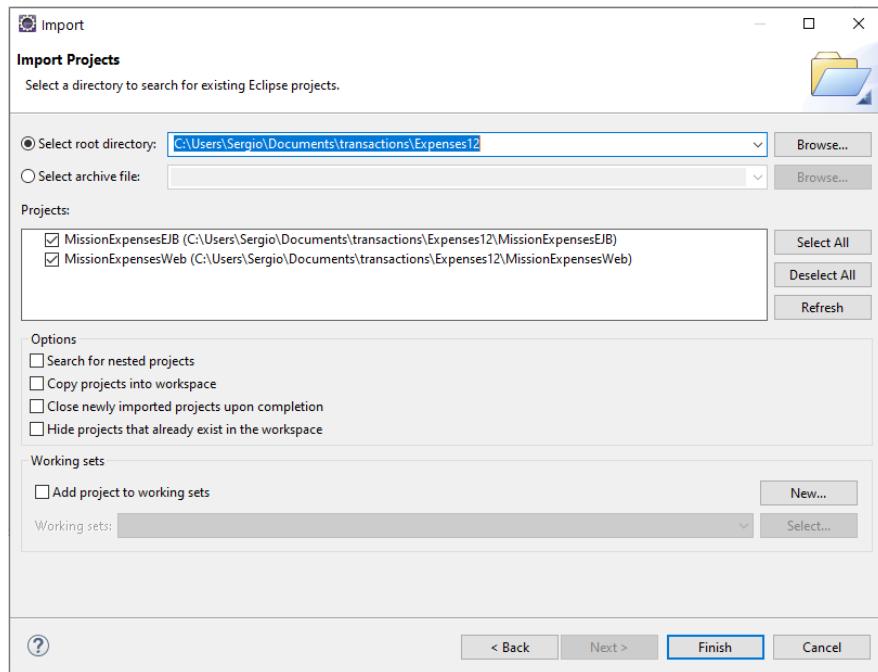


Figure 86

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

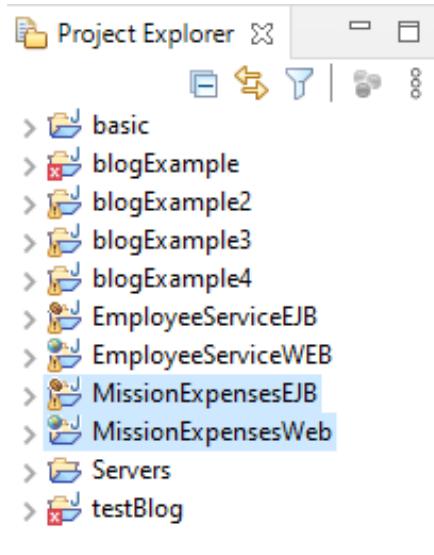


Figure 87

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.

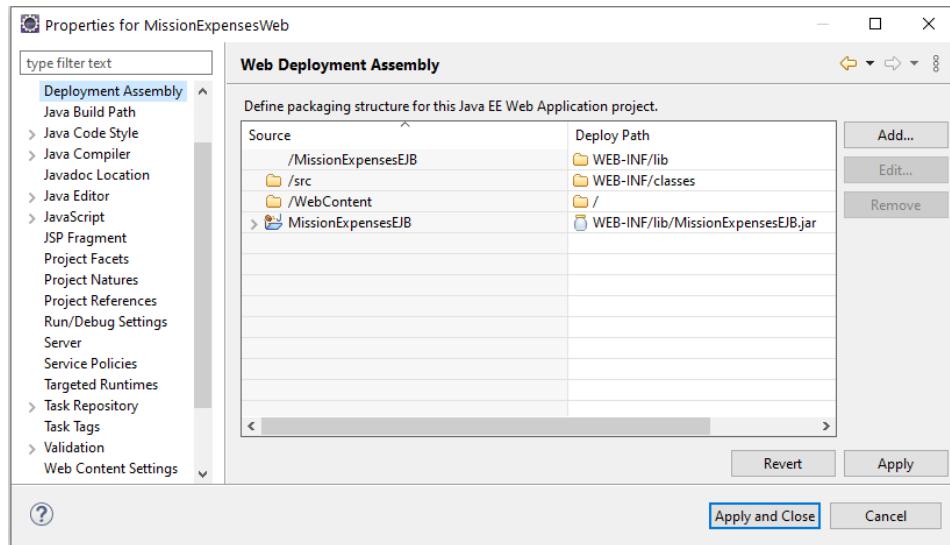


Figure 88

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). If something is marked as “**unbound**” then you must edit it with the current version installed inside your environment.
7. In the project options, “Targeted runtimes” check that the correct runtime is selected.
8. Check also that the java version in the build path, in the project facets and in the java compiler are the same (java 16).
9. If any other error is present, please check the “markers” section inside eclipse.
10. Sometimes, in Linux, table names are case sensitive. If that is your case, you must go in each entities source files and put the `@Table(name="yourtable")` annotation just under the `@Entity` annotation. For each entity, “`yourtable`” is the table name that you see inside mysql workbench.
11. 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>