

EPAW – LAB 1

Goal

In this first assignment, you will learn how to use the environment in which you will work during the course. This includes the installation of different resources and implement a Hello World application plus a connection with a relational MySQL database.

Installing the Software

After this lab, you should be able to have all the software installed and your first dynamic (server side) application running.

1. JDK (Java Development Kit)

<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

The Java Development Kit (JDK) is an implementation of either one of the Java SE, Java EE or Java ME platforms released by Oracle Corporation in the form of a binary product aimed at Java developers on Solaris, Linux, Mac OS X or Windows.

For the course, you should install either Java 6, 7 or 8, and your code should be compatible within versions.

2. Eclipse for Java EE

<https://www.eclipse.org/downloads/packages/eclipse-ide-java-ee-developers/neon3>

Eclipse is an integrated development environment (IDE). It contains a base workspace and an extensible plug-in system for customizing the environment. Written mostly in Java, Eclipse can be used to develop applications. In EPAW we will use it to develop Java Dynamic Web projects and therefore you must download the Java EE version.

For the course, you must use Eclipse for Java EE.

3. Apache Tomcat

<https://tomcat.apache.org/download-70.cgi>

How to install:

<http://tomcat.apache.org/tomcat-7.0-doc/setup.html>

This might be tricky so be patient :) You can find videos and tutorials explaining how to make it work.

Alternatively, you can avoid the installation of Apache Tomcat and configure Eclipse to do the Job (only for developing purposes). In that case, you should download the zip file of the Core version.

<http://www.coreservlets.com/Apache-Tomcat-Tutorial/tomcat-7-with-eclipse.html>

For the course, you can either use tomcat 7 or 8.

4. MySQL

MySQL is a relational database management system (RDBMS), and ships with no GUI tools to administer MySQL databases or manage data contained within the databases.

<http://dev.mysql.com/downloads/>

We will also download the workbench that will help you to interact with the MySQL (as in the first-year Databases subject)

<http://dev.mysql.com/downloads/workbench/>

(this part is optional, you may or may not use the workbench but it is recommended if you are not use to work with MySQL).

Once we have all the software installed and running we will create our first Dynamic Web Project.

1. Open Eclipse IDE for Java EE and create a New Dynamic Web Project:

File --> New --> Other --> Web --> Dynamic Web Project.

We will use our Apache Tomcat as selected server and the Java Runtime Environment.

2. Hello World! Application.

Your first Hello World! Application will be a Servlet; thus, you should create a Servlet within your dynamic web project. Eclipse will generate all the stuff we need. In the `doGet()` method you should generate an HTML output which is a correct webpage that shows the message: Hello World! This is my first dynamic web.

Something like this (see the examples given in theory classes):

```
PrintWriter out = response.getWriter();

    out.println
        (ServletUtilities.headWithTitle(title) +
         "<body>\n" +
         "<h1>" + title + "</h1>\n" +
         "<table style='border: 1px solid black;'" );

...
```

After that, run it in the server and check whether it works.

3. Incorporating a relational MySQL database.

You need to download the MySQL connector:

<http://dev.mysql.com/downloads/connector/j/5.0.html>

once it is downloaded you should include it in your project, you should save it in

-project-/WEB-INF/lib

(where -project- is your project name), and make sure it is included in the web.xml file. You must use eclipse for that.

Before accessing the DB from Servlets, we will play a bit with the MySQL command line client. You should open it, log in with root and create a user which is not “root”, we will call it “mysql”

```
GRANT ALL PRIVILEGES ON *.* TO mysql@localhost IDENTIFIED BY 'prac';
```

Once we do that we should log in with the user:

```
mysql mysql -u mysql -p
```

To launch the mysql client you have to go to the path where the binary is located. Once we are logged, we should create our database and a table:

```
CREATE DATABASE ts1;

USE ts1;

CREATE TABLE taula (
    nom CHAR(24),
    descripcio VARCHAR(255),
    id INTEGER NOT NULL AUTO_INCREMENT,
    telefon INTEGER,
    PRIMARY KEY (id) );
```

Next steps would be to include some data by using an INSERT command several times:

http://www.w3schools.com/sql/sql_insert.asp

Then, you can check its content with:

```
SELECT * FROM TAULA;
```

or use the update command to change it:

http://www.w3schools.com/sql/sql_update.asp

Once we have our database checked and filled with data, we will create a Servlet which will access MYSQL and will basically print (with a PrintWriter) the contents of the DB.

But first we need a java class to connect with the DB (this would be instantiated in our servlet) Something like this will do:

```
import java.sql.*;

public class DAO {

    private Connection connection;
    private Statement statement;

    public DAO() throws Exception {

        String user = "mysql";
        String password="prac";
        Class.forName("com.mysql.jdbc.Driver").newInstance();
        connection=DriverManager.getConnection("jdbc:mysql://localhost/ts1?use
r="+user+"&password="+password);
```

```
        statement=connection.createStatement();
    }

    //execute queries
    public ResultSet executeSQL(String query) throws SQLException{
        return statement.executeQuery(query);
    }
    //TODO: code for updates for Assignments 2, 3 and 4.
    public void disconnectBD() throws SQLException{
        statement.close();
        connection.close();
    }
}
```

Now, the last step is to change the code of the HelloWorld servlet to make it show the data included in the database. You can use an HTML table, list or whatever you want.

You must submit a zip file containing:

1. A WAR file with your application:

Read this to learn how to export your JAVA EE application in a war file:

http://help.eclipse.org/indigo/index.jsp?topic=/org.eclipse.wst.webtools.doc.user/topics/twcr_ewar.html

2. A .sql file with the contents of your database, by exporting it with MYSQL.