ETSF20: Webserver lab

version 1.4.5, Windows

January 30, 2020

1 Introduction

In order to execute the lab assignments you need to know Java, since the programs (Servlets) that run on the Tomcat server are written in Java. You will also need to write some SQL-statements in order to access the database, but no advanced SQL programming will be needed. You do not need any prior knowledge about SQL.

Do not hesitate to ask the teachers if you have problems with the assignments.

2 The system

The system that you will work with is a Tomcat web server with a MySQL database, which is the same kind of system that you will work with in the project.

During the lab, you will use a MySQL DBMS which is installed on a computer at the department (vm23). The Tomcat server will run on a Windows machine in the computer rooms at Campus Helsingborg.

In the lab, Eclipse will be used. For this kind of system, Eclipse EE is preferably used instead of the "normal" Eclipse. One advantage of Eclipse EE compared to the normal Eclipse is that it is possible to run the webserver from within the environment.

3 Lab assignments

You should do all assignments in groups of two students.

You can use the course homepage, laboration, to find the downloads needed during the lab.

3.1 Startup

1. Make sure that you know your log-in information to *vm23 and* MySQL on vm23. If you do not know this, the teacher at the lab session can help you. You need to know the following:

- *username and password on vm23 (all students will use the same account)*
- username, password and database in *MySQL on* vm23 (you will get individual accounts and databases)

2. Log in to a Windows computer.
3. Create a working folder, e.g. `Documents\pusplab`
4. Download Tomcat (zip) from `http://fileadmin.cs.lth.se/cs/Education/etsf20/lab/apache-tomcat-9.0.30-windows-x64.zip` to your working folder
   (`http://tomcat.apache.org/download-90.cgi` 64-bit)
5. Unzip the source code for the Tomcat server in your working folder
6. You need to provide a jdbc-file in order to access the database from java. Put the following file to the `lib`-directory of the Tomcat server:
   `http://fileadmin.cs.lth.se/cs/Education/etsf20/lab/mysql-connector-java-8.0.19.jar`
   (`http://dev.mysql.com/downloads/connector/j/`)
7. Download the example system to your working folder from
   `http://fileadmin.cs.lth.se/cs/Education/etsf20/lab/test.war`
   (Check file extension, should be `war`)
8. Set the environment variables (Win 10):
   1. Click Start menu
   2. Search for environment
   3. Choose Edit environment variables for your account
   4. Choose New
   5. Add:

   ```
   Variable name: JRE_HOME
   Variable value: C:\Program Files (x86)\Java\jre8.0_191
   ```
9. Start Eclipse EE
10. Choose your working folder as your workspace
11. If there is a welcome screen, press "Workbench" in the upper right corner of the window.
12. Let Eclipse know about your server by clicking on the "Servers"-tab on the bottom, and then right click in the bottom pane and choosing "new" → "Server". Select "Apache Tomcat v.9.0" under server type in the first dialogue window. In the next dialogue window provide the path to the Tomcat server on your computer.
13. Import the example system, `test.war`, into Eclipse by choosing "File" → "Import" in the menu, and then follow the instructions. (The file type that you want to import is "Web" → "WAR file".)

3.2 Define the database

Start MySQL Workbench, create a connection to vm23.cs.lth.se and go to pt. 17! Username and password are the same as the ones you got to Epuss

14. Download Putty to your working folder from
    http://www.chiark.greenend.org.uk/ sgtatham/putty/latest.html (putty.exe)
    (if needed)

    Alt. use MySqlWorkbench

15. Start a Putty window and log in to the computer at CS.
    `> vm23.cs.lth.se`
    `> User: etsf20`
    `> Password:` *see mail/ask teacher...*

16. Start MySQL:
    `> mysql -u <user_name> -p`
    `> Password:` *see mail/ask teacher*

17. Show databases that are available (this should include your database):
    `mysql> show databases;`

18. Create a new table in the database:

    ```
    mysql> use <database>;
    mysql> create table Respondents
           (name varchar(100), primary key (name));
    ```

19. Show tables in the database:
    `mysql> show tables;`

20. Show information about the table Respondents:
    `mysql> describe Respondents;`

3.3 Understanding the source code and starting the server

21. Read through all the code and try to understand every part of it. You find the java-files in the explorer pane to the left under "test" → "Java Resources" → "src" → "Default Package". Notice that there are three classes:

    Survey which is the main Servlet

    FormGenerator which is used to build HTML forms

    Database which manages the database connection and communication with the database

22. In the Database-class the login-information to MySQL must be correct. Update this class with your personal log-in information and database name.

23. Select the "Survey"-tab. Press the run-button  in the top-pane of Eclipse. This should start the server (if it is not already started) and start a simple web-browser with the application inside Eclipse.

24. Try out the application. (Try both with a name that never has been used before, and with a name that has already been used.)

25. Stop the webserver by pressing the "Servers"-tab, and then right-clicking on the server and choosing "Stop".

26. Retrieve information about respondents from the database in *the console window (putty)*/MySQL Workbench:
```
mysql> select * from Respondents;
```

3.4 Making changes to the system

27. As you have seen, the system is intended to implement a simple survey, where the user first enters his/her name and then answers four questions concerning the success of a project. The name is stored in the database, but the answers to the four questions are only used to calculate the sum of the four values (which is not meaningful).

    Change the system so that the answers to the four questions are stored in the database. (Define a new table.)

    Test that the updated system works, and look in the database.

28. The system that you have been given first asks for a name and then four questions about project success. The value of the answers about success is however limited since nothing is known about the project.

    Change the system so that it asks two questions that characterize the project before it asks about the success. That is, there should be three different sets of questions:

    - Name (this was implemented from the beginning)
    - Questions characterizing the project (you should implement this. Define a new table)
    - Questions about success (this was implemented from the beginning, but you have made changes so the answers are stored in the database)[1]

    Make the changes in the following steps:

[1] The four metrics were influenced by A. J. Shenar et al. "Refining the search for project success factors: a multivariate, typological approach", R&D Management, 32:2, 2002, pp. 111-126.

(a) Make a "design"/"description" of the changes you want to make. Use pen and paper.

(b) Implement the changes.

29. Make a new servlet that displays the contents of the database.

3.5 Starting and stopping the webserver outside Eclipse

30. If the server in Eclipse is still running, stop it.

31. In a command window, go to the Tomcat bin-folder and run the startup-command:
`> startup.bat` (shutdown.bat)

32. Go to the following URL in a web browser: `http://localhost:8080/`
This should display the start page of the Tomcat server.

33. Export the project as a war-file from Eclipse by choosing "File" → "Export", and then "WEB" → "WAR file", and then choosing the test-project and the `webapps`-folder in the Tomcat folder as project and destination.

34. Go to the following URL: `http://localhost:8080/test/Survey` This should start the application in the web browser.

3.6 Base system - *if you have time*

35. Download the base system to your working folder from
`http://fileadmin.cs.lth.se/cs/Education/etsf20/lab/BaseBlockSystem.war`
(Check file extension, should be `war`)
System documentation, see course homepage.

36. Import the base system into Eclipse - use the same steps as in the previous parts of the exercise.

37. Implement the following functionality: The system should support setting users as inactive/active. All informatiom regarding an inactive user should remain in the database. The inactive user should be refused to log in. For information about the database structure, see servletBase.java.

3.7 End of lab session

38. Make a list of suggested improvements to the test system that you were given. That is, what changes do you suggest that we make until next year?

39. Show the teacher that the system is working.

40. In a command window, go to the Tomcat folder and run the startup-command:
    ```
    > shutdown.bat
    ```
    It is important that you do this.