

How to Set Up Development Environment

1. Install Software/Dependencies
 - a. Install Eclipse IDE
 - i. The installation file can be found on the following website:
 1. <https://www.eclipse.org/downloads/packages/eclipse-ide-java-developers/oxygen1a>
 - ii. Ensure that the correct OS installation file for your machine is selected (can be found below 'Download Links' on website).
 - b. Install MySQL Workbench
 - i. The installation file can be found on the following website:
 1. <https://dev.mysql.com/downloads/workbench/>
 - ii. Ensure that the correct OS is selected (can be found below 'Select Operating System: ').
 - iii. Click 'Download' button
 - c. Install MySQL Server
 - i. The installation file can be found on the following website:
 1. <https://dev.mysql.com/downloads/mysql/>
 - ii. Ensure that the correct OS is selected (can be found below 'Select Operating System: ').
 - iii. Click 'Download' button
 - d. Install webapprunner
 - i. The direct download of the .jar file can be found on the following website:
 1. <http://central.maven.org/maven2/com/github/jsimone/webapp-runner/7.0.82.0/webapp-runner-7.0.82.0.jar>
 - ii. Once the .jar has downloaded to your machine, change its location to one that is easy to find for you.
 1. The .jar is used in instructions below (*How to Run the System*).
2. Get Project Files on Machine
 - a. Open the terminal/command line prompt for your machine.
 - i. iTerm or any other terminal/command line prompt replacements are acceptable if this is your preference.
 - b. Enter 'git clone https://github.com/ashhcree17/cs414-f17-801-Ashton.git' into the terminal/command line prompt on your machine.
 - i. This step copies the project source code onto your machine.
3. Get Project Files into Eclipse IDE
 - a. Open Eclipse IDE.
 - b. A prompt to enter an Eclipse workspace name will appear. This workspace name is not important for successful access to the project, so its name can be anything (e.g., "eclipse-workspace", "cs414-workspace", etc).

- c. Select 'File'
 - d. Select 'Import...'
 - e. Select 'Gradle'
 - f. Select 'Existing Gradle Project'
 - g. Select 'Browse...' (location: to right of 'Project root directory' input bar).
 - h. Navigate to the location of the previously cloned git repository folder.
 - i. Click 'Open' button (project source files will appear)
4. Start MySQL Server
- a. For a Windows machine, enter ' "C:\Program Files\MySQL\MySQL Server 5.7\bin\mysqld" --console' (adjust path name as necessary)
 - b. For a Macintosh machine, enter 'sudo launchctl load /Library/LaunchDaemons/mysql.agent.plist' (path name should not need to be adjusted)
5. Get Project Database into MySQL Workbench
- a. Open MySQL Workbench.
 - b. Create New Connection (and Connect)
 - i. Click [+] button to right of 'MySQL Connections' title seen on the home screen.
 - ii. In the resulting 'Setup New Connection' form that displays, do not enter information or make any changes from what is default.
 - iii. Click 'OK' button.
 - iv. Select the newly created 'Local instance 3306' database to open.
 - c. Import Data
 - i. Select 'Data Import/Restore' under the 'Management' section of the Workbench left side panel.
 - ii. Click the radio button to the left of 'Import from Self-Contained File'.
 - iii. Check if the input field found to the right of 'Import from Self-Contained File' has a file auto-populated in it.
 - 1. If this file is the .sql file found in the project git repository previously cloned onto your machine, you may continue to Step iv.
 - 2. If this file is not the .sql file or there is no file auto-populated, then click the '...' button.
 - a. Navigate to the location of the previously cloned git repository folder.
 - b. Expand the 'docs' folder in the git directory
 - c. Select the file named 'globo_gym_db.sql'.
 - iv. Click 'Start Import' button at bottom right of screen.

How to Run the System

1. Open the terminal/command line prompt for your machine.

- a. iTerm or any other terminal/command line prompt replacements are acceptable if this is your preference.
2. Enter 'gradle clean build' into command line.
 - a. This step builds the WAR file
 - i. WAR (Web application ARchive) files are used to distribute Java-based web applications and have same file structure as a JAR file.
3. Navigate to the globoGymMS/build directory that is created on your machine (location of the git repository previously cloned)
 - a. Navigation to this directory takes place in your machine's file explorer.
4. Copy (DON'T cut. Please only copy) webapp-runner-8.5.23.1.jar from its location on your machine to the libs folder found in globoGymMS/build directory.
5. Enter 'java -jar libs/webapp-runner-7.0.82.0.jar libs/*.war'
6. Visit 'localhost:8080/globoGymMS/' into address bar of an Internet browser.
7. Steps 2-5 must be repeated before Step 6 if you make changes to the code and want to see them reflected/built into the WAR file.

How to Run Tests

1. By Command Line
 - a. Enter 'gradle test' into command line
2. By Eclipse
 - a. Expand 'src' folder in 'globoGymMS' directory
 - i. Expand src --> test --> java --> globoGymMS, AND
 - ii. Expand src --> integrationTest --> java --> globoGymMS
 - b. Right-click the folder 'globoGymMS' for src/test/java
 - c. Select 'Run As' --> 'JUnit Test'
 - d. Repeat Steps b. and for src/integrationTest/java

References

1. <https://dev.mysql.com/doc/workbench/en/wb-admin-export-import-management.html>
2. <https://dev.mysql.com/doc/mysql-monitor/3.4/en/mem-starting-agent-osx.html>
3. <https://dev.mysql.com/doc/refman/5.7/en/windows-server-first-start.html>
4. <https://dev.mysql.com/doc/workbench/en/wb-mysql-connections-new.html>
5. <https://stackoverflow.com/questions/17371639/how-to-store-arrays-in-mysql>
6. https://www.w3schools.com/sql/sql_primarykey.asp
7. <http://www.joinfu.com/2005/12/managing-many-to-many-relationships-in-mysql-part-1/>
8. <https://www.journaldev.com/3524/spring-hibernate-integration-example-tutorial>
9. <https://www.journaldev.com/3531/spring-mvc-hibernate-mysql-integration-crud-example-tutorial>

10. <https://stackoverflow.com/questions/5862680/whats-the-advantage-of-persist-vs-save-in-hibernate>
11. <https://stackoverflow.com/questions/5776098/hibernate-session-delete-an-object-if-exists>
12. <https://devcenter.heroku.com/articles/java-webapp-runner>
13. <https://stackoverflow.com/questions/28934952/unable-to-start-spring-boot-application>
14. <https://stackoverflow.com/questions/8955074/generatedvaluestrategy-identity-vs-generatedvaluestrategy-sequence>
15. <https://stackoverflow.com/questions/8955074/generatedvaluestrategy-identity-vs-generatedvaluestrategy-sequence>
16. <https://www.journaldev.com/3531/spring-mvc-hibernate-mysql-integration-crud-example-tutorial>
17. <https://docs.oracle.com/javaee/7/api/javax/persistence/ManyToMany.html>