**Setting up ORM for Java/MySQL**

Suggested way to set up ORM for Java/MySQL.

1. Use the IDE Eclipse
2. In Eclipse, navigate to Help -> Install New Software.
   1. Enter "http://download.jboss.org/jbosstools/neon/stable/updates/" in the Work With box.
   2. Enter "Hibernate" into the filter box.
   3. Select Hibernate Tools.
   4. Continue to follow the wizard for installation

Graphical user interface, text, application, email

Description automatically generated

1. Now you will need to go to the MySQL page and download (either install or upgrade) **MySQL Connector**/**J** -- the official JDBC driver for MySQL

Graphical user interface, application

Description automatically generated

1. Go to the Window → Preferences in Eclipse. Expand the Data Management → Connectivity → Driver Definitions and click on the Add button.
2. In the appeared New Driver Definition dialog, select the MySQL in the vendor filter combo box and select the latest MySQL JDBC Driver from the listed drivers.  
     
   Graphical user interface, table

   Description automatically generated
3. Switch to the JAR List tab and add the mysql-connector jar.  
     
   Graphical user interface, application

   Description automatically generated  
     
     
   Edit the Jar and find the mysql connector jar you downloaded previously.  
     
   Graphical user interface, text, application

   Description automatically generated
4. Graphical user interface, text, application, chat or text message

   Description automatically generatedNow make a new Maven project called hibernate-core that looks like this  
     
     
   The code for the two java classes and the xml files can be found here: <http://jackmyers.info/db/files/orm_java_mysql/>

Graphical user interface, text, application

Description automatically generated

1. You will need to set up MySQL (local) like this. (If you are installing MySQL locally, play attention to the wizard so that you write down your root password!)  
     
   Table

   Description automatically generated
2. Make sure that your hibernate.cfg.xml file properly sets up your connection to the database.  
     
   <?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE hibernate-configuration PUBLIC

"-//Hibernate/Hibernate Configuration DTD 3.0//EN"

"http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">

<hibernate-configuration>

<session-factory>

<!-- Database connection settings -->

<property name="connection.driver\_class">com.mysql.jdbc.Driver</property>

<property name="connection.url">jdbc:mysql://localhost:3306/music</property>

<property name="connection.username">root</property>

<property name="connection.password">p@ssw0rd</property>

<property name="show\_sql">true</property>

<mapping class="Song" />

</session-factory>

</hibernate-configuration>

1. When you can successfully run the Driver, your environment is set up for hibernate, java, jpa and MySQL.x
2. --------------------------------------------------------------------------------
3. Hibernate: insert into song (artist, length\_sec, song\_name) values (?, ?, ?)
4. Hibernate: insert into song (artist, length\_sec, song\_name) values (?, ?, ?)
5. --------------------------------------------------------------------------------
6. Title: I Feel Like I'm Drowning
7. Artist: Two Feet
8. Length: 3:05
9. --------------------------------------------------------------------------------
10. Title: Under Pressure
11. Author: Queen
12. Length: 4:06
13. --------------------------------------------------------------------------------
14. Hibernate: update song set artist=?, length\_sec=?, song\_name=? where song\_id=?
15. --------------------------------------------------------------------------------
16. Title: Under Pressure
17. Author: Queen [feat. David Bowie]
18. Length: 4:06
19. --------------------------------------------------------------------------------
20. Hibernate: delete from song where song\_id=?