# Town Wizard DB Application Development Environment Setup

Town Wizard DB is a Java application which makes use of the following technologies:

1. Java SE 7
2. Eclipse (version 4 “Juno” for Java EE at the time or writing, but earlier versions will do, too) (<http://www.eclipse.org>)
3. Maven 3 (<http://maven.apache.org>)
4. Maven plugin for Eclipse (<http://www.sonatype.org/m2eclipse>)
5. Apache Tomcat 7 (<http://tomcat.apache.org>)
6. Sysdeo Tomcat plugin for Eclipse (<http://www.eclipsetotale.com/tomcatPlugin.html>)
7. Java frameworks
   1. Jersey Framework (<http://jersey.java.net>)
   2. Hibernate 4 (<http://hibernate.org>)
   3. Spring 3 (<http://www.springsource.org/spring-framework>)

Special installation is needed for all except for java frameworks, which are downloaded by maven.

This document also assumes that git and MySQL are already installed in the system.

## Java SE 7

1. Download JDK for your system/architecture from <http://www.oracle.com/technetwork/java/javase/downloads/index.html>. Make sure to download JDK not JRE.
2. On Windows, after installation, set the following environment variables:  
   JAVA\_HOME=<jdk\_installation\_directory>  
   PATH=<your\_current\_path>;%JAVA\_HOME%\bin
3. On Mac, after installation go to System Preferences -> Java and switch to Java 7.
4. Check: in terminal window run  
   javac –version  
   Should report version 1.7.x\_xx

## Eclipse

1. Download Eclipse 4 (Juno) for Java EE for your system from <http://www.eclipse.org/downloads/>
2. Unzip the content of the archive (zip or tar) to the directory of your choice. On my system it is  
   ~/dev/eclipse
3. Start it by double clicking on the eclipse.exe on Windows or eclipse on Mac (you could also make a shortcut for it)
4. Once started, if you see Eclipse welcome page, click on Workbench link in the right upper corner to close it.
5. Switch to Java perspective rather than Java EE perspective (Windows -> Open Perspective -> Java) and close Java EE perspective
6. Point Eclipse to Java 7 installation. (Eclipse/Edit -> Preferences -> Java -> Installed JREs). Make sure the Java SE 7 is present and checked. If not present, click on the Add button, and navigate to your JAVA\_HOME directory (on my Mac it is /Library/Java/JavaVirtualMachines/jdk1.7.7\_07.jdk/Contents/Home)
7. Set up editor setting to use spaces instead of tabs
   1. Preferences -> General -> Editors -> Text Editors -> Check Insert spaces for tabs (also, make sure the tab width is 4). It is also useful to check Show line number and Show print margin on that page (set print margin to 100)
   2. Preferences -> Java -> Code Style -> Formatter -> New -> Give a new profile any name (for example Salzinger) -> Ok -> Check Use spaces to indent wrapped lines and change Tab policy to Spaces only -> Ok
   3. Preferences -> XML -> XML Files -> Editor -> Check Indent using spaces and set indentation size to 2 (2 – for XML, 4 – for Java code)
   4. In general, whenever new type of files is being edited by Eclipse, one should make sure tabs are not used for indentation
8. Disable spelling: Preferences -> General -> Editors -> Text Editors -> Spelling -> Uncheck Enable spell checking
9. Keep eclipse open

## Maven

1. First, check if Maven is already installed in your system (which might happen for Macs). Run from terminal:  
   mvn –version  
   If reports something like Apache Maven 3.0.3, then you might skip the download and installation steps
2. Download maven version 3 zip file from <http://maven.apache.org/download.html>.
3. Unzip it in a directory of your choice (mine is ~/dev/apache-maven-3.0.4)
4. On Windows (and on Mac if you decide to go with the downloaded version rather then preinstalled one), set the following environment variables
   1. MAVEN\_HOME=<maven\_installation\_directory>
   2. PATH=<current\_path>;%MAVEN\_HOME%\bin
5. Check: mvn –version

## Maven Plugin for Eclipse

1. In Eclipse, go to Help -> Install New Software -> Add -> Type m2eclipse as name
2. Go to <http://eclipse.org/m2e/download/> and copy the link on the page for the latest version under Update Sites to the Location field (which is <http://download.eclipse.org/technology/m2e/releases> at the time of writing).
3. Maven Integration for Eclipse option should appear in the box. Select everything -> Next -> Next -> Accept terms of license agreement -> Finish. Wait till it installs, and say yes to restart Eclipse
4. Point maven plugin to your Maven installation (Eclipse will use its embedded maven installation otherwise, which will be different, and it’s not a good idea to use different maven installations from inside and outside of eclipse): Preferences -> Maven -> Installation -> Add -> Point to your Maven home directory (if you use Mac preinstall version it might look something like /usr/share/maven). Make sure it is checked. Click Ok.

## Tomcat

1. Download Tomcat version 7 from <http://tomcat.apache.org/> -> Download Tomcat 7 -> Core zip
2. Unzip to the directory of your choice (mine is ~/dev/apache-tomcat-7.0.32)
3. Make Tomcat startup/shutdown scripts executable (on Mac):  
   chmod +x <tomcat\_installation\_directory>/bin/\*.sh
4. Start it:   
   <tomcat\_installation\_directory>/bin/startup.sh
5. Navigate in the browser to localhost:8080. You should see the Tomcat welcome page
6. Stop it:  
   <tomcat\_installation\_directory>/bin/shutdown.sh
7. Also, verify that Tomcat works out of JDK 7, that is look at the environment variables Tomcat reports on startup/shutdown, like:  
   Using CATALINA\_BASE: /Users/j2vm/dev/apache-tomcat-7.0.32  
   Using CATALINA\_HOME: /Users/j2vm/dev/apache-tomcat-7.0.32  
   Using CATALINA\_TMPDIR: /Users/j2vm/dev/apache-tomcat-7.0.32/temp  
   Using JRE\_HOME: /Library/Java/JavaVirtualMachines/jdk1.7.0\_07.jdk/Contents/Home  
     
   If JRE\_HOME is incorrect export the correct JAVA\_HOME (yes, JAVA, not JRE) as  
   export JAVA\_HOME=/Library/Java/JavaVirtualMachines/jdk1.7.0\_07.jdk/Contents/Home  
     
   You can run it as a separately before lunching Tomcat or add this line at the top of the **<tomcat\_home>/bin/catalina.sh** Tomcat startup script.
8. Optionally, increase Tomcat memory by creating **setenv.sh** file in the Tomcat’s bin directory:  
   vim /Users/j2vm/dev/apache-tomcat-7.0.32/bin/setenv.sh  
   --add line  
   export CATALINA\_OPTS="-Xms256m -Xmx512m"  
   sudo chmod +x setenv.sh

## Tomcat Plugin for Eclipse

1. Download the latest plugin from <http://www.eclipsetotale.com/tomcatPlugin.html> (tomcatPluginV33.zip at the time of writing)
2. Unzip and move the entire **com.sysdeo.xxx** directory to <eclipse\_directory>/**dropins** directory.
3. Restart Eclipse
4. Check: On Eclipse bar you should see three additional buttons to start/stop/restart Tomcat
5. Adjust Tomcat settings in Eclipse
   1. Preferences -> Tomcat -> Check radio Version 7.x
   2. Preferences -> Tomcat -> Point Tomcat home to your Tomcat home directory
   3. Preferences -> Tomcat -> Make sure Context files is chosen as Context declaration mode, and context directory points to <tomcat\_home>/conf/Catalina/localhost
   4. Check: click start Tomcat button on the Eclipse toolbar. Go to localhost:8080 and see the Tomcat home page. Stop Tomcat.

## Import Town Wizard DB Application into Eclipse

1. Download the application from github if not done.
2. Run SQL scripts against your local MySQL database. The database scripts are organized as migrations (the idea borrowed from Rails), where a migration is a separate discrete state of DB schema (and possibly some configuration or read-only data). Current DB migration number is saved in the **Migration** table, so at each point it is clear, what is the current state of the DB schema. For each migration, there is an UP script which brings DB to the next migration, and DOWN script, which rolls back the changes.  
     
   For now, you need to run all migrations manually as:  
   mysql –uroot –p < <your\_project\_directory>/sql/0/up.sql  
   mysql –uroot –p < <your\_project\_directory>/sql/1/up.sql  
     
   To “rollback” migration 1, for example, run:  
   mysql –uroot –p < <your\_project\_directory>/sql/1/down.sql  
     
   There is no DOWN script for migration 0, because it only creates the Migration table.
3. Insert some sample data:  
   mysql –uroot –p < <your\_project\_directory>/sql/sample\_data/up.sql  
     
   The DB should have now at least one user with address.
4. Import application project into Eclipse.
   1. File -> Import -> Maven -> Existing Maven Projects -> Next
   2. Browse to your project directory – a checked /pom.xml file should appear in the box. Hit Next and then Finish.
5. Build the project in Eclipse: Right click on the project in the project explorer -> Ran As -> Maven install. Watch maven console. For the first time it may take a while to build the project, for Maven needs to download and install all the project dependencies from the Internet. Make sure Maven reports BUIlD SUCCESS in the end.  
     
   The result of Maven install command will be a “target” directory under your project filled with Maven artifacts:
   1. **townwizard-db.war** file which is a Tomcat deployable unit (the entire application in one file with all the Java code and dependencies)
   2. **townwizard-db-<version>** directory, which contains the expanded version of the townwizard-db.war file
   3. **surefire-reports** directory with the result of unit tests
   4. some other artifacts

## Running the Application in Tomcat

At this point the application is ready for deployment. There are two basic deployment options:

* Copying the war file under our Tomcat installation (this is how we may install it on production)
* Pointing Tomcat to the directory with “expanded” version of the application, created by Maven (this is what we’d like to do for development, since any changes in Java classes will be picked up by Tomcat and server restart will not be required)

For the second option, do:

1. Create the following file:  
   <tomcat\_installation\_directory>/conf/Catalina/localhost/tw.xml
2. Add this one file to that file, close and save  
   <Context docBase=”<your\_project\_directory>/target/townwizard-db-1.0.0”/>

Now you’re ready to run it. To run from terminal by starting Tomcat, execute in terminal  
<tomcat\_installation\_directory>/bin/startup.sh

Navigate to a sample service in your browser as: **localhost:8080/tw/users/1**. You should see JSON representing a user and his address.

Now, let’s try running it from Eclipse.

Shutdown tomcat: <tomcat\_installation\_directory>/bin/shutdown.sh. The service will no longer be accessible in the browser.

In Eclipse, click on the start Tomcat button on top. Watch Eclipse console till it reports that the server has started, and the sample web service should be available in your browser again.

You may shut down Tomcat in Eclipse by clicking on the stop Tomcat button, or just by clicking on the red square button in the Console window.

## Running the Application in Jersey standalone server (optional)

It’s also possible to run the application as a standalone server. For that,

1. Make sure the Tomcat is stopped, and then right-click on the project -> Ran As -> Java Application.
2. The dialog box will show you all available java applications. Choose  
   **Standalone** – **com.townwizard.db.application.** You only have to do it once. Once you’ve done it the Standalone menu item will appear under “Run” button Eclipse dropdown in Eclipse toolbar.
3. Once it starts, the sample web service is available in your browser. To stop it just hit the red square button.

## Enable Virtual Hosting in Apache Server

For Facebook connect, the townwizard db Tomcat installation must be available as a public server (**tw‑db.com** for now), and Apache which is used to run townwizard PHP application needs to know how to redirect calls to that server URL to Tomcat. To make work on a development environment one might do:

1. Add tw-db.com to hosts file as:   
   127.0.0.1 www.townwizardconnect.com  
   127.0.0.1 townwizardconnect.com  
   127.0.0.1 www.townwizardconnectinternal.com  
   127.0.0.1 townwizardconnectinternal.com
2. Modify /etc/apache2/extra/httpd-vhosts.conf.  Add the following lines:  
      
   <VirtualHost \*:80>  
      ServerName demo.townwizard.com  
   </VirtualHost>  
     
   <VirtualHost \*:80>  
    ServerName masterdefault.com  
    ServerAlias www.masterdefault.com  
   </VirtualHost>  
     
   <VirtualHost \*:80>  
      ServerName townwizardconnect.com  
      ServerAlias www.townwizardconnect.com  
      ProxyPass / http://localhost:8080/tw/  
      ProxyPassReverse / http://localhost:8080/tw/  
   </VirtualHost>  
     
   <VirtualHost 127.0.0.1:80>  
      ServerName townwizardconnectinternal.com  
      ServerAlias www.townwizardconnectinternal.com  
      ProxyPass / http://localhost:8080/tw/  
      ProxyPassReverse / http://localhost:8080/tw/  
     
    RewriteEngine on  
    RewriteRule !^/login /forbidden  
   </VirtualHost>  
     
   Add more virtual host declarations if more Townwizard partner URLs are used.
3. Make sure this configuration file is parsed by Apache at startup. Look at the */etc/apache2/httpd.conf* and uncomment the following line if commented:  
     
   Include /private/etc/apache2/extra/httpd-vhosts.conf
4. Restart apache

To check, paste this URL into your browser: [*http://*townwizardconnect.com*/users/1*](http://townwizardconnect.com/users/1)*.* This should display either empty JSON or a user JSON representation (providing your Tomcat is up).