Component-based Software Development

Introduction to Computing Platform Java, Tomcat, Eclipse, JPA configurations

Dr. Vinod Dubey
SWE 645
George Mason University

JDK:
Installation and Setup

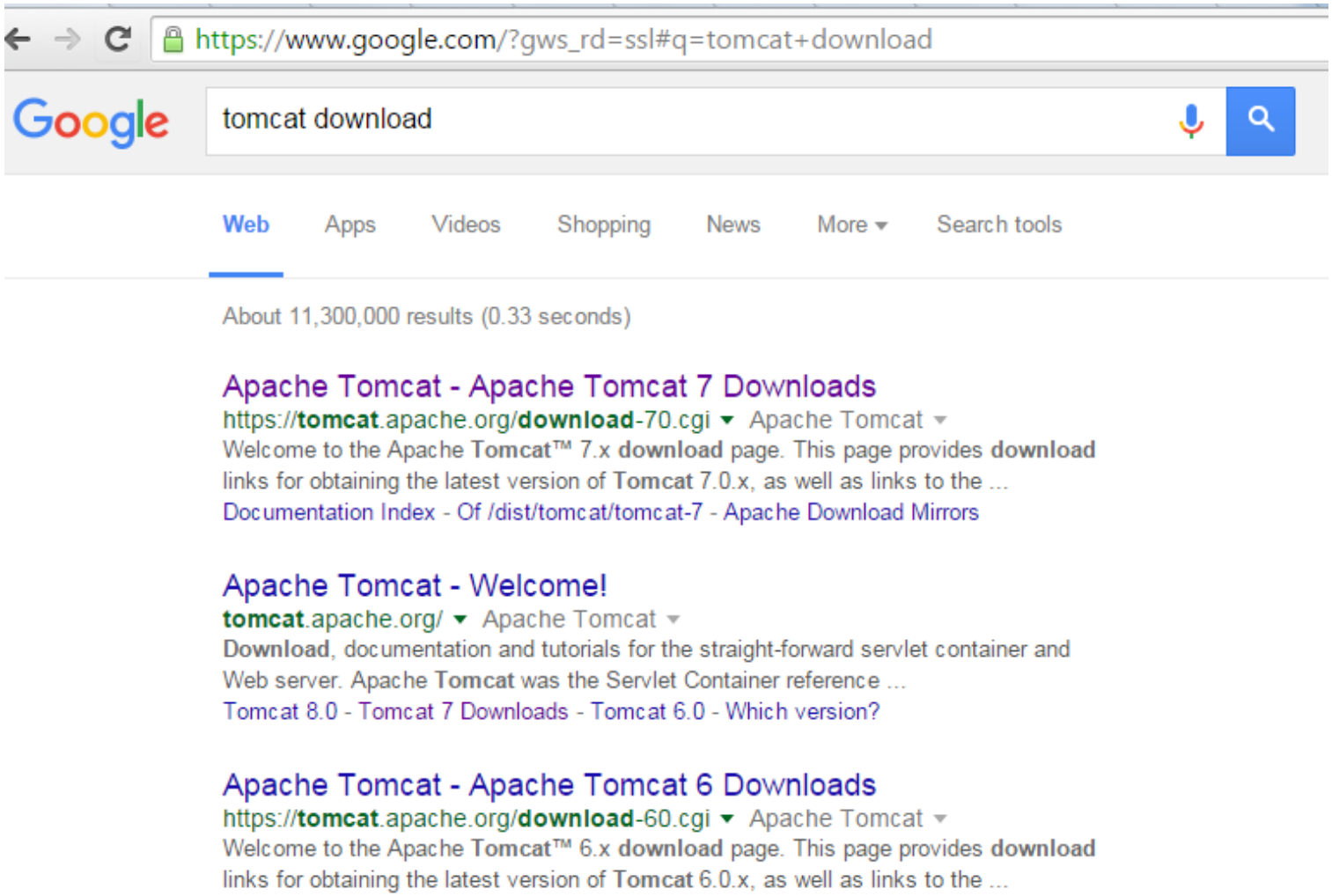
Download and Install JDK1.7

- **Tomcat or JBoss requires JVM environment**
  - The first mandatory requirement is to install a JDK 1.7 environment where Tomcat or JBoss will run
  - **The Java SE download site can be found at**

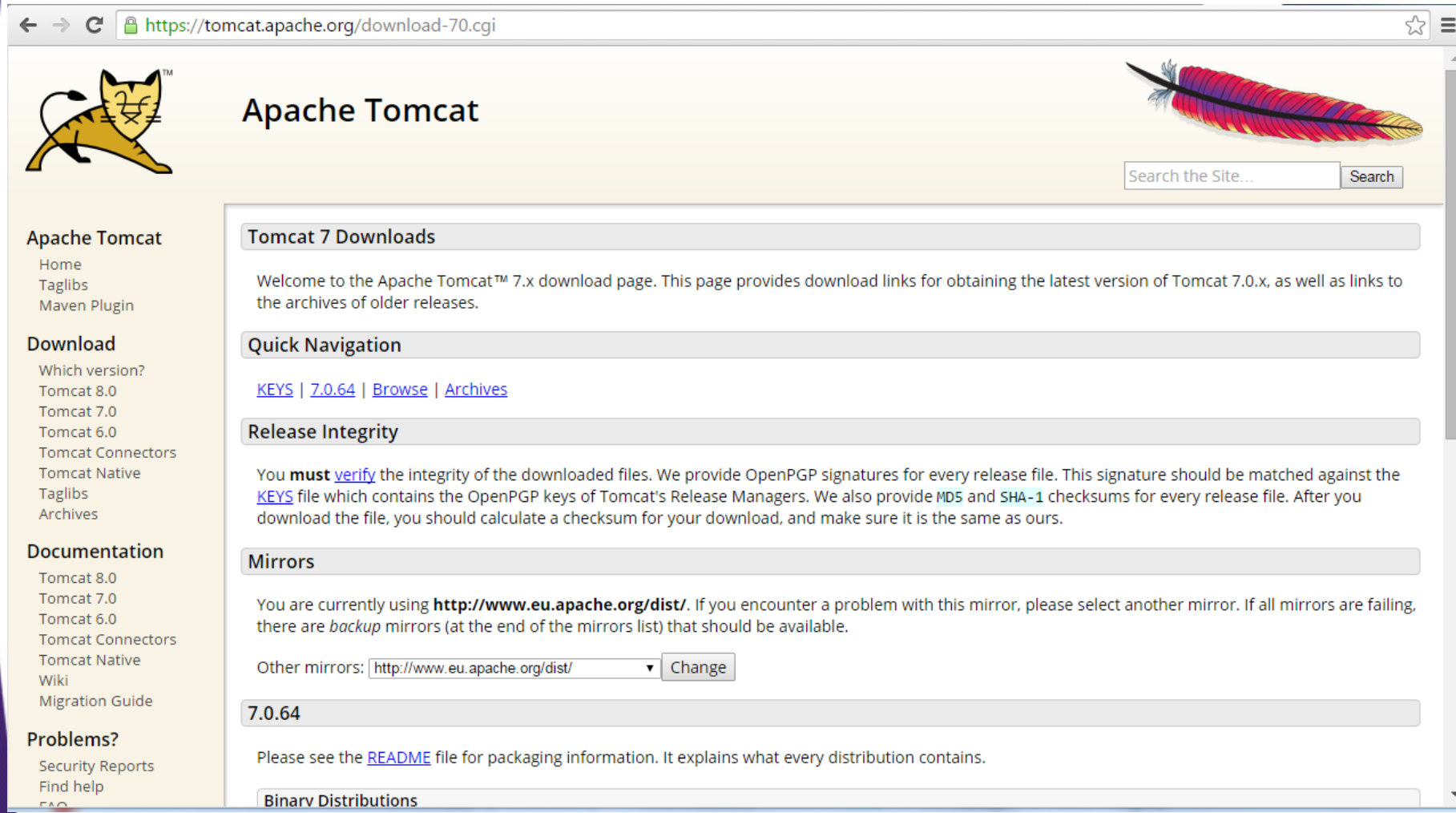
    http://www.oracle.com/technetwork/java/javase/downloads/index.html
- **Set up the JAVA_HOME environment variable that points to path of your JDK installation**
  - For example: JAVA_HOME=c:\Java\jdk1.7

Tomcat: Installation and Setup

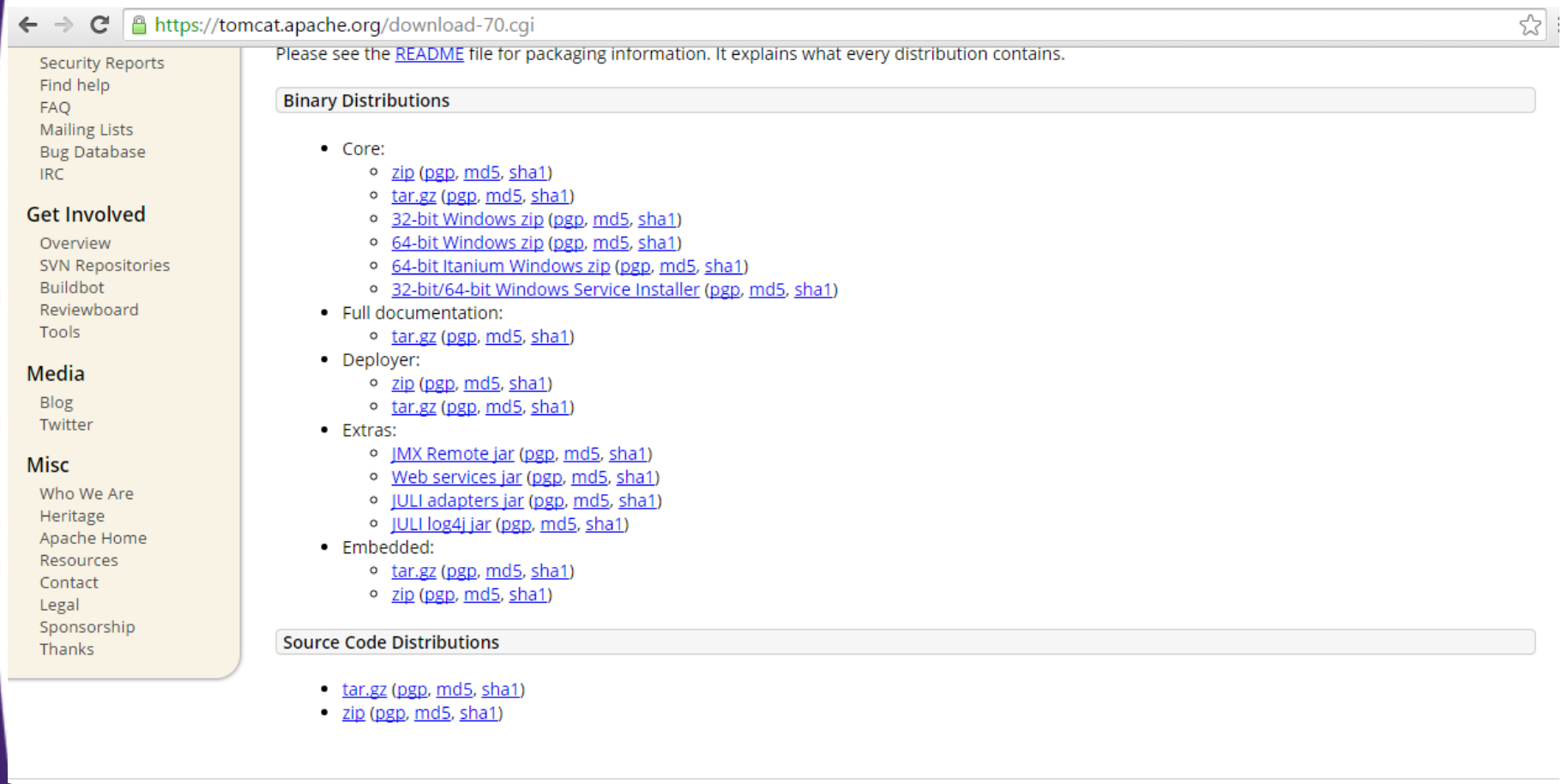
Tomcat 7 Download and Installation



Tomcat 7 Download and Installation

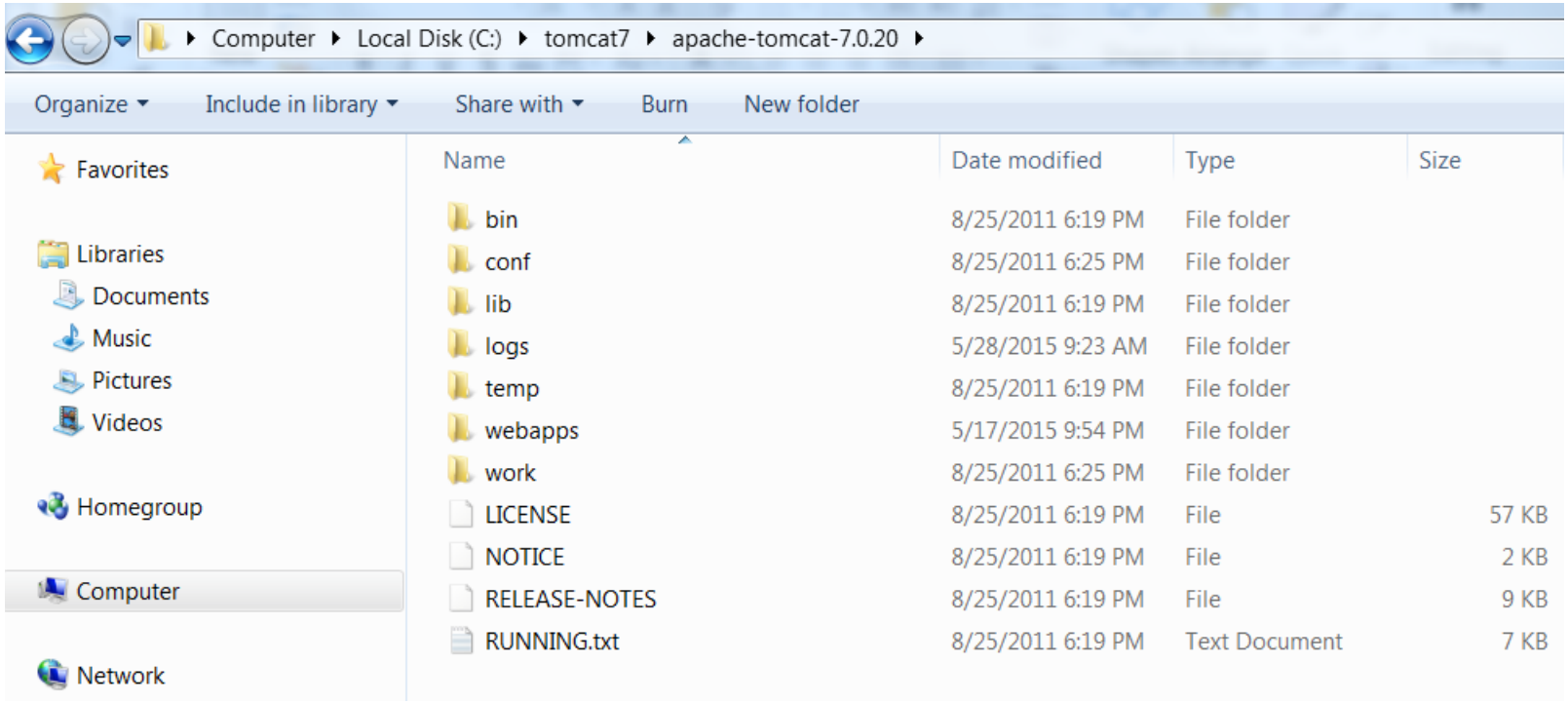


Tomcat 7 Download and Installation



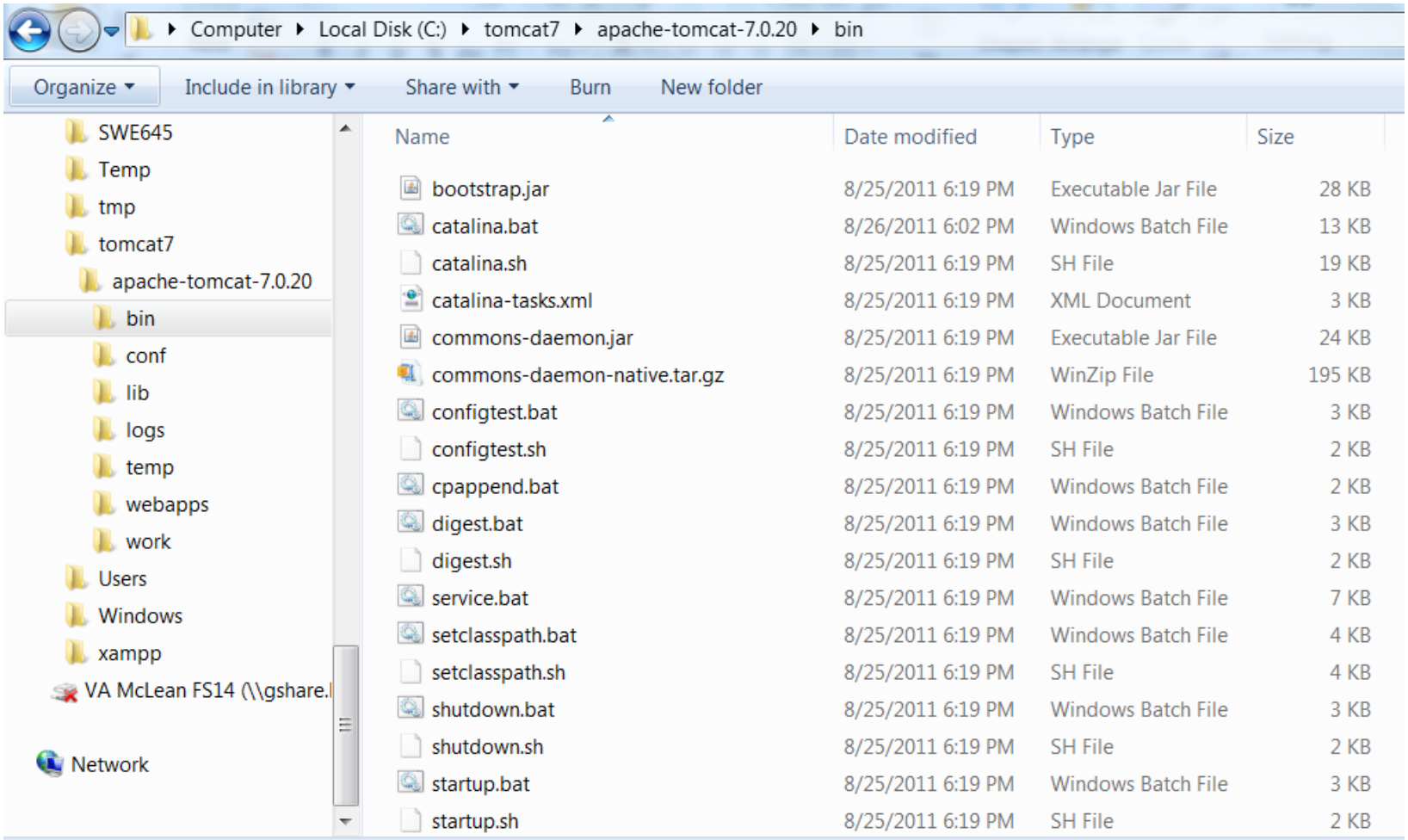
Download and save the zip file (32 bit or 64 bit version depending on your O/S

Tomcat 7 Download and Installation



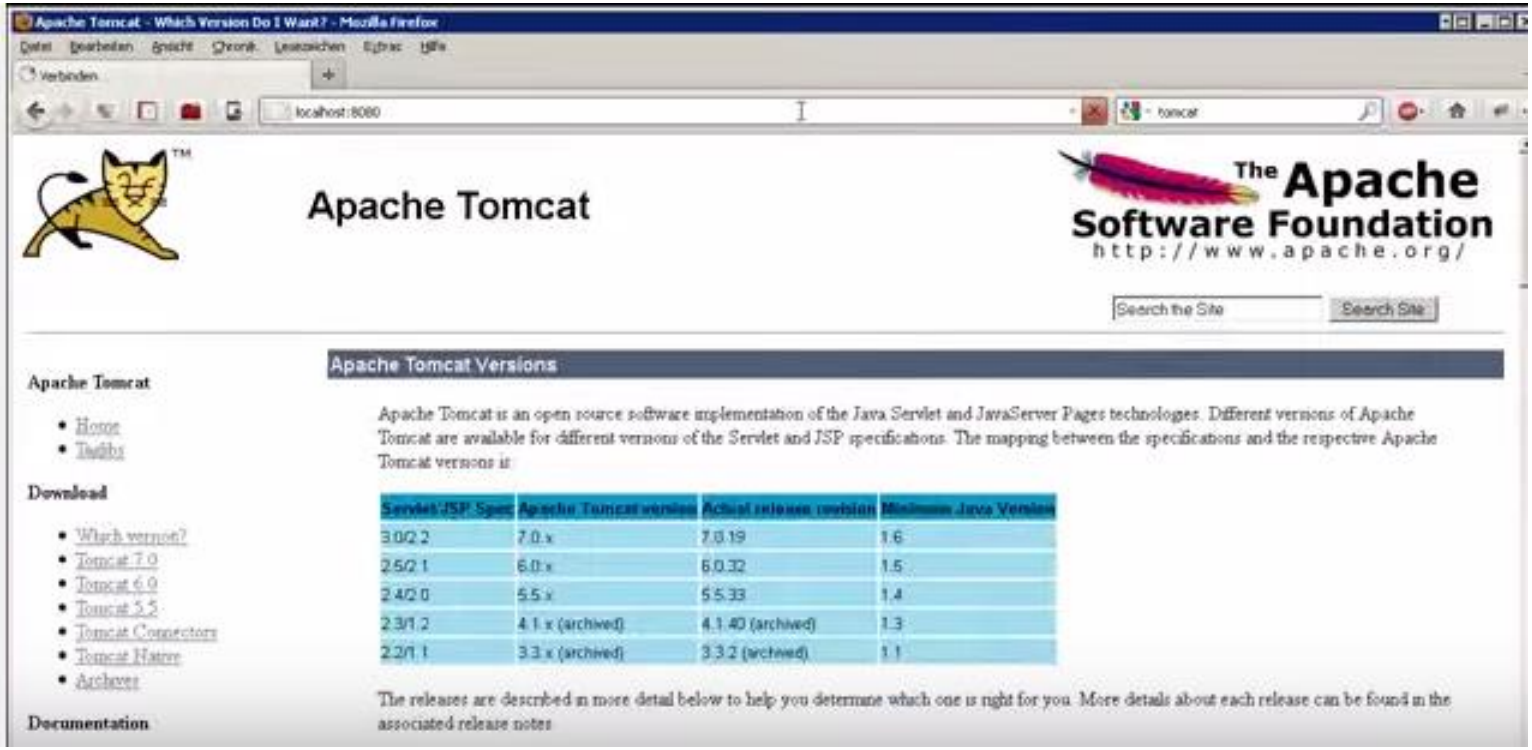
Unzip/extract the downloaded zip file

Tomcat 7 Download and Installation



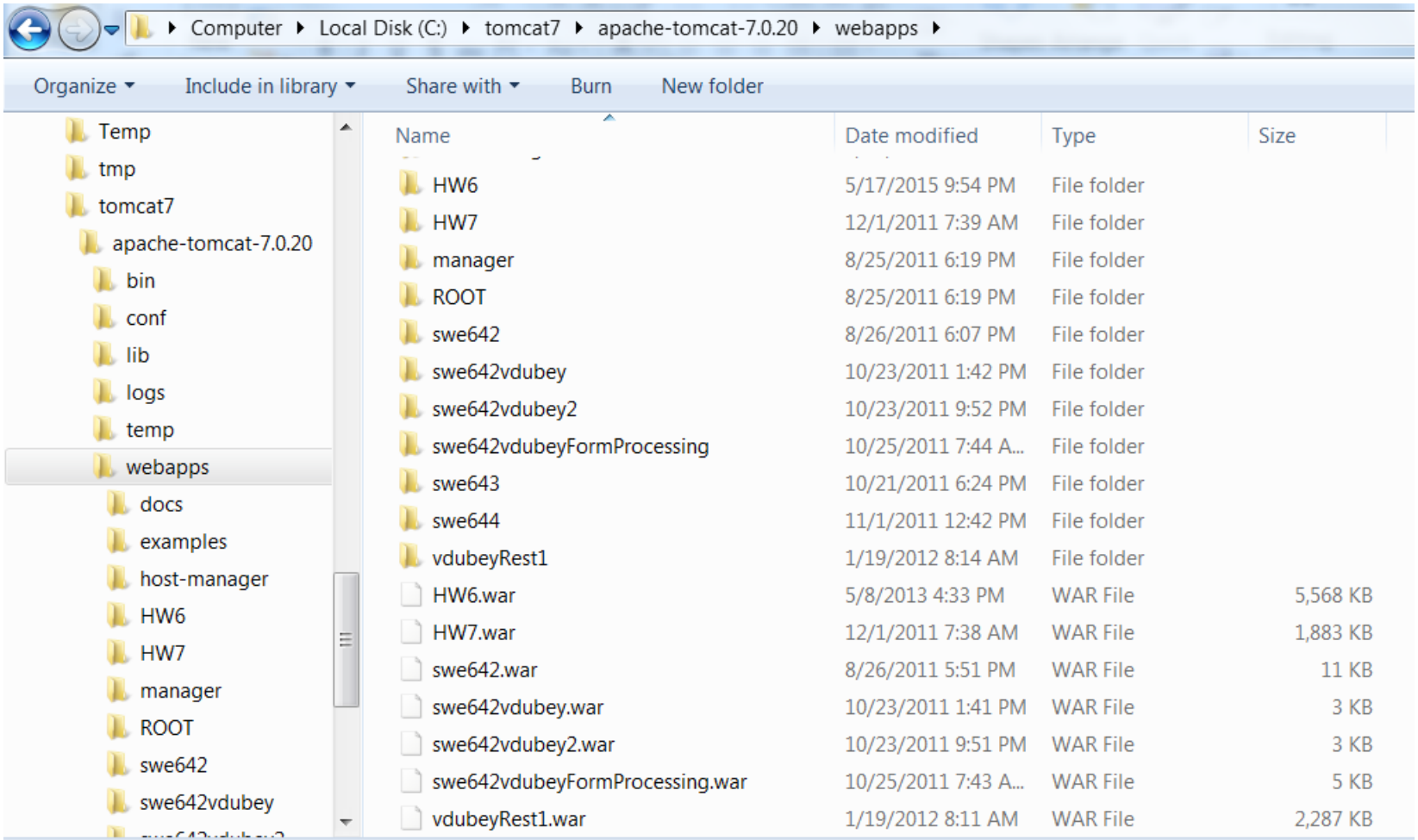
Bin directory contains startup/shutdown scripts

Download and Install Tomcat



Once started, tomcat will be up and running at http://localhost:8080

Tomcat 7 Download and Installation



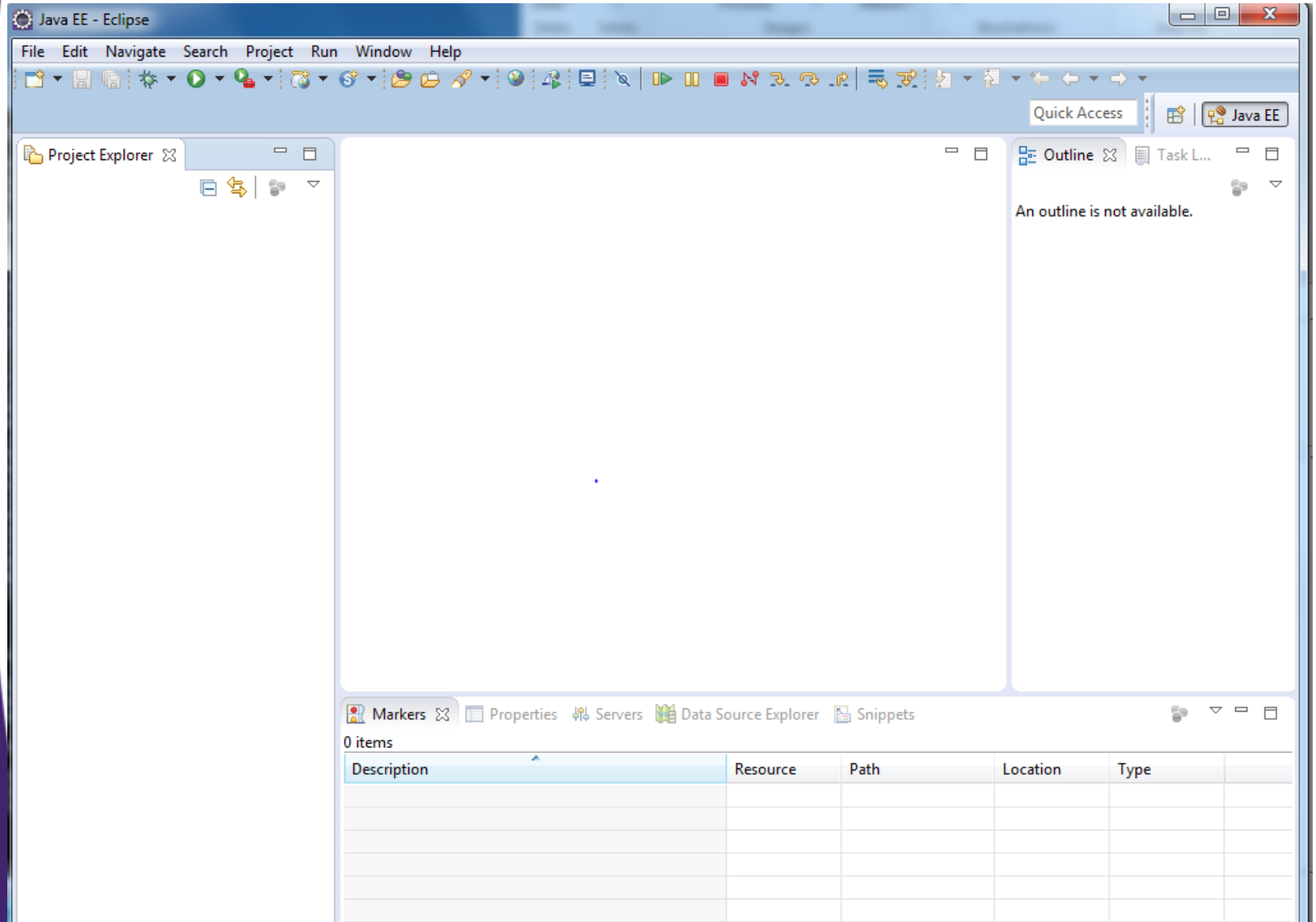
webapps folder is used for deploying war files

Eclipse:
Installation and Setup

Installing and configuring Eclipse environment

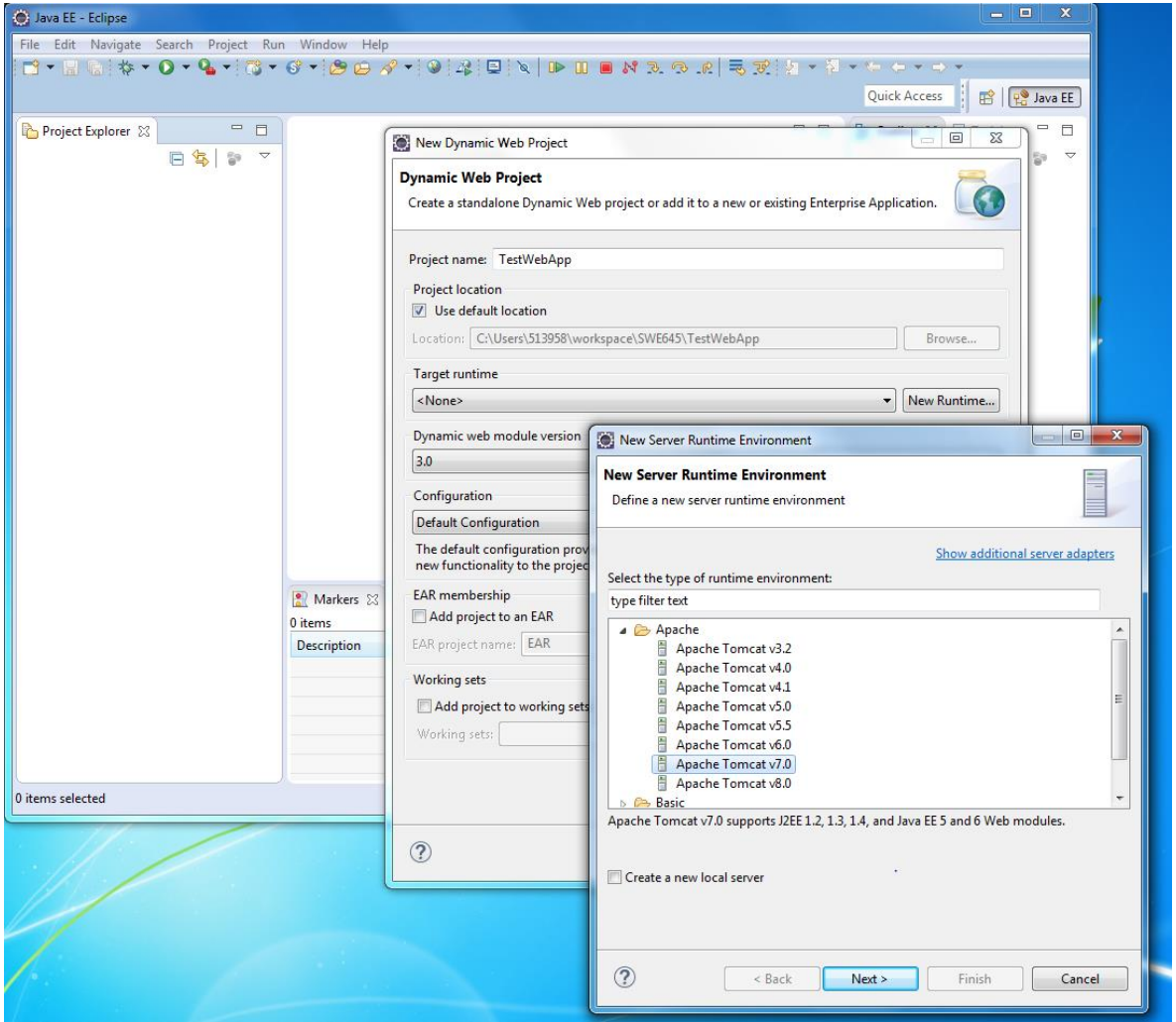
- Download the latest **Enterprise Edition** of **Eclipse IDE** from http://www.eclipse.org

Open the Eclipse IDE



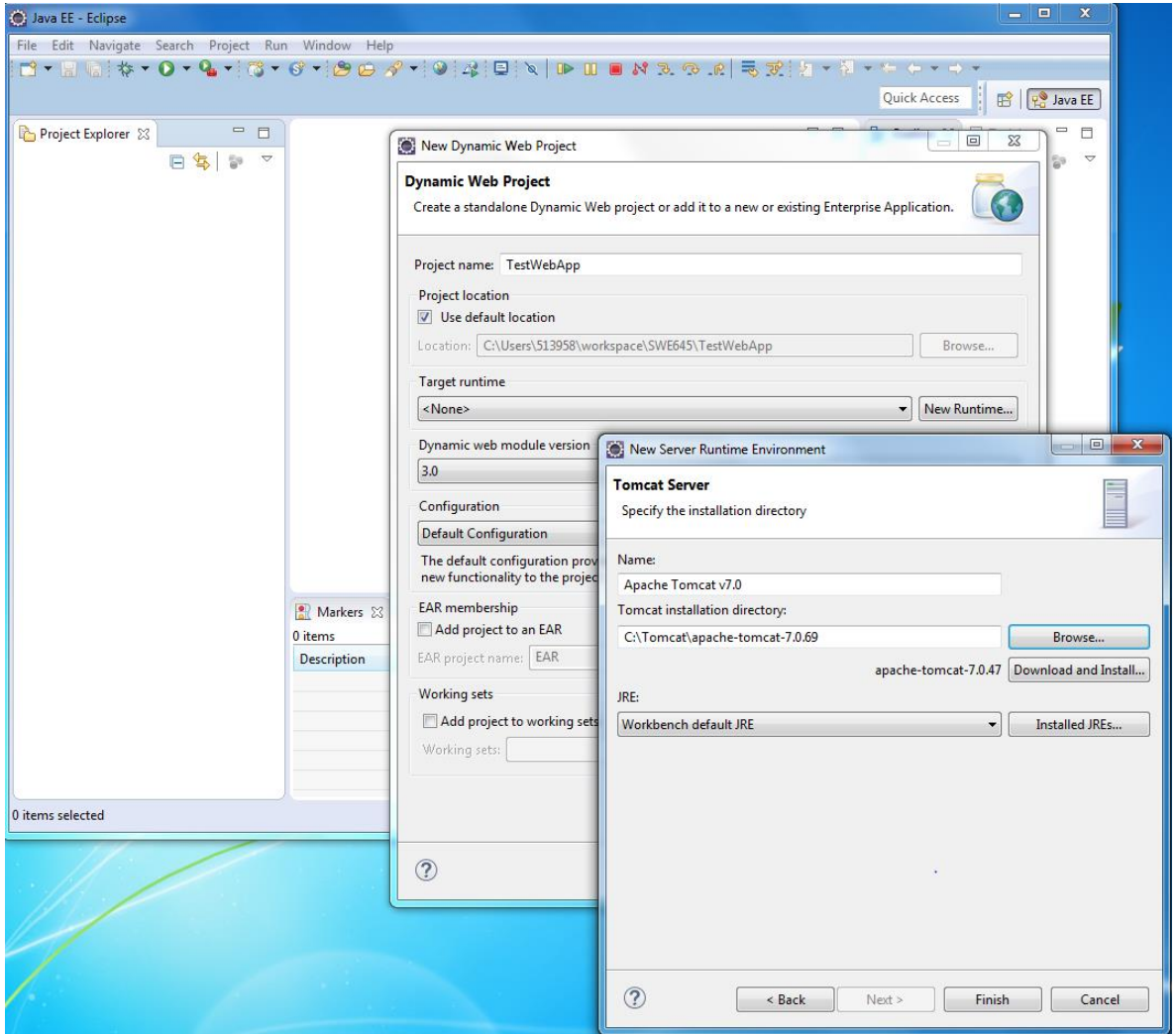
Created a Dynamic Web Project

- File -> New -> Dynamic Web Project



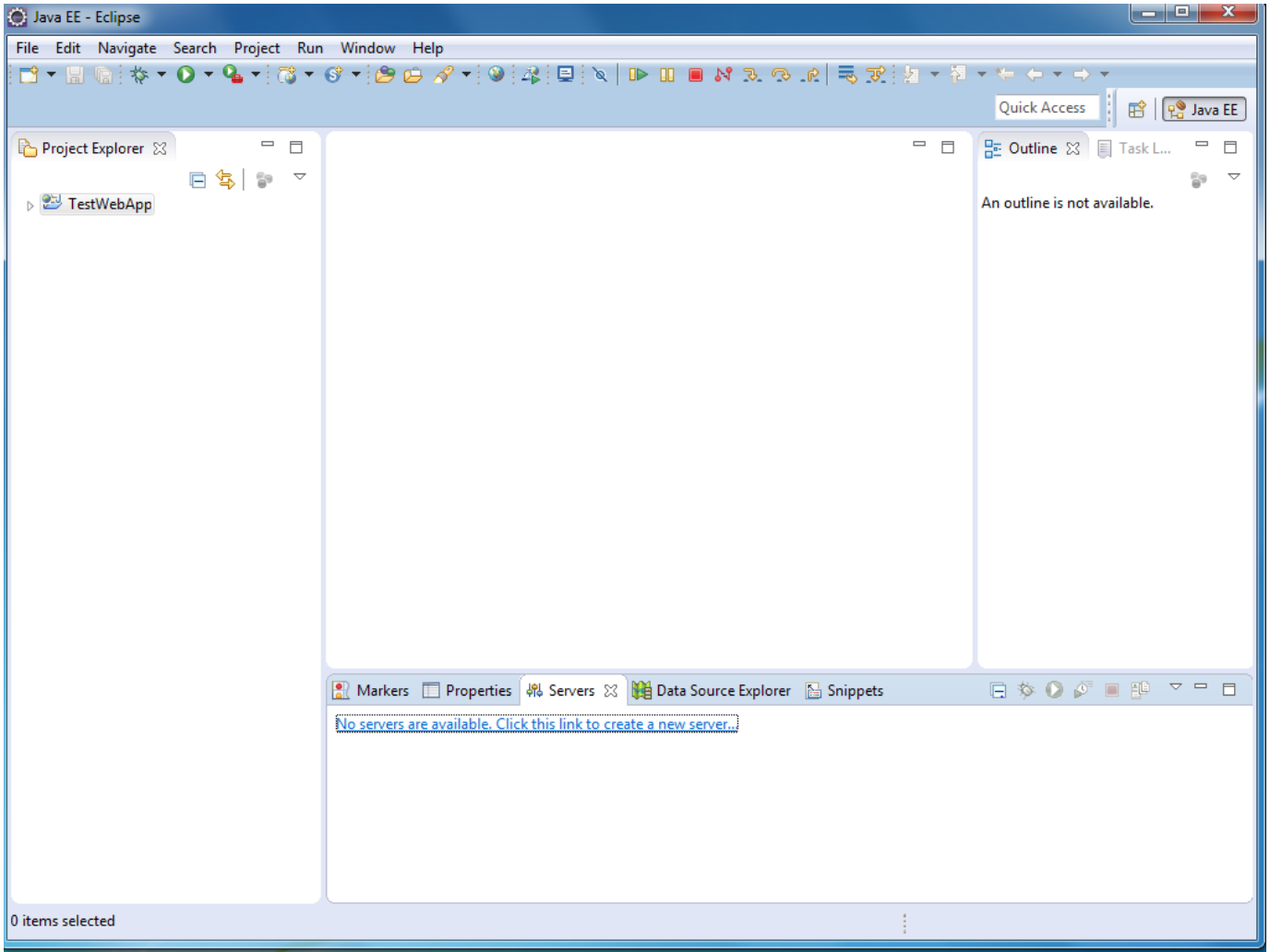
For the Target runtime, select Tomcat

- **Browse to point to the Tomcat installation directory**



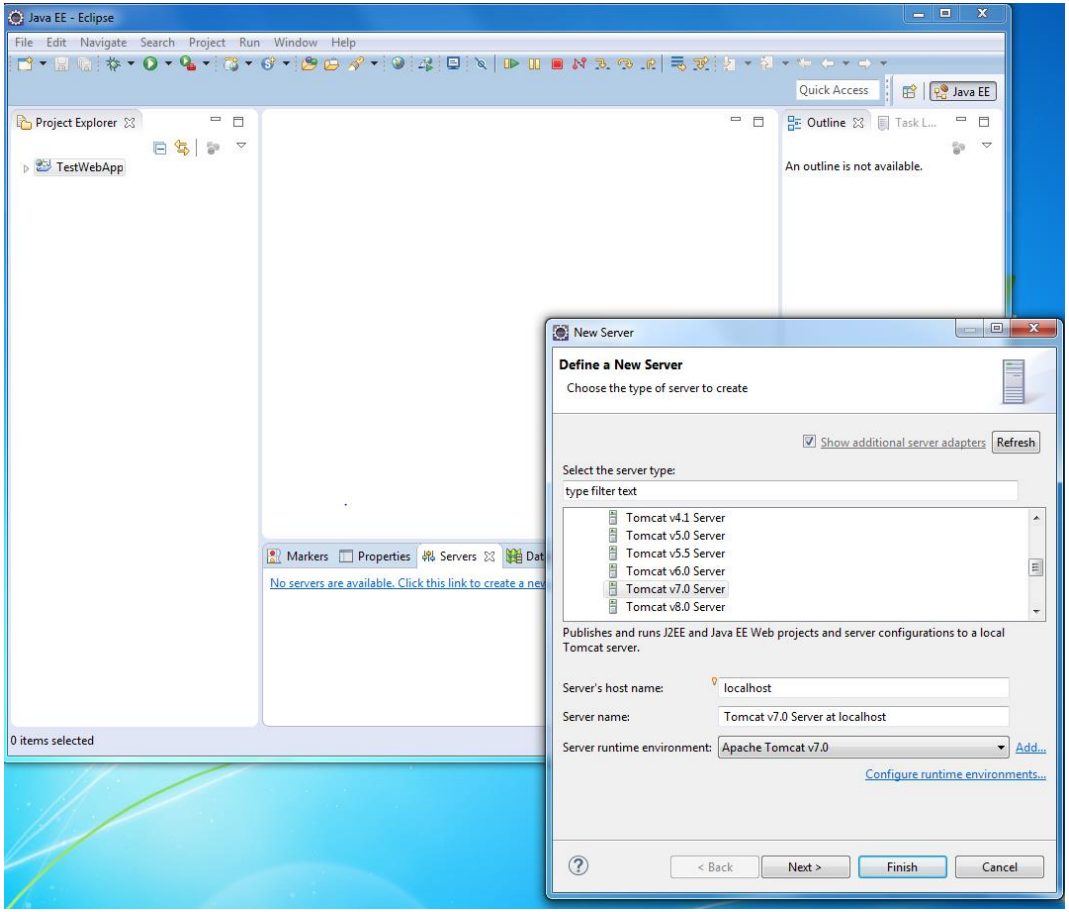
Configure Tomcat in Eclipse

- **Initially there is nothing under the Servers tab**

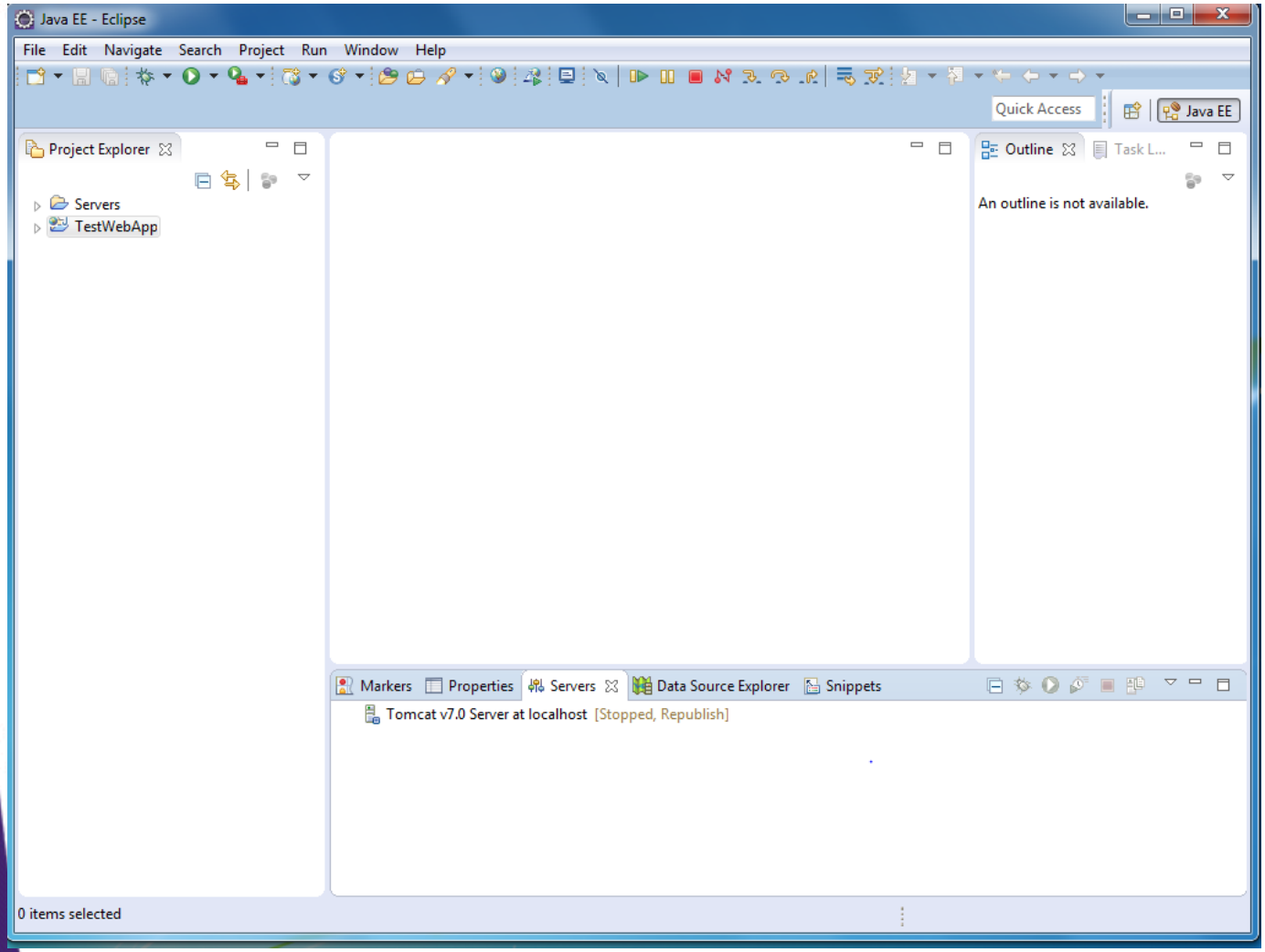


Configure Tomcat in Eclipse

- **R-click in the space under Servers tab and select New to define new server – select Tomcat**

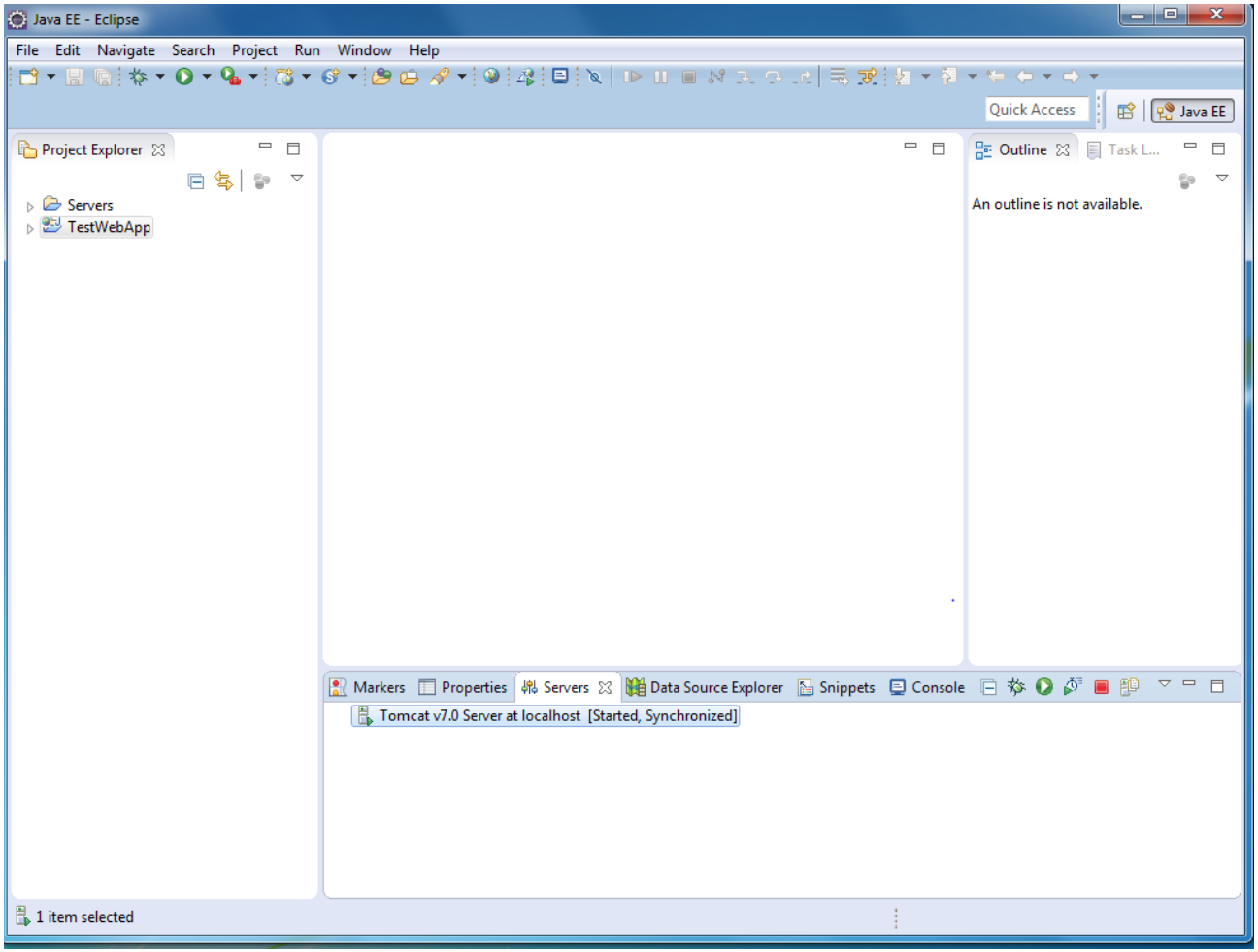


Configure Tomcat in Eclipse



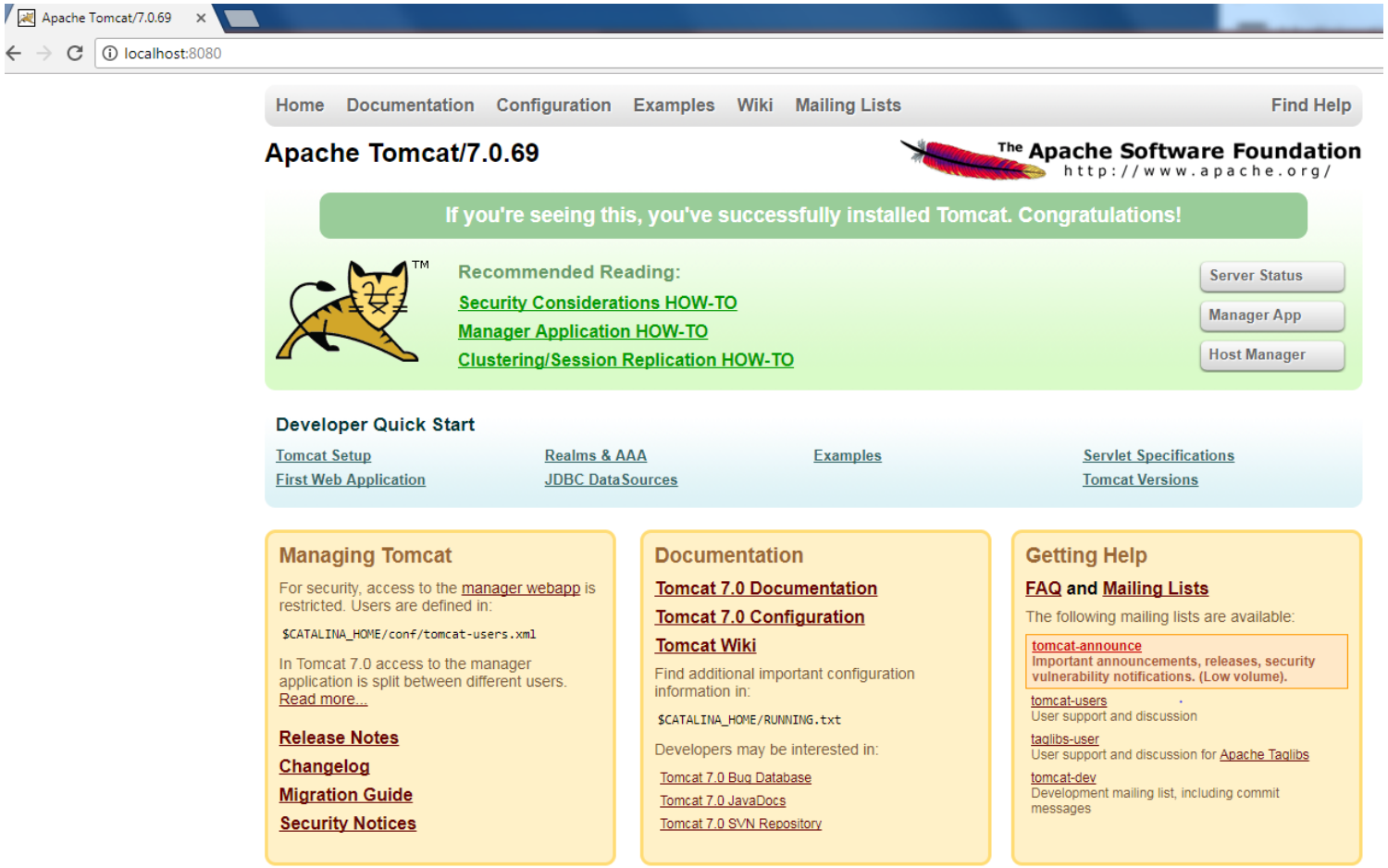
Start the Tomcat

- **R-click on the Tomcat server and select Start**



Access Tomcat homepage

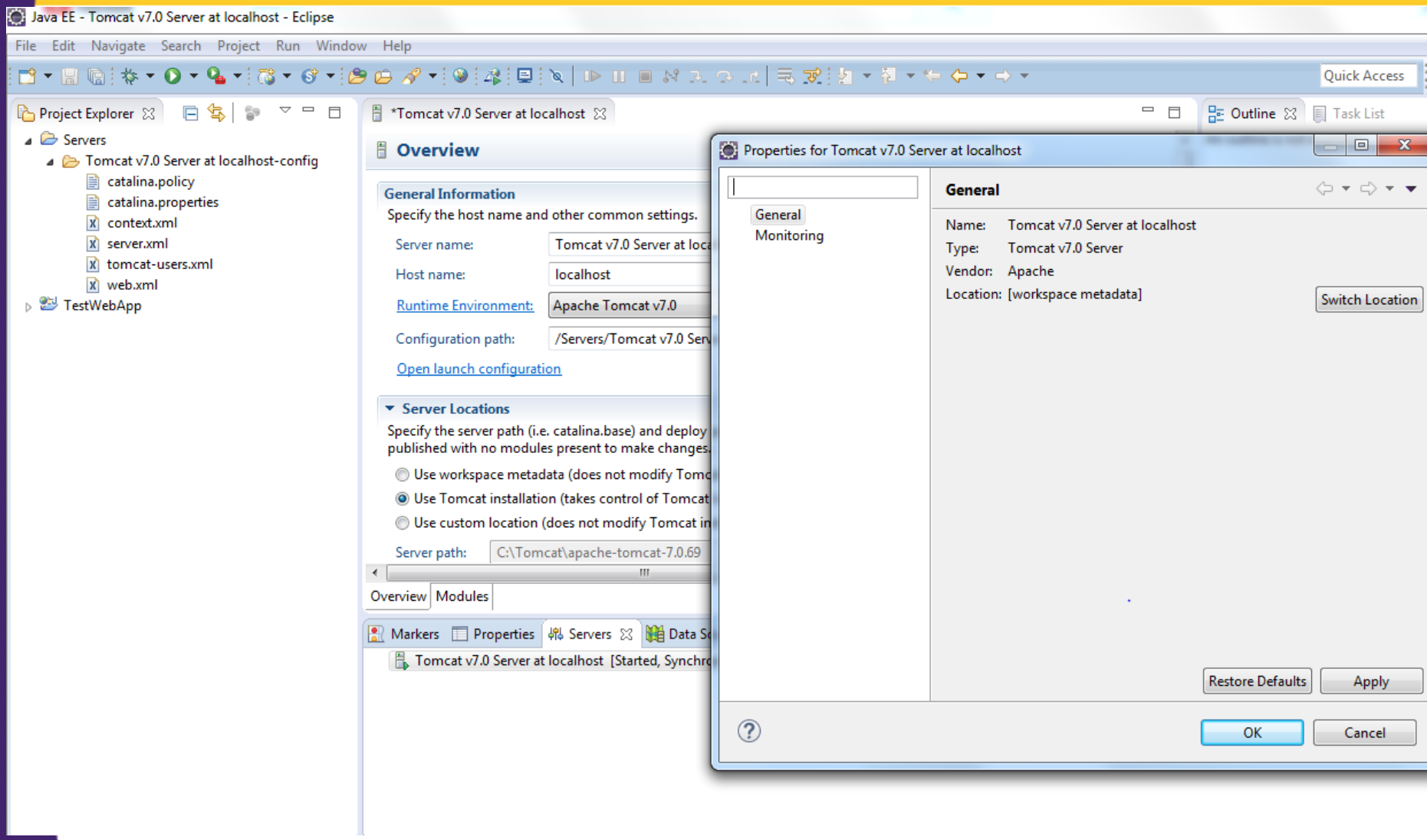
- **Once the server is running, you can access the homepage at localhost:8080**



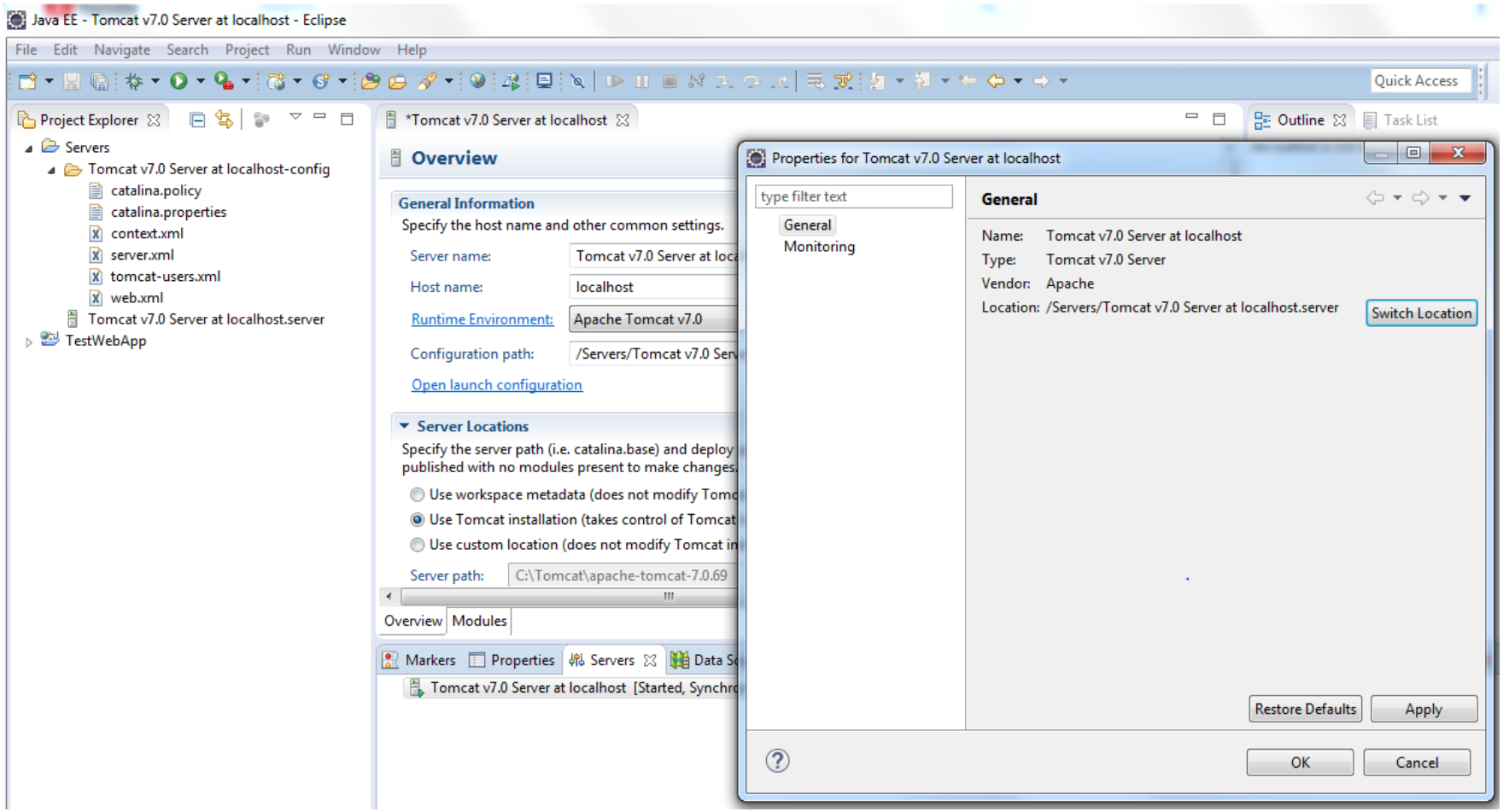
Trouble shooting steps

- **If you don't see the Server tab in the bottom portion of the Eclipse IDE, do the following:**
  - Window-> Show View -> Other - > Servers
- **Once the server is started, you can test by going to localhost:8080**
- **If you get HTTP Status 404 – page not found error, do the following two things:**
  - R-click on the Tomcat v7.0 Server and select Properties, and press the button "Switch Location" so that the Location is changed from "Workspace Metadata" to "/Servers/Tomcat Server v7.0 at localhost server [STEP 1]
  - Double click on the Tomcat v7.0 Server and for the server location make sure use select radio button for "Use Tomcat Installation" and NOT the "Use workspace metadata" [STEP 2]
  - Now Save the changes or try deleting the page with newly selected radio button option, it will ask you to save the settings, if so, select Yes
  - Restart the server and now you should be able to see the tomcat startup page at localhost:8080
- **Now create a Dynamic Web Project (make sure checkbox for the deployment descriptor file is checked), and create a default welcome page index.html (in Web Content folder) as listed in web.xml**
- **To deploy this project on tomcat server, R-click on the project name -> Run As -> Run on server; select the Tomcat server (already configured/selected), go to Next, make sure your Dynamic Web Project is in the Configured box on the right hand side, select Finish, then select OK to Restart Server.**
  - Now, in your browser, you can go to your localhost:8080/TestWebApp where the TestWebApp is your project name. This should render the content of the indext.html

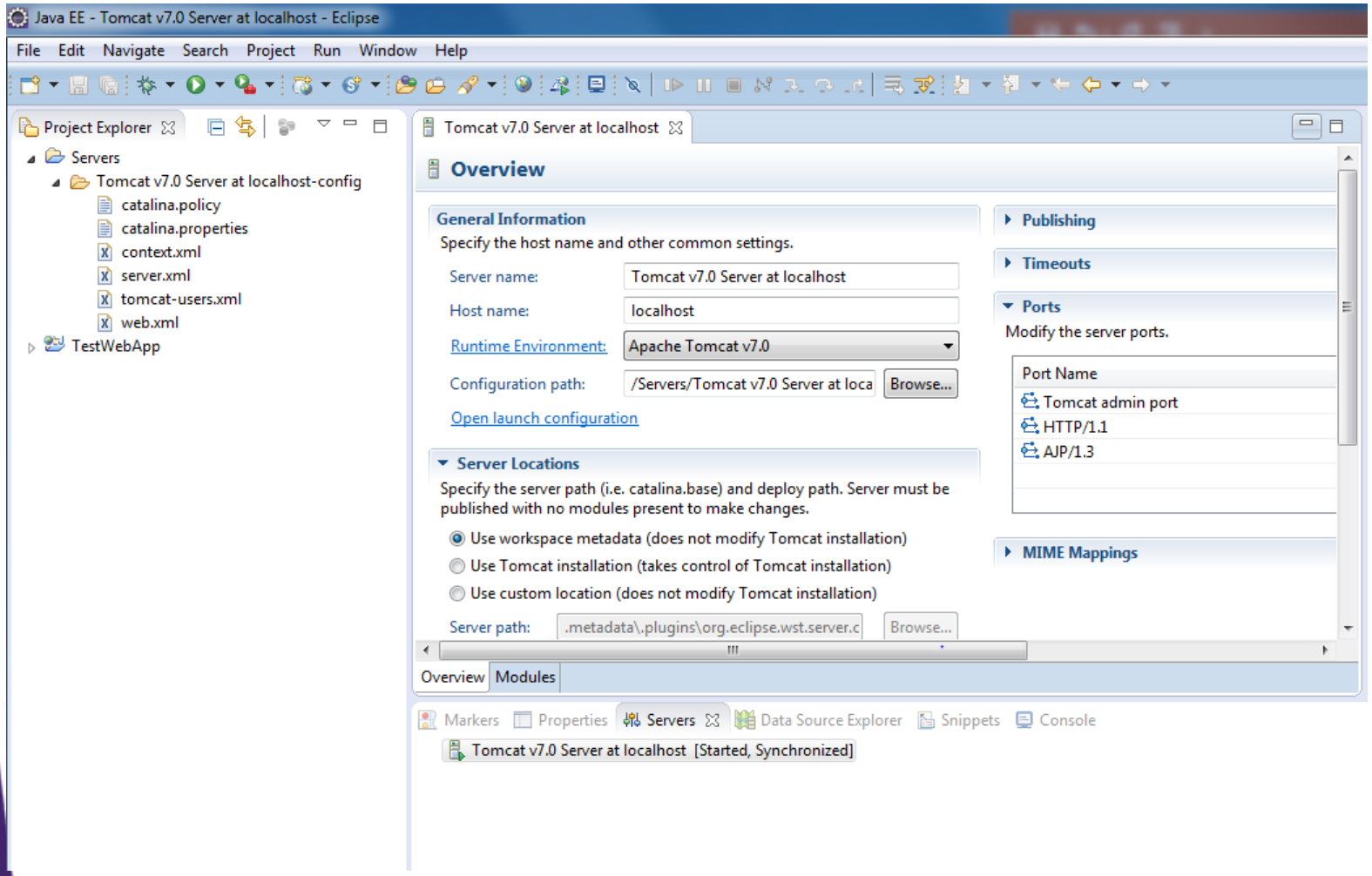
STEP 1


Java EE - Tomcat v7.0 Server at localhost - Eclipse
File Edit Navigate Search Project Run Window Help
Quick Access
Project Explorer
Servers
Tomcat v7.0 Server at localhost-config
catalina.policy
catalina.properties
context.xml
server.xml
tomcat-users.xml
web.xml
TestWebApp
*Tomcat v7.0 Server at localhost
Overview
General Information
Specify the host name and other common settings.
Server name:
Tomcat v7.0 Server at loca
Host name:
localhost
Runtime Environment:
Apache Tomcat v7.0
Configuration path:
/Servers/Tomcat v7.0 Serv
Open launch configuration
Server Locations
Specify the server path (i.e. catalina.base) and deploy
published with no modules present to make changes.
Use workspace metadata (does not modify Tomc
Use Tomcat installation (takes control of Tomcat
Use custom location (does not modify Tomcat in
Server path:
C:\Tomcat\apache-tomcat-7.0.69
Overview
Modules
Markers
Properties
Servers
Data S
Tomcat v7.0 Server at localhost [Started, Synchro
Outline
Task List
Properties for Tomcat v7.0 Server at localhost
General
Monitoring
General
Name: Tomcat v7.0 Server at localhost
Type: Tomcat v7.0 Server
Vendor: Apache
Location: [workspace metadata]
Switch Location
Restore Defaults
Apply
OK
Cancel

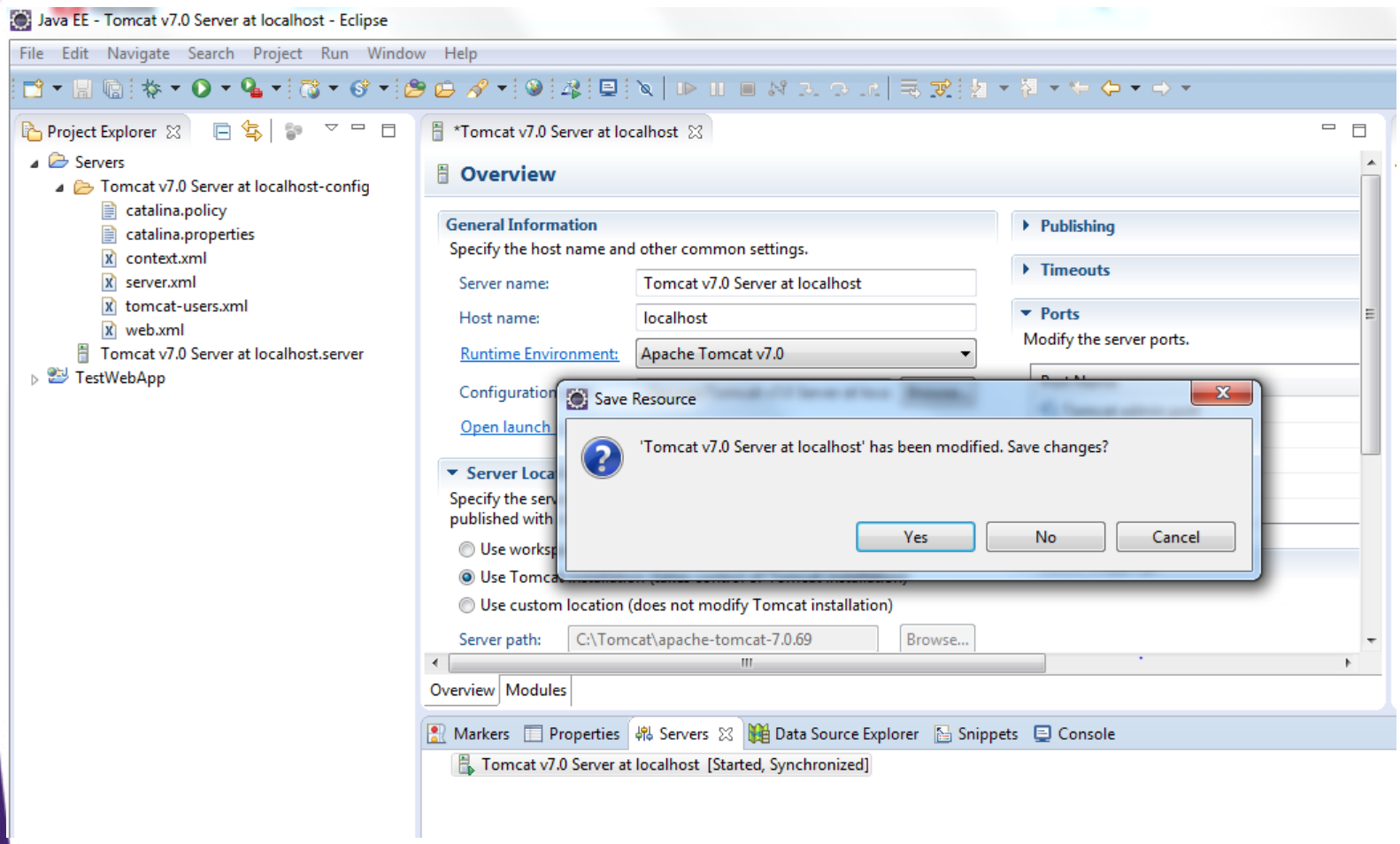
STEP 1


Java EE - Tomcat v7.0 Server at localhost - Eclipse
File Edit Navigate Search Project Run Window Help
Quick Access
Project Explorer
Servers
Tomcat v7.0 Server at localhost-config
catalina.policy
catalina.properties
context.xml
server.xml
tomcat-users.xml
web.xml
Tomcat v7.0 Server at localhost.server
TestWebApp
*Tomcat v7.0 Server at localhost
Overview
General Information
Specify the host name and other common settings.
Server name:
Host name:
localhost
Runtime Environment:
Apache Tomcat v7.0
Configuration path:
Open launch configuration
Server Locations
Use Tomcat installation (takes control of Tomcat
Use custom location (does not modify Tomcat in
Server path:
C:\Tomcat\apache-tomcat-7.0.69
Overview
Modules
Markers
Properties
Servers
Tomcat v7.0 Server at localhost [Started, Synchro
Outline
Task List
Properties for Tomcat v7.0 Server at localhost
type filter text
General
Monitoring
General
Name: Tomcat v7.0 Server at localhost
Type: Tomcat v7.0 Server
Vendor: Apache
Location: /Servers/Tomcat v7.0 Server at localhost.server
Switch Location
Restore Defaults
Apply
OK
Cancel

STEP 2


Java EE - Tomcat v7.0 Server at localhost - Eclipse
File Edit Navigate Search Project Run Window Help
Project Explorer
Servers
Tomcat v7.0 Server at localhost-config
catalina.policy
catalina.properties
context.xml
server.xml
tomcat-users.xml
web.xml
TestWebApp
Tomcat v7.0 Server at localhost
Overview
General Information
Specify the host name and other common settings.
Server name: Tomcat v7.0 Server at localhost
Host name: localhost
Runtime Environment: Apache Tomcat v7.0
Configuration path: /Servers/Tomcat v7.0 Server at loca
Browse...
Open launch configuration
Server Locations
Specify the server path (i.e. catalina.base) and deploy path. Server must be published with no modules present to make changes.
Use workspace metadata (does not modify Tomcat installation)
Use Tomcat installation (takes control of Tomcat installation)
Use custom location (does not modify Tomcat installation)
Server path: .metadata\.plugins\org.eclipse.wst.server.c
Browse...
Publishing
Timeouts
Ports
Modify the server ports.
Port Name
Tomcat admin port
HTTP/1.1
AJP/1.3
MIME Mappings
Overview
Modules
Markers
Properties
Servers
Data Source Explorer
Snippets
Console
Tomcat v7.0 Server at localhost [Started, Synchronized]

STEP 2

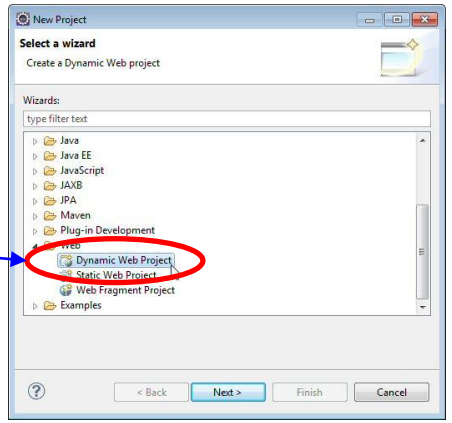

Java EE - Tomcat v7.0 Server at localhost - Eclipse
File Edit Navigate Search Project Run Window Help
Project Explorer
Servers
Tomcat v7.0 Server at localhost-config
catalina.policy
catalina.properties
context.xml
server.xml
tomcat-users.xml
web.xml
Tomcat v7.0 Server at localhost.server
TestWebApp
*Tomcat v7.0 Server at localhost
Overview
General Information
Specify the host name and other common settings.
Server name:
Tomcat v7.0 Server at localhost
Host name:
localhost
Runtime Environment:
Apache Tomcat v7.0
Publishing
Timeouts
Ports
Modify the server ports.
Save Resource
'Tomcat v7.0 Server at localhost' has been modified. Save changes?
Yes
No
Cancel
Use custom location (does not modify Tomcat installation)
Server path:
C:\Tomcat\apache-tomcat-7.0.69
Browse...
Overview
Modules
Markers
Properties
Servers
Data Source Explorer
Snippets
Console
Tomcat v7.0 Server at localhost [Started, Synchronized]

Dynamic Web Projects

- **Dynamic Web Projects**
  - Purpose
    - Servlets, JSP, JSF, Struts, and other Web apps.
  - Creating
    - File, New, Dynamic Web Project

Making Dynamic Web Project in Eclipse

- **Create project**
  - File → New → Project → Web → Dynamic Web Project
    - Next time, you can do File → New → Dynamic Web Project

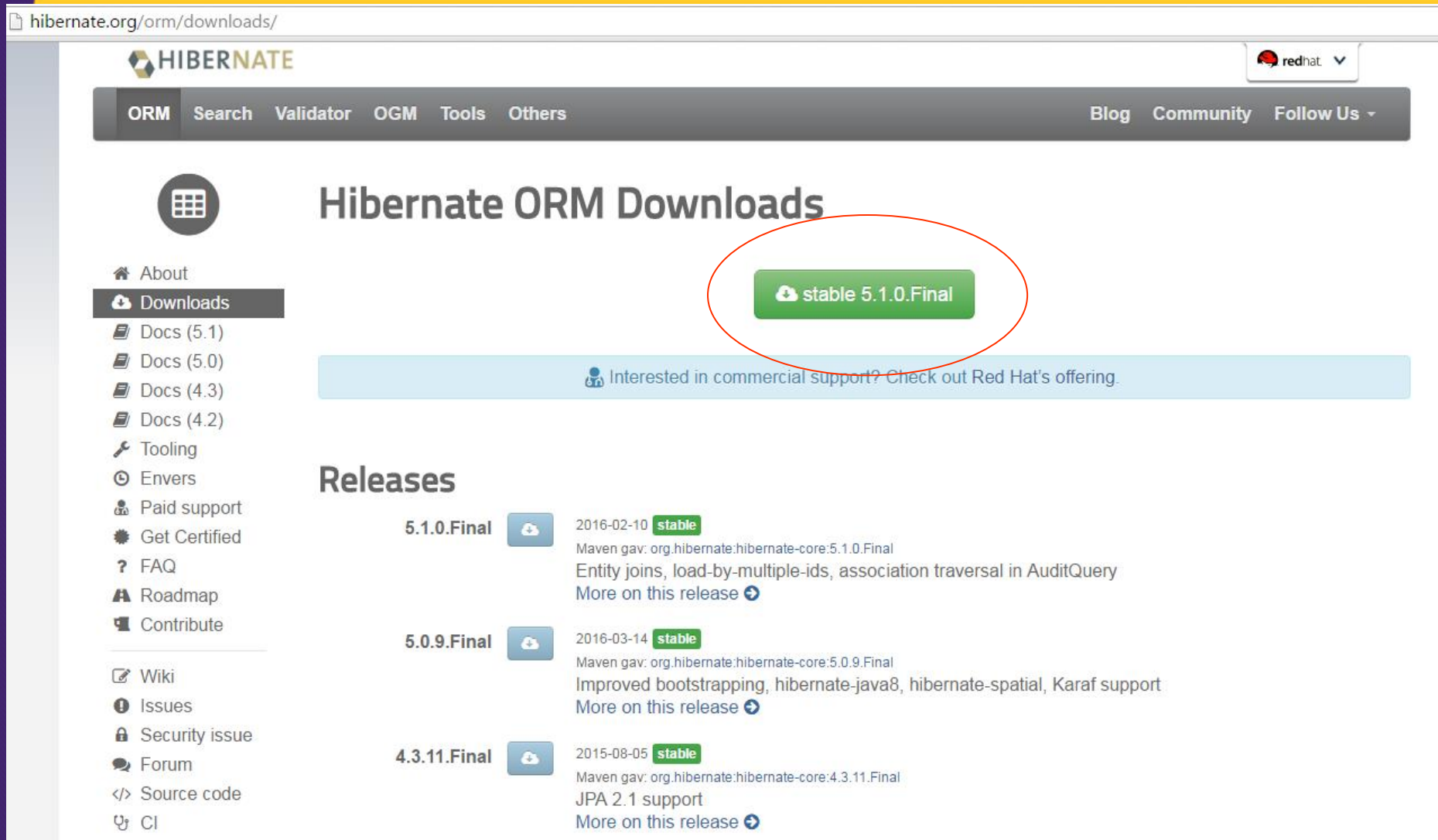


JPA Configuration With Hibernate

JPA Providers

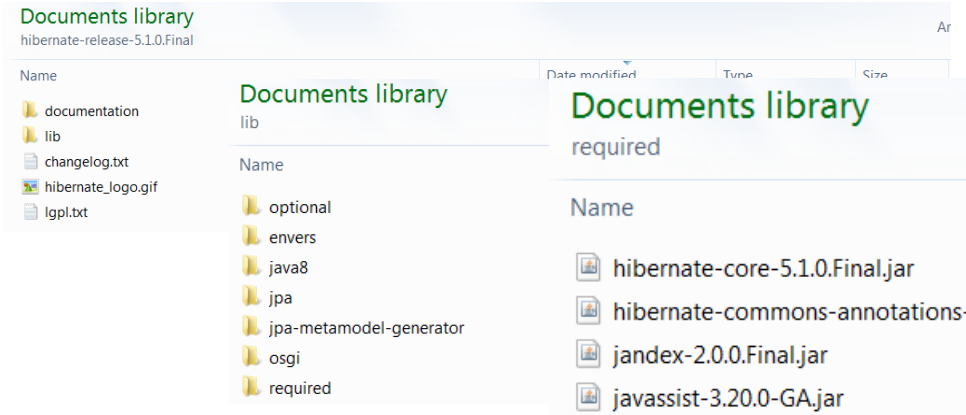
- **Hibernate**
  - To use hibernate, please Download hibernate (zip file) www.hibernate.org/downloads and jar files to project build path/classpath and add them to lib folder of your application
  - On download page, click on release bundle, e.g., 3.6.4…..zip and, which has
    - has lib folder which has jars needed by hibernate
    - You will need all jars in lib/jpa and lib/required folders
  - Also, download JDBC driver (of database choice) and add it to your project's java build path
    - This configures hibernate to use JDBC driver (jar) to connect to database

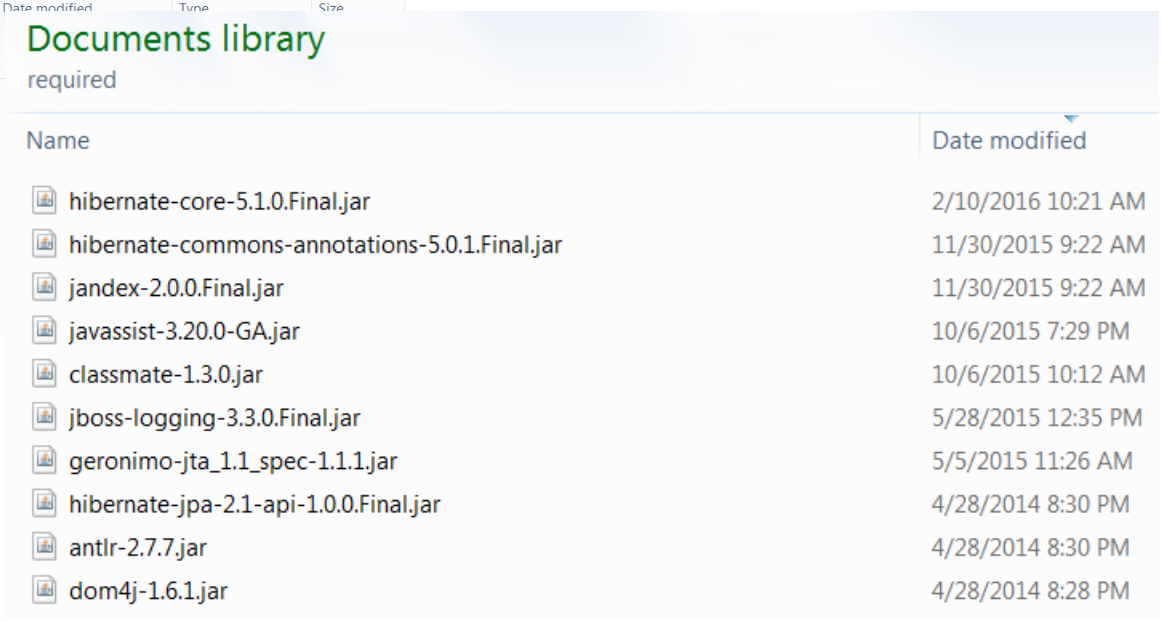
JPA Providers

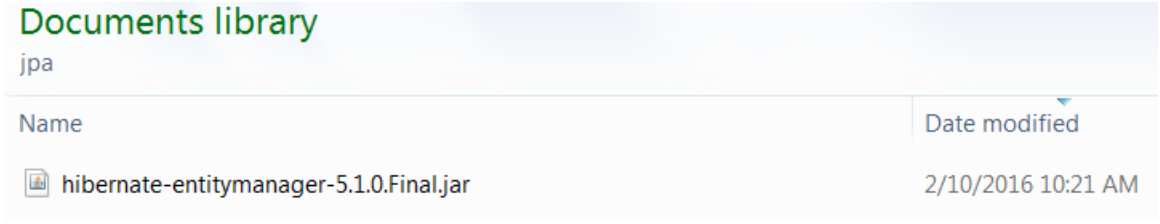


Adding Hibernate/JPA jars to your classpath in eclipse

- **Unzip the downloaded file. Your jars are inside lib folder, specifically inside lib/required and lib/jpa**

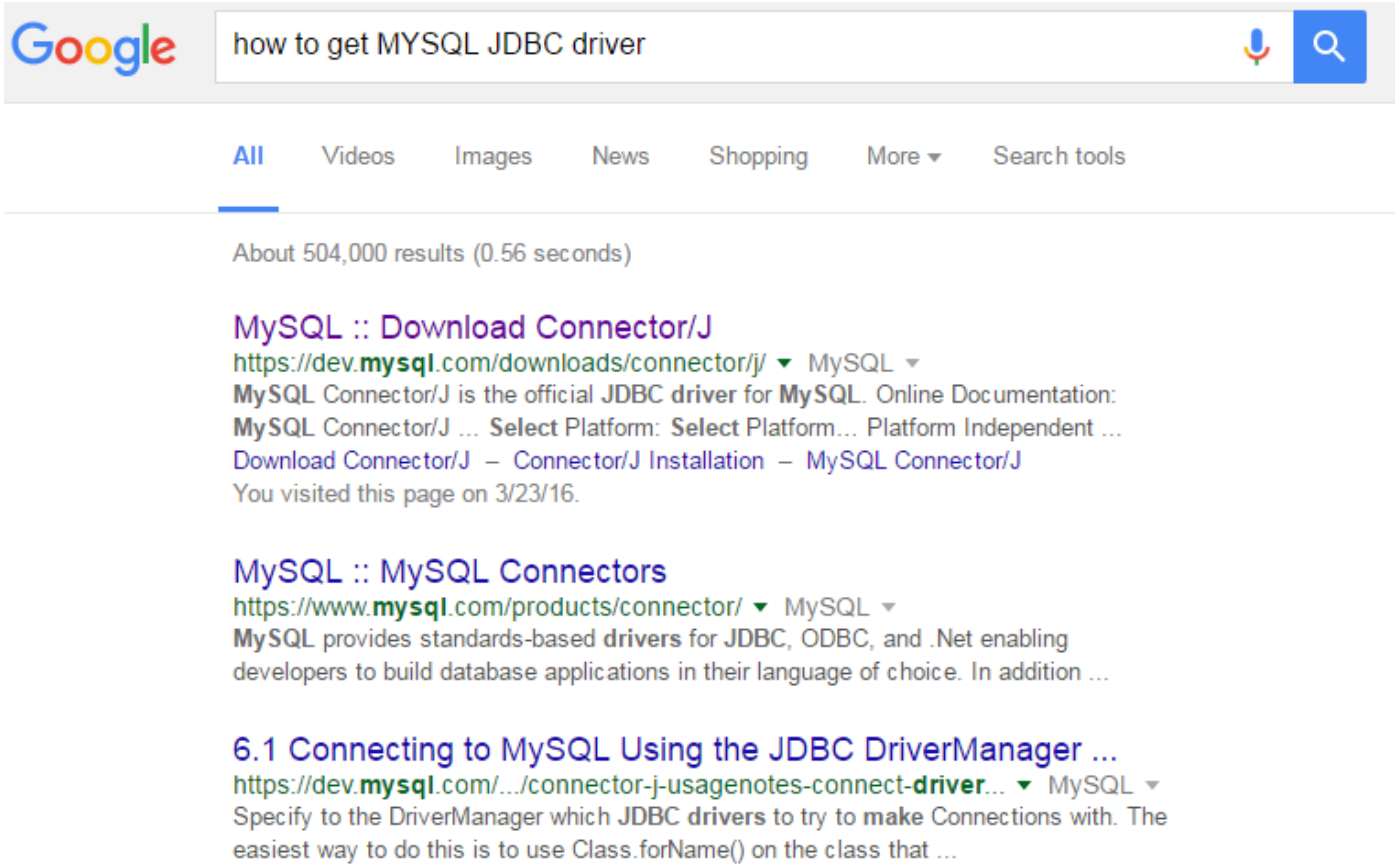






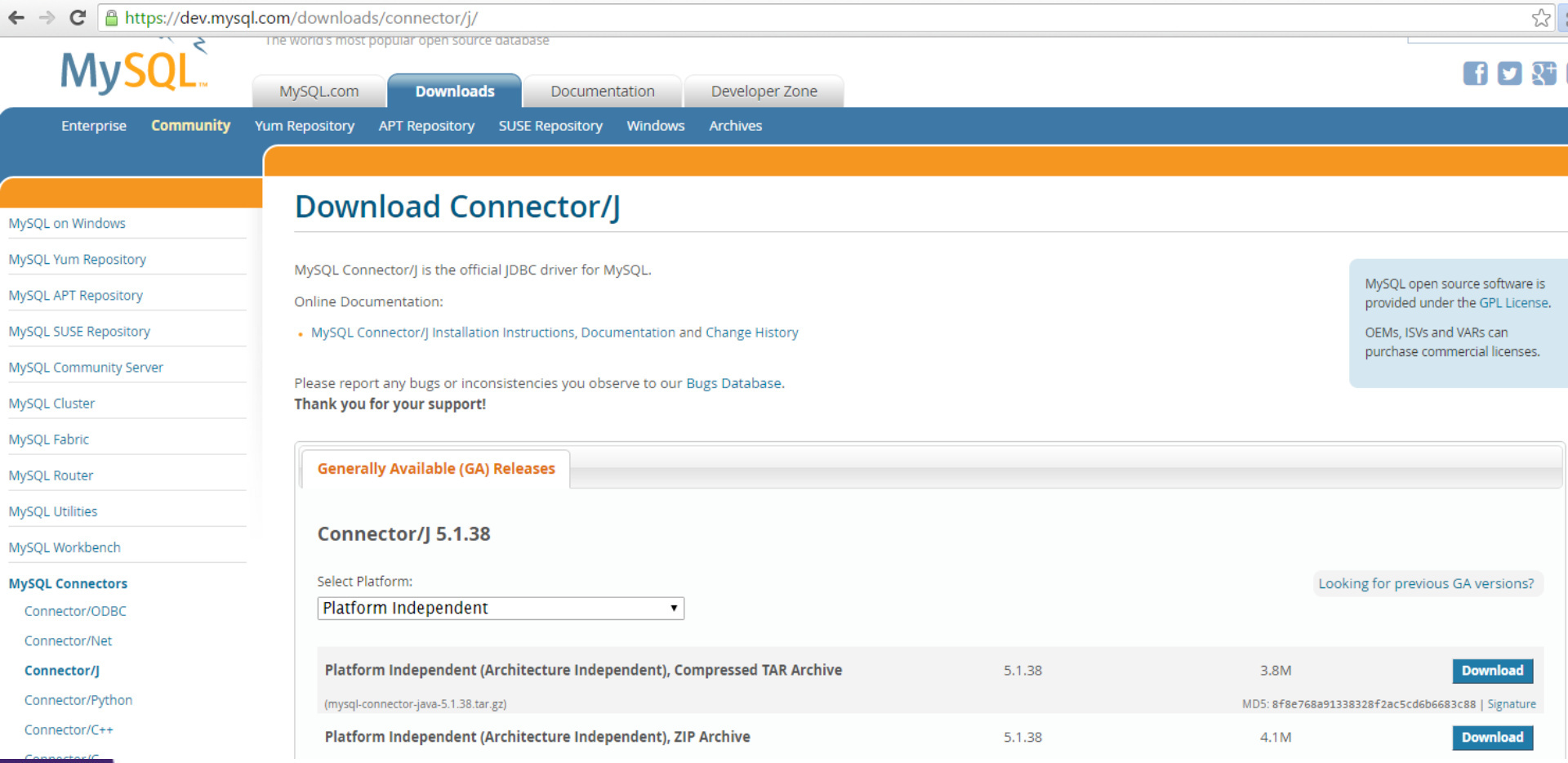
Download MySQL JDBC driver and add it to Java Build Path of your project in Eclipse

- **Here is what I got in Google search – click the first link "MySQL::Download Connector/J"**



Download MySQL JDBC driver and add it to Java Build Path of your project in Eclipse

- **Download the zip file (the 2nd Download link)**

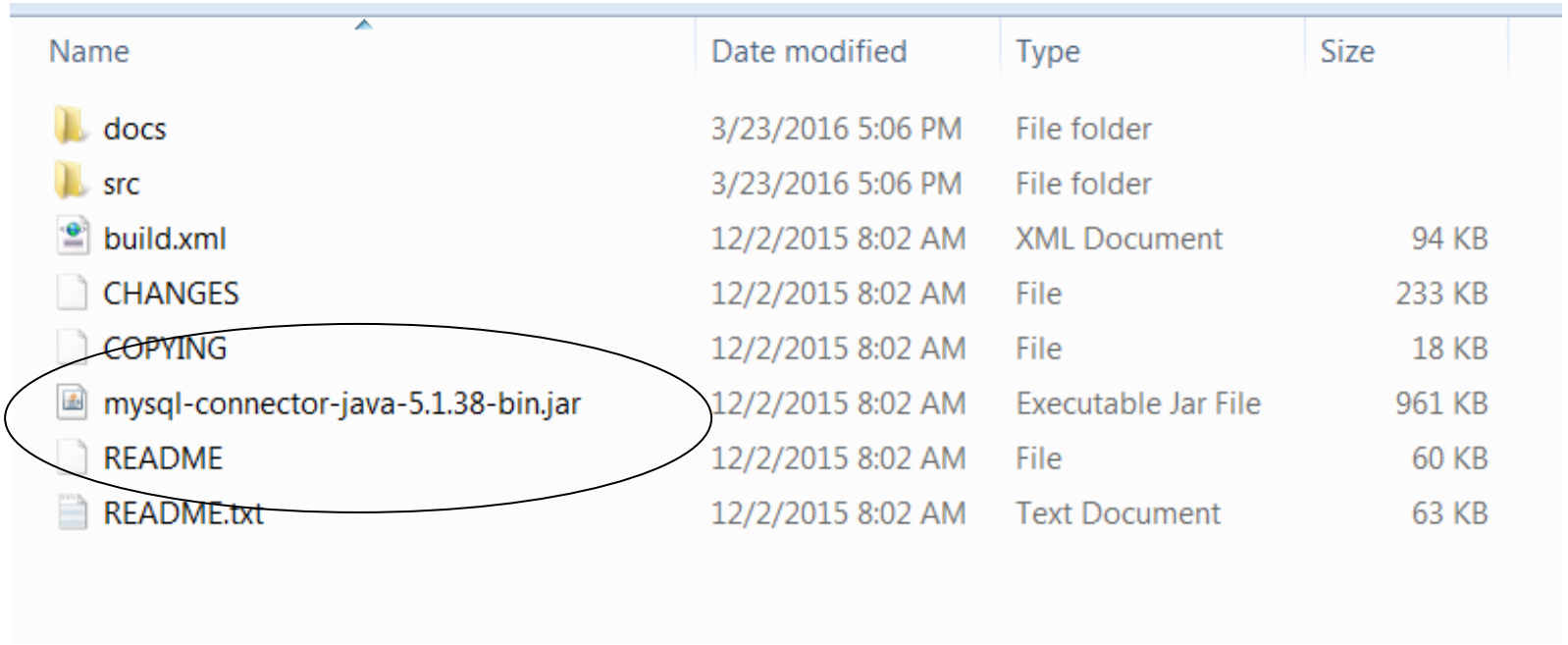


Download MySQL JDBC driver and add it to Java Build Path of your project in Eclipse

- On the page that follows, click on "No thanks, just start my download"

Download MySQL JDBC driver and add it to Java Build Path of your project in Eclipse

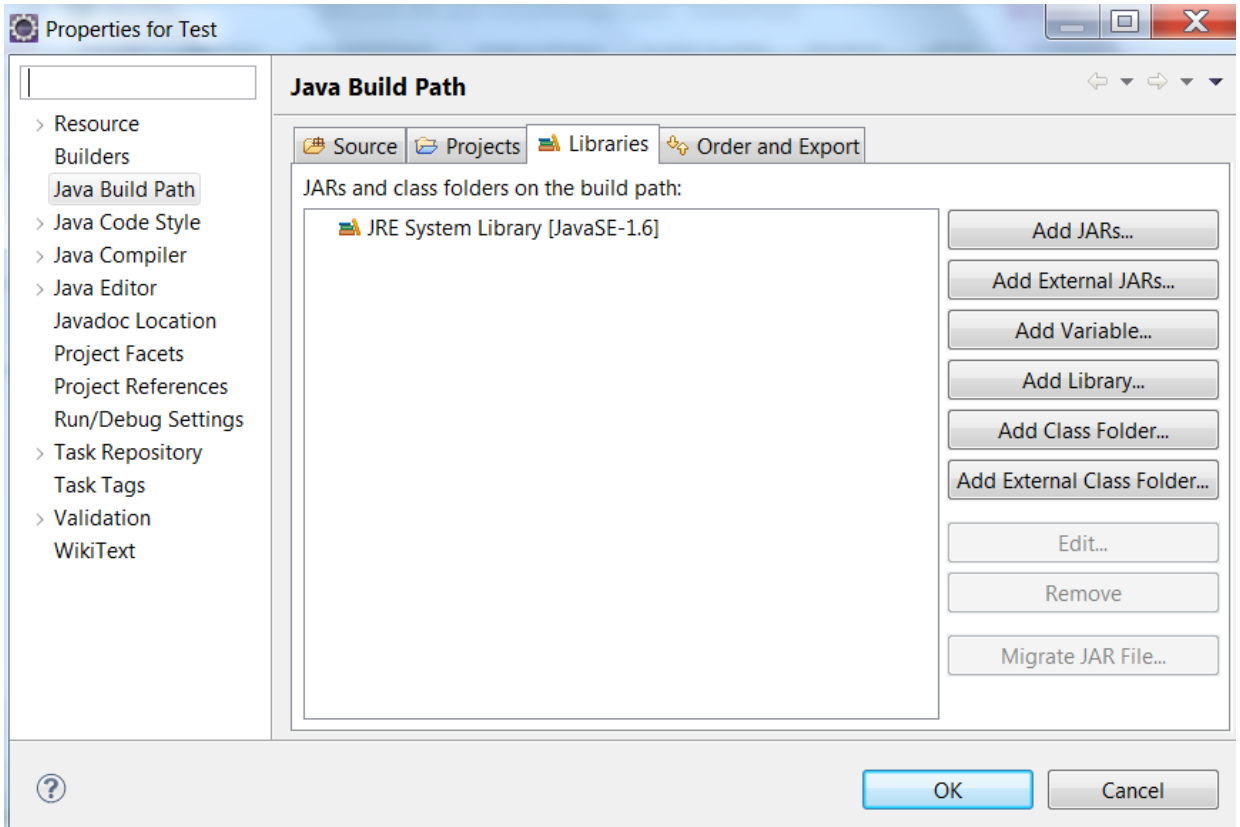
- **Unzipped downloaded zip file contains one jar file, which is your jdbc driver**
- **You must add this jar file to Java Build Path of your project in Eclipse**



Name	Date modified	Type	Size
docs	3/23/2016 5:06 PM	File folder	
src	3/23/2016 5:06 PM	File folder	
build.xml	12/2/2015 8:02 AM	XML Document	94 KB
CHANGES	12/2/2015 8:02 AM	File	233 KB
COPYING	12/2/2015 8:02 AM	File	18 KB
mysql-connector-java-5.1.38-bin.jar	12/2/2015 8:02 AM	Executable Jar File	961 KB
README	12/2/2015 8:02 AM	File	60 KB
README.txt	12/2/2015 8:02 AM	Text Document	63 KB

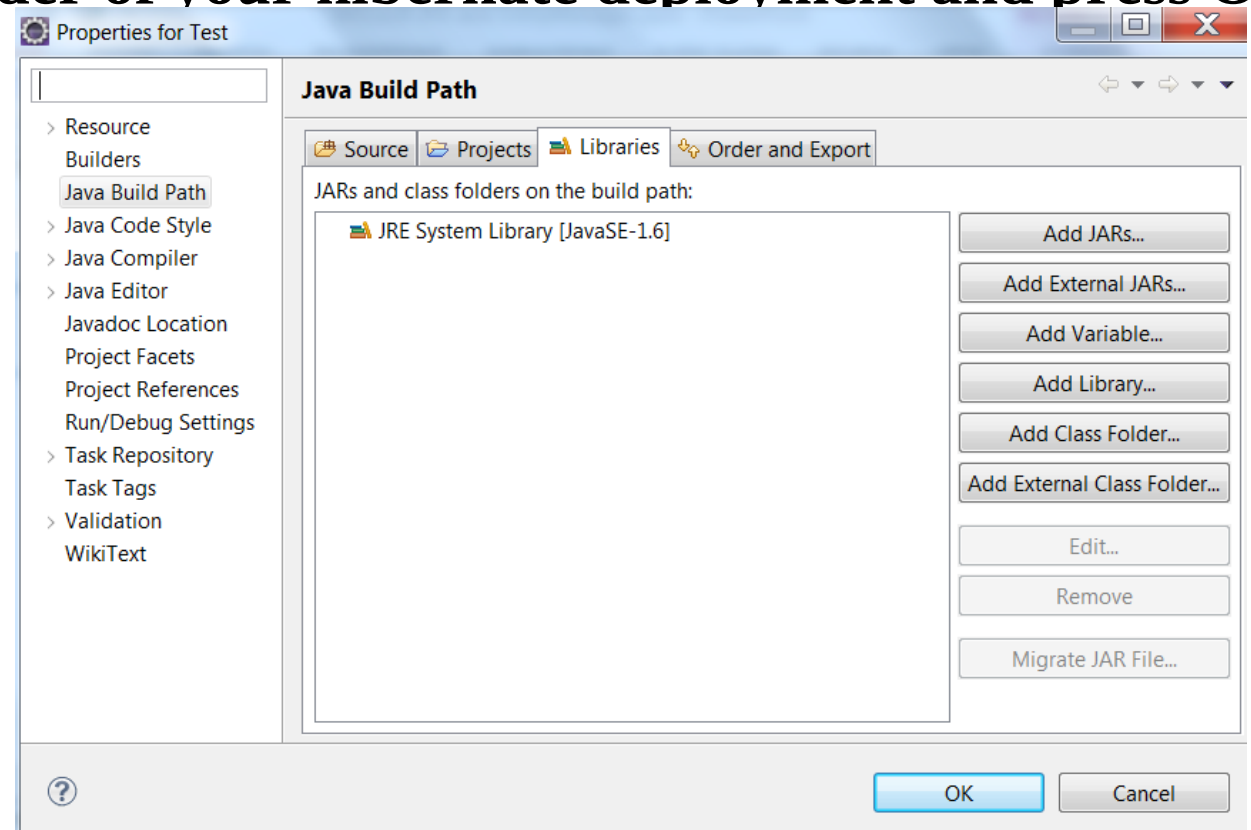
Adding Hibernate/JPA jars to your classpath in eclipse

- **Select or Click on your Java Project or Dynamic Web Project name in Eclipse, and the select Project->Properties for your project**
  - **You will see something like this**



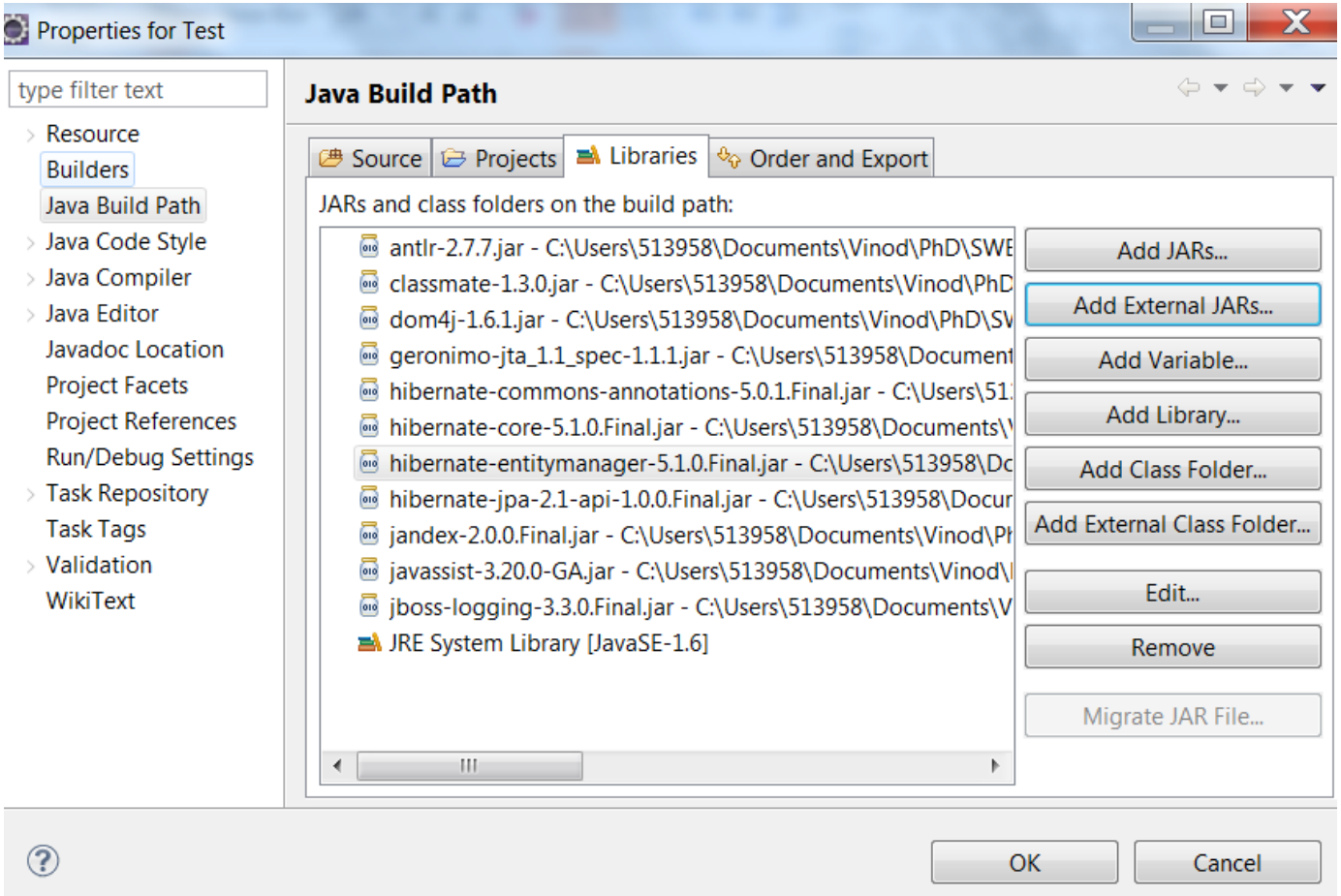
Adding Hibernate/JPA jars to your classpath in eclipse

- Now select **Java Build Path** (in the left panel) and the **Library** tab (as shown below)
- Then use **Add External Jars** link to add all jars in **lib/required** and **lib/jpa** folder of your hibernate deployment and press OK



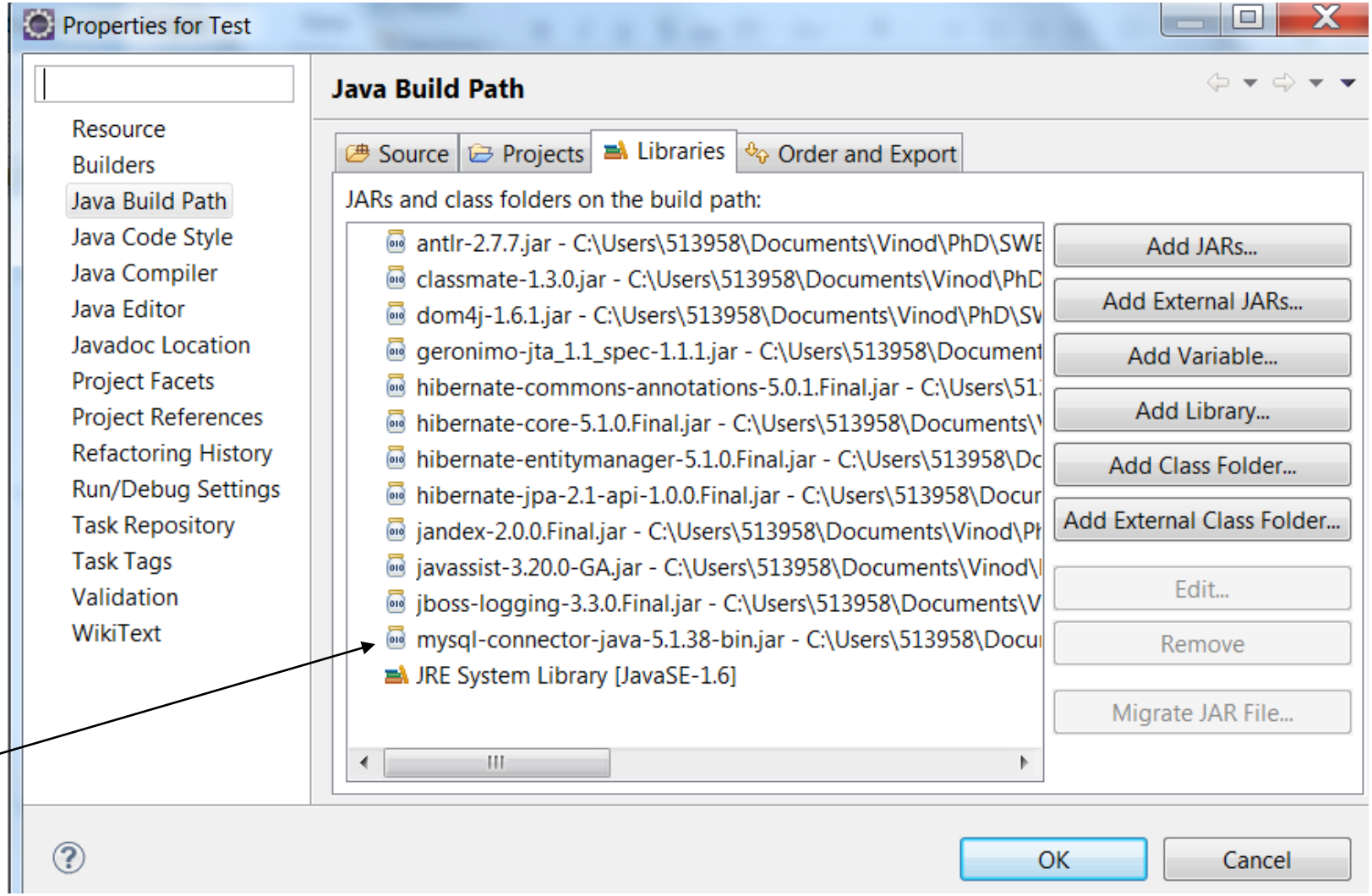
Adding Hibernate/JPA jars to your classpath in eclipse

- **After adding the jar files, you should see something like as shown below – now your project's class path has hibernate-core jar and all its dependencies jars as well as jpa jars!**



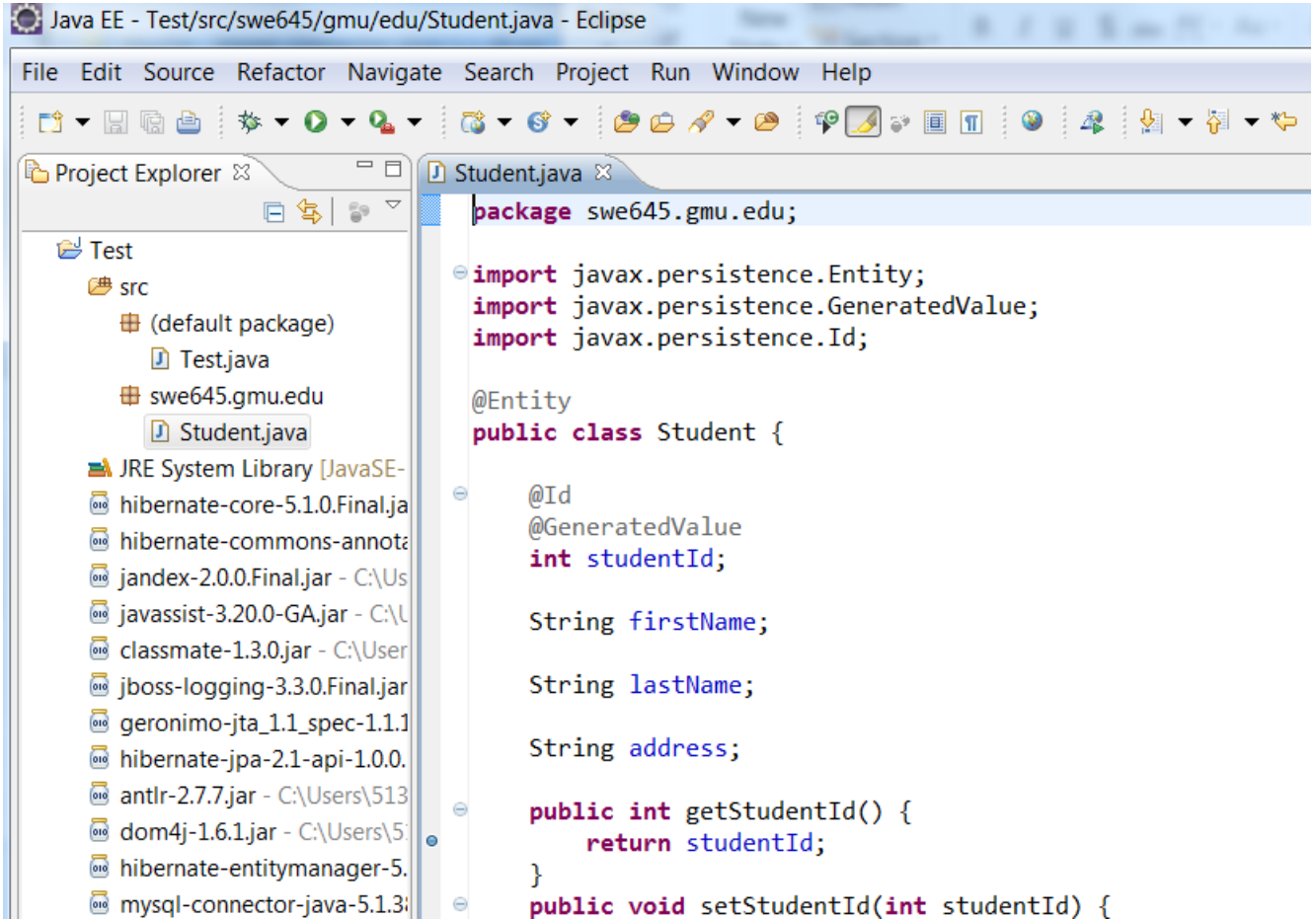
Adding Hibernate/JPA jars to your classpath in eclipse

- **Repeat the same steps to add MySQL JDBC Driver jar**



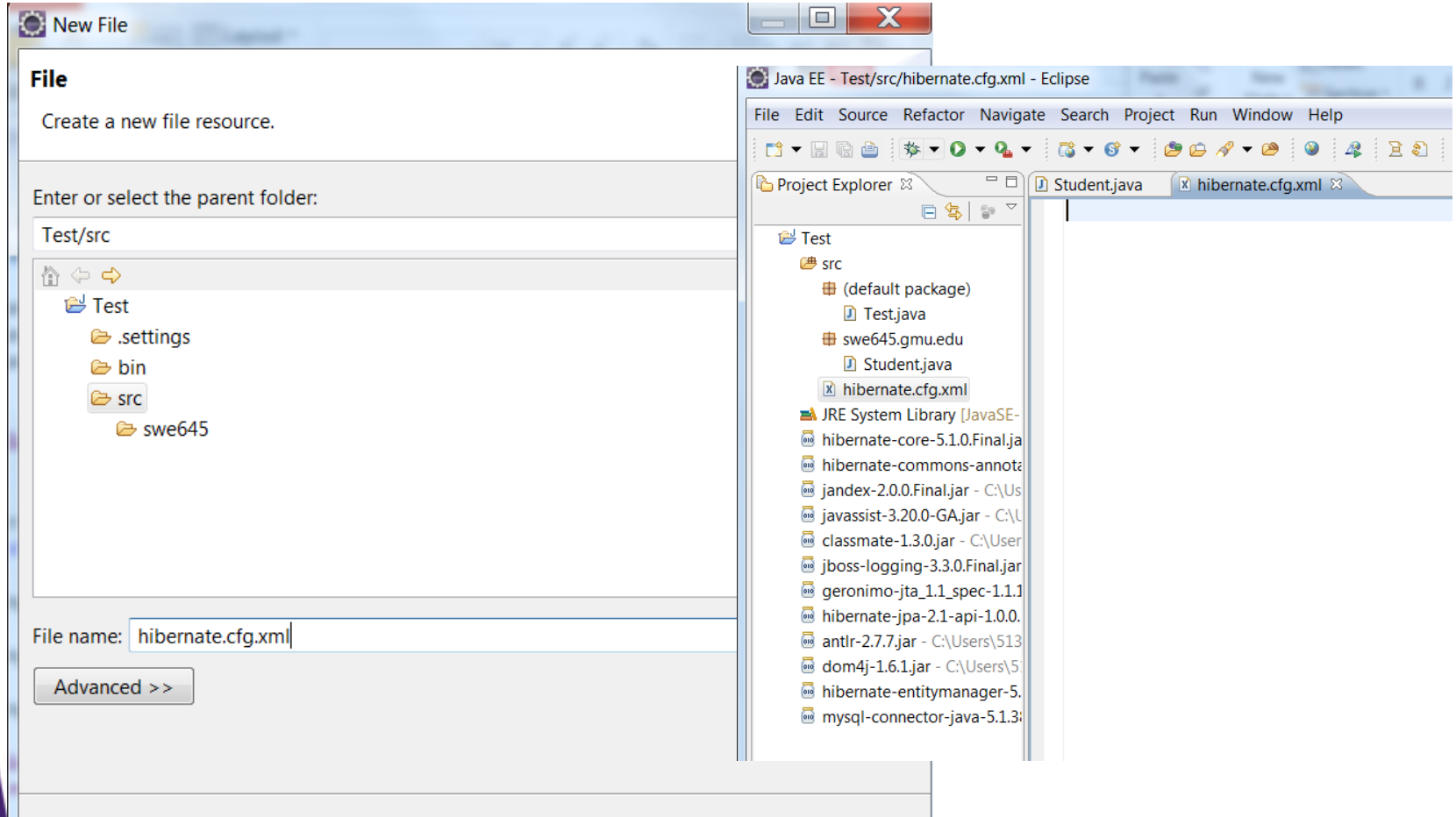
Your project is now ready to create JPA/Hibernate entity

- **You can now create POJO and use JPA/hibernate annotations to create your JPA or hibernate entity**
  - **Make sure to use import javax.persistence.***



Create Hibernate JPA Configuration file: persistence.xml

- **R-Click on your Java Project in Eclipse, select New-File and give it a name, persistence.xml, save it in src/META folder**



Create Hibernate JPA Configuration file: persistence.xml

- **Below is a template example of persistence.xml, which resides in the META-INF folder.**

```
<persistence xmlns="http://java.sun.com/xml/ns/persistence"
             xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
             xsi:schemaLocation="http://java.sun.com/xml/ns/persistence http://java.sun.com/xml/ns/persistence/persistence_2_0.xsd"
             version="2.0">
<persistence-unit name="sample">
   <provider>org.hibernate.ejb.HibernatePersistence</provider>
   <jta-data-source>java:/DefaultDS</jta-data-source>
   <mapping-file>ormap.xml</mapping-file>
   <jar-file>MyApp.jar</jar-file>
   <class>org.acme.Employee</class>
   <class>org.acme.Person</class>
   <class>org.acme.Address</class>
   <properties>
      <property name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
      <property name="hibernate.connection.password">XXXXXX</property>
      <property name="hibernate.connection.url">jdbc:mysql://<hostname>/<database></property>
      <property name="hibernate.connection.username">XXXXX</property>
      <property name="hibernate.default_schema">XXXXXX</property>
      <property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>
   </properties>
  </persistence-unit>
</persistence>
```

Create Hibernate JPA Configuration file: persistence.xml

- **Below is an example of persistence.xml, which resides in the src/META-INF folder. Here swe645 is the MySQL database name created via Amazon RDS.**

```
<?xml version="1.0" encoding="UTF-8"?>
- <persistence xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence
http://xmlns.jcp.org/xml/ns/persistence/persistence_2_1.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://xmlns.jcp.org/xml/ns/persistence" version="2.1">
  - <persistence-unit transaction-type="RESOURCE_LOCAL" name="assign3">
      <provider>org.hibernate.ejb.HibernatePersistence</provider>
      <class>assignment.Student</class>
    - <properties>
        <property name="hibernate.dialect" value="org.hibernate.dialect.MySQLDialect"/>
        <property name="hibernate.hbm2ddl.auto" value="update"/>
        <property name="hibernate.show_sql" value="true"/>
        <property name="hibernate.connection.driver_class" value="com.mysql.jdbc.Driver"/>
        <property name="hibernate.connection.url" value="jdbc:mysql://127.0.0.1:3306/swe645"/>
        <property name="hibernate.connection.username" value="nsimon2"/>
        <property name="hibernate.connection.password" value="nopassword"/>
      </properties>
  </persistence-unit>
</persistence>
```

Create Hibernate JPA Configuration file: persistence.xml

- **Below is an example of persistence.xml using GMU's Oracle database.**

```
<?xml version="1.0" encoding="UTF-8"?>
- <persistence xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence
http://xmlns.jcp.org/xml/ns/persistence/persistence_2_1.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://xmlns.jcp.org/xml/ns/persistence" version="2.1">
  - <persistence-unit transaction-type="RESOURCE_LOCAL" name="SWE645HW4">
      <class>Student</class>
    - <properties>
        <property name="hibernate.connection.url"
            value="jdbc:oracle:thin:@apollo.ite.gmu.edu:1521:ite10g"/>
        <property name="hibernate.connection.driver_class" value="oracle.jdbc.driver.OracleDriver"/>
        <property name="hibernate.connection.username" value="sshres18"/>
        <property name="hibernate.connection.password" value="esoals"/>
        <property name="hibernate.archive.autodetection" value="class"/>
        <property name="hibernate.show_sql" value="true"/>
        <property name="hibernate.format_sql" value="true"/>
        <property name="hbm2ddl.auto" value="update"/>
      </properties>
  </persistence-unit>
</persistence>
```