CSI5380: Systems & Architectures for

Electronic Commerce

Project Part 1: “Kill-9” CD Store –ReadMe Documentation

|  |  |  |
| --- | --- | --- |
| **Team Members** | **Student ID** | **Task** |
| Michele Belanger-Petrucci (Team Leader) | 566220 |  |
| Jiaren Suo |  |  |
| Yuan |  |  |
| Kaifan Zhu |  |  |
| Cynthia |  |  |

## Instructions for Installation

### Installation Eclipse for Windows

* **Step 0:** Install JDK

To use Eclipse for Java programming, you need to first install Java Development Kit (JDK).

* **Step 1:** Download

Download Eclipse from <https://www.eclipse.org/downloads/>

Choose "Eclipse IDE for Java Developers" (32-bit or 64-bit) (e.g., "eclipse-java-luna-SR1a-win32-x86\_64.zip").

* **Step 2:** Unzip

To install Eclipse, simply unzip the downloaded file into a directory of your choice. There is no need to run any installer.

When you install it successfully, you will see the welcome page as below.

Figure **1**: Successful installation of Eclipse IDE

### Installation Tomcat 8

* **Step 1:** Download

Download Tomcat from  http://tomcat.apache.org .

Click on Tomcat 8.0 link in Download section on the left the website, then choose zip in Cores section. In this moment your tomcat server is installed. But it is time configure it.

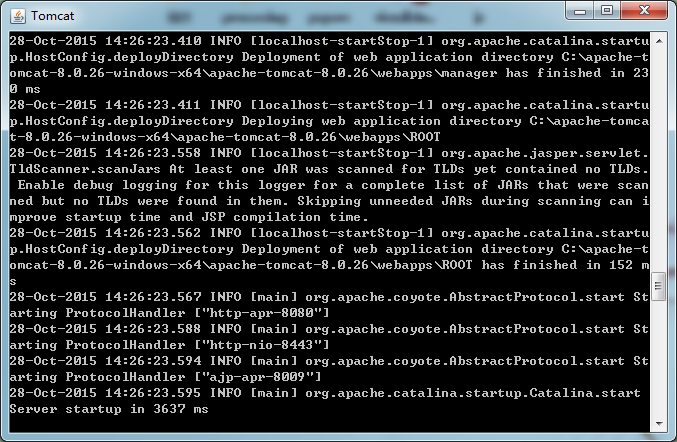
* **Step 2**: Include the Servlet library in the class path

So go to Start -> Control Panel -> System -> Advanced system settings. Switch to Advanced tab -> Environment Variables. In System Variables section click the new button.

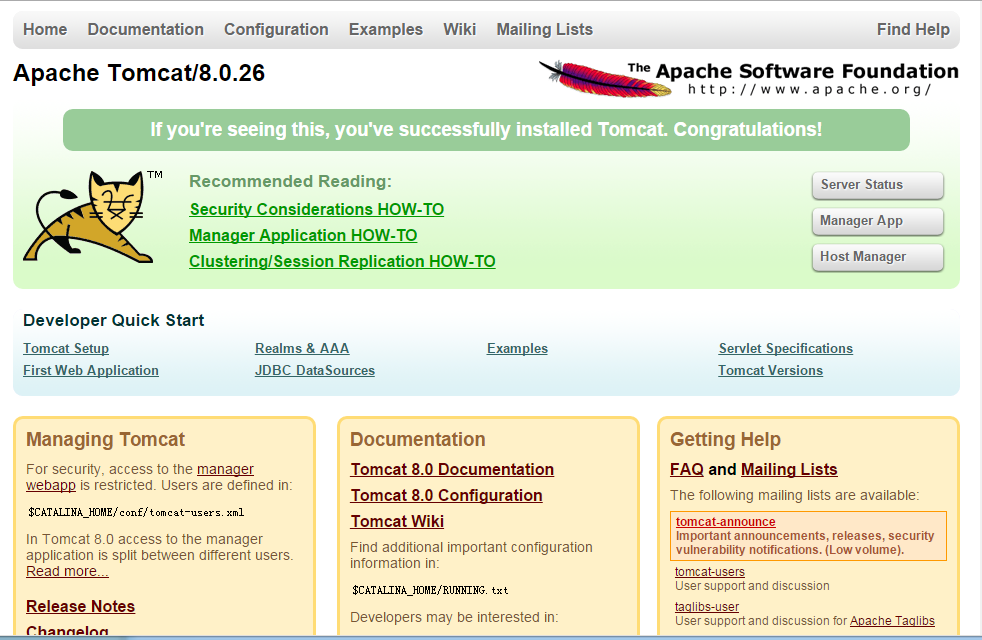
In Variable Name field write JAVA\_HOME and in Variable Value write your JDK installed directory. In System Variables section another time click the New button. In Variable Name field write JAVA\_JRE and in Variable Value write your JRE installed directory.

* **Step 3:** start tomcat service

Navigate to Tomcat's bin ("C:\apache-tomcat-8.0.26-windows-x64\apache-tomcat-8.0.26\bin”) folder and start the tomcat service.



#### Figure 2 : Tomcat Server



#### Figure 3: Tomcat running successfully

### Configure Tomcat on Eclipse

* **Step 1**: Start Eclipse and go to the Workbench. Then, click on Servers tab at bottom. (If you don't see Servers tab, add the tab via Window, Show View, Servers.) R-click on Servers tab, New, Server, Apache, Tomcat v8.0, navigate to the folder where you unzipped Tomcat, OK. You should now see "Tomcat v8.0 Server at localhost" listed under the Servers tab at the bottom.
* **Step 2:** Click on Servers tab at bottom. R-click on Tomcat v8.0, choose "Start". Open http://localhost:8080/.You will see a 404 error message, but at least the message comes from Tomcat.
* **Step 3:** Copy the ROOT Web App into Eclipse to rectify the error message.

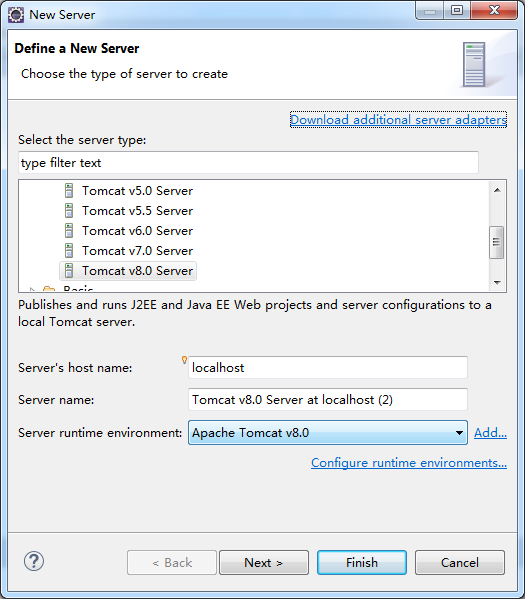


Figure **4:**

### Installing MySQL

### We use MySQL workbench for creating and managing SQL databases. We download and install MySQL 5.6, JConnector 5.1 from <http://www.mysql.com/downloads/> .

Figure 5: MySQL workbench

### SSL configuration

* **Step 1:** using keytool to create keystore to store the server's private key and self-signed certificate. "Keytool" included in the JDK is used to create keystore.

"%JAVA\_HOME%\bin\keytool" -genkey -alias tomcat -keyalg RSA

* http://tomcat.apache.org/tomcat-6.0-doc/images/void.gif**Step 2:** Copy the keystore file generated to tomcat folder (copy it to C:\apache-tomcat-8.0.26-windows-x64\apache-tomcat-8.0.26\conf/.keystore)
* **Step 3:** Configure Tomcat's SSL connector by changing "Server.xml" file content in the conf folder of tomcat(as Figure 6 )

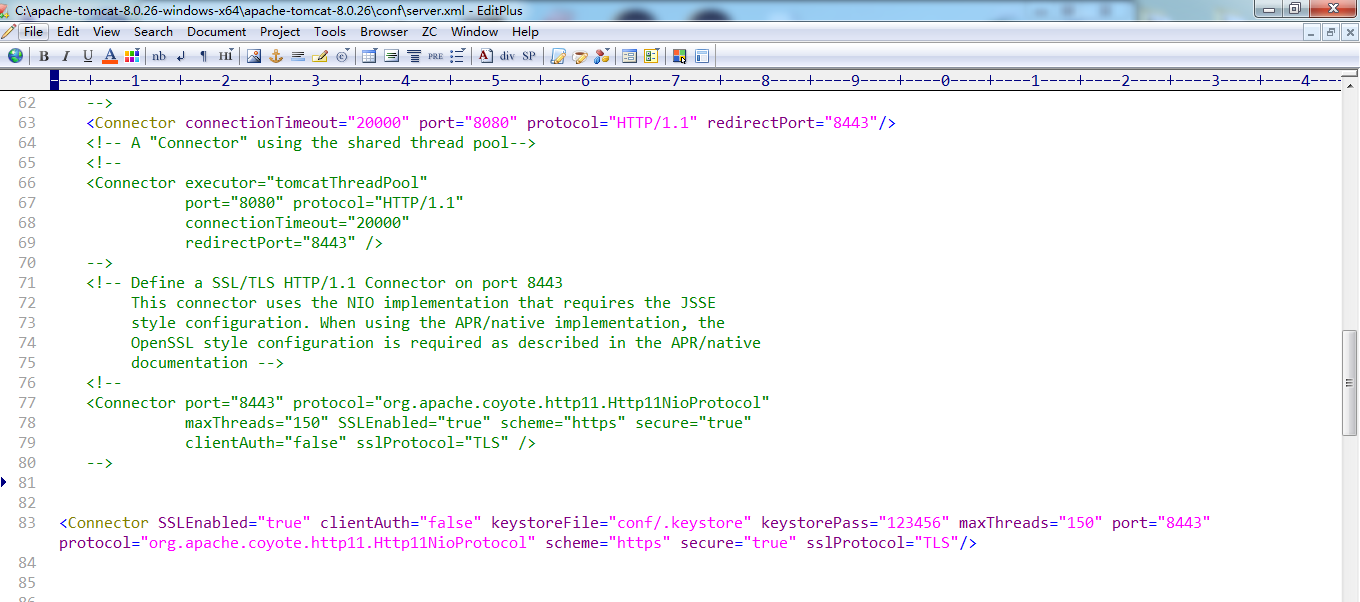


Figure 6: Configuring Tomcat's SSL Connector

* **Step 4:** Check if the Tomcat is successfully configured for https. Provide "[https://localhost:8443](https://localhost:8443/)" in the Browser URL and verify that you will see tomcat up and running successfully.

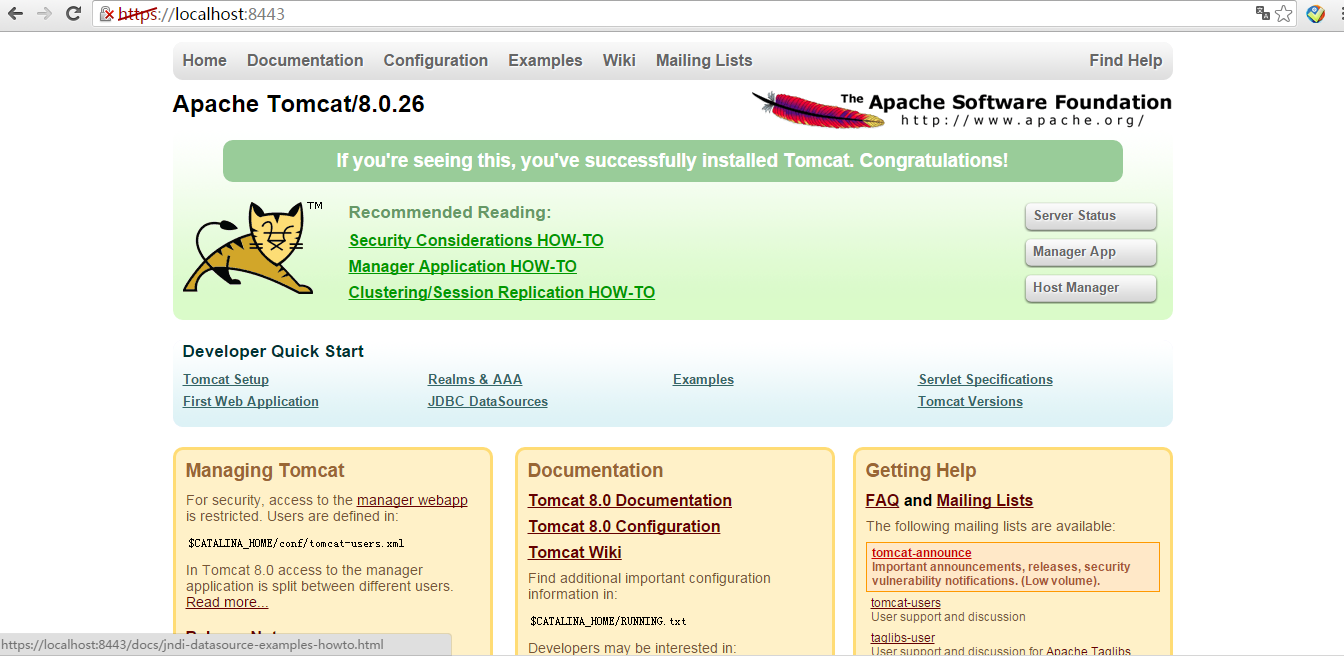


Figure 7 : Tomcat successfully configured to run on https

## 1.6 Connection Pool

To set up connection pooling, include the following commands in the **hibernate.config.xml** in order to leverage Tomcat’s connection pooling.

<?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>

<property name=*"hibernate.connection.datasource"*>java:/comp/env/jdbc/cd\_db</property>

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

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

<property name=*"hibernate.connection.password"*>password</property>

<property name=*"hibernate.connection.url"*>jdbc:mysql://localhost:3306/cd\_db</property>

<property name=*"hibernate.dialect"*>org.hibernate.dialect.MySQLDialect</property>

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

<property name=*"hibernate.hbm2ddl.auto"*>update</property>

<property name=*"hibernate.current\_session\_context\_class"*>thread</property>

<mapping class=*"com.K9.hibernate.bean.Account"*/>

<mapping class=*"com.K9.hibernate.bean.Address"*/>

<mapping class=*"com.K9.hibernate.bean.Category"*/>

<mapping class=*"com.K9.hibernate.bean.CD"*/>

<mapping class=*"com.K9.hibernate.bean.OrderItem"*/>

<mapping class=*"com.K9.hibernate.bean.Orders"*/>

</session-factory>

</hibernate-configuration>

In order to set up connection pooling in Tomcat, a context.xml file has to be created in the .. \WebContent\META-INF directory of the eclipse dynamic web project.

<?xml version=*'1.0'* encoding=*'utf-8'*?>

<Context>

<Resource name=*"jdbc/cd\_db"*

auth=*"Container"*

type=*"javax.sql.DataSource"*

factory=*"org.apache.tomcat.jdbc.pool.DataSourceFactory"*

validationInterval=*"30000"*

timeBetweenEvictionRunsMillis=*"30000"*

maxActive=*"100"*

minIdle=*"10"*

maxWait=*"10000"*

initialSize=*"10"*

username=*"root"*

password=*"password"*

driverClassName=*"com.mysql.jdbc.Driver"*

url=*"jdbc:mysql://localhost:3306/cd\_db"*/>

</Context>

## 1.7 Web Service Configuration

Web services are application services that allow different applications from different sources to communicate with each other. We have implemented two web services in our project.

* ProductCatalogService: Used for obtaining all the products and their details.
* OrderProcessingService: Used to get user details and also to add order details with the user selected items into database successfully.

Creating Web Services:

The web service we have implemented in our project is **SOAP**.

## 1.8 war file

In order to deploy the project onto a web server, a WAR file is created. This WAR file, in Tomcat, is copied to the \webapps directory. When Tomcat is running, the WAR file will automatically be opened and the proper directory structure will be created which will contain the necessary runtime files and JARs.