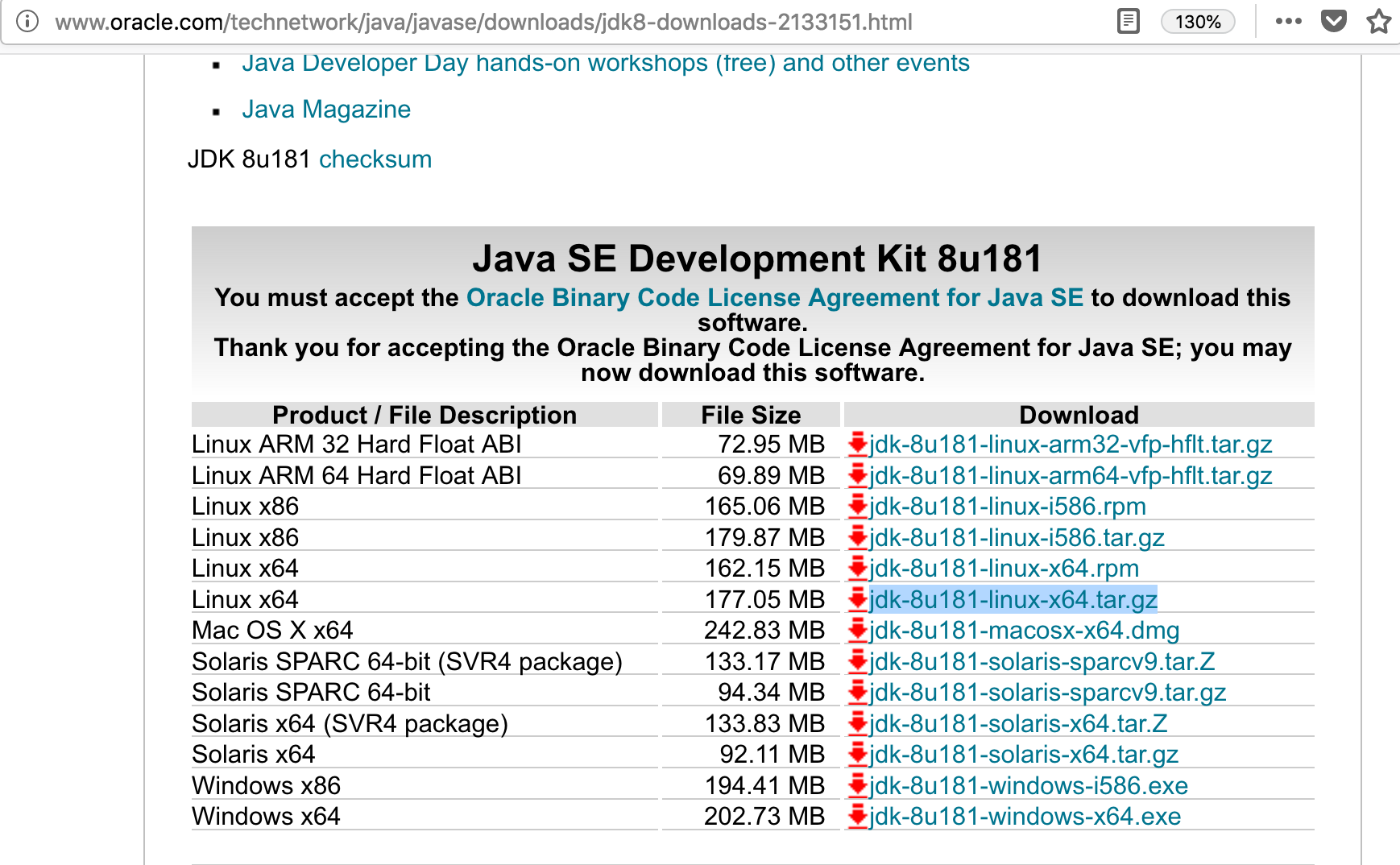
HTTP Cookies and Protecting them with Flags

Download Oracle JDK 8

<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

Download the (Linux x64) dk-8u181-linux-x64.tar.gz file for Fedora



Extract the JDK file in **/home/student/ssdlab3** folder, so your Java Home becomes **/home/student/ssdlab3/jdk1.8.0\_181**

Download Apache Maven and extract to /home/student/ssdlab3 folder.

<http://www-eu.apache.org/dist/maven/maven-3/3.5.4/binaries/apache-maven-3.5.4-bin.zip>

So, your Maven Home is **/home/student/ssdlab3/apache-maven-3.5.4**

Download Apache Tomcat and extract to /home/student/ssdlab3 folder.

<http://www-us.apache.org/dist/tomcat/tomcat-7/v7.0.90/bin/apache-tomcat-7.0.90.zip>

So, your Tomcat home is **/home/student/ssdlab3/apache-tomcat-7.0.90**

Open the .bashrc file in /home/student and add following content at the end of the file.

|  |
| --- |
| export JAVA\_HOME=**/home/student/ssdlab3/jdk1.8.0\_181**  export JRE\_HOME=**/home/student/ssdlab3/jdk1.8.0\_181/jre**  export M2\_HOME=**/home/student/ssdlab3/apache-maven-3.5.4**  export CATALINA\_HOME=**/home/student/ssdlab3/apache-tomcat-7.0.90**  export PATH=**$JAVA\_HOME/bin:$M2\_HOME/bin:$PATH** |

Close all terminals and open a new terminal, or execute the “source ~/.bashrc” command to apply above changes to environment variables.

Check if maven is working by typing “**mvn**” command in terminal.

Configuring HTTPS in tomcat

From **/home/student/ssdlab3** path in your terminal, execute following command.

**keytool -genkey -alias server -keyalg RSA -keystore keystore.jks -keysize 2048 -validity 365 -sigalg SHA256withRSA -storepass password -keypass password -dname "CN=server, OU=ID, O=IBM, L=Hursley, S=Hants, C=GB"**

This will create keystore.jks file in your /home/student/ssdlab3 path.

Open the **/home/student/ssdlab3/apache-tomcat-7.0.90/conf/server.xml** file.

It has following configuration commented out.

|  |
| --- |
| <!--  <Connector port="8443" protocol="org.apache.coyote.http11.Http11Protocol"  maxThreads="150" SSLEnabled="true" scheme="https" secure="true"  clientAuth="false" sslProtocol="TLS" />  --> |

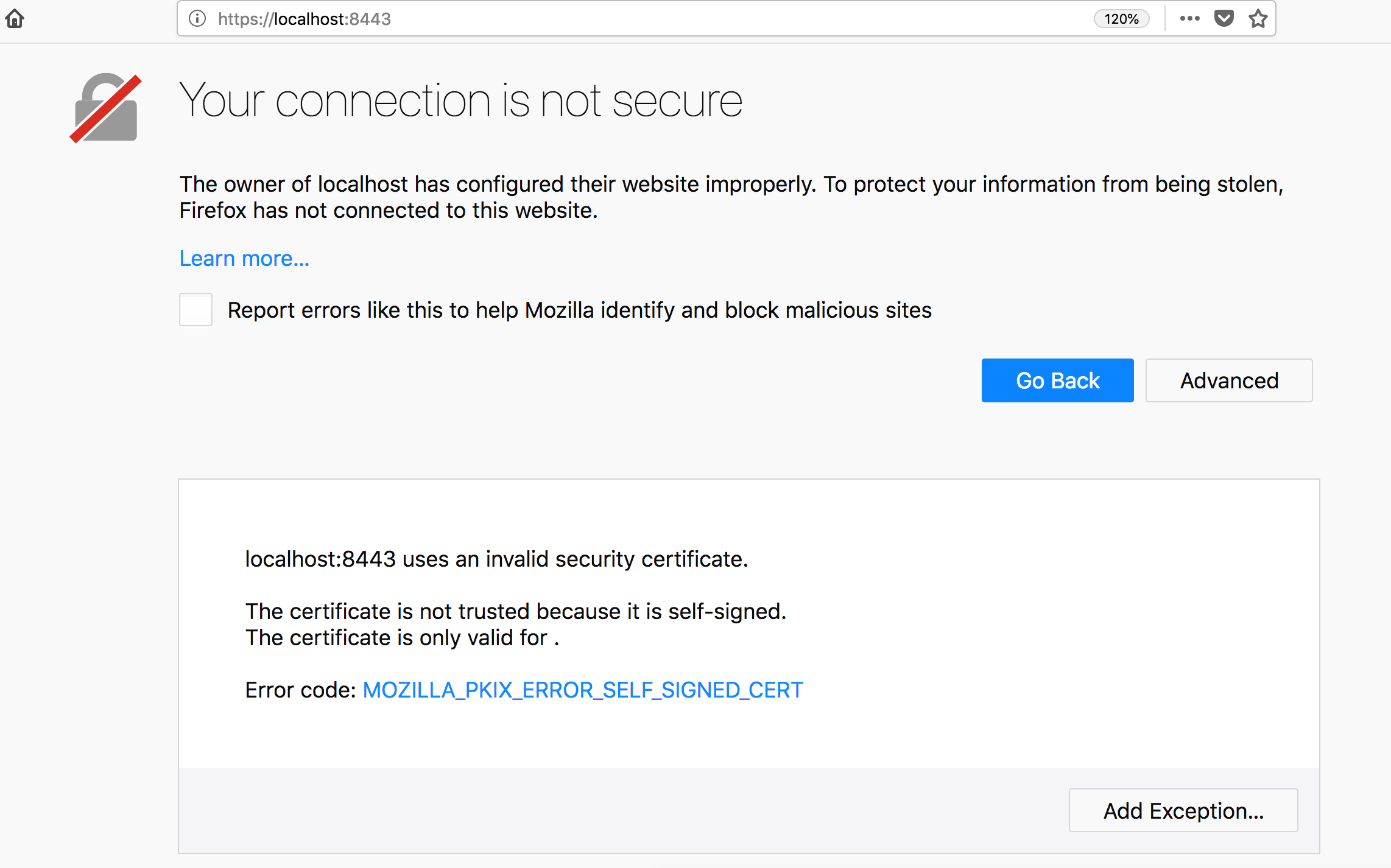
Under above, put following configuration.

|  |
| --- |
| <Connector  protocol="org.apache.coyote.http11.Http11NioProtocol"  port="8443" maxThreads="200"  scheme="https" secure="true" SSLEnabled="true"   **keystoreFile="/home/student/ssdlab3/keystore.jks" keystorePass="password"**  clientAuth="false" sslProtocol="TLS" ciphers="TLS\_ECDHE\_RSA\_WITH\_AES\_128\_CBC\_SHA256,TLS\_ECDHE\_RSA\_WITH\_AES\_128\_CBC\_SHA,TLS\_ECDHE\_RSA\_WITH\_AES\_256\_CBC\_SHA384,TLS\_ECDHE\_RSA\_WITH\_AES\_256\_CBC\_SHA,TLS\_ECDHsE\_RSA\_WITH\_RC4\_128\_SHA,TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA256,TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA,TLS\_RSA\_WITH\_AES\_256\_CBC\_SHA256,TLS\_RSA\_WITH\_AES\_256\_CBC\_SHA,SSL\_RSA\_WITH\_RC4\_128\_SHA"  /> |

Go into **/home/student/ssdlab3/apache-tomcat-7.0.90/bin** path from terminal and execute the following command.

**sh catalina.sh run**

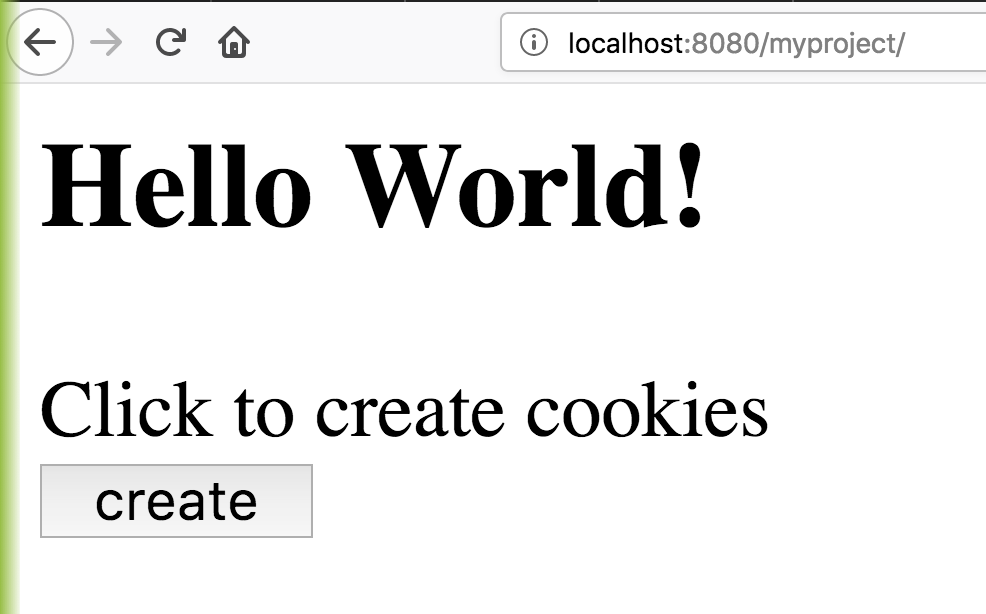
Open the browser and go to [**https://localhost:8443**](https://localhost:8443) and it should give SSL Warning due to using self-signed certificate. Add SSL Exception and view the tomcat page.



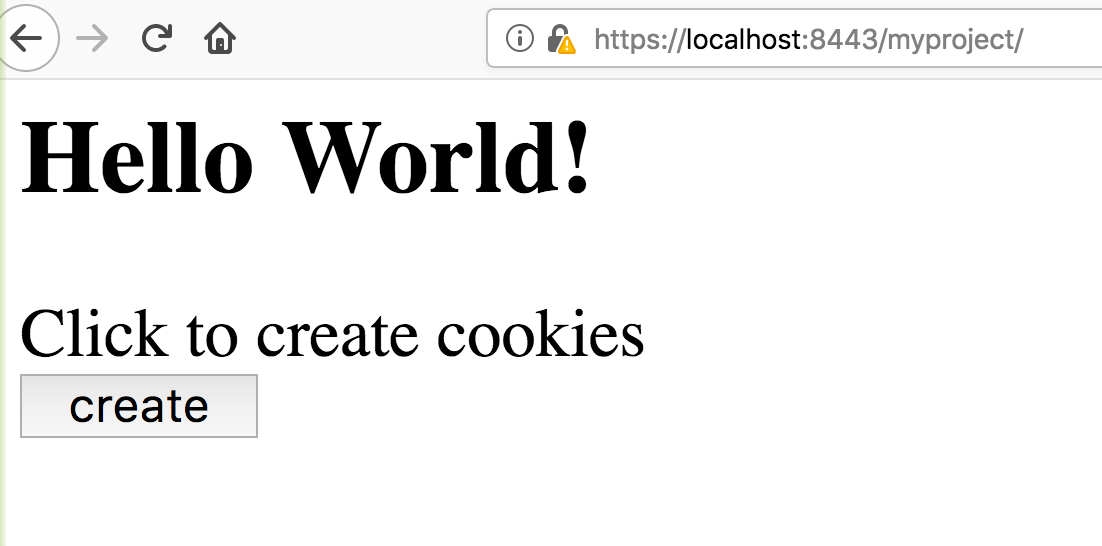
GitHub Sample Project

<https://github.com/thariyarox/java-cookies-sample>

<http://localhost:8080/myproject>



<https://localhost:8443/myproject>



<https://stackoverflow.com/questions/13147113/setting-an-httponly-cookie-with-javax-servlet-2-5>