Steps to implement HTTPS in tomcat server and Spring Framework

The SSL has been imported in the Project doing the following steps:

First, we opened the terminal and we type cd %JAVA_HOME%/bin, this command redirects us to the java directory where we have installed the tool “Keytool”.

We type: keytool -genkeypair -alias Mycert -keyalg RSA -keystore “C:\SSL\Mycert.cert” and give the follow information:

```
Enter keystore password:  pass123
Re-enter new password: pass123
What is your first and last name?
  [Unknown]:  Javi
What is the name of your organizational unit?
  [Unknown]:  us
What is the name of your organization?
  [Unknown]:  us
What is the name of your City or Locality?
  [Unknown]:  Sevilla
What is the name of your State or Province?
  [Unknown]:  Sevilla
What is the two-letter country code for this unit?
  [Unknown]:  ES
Is CN=Javi, OU=US, O=US, L=Sevilla, ST=Sevilla, C=ES correct?
  [no]:  yes
```

This it will create a keystore file named Mycert in path “C:\SSL\Mycert.cert”.

Next we should modify the file server.xml from our Tomcat server directory (/src/main/webapp/WEB-INF/web.xml) replacing this lines:

```
  <Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"
               maxThreads="150" scheme="https" secure="true"
               clientAuth="false" sslProtocol="TLS" />
```

by these:

```
<Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"
               maxThreads="150" scheme="https" secure="true"
               clientAuth="false" sslProtocol="TLS"  keystoreFile="C:/SSL/Mycert.cert"
keystorePass="pass123" />
```

Next we need to add these lines to “web.xml” file of our project (src/main/webapp/WEB-INF/)

```
<security-constraint>
<web-resource-collection>
<web-resource-name>Acme-Madruga</web-resource-name>
<url-pattern>/*</url-pattern>
</web-resource-collection>
<user-data-constraint>
<transport-guarantee>CONFIDENTIAL</transport-guarantee>
</user-data-constraint>
</security-constraint>
```

Please, If you want to deploy the project with https on your computer, add the Mycert file (included in the https.zip file in item 2, 4 and 5) in your folder “C:\SSL” and replace the lines from “server.xml” in you Tomcat Server folder (C:\Program Files\Apache Software Foundation\Tomcat 7.0\conf\server.xml) (like the example from upside).

We include both files used in our project in https.zip (item 2,4 and 5).

Remember you should start the application in port 8443 if you want to use https (https://localhost:8443/).