

Appendix J Deploying SOBA on GlassFish 3.1

Some readers are interested in deploying SOBA on GlassFish. Assuming that you have followed the instructions detailed in Chapter 2 and Appendix E and made sure that SOBA runs on Tomcat 7 in your environment, this appendix describes the steps for deploying SOBA 3.2 (Maven version) on GlassFish 3.1.

Step 1: In SOBA's `web.xml` file, replace the “*default*” servlet with the “*DefaultServlet*” as shown below if it's not already replaced (the former only works on Tomcat):

```
<!-- this is necessary in order to make SOBA run on glassfish and jboss -->
<servlet>
  <servlet-name>DefaultServlet</servlet-name>
  <servlet-class>org.apache.catalina.servlets.DefaultServlet</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>DefaultServlet</servlet-name>
  <url-pattern>*.jpg</url-pattern>
</servlet-mapping>
<!--
<servlet-mapping>
  <servlet-name>default</servlet-name>
  <url-pattern>*.jpg</url-pattern>
</servlet-mapping>
<servlet-mapping>
  <servlet-name>default</servlet-name>
  <url-pattern>*.gif</url-pattern>
</servlet-mapping>
-->
```

Step 2: Open up a command prompt and execute the command `asadmin` from the `bin` directory of your GlassFish installation (you need to start up your domain by executing `asadmin> start-domain` first if your GlassFish server is not running):

```
asadmin> create-ssl --type http-listener --certName sampleCert http-listener-2
```

Step 3: Verify the http listener created in the previous step by logging into your GlassFish admin console at <http://localhost:4848/> and then navigating to *Configurations* → *server-config* → *Network Config* → *Network Listeners* → *http-listener-2*. Similar to Figure J.1, it should show the port 8181 for the HTTPS protocol (remember that later you need to use this port number of 8181 instead of 8443 to start up SOBA).

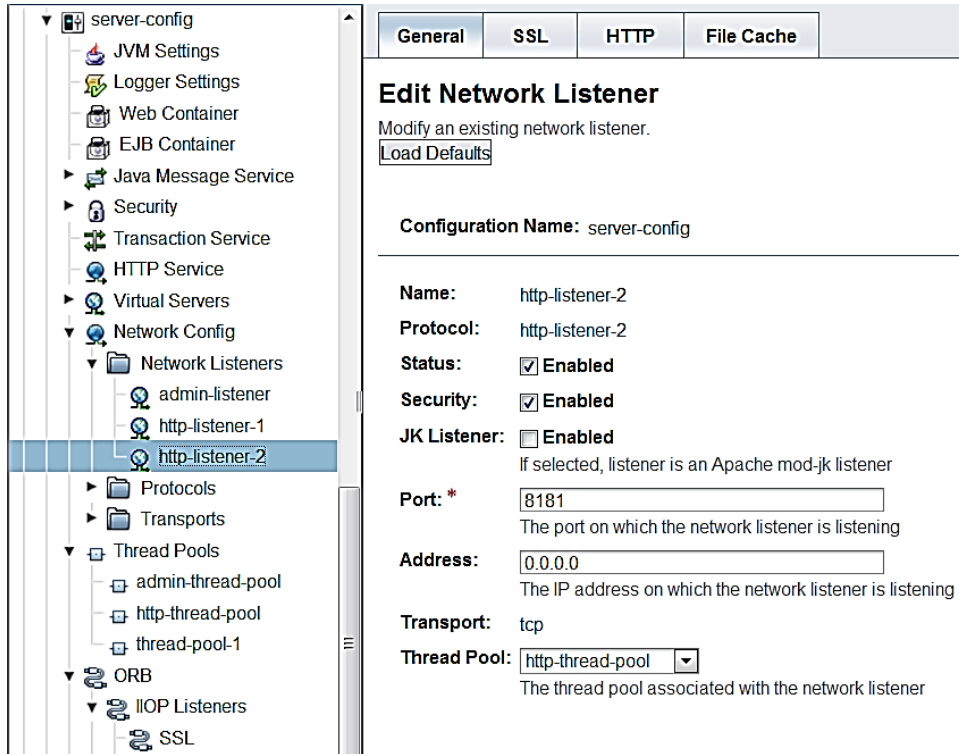


Figure J.1 GlassFish HTTPS/SSL configured to run at port 8181.

Step 4: Copy the *soba* folder (you need to rename it to *soba* from its default name of *soba3.2-3.2-mvn*) in your *target* directory built with Maven to the *domains/domain1/autodeploy* folder of your GlassFish installation (you can also use the *soba3.2-3.2-mvn.war* file but you need to rename it to *soba.war*).

Step 5: stop and restart your GlassFish server to be sure that SOBA is deployed successfully as follows:

```
asadmin> stop-domain
asadmin> start-domain
```

Now try <https://localhost:8181/soba/login.jsp> and you should see the normal SOBA login page (note that the context-root setting is ignored somehow by GlassFish, and <https://localhost:8181/soba> without adding `/login.jsp` would not work). That's all about how to deploy SOBA on GlassFish 3.1.

Appendix K Deploying SOBA on JBoss 7.1

Some readers are interested in deploying SOBA on JBoss. Assuming that you have followed the instructions detailed in Chapter 2 and Appendix E and made sure that SOBA runs on Tomcat 7 in your environment, this appendix describes the steps for deploying SOBA 3.2 (Maven version) on Jboss 7.1.

Step 1: In SOBA's `web.xml` file, replace the “*default*” servlet with the “*DefaultServlet*” as shown below if it's not already replaced (the former only works on Tomcat):

```
<!-- this is necessary in order to make SOBA run on glassfish and jboss -->
<servlet>
  <servlet-name>DefaultServlet</servlet-name>
  <servlet-class>org.apache.catalina.servlets.DefaultServlet</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>DefaultServlet</servlet-name>
  <url-pattern>*.jpg</url-pattern>
</servlet-mapping>
<!--
<servlet-mapping>
  <servlet-name>default</servlet-name>
  <url-pattern>*.jpg</url-pattern>
</servlet-mapping>
<servlet-mapping>
  <servlet-name>default</servlet-name>
  <url-pattern>*.gif</url-pattern>
</servlet-mapping>
-->
```

Step 2: In your `standalone.xml` file located in the `standalone/configurations` directory of your Jboss 7.1 installation directory, add the https connector as shown as highlighted below:

```
<subsystem xmlns="urn:jboss:domain:web:1.1"
  default-virtual-server="default-host" native="false">
  <connector name="http" protocol="HTTP/1.1" scheme="http"
    socket-binding="http"/>
  <connector name="https" protocol="HTTP/1.1" scheme="https"
    socket-binding="https" secure="true">
    <ssl name="https" password="changeit"
      certificate-key-file=" ../standalone/configuration/.keystore"/>
  </connector>
  <virtual-server name="default-host" enable-welcome-root="true">
    <alias name="localhost"/>
    <alias name="example.com"/>
  </virtual-server>
</subsystem>
```

Step 3: Copy your `.keystore` file from your `C:\Users\<login-name>` directory to the `standalone/configurations` directory of your Jboss 7.1 installation. Refer to Section 2.4.2 if you missed the step of creating a SSL testing certificate. Tomcat and Jboss require this certificate be created *a priori*, whereas the GlassFish has a built-in certificate and therefore doesn't require this certificate be created.

Step 4: Copy the `soba.war` file (you need to rename it to `soba.war` from its default name of `soba3.2-3.2-mvn.war`) in your `target` directory built with Maven to the `standalone/deployments` folder of your Jboss installation.

Step 5: Start your Jboss server to get SOBA deployed successfully by executing the follow command (you may need to set your `JBOSS_HOME` environment variable first):

```
<jboss-install>/bin> set JBOSS_HOME=<yourPath>/jboss-as-7.1.1.Final
<jboss-install>/bin> standalone.bat
```

Now try <https://localhost:8443/soba/login.jsp> and you should see the normal SOBA login page (note that the `context-root` setting is ignored somehow by Jboss, and <https://localhost:8443/soba> without adding `/login.jsp` would not work). That's all about how to deploy SOBA on JBoss 7.1