SHRINE 3.2.0 Chapter 12.4 - Sample server.xml File

After importing the signed certificates in addition to the Hub CA and HTTPS certificate, configure your Tomcat server.xml file to use the correct certificate to serve SHRINE https requests. Tomcat normally uses port 6443 to serve SHRINE.

To serve SSL find this section and change it to use the right keystore password and key alias to serve https from tomcat. Although you can use the same keyAlias to sign shrine queries and to support TLS for https most sites choose to use their own cert signed by a CA in a public cert tree. This prevents dire warnings from browsers. Follow standard procedures for serving https via TLS from tomcat. Find the example in shrine-setup/server.xml .

```
    <!-- Define a SSL/TLS HTTP/1.1 Connector on port 8443
         This connector uses the NIO implementation that requires the JSSE
         style configuration. When using the APR/native implementation, the
         OpenSSL style configuration is required as described in the APR/native
         documentation -->

    <Connector port="6443" protocol="org.apache.coyote.http11.Http11NioProtocol"
               maxThreads="150" SSLEnabled="true" scheme="https" secure="true"
               clientAuth="false" sslProtocol="TLS"
               keystoreFile="/opt/shrine/shrine.keystore"
               keystorePass="password"
               keyAlias="<name_of_keystore_PrivateKeyEntry>" />
```

The URLs to the SHRINE webclient, steward, and changed in SHRINE 2.0.0. If you would like to add redirects to the old URLs, add a RewriteValve to the server.xml file and copy the shrine-setup/rewrite.config file to **/opt/shrine/tomcat/conf/Catalina/localhost**.

server.xml

```
<Host name="localhost"  appBase="webapps" unpackWARs="true" autoDeploy="true">

    <Valve className="org.apache.catalina.valves.rewrite.RewriteValve" />
........
```

Here's a sample server.xml file (note the 'Connector port = "6443"' section):

server.xml

```
<?xml version='1.0' encoding='utf-8'?>
<!--
  Licensed to the Apache Software Foundation (ASF) under one or more
  contributor license agreements.  See the NOTICE file distributed with
  this work for additional information regarding copyright ownership.
  The ASF licenses this file to You under the Apache License, Version 2.0
  (the "License"); you may not use this file except in compliance with
  the License.  You may obtain a copy of the License at

      http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License.
-->


<!-- Note:  A "Server" is not itself a "Container", so you may not
     define subcomponents such as "Valves" at this level.
     Documentation at /docs/config/server.html
 -->

<Server port="8005" shutdown="SHUTDOWN">
```

```
<Listener className="org.apache.catalina.startup.VersionLoggerListener" />
<!-- Security listener. Documentation at /docs/config/listeners.html
<Listener className="org.apache.catalina.security.SecurityListener" />
-->

<!--APR library loader. Documentation at /docs/apr.html -->
<Listener className="org.apache.catalina.core.AprLifecycleListener" SSLEngine="on" />
<!-- Prevent memory leaks due to use of particular java/javax APIs-->
<Listener className="org.apache.catalina.core.JreMemoryLeakPreventionListener" />
<Listener className="org.apache.catalina.mbeans.GlobalResourcesLifecycleListener" />
<Listener className="org.apache.catalina.core.ThreadLocalLeakPreventionListener" />

<!-- Global JNDI resources
     Documentation at /docs/jndi-resources-howto.html
-->

<GlobalNamingResources>
  <!-- Editable user database that can also be used by
       UserDatabaseRealm to authenticate users
  -->

  <Resource name="UserDatabase" auth="Container"
            type="org.apache.catalina.UserDatabase"
            description="User database that can be updated and saved"
            factory="org.apache.catalina.users.MemoryUserDatabaseFactory"
            pathname="conf/tomcat-users.xml" />
</GlobalNamingResources>

<!-- A "Service" is a collection of one or more "Connectors" that share
     a single "Container" Note:  A "Service" is not itself a "Container",
     so you may not define subcomponents such as "Valves" at this level.
     Documentation at /docs/config/service.html
 -->

<Service name="Catalina">
  <!--The connectors can use a shared executor, you can define one or more named thread pools-->
  <!--
  <Executor name="tomcatThreadPool" namePrefix="catalina-exec-"
      maxThreads="150" minSpareThreads="4"/>
  -->

  <!-- A "Connector" represents an endpoint by which requests are received
       and responses are returned. Documentation at :
       Java HTTP Connector: /docs/config/http.html (blocking & non-blocking)
       Java AJP  Connector: /docs/config/ajp.html
       APR (HTTP/AJP) Connector: /docs/apr.html
       Define a non-SSL/TLS HTTP/1.1 Connector on port 8080
  -->

  <Connector port="6060" protocol="HTTP/1.1"
             connectionTimeout="20000"
             redirectPort="6443" />


  <!-- A "Connector" using the shared thread pool-->
  <!--
  <Connector executor="tomcatThreadPool"
             port="8080" protocol="HTTP/1.1"
             connectionTimeout="20000"
             redirectPort="8443" />
  -->

  <!-- Define a SSL/TLS HTTP/1.1 Connector on port 8443
       This connector uses the NIO implementation that requires the JSSE
       style configuration. When using the APR/native implementation, the
       OpenSSL style configuration is required as described in the APR/native
       documentation -->

  <Connector port="6443" protocol="org.apache.coyote.http11.Http11NioProtocol"
             maxThreads="150" SSLEnabled="true" scheme="https" secure="true"
             clientAuth="false" sslProtocol="TLS"
```

```
               keystoreFile="/opt/shrine/shrine.keystore"
               keystorePass="password"
               keyAlias="<name_of_keystore_PrivateKeyEntry>"
/>

    <!-- Define an AJP 1.3 Connector on port 8009 -->
    <Connector port="6009" protocol="AJP/1.3" redirectPort="8443" />


    <!-- An Engine represents the entry point (within Catalina) that processes
         every request.  The Engine implementation for Tomcat stand alone
         analyzes the HTTP headers included with the request, and passes them
         on to the appropriate Host (virtual host).
         Documentation at /docs/config/engine.html -->

    <!-- You should set jvmRoute to support load-balancing via AJP ie :
    <Engine name="Catalina" defaultHost="localhost" jvmRoute="jvm1">
    -->

    <Engine name="Catalina" defaultHost="localhost">


      <!--For clustering, please take a look at documentation at:
          /docs/cluster-howto.html  (simple how to)
          /docs/config/cluster.html (reference documentation) -->
      <!--
      <Cluster className="org.apache.catalina.ha.tcp.SimpleTcpCluster"/>
      -->

      <!-- Use the LockOutRealm to prevent attempts to guess user passwords
           via a brute-force attack -->


      <Realm className="org.apache.catalina.realm.LockOutRealm">
        <!-- This Realm uses the UserDatabase configured in the global JNDI
             resources under the key "UserDatabase".  Any edits
             that are performed against this UserDatabase are immediately
             available for use by the Realm.  -->


        <Realm className="org.apache.catalina.realm.UserDatabaseRealm"
               resourceName="UserDatabase"/>
      </Realm>

      <Host name="localhost"  appBase="webapps"
            unpackWARs="true" autoDeploy="true">


      <Valve className="org.apache.catalina.valves.rewrite.RewriteValve" />

        <!-- SingleSignOn valve, share authentication between web applications
             Documentation at: /docs/config/valve.html -->
        <!--
        <Valve className="org.apache.catalina.authenticator.SingleSignOn" />
        -->

        <!-- Access log processes all example.
             Documentation at: /docs/config/valve.html
             Note: The pattern used is equivalent to using pattern="common" -->


        <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
               prefix="localhost_access_log" suffix=".txt"
               pattern="%h %l %u %t &quot;%r&quot; %s %b" />


      </Host>
    </Engine>
  </Service>
</Server>
```