How to configure Tomcat to support SSL or https

1. Generate Keystore

First, uses "keytool" command to create a self-signed certificate. During the keystore creation process, you need to assign a password and fill in the certificate's detail.

```
$Tomcat\bin>keytool -genkey -alias ecicert -keyalg RSA -keystore /home/gist/ecicert
Enter keystore password:
Re-enter new password:
What is your first and last name?
  [Unknown]:  Write your full name
What is the name of your organizational unit?
  //omitted to save space
  [no]:  yes

Enter key password for <ecicert>
        (RETURN if same as keystore password):
Re-enter new password:

$Tomcat\bin>
```

Here, you just created a certificate named "**ecicert**", which locate at "**/home/gist**".

Certificate Details
You can use same "keytool" command to list the existing certificate's detail

```
$Tomcat\bin>keytool -list -keystore /home/gist
Enter keystore password:

Keystore type: JKS
Keystore provider: SUN

Your keystore contains 1 entry

ecicert, 14 November 2014, PrivateKeyEntry,
Certificate fingerprint (MD5): C8:DD:A1:AF:9F:55:A0:7F:6E:98:10:DE:8C:63:1B:A5

$Tomcat\bin>
```

2. Connector in server.xml

Next, locate your Tomcat's server configuration file at $Tomcat\conf\server.xml, modify it by adding a connector element to support for SSL or https connection.

File : $Tomcat\conf\server.xml

```
//...
<!-- Define a SSL HTTP/1.1 Connector on port 443
     This connector uses the JSSE configuration, when using APR, the
     connector should be using the OpenSSL style configuration
     described in the APR documentation -->

<Connector port="443" protocol="HTTP/1.1" SSLEnabled="true"
          maxThreads="150" scheme="https" secure="true"
          clientAuth="false" sslProtocol="TLS"
              keystoreFile="/home/gist/ecicert"
              keystorePass="password" />
//...
```

Note

keystorePass="password" is the password you assigned to your keystore via “keytool” command.

3. Redirect to port 443 from port 80/8080 :

Update server.xml configuration file in Tomcat home directory and change the following part of its configuration:

```
<Connector port="8080" protocol="HTTP/1.1"
          connectionTimeout="20000"
         URIEncoding="UTF-8"
         redirectPort="443" />
```

to what's shown below:

```
<Connector port="8080" enableLookups="false"
```

```
        redirectPort="443" />
```

Now, **Update web.xml configuration file** in Tomcat home directory and add the following content into the end **before the closing </web-app> markup**:

```
<!-- Done for HTTP to HTTPS redirection. The Tomcat always requires secure connection now. --
>
        <security-constraint>
      <web-resource-collection>
        <web-resource-name>HTTPSOnly</web-resource-name>
        <url-pattern>/*</url-pattern>
      </web-resource-collection>
      <user-data-constraint>
        <transport-guarantee>CONFIDENTIAL</transport-guarantee>
      </user-data-constraint>
    </security-constraint>
    <security-constraint>
      <web-resource-collection>
        <web-resource-name>HTTPSOrHTTP</web-resource-name>
        <url-pattern>*.ico</url-pattern>
        <url-pattern>/images/*</url-pattern>
        <url-pattern>/css/*</url-pattern>
      </web-resource-collection>
      <user-data-constraint>
        <transport-guarantee>NONE</transport-guarantee>
      </user-data-constraint>
    </security-constraint>
```

Restart Tomcat servlet container. **The Tomcat always requires secure connection now.**