Here's how to set up **Client-Certificate Authentication (Mutual TLS)** on **Apache Tomcat in Windows 11**, with full configuration, key generation steps, and a working web application.

**✅ Overview of Steps on Windows 11**

| **Step** | **Task** |
| --- | --- |
| 1️⃣ | Generate server and client certificates using keytool |
| 2️⃣ | Configure Tomcat's server.xml with SSL & clientAuth="true" |
| 3️⃣ | Write web.xml with CLIENT-CERT authentication |
| 4️⃣ | Create a protected JSP resource |
| 5️⃣ | Import client certificate into your browser |
| 6️⃣ | Test with a browser or curl |

**📁 Folder Setup**

Create a working folder like:

C:\TomcatCertAuth\

Inside:

* server.keystore
* server.truststore
* client.keystore
* WAR folder for your web app

**🔐 1. Generate Certificates using keytool (in CMD or PowerShell)**

🧰 Make sure keytool is in your system PATH (comes with JDK).

**Open Command Prompt and run:**

cd C:\TomcatCertAuth

**1.1 Generate Server Key + Keystore**

keytool -genkeypair -alias server -keyalg RSA -keysize 2048 -validity 365 ^

-keystore server.keystore -storepass changeit ^

-dname "CN=localhost, OU=Dev, O=MyOrg, L=City, ST=State, C=IN"

**1.2 Export Server Certificate**

keytool -exportcert -alias server -keystore server.keystore ^

-file server.crt -storepass changeit

**1.3 Generate Client Key + Keystore**

keytool -genkeypair -alias client -keyalg RSA -keysize 2048 -validity 365 ^

-keystore client.keystore -storepass changeit ^

-dname "CN=client, OU=Client, O=Org, L=City, ST=State, C=IN"

**1.4 Export Client Certificate**

keytool -exportcert -alias client -keystore client.keystore ^

-file client.crt -storepass changeit

**1.5 Create Server Truststore (to trust client)**

keytool -importcert -alias client -file client.crt ^

-keystore server.truststore -storepass changeit -noprompt

**⚙️ 2. Configure Tomcat server.xml**

**File:**

C:\Tomcat\conf\server.xml

**Add SSL connector under <Service>:**

<Connector port="8443"

protocol="org.apache.coyote.http11.Http11NioProtocol"

maxThreads="200"

SSLEnabled="true"

scheme="https"

secure="true"

clientAuth="true"

sslProtocol="TLS"

keystoreFile="C:/TomcatCertAuth/server.keystore"

keystorePass="changeit"

truststoreFile="C:/TomcatCertAuth/server.truststore"

truststorePass="changeit" />

✅ Use **forward slashes (/)** or escape backslashes (\\) in paths.

**🛡️ 3. Web Application Files**

**📁 Folder structure:**

C:\Tomcat\webapps\ClientCertApp\

│

├── index.jsp

├── WEB-INF\

│ └── web.xml

└── secure\

└── secret.jsp

**🔹 web.xml**

<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee" version="3.1">

<security-constraint>

<web-resource-collection>

<web-resource-name>CertArea</web-resource-name>

<url-pattern>/secure/\*</url-pattern>

</web-resource-collection>

<auth-constraint>

<role-name>ssluser</role-name>

</auth-constraint>

<user-data-constraint>

<transport-guarantee>CONFIDENTIAL</transport-guarantee>

</user-data-constraint>

</security-constraint>

<login-config>

<auth-method>CLIENT-CERT</auth-method>

<realm-name>CertRealm</realm-name>

</login-config>

<security-role>

<role-name>ssluser</role-name>

</security-role>

</web-app>

**🔹 secure/secret.jsp**

<html>

<body>

<h2>Client Certificate Authenticated!</h2>

<p>Welcome <%= request.getRemoteUser() %></p>

<p>User Principal: <%= request.getUserPrincipal() %></p>

</body>

</html>

**🌐 4. Import Client Certificate to Browser**

To test in Chrome or Firefox:

**Convert client.keystore to PKCS12 format:**

keytool -importkeystore ^

-srckeystore client.keystore -srcstoretype JKS -srcstorepass changeit ^

-destkeystore client.p12 -deststoretype PKCS12 -deststorepass changeit

**Import into browser:**

1. Open Chrome → Settings → Privacy and Security → Security → Manage Certificates
2. Import client.p12 (choose Personal/Trusted)
3. Password: changeit

**🧪 5. Test in Browser**

* Restart Tomcat
* Open:  
  👉 https://localhost:8443/ClientCertApp/secure/secret.jsp
* You’ll be prompted to select a certificate.
* If valid, you’ll see the authenticated message.

**🧰 Optional: Test with Curl (Windows)**

curl --cert client.p12 --cert-type P12 --pass changeit ^

https://localhost:8443/ClientCertApp/secure/secret.jsp -k

**✅ Summary**

| **File** | **Purpose** |
| --- | --- |
| server.keystore | Server certificate & key |
| client.keystore | Client certificate & key |
| server.truststore | Trusts client cert |
| server.xml | SSL connector with clientAuth=true |
| web.xml | CLIENT-CERT authentication |
| secret.jsp | Secured resource |

**Google Chrome** on Windows **does not have a built-in certificate import interface** like Firefox. It relies on the **Windows Certificate Store**. To use client certificates (like client.p12) in Chrome, you must import them via **Windows built-in Certificate Manager**.

**✅ Solution: Import .p12 into Windows Certificate Store**

**🧰 Steps (Windows 11 / 10):**

1. **Press Windows + R** → type:

certmgr.msc

→ Press **Enter**  
(This opens the Windows Certificate Manager for the current user.)

1. Navigate to:

Personal → Certificates

1. **Right-click** on Certificates →  
   Select **All Tasks → Import...**
2. The **Certificate Import Wizard** will appear:
   * Click **Next**
   * Browse to your file: client.p12
   * Click **Next**
   * Enter password: changeit (or your own password used with keytool)
   * Keep the option: **Place all certificates in the following store** → **Personal**
   * Click **Finish**
3. You should see a success message:  
   *"The import was successful."*

**✅ Test in Chrome**

1. Open Chrome.
2. Visit:

https://localhost:8443/ClientCertApp/secure/secret.jsp

1. Chrome should now prompt:  
   *"Select a certificate to authenticate yourself"*

If Chrome doesn’t prompt, ensure:

* Tomcat SSL is running with clientAuth="true"
* The certificate is issued to **CN=client**
* The client cert is trusted by Tomcat’s truststore

The net::ERR\_CERT\_AUTHORITY\_INVALID error in Google Chrome means that the **server's SSL certificate is not trusted** by the browser. This is common in **test/dev environments** using **self-signed certificates** (like the one we generated with keytool).

**✅ Why This Happens**

When you generate a certificate using keytool, it is **self-signed**, meaning:

* It was not issued by a trusted Certificate Authority (CA).
* Browsers like Chrome or Firefox do **not trust it by default**.

**✅ Safe Ways to Fix for Localhost Testing**

**🔹 Option 1: Proceed Manually (Dev Only)**

On https://localhost:8443, Chrome may show a warning:

*"Your connection is not private"*  
NET::ERR\_CERT\_AUTHORITY\_INVALID

Click:

* **Advanced** → **Proceed to localhost (unsafe)**

This is OK for development but **not suitable for production**.

**🔹 Option 2: Trust the Self-Signed Certificate**

You can **manually trust the server certificate** in Windows so Chrome stops showing the error.

**✅ Steps:**

1. Export the server certificate (you already did):

keytool -exportcert -alias server -file server.crt -keystore server.keystore -storepass changeit

1. Open **certmgr.msc**:
   * Press Windows + R, type certmgr.msc, press **Enter**.
2. In **left panel**, go to:

Trusted Root Certification Authorities → Certificates

1. Right-click on **Certificates** → **All Tasks → Import...**
2. Import server.crt:
   * Choose **Place all certificates in the following store**:  
     → ✅ Trusted Root Certification Authorities
   * Complete the wizard.
3. Restart Chrome.

**🧪 Verify It Works:**

1. Restart Tomcat.
2. Open Chrome:

https://localhost:8443/ClientCertApp/secure/secret.jsp

1. The error should be gone, and you'll be prompted to select your client certificate.