

# TestLink Plugin SSL support

In order to use SSL with TestLink Plugin you have to import the certificate of your TestLink server to the list of trusted certificates of your JRE. If you try to use the plug-in without importing your certificate you may see an error similar to the one in the following screen.

```
Started by user admin
[testng] $ cmd /c call C:\WINDOWS\TEMP\hudson7504627367292723478.bat

C:\dev\java\qa_workspace\testng>mvn -f pom-test.xml clean
[INFO] Scanning for projects...
[INFO] -----
[INFO] Building TestNG
[INFO]   task-segment: [clean]
[INFO] -----
[INFO] [clean:clean {execution: default-clean}]
[INFO] -----
[INFO] BUILD SUCCESSFUL
[INFO] -----
[INFO] Total time: < 1 second
[INFO] Finished at: Sat Apr 09 00:41:45 BRT 2011
[INFO] Final Memory: 4M/15M
[INFO] -----
Preparing TestLink client API.
Using TestLink URL: https://localhost/testlink-1.9.2/lib/api/xmlrpc.php

FATAL: Error verifying developer key: I/O error while communicating with HTTP server: sun.security.validator.ValidatorException: PKIX p
sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested target
br.eti.kinoshita.testlinkjavaapi.TestLinkAPIException: Error verifying developer key: I/O error while communicating with HTTP server:
sun.security.validator.ValidatorException: PKIX path building failed: sun.security.provider.certpath.SunCertPathBuilderException: unabl
certification path to requested target
    at br.eti.kinoshita.testlinkjavaapi.MiscService.checkDevKey(MiscService.java:71)
    at br.eti.kinoshita.testlinkjavaapi.TestLinkAPI.<init>(TestLinkAPI.java:128)
    at hudson.plugins.testlink.TestLinkService.<init>(TestLinkService.java:97)
    at hudson.plugins.testlink.TestLinkBuilder.perform(TestLinkBuilder.java:447)
    at hudson.tasks.BuildStepMonitor$1.perform(BuildStepMonitor.java:19)
    at hudson.model.AbstractBuild$AbstractRunner.perform(AbstractBuild.java:649)
```

First you have to obtain a local copy of your certificate. You can do it with Firefox, for instance. Just open your https URL, right click on the displayed page, Show Info / Security, look for the export button and save it to somewhere in your computer.

Now you need to import this certificate into a keystore.

```
keytool -importcert -keystore jssecacerts -trustcacerts -alias localhost -file localhost.crt
```

```
C:\WINDOWS\system32\cmd.exe

C:\opt\java\jdk1.6.0_24\jre\lib\security>keytool -importcert -keystore jssecacerts -trustcacerts -alias localhost -file localhost.crt
Enter keystore password:
Re-enter new password:
Owner: CN=localhost
Issuer: CN=localhost
Serial number: b5c752c98781b503
Valid from: Tue Nov 10 21:48:47 BRST 2009 until: Fri Nov 08 21:48:47 BRST 2019
Certificate fingerprints:
    MD5:  A0:A4:4C:C9:9E:84:B2:6F:9E:63:9F:9E:D2:29:DE:E0
    SHA1: B0:23:8C:54:7A:90:5B:FA:11:9C:4E:8B:AC:CA:EA:CF:36:49:1F:F6
    Signature algorithm name: SHA1withRSA
    Version: 1
Trust this certificate? [no]: yes
Certificate was added to keystore

C:\opt\java\jdk1.6.0_24\jre\lib\security>keytool -list -keystore jssecacerts
Enter keystore password:

Keystore type: JKS
Keystore provider: SUN

Your keystore contains 1 entry

localhost, Apr 9, 2011, trustedCertEntry,
Certificate fingerprint (MD5): A0:A4:4C:C9:9E:84:B2:6F:9E:63:9F:9E:D2:29:DE:E0

C:\opt\java\jdk1.6.0_24\jre\lib\security>_
```



My locale was set to Brazil and language to Brazilian Portuguese and for some reason after importing the certificate I faced issues when trying to list it. Then I changed by region settings to United States and American English, imported the certificate again and it worked. 🤔 Tricky no?

Now you have to tell the JRE that it has a new place to look for certificates (your generated keystore). Open %JENKINS\_HOME%\jenkins.xml and look for the arguments tag (<arguments>). Add the following properties into it.

```
-Djavax.net.ssl.keyStoreType=jks
-Djavax.net.ssl.keyStore=C:\opt\java\jdk1.6.0_24\jre\lib\security\jssecacerts
-Djavax.net.ssl.keyStorePassword=kinoshita
-Djavax.net.ssl.trustStore=C:\opt\java\jdk1.6.0_24\jre\lib\security\jssecacerts
-Djavax.net.ssl.trustStorePassword=kinoshita
```

It will look like follows after you are done.

```

36 <!--
37   if you'd like to run Hudson with a specific version of Java, specify a full path to java.exe.
38   The following value assumes that you have java in your PATH.
39   -->
40 <executable>java</executable>
41 <arguments>-Xrs -Xmx256m -Dhudson.lifecycle=hudson.lifecycle.WindowsServiceLifecycle
-Djavax.net.ssl.keyStoreType=jks
-Djavax.net.ssl.keyStore=C:\opt\java\jdk1.6.0_24\jre\lib\security\jssecacerts
-Djavax.net.ssl.keyStorePassword=kinoshita
-Djavax.net.ssl.trustStore=C:\opt\java\jdk1.6.0_24\jre\lib\security\jssecacerts
-Djavax.net.ssl.trustStorePassword=kinoshita -jar "%BASE%\jenkins.war" --httpPort=3755</arguments>
42 <!--
43   interactive flag causes the empty black Java window to be displayed.
44   I'm still debugging this.
45 <interactive />
46 -->
47 <logmode>rotate</logmode>

```

Now just start jenkins and execute your build and start using the plug-in with SSL.

```

Started by user anonymous
[testng] $ cmd /c call C:\WINDOWS\TEMP\hudson3808706380864786760.bat

C:\dev\java\qa_workspace\testng>mvn -f pom-test.xml clean
[INFO] Scanning for projects...
[INFO] -----
[INFO] Building TestNG
[INFO]   task-segment: [clean]
[INFO] -----
[INFO] [clean:clean {execution: default-clean}]
[INFO] -----
[INFO] BUILD SUCCESSFUL
[INFO] -----
[INFO] Total time: < 1 second
[INFO] Finished at: Sat Apr 09 12:54:08 BRT 2011
[INFO] Final Memory: 4M/15M
[INFO] -----
Preparing TestLink client API.
Using TestLink URL: https://localhost/testlink-1.9.2/lib/api/xmlrpc.php.

Retrieving TestLink details about test project, test plan and build.
Using TestLink Test Project: [Sample project], ID: [1].
Using TestLink Test Plan: [Sample plan], ID: [10].
Using TestLink Build: [Build 1.0], ID: [1].

Retrieving list of automated test cases from TestLink.
Found [1] TestLink Automated Test Cases.

Found TestLink test case: TestCase [id=17, name=Guice Test, testSuiteId=null, testProjectId=null, authorLogin=null, summary=, steps=[], precor
testImportance=null, executionType=2, order=null, internalId=null, checkDuplicatedName=null, actionOnDuplicatedName=null, versionId=18, versio
parentId=null, customFields=[], executionStatus=p].
Retrieving list of custom fields for test case.
Retrieving custom field SampleCustomField.
Custom field SampleCustomField value: test.guice.GuiceTest.

```

Depending on your environment settings these instructions may vary a little. But the basic idea is add your server certificate into the list of trusted certificates of your JRE. Hope it helps :-).



There is another way to use the plug-in with SSL, it would require a small modification in the code that would accept any SSL connection, not minding about certificates, as pointed in <http://ws.apache.org/xmlrpc/ssl.html>. However it would be a risky in productive environments.