VBooking

Enable HTTPS traffic between Tomcat and the LogonService on WAS

This document will detail the steps to enable secure HTTP connections in the VBooking application.

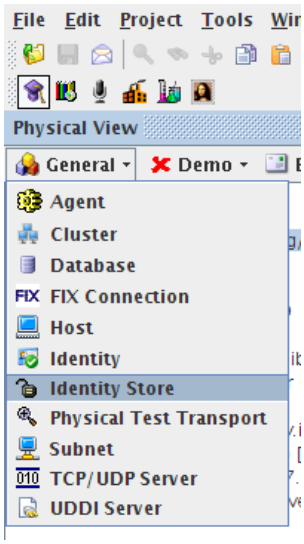
Overview

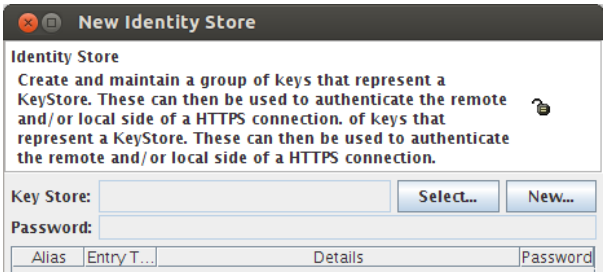
Assumption : Tomcat is accessing the webservice via the GreenHat HTTP/S proxy. (If Tomcat is accessing the service directly, you will have to do the optional step below to get the certificate from the WAS server)

- Use Rational Integration Tester to create a certificate truststore that Tomcat can use to 'trust' the called webservice.
- Configure Tomcat to use the truststore
- Configure the vbooking_booking application on Tomcat to access the WAS webservice using HTTPS
- (OPTIONAL) Setup a custom certificate on WAS to authenticate the webservice. Add the certificate to the truststore that Tomcat uses.

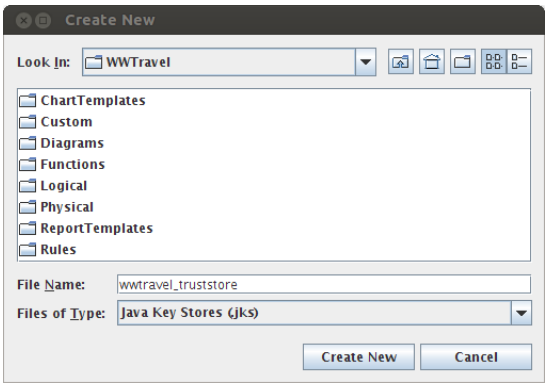
1. Create a trust store

Start Rational Integration Tester. In the Physical View of Architecture School, start the new wizard for General > Identity Store.

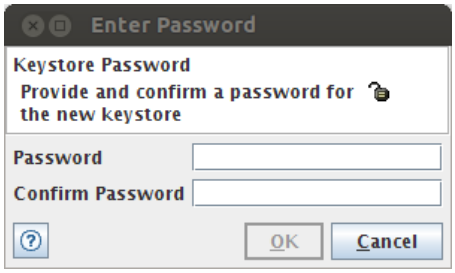




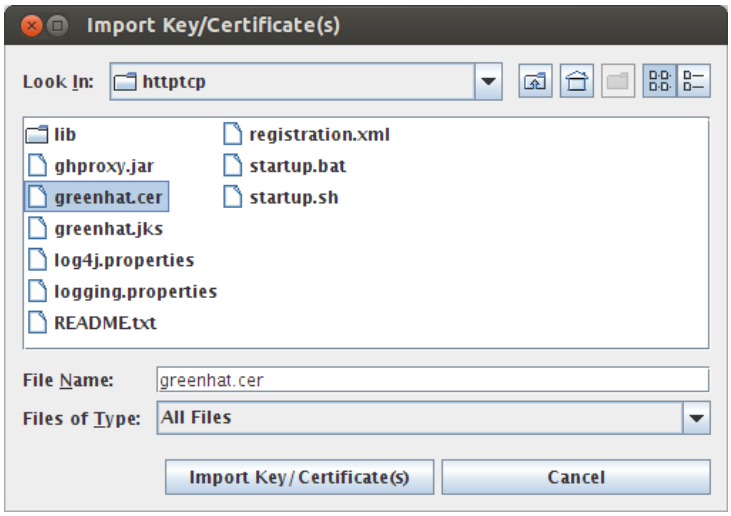
Select New on the top line, and give the store a filename and location in the dialog that appears, e.g. “vbooking_truststore”. Select “Create New”.



Enter a password for the keystore. (Remember this :-))

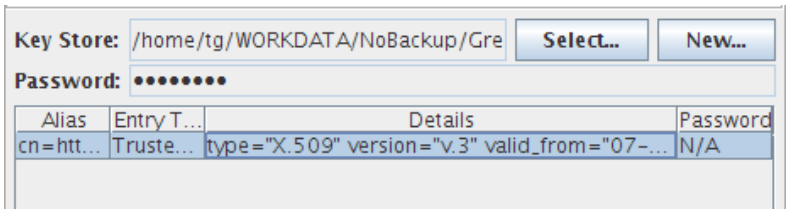


Select Import Key/Certificates and navigate to the Rational Integration Tester Platform Pack install directory to find the GreenHat proxy certificate. The certificate can be found at <Platform Pack Install Directory>/httptcp/greenhat.cer.



Select Import and the certificate should be imported and visible within the Identity Store. Press OK to finish editing the store.

Make a note of the location of the truststore location (it's a '.jks' file) as you will need it for the next step.



2. Configure Tomcat to use the truststore

Copy the truststore to the Tomcat conf/ directory.

Add the following line of startup options to Tomcat, replacing the path and password appropriate for your system (edit <Tomcat Install Dir>/bin/catalina.bat)

set JAVA_OPTS=%JAVA_OPTS% -Djavax.net.ssl.trustStore="**C:\Fully\Qualified\Path\To\Truststore.jks**" -Djavax.net.ssl.trustStorePassword="**Identity_Store_Password**" -Djavax.net.ssl.trustStoreType=jks

Also ensure that the https proxy settings have been added to Tomcat.

set JAVA_OPTS=%JAVA_OPTS% -Dhttps.proxyHost=localhost -Dhttps.proxyPort=3129

Restart Tomcat (or wait until after step 3. to restart it).

3. Configure the vbooking-booking app to use the https connection

Edit the web.xml file for the vbooking-booking app on Tomcat.

Path will be */webapps/vbooking_booking/WEB-INF/web.xml

Find the section for the Logon service which looks like this :

<context-param>

<param-name>**logonWSURL**</param-name>

<param

value>http://localhost:9080/com.vbooking.mybooking/services/**LogonServiceImpl**</param-value>

/context-param>

Edit it to use the WAS secure port and https protocol :

<context-param>

<param-name>logonWSURL</param-name>

<param

value>http**s**://localhost:**9443**/com.vbooking.mybooking/services/LogonServiceImpl</param-value>

/context-param>

Save, and restart the Tomcat server.

Usage

Because we have just changed the URL that the vbooking_booking application uses to access the Logon webservice, we can enable Tomcat<>WAS HTTPS communication, just by using a secure URL.

From the VBooking Booking main page, select the “Amend my flight booking” button.

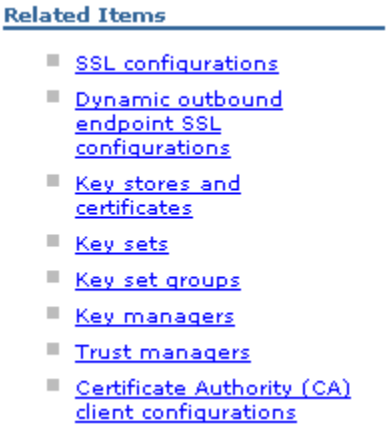
This accesses the LogonService when the logon page is loaded, and if you get a page returned with username and password boxes, then the service has been invoked.

(OPTIONAL) Create a certificate to authenticate directly between Tomcat and WAS (without the GreenHat proxy in-between)

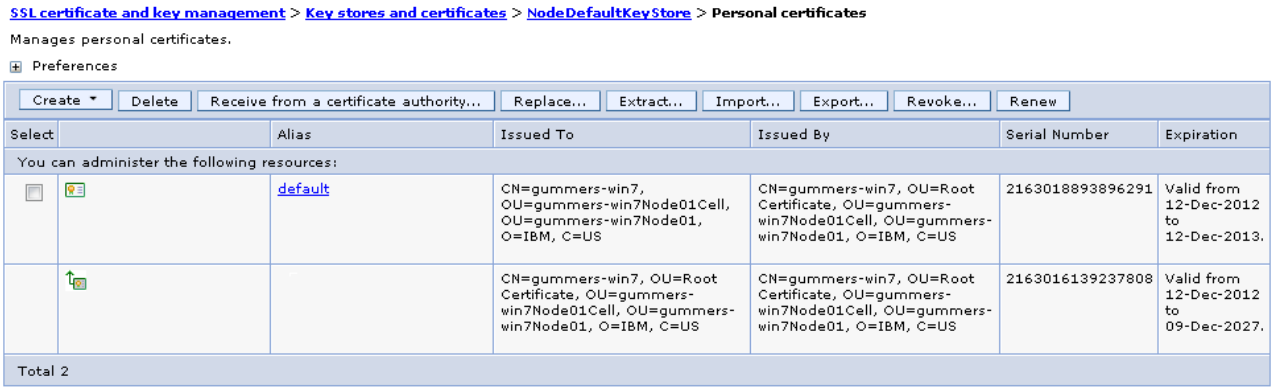
Logon to the Administrative console of the WAS instance

Expand the 'Security' section and select 'SSL certificate and key management'

Select 'Key stores and certificates' on the right hand side

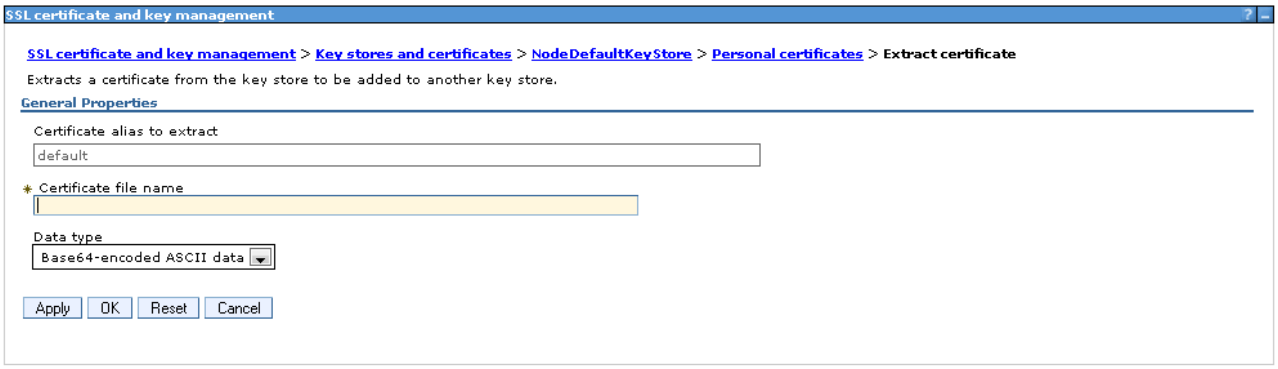


Select the 'NodeDefaultKeyStore' entry then 'Personal Certificates' in the Additional Properties section.



You can either just export the default certificate created when you created this WAS profile, or you can create a new Self-signed Certificate using the Create wizard at the top of the table.

Tick the checkbox of the certificate you want to export and select the 'Extract' button. ('Export' button would export the certificate in a key store already, but it is simpler to import the certificate to our existing store using RIT)



Enter a name, and keep the default Base64/ASCII data encoding. Press OK to export the certificate to file.

The certificate will be exported to the WAS */profiles/<ProfileName>/etc/ directory.

Copy it to a place you can access it with RIT to import to your Tomcat trust store.

> NB : If you created your own certificate, you may need to configure WAS to use it. From the 'SSL certificate and key management page', select 'SSL configurations' and then the 'NodeDefaultSSLSettings' entry. The 'Default server certificate alias' can be set here to your custom certificate.

In RIT, as for Step 1., use the Identity Store you created to import the WAS certificate (to join the GreenHat proxy certificate) into the truststore. Then copy the trust store to Tomcat and restart Tomcat. Tomcat can now access the HTTPS service directly when the proxy is not enabled.