Camel http4 component HTTPS

Issue: javax.net.ssl.SSLHandshakeException: sun.security.validator.ValidatorException: PKIX path building failed:

sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested target.

<http://magicmonster.com/kb/prg/java/ssl/pkix_path_building_failed.html>

1. Set java system parameter

(1) <http://blog.csdn.net/liuxiao723846/article/details/52695549>

keytool --import -alias devcerts -file ./sbybz2070.sby.ibm.com.cer -keystore dev.keystore

keytool --import -alias esbcerts -file ./sbybz2070.sby.ibm.com.cer -keystore certs.ks

System.setProperty("javax.net.ssl.trustStorePassword", "Passw0rd@");

System.setProperty("javax.net.ssl.trustStore", "src/test/resources/keystore/dev.keystore");

After test:

System.clearProperty("javax.net.ssl.trustStorePassword");

System.clearProperty("javax.net.ssl.trustStore");

1. <https://stackoverflow.com/questions/5706166/apache-camel-http-and-ssl>

**Setting up SSL for HTTP Client**

**Using the JSSE Configuration Utility**

[**http://camel.apache.org/http4.html#HTTP4-UsingtheJSSEConfigurationUtility**](http://camel.apache.org/http4.html#HTTP4-UsingtheJSSEConfigurationUtility)

The following code does actually work. The official camel web site can not work.

KeyStoreParameters ksp = new KeyStoreParameters();

ksp.setResource("src/test/resources/certs.ks");

ksp.setPassword("changeit");

TrustManagersParameters trustp = new TrustManagersParameters();

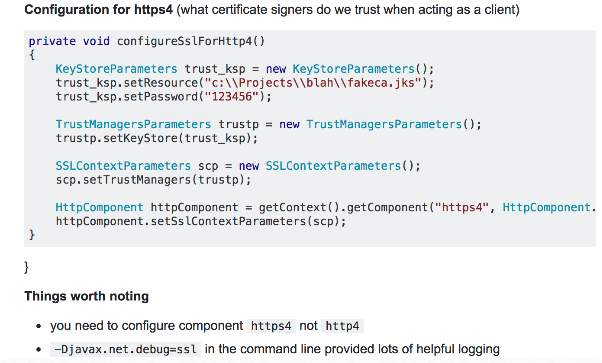
trustp.setKeyStore(ksp);

SSLContextParameters scp = new SSLContextParameters();

scp.setTrustManagers(trustp);

HttpComponent httpComponent = camelContext.getComponent("https4", HttpComponent.class);

httpComponent.setSslContextParameters(scp);



1. 系统JVM 导入证书到keystore

Mac上Java路径：

/Library/Java/JavaVirtualMachines/jdk1.8.0\_121.jdk/Contents/Home/jre/lib/security

<https://stackoverflow.com/questions/21076179/pkix-path-building-failed-and-unable-to-find-valid-certification-path-to-requ>



1. Configuring Apache HTTP Client Directly

<https://egkatzioura.com/2015/06/25/apache-camel-ssl-on-http4/>

SSLHttpClientConfigurer.java

*package org.eventfully.wmbtesting;*

*import org.apache.camel.component.http4.HttpClientConfigurer;*

*import org.apache.http.config.Registry;*

*import org.apache.http.config.RegistryBuilder;*

*import org.apache.http.conn.HttpClientConnectionManager;*

*import org.apache.http.conn.scheme.Scheme;*

*import org.apache.http.conn.scheme.SchemeRegistry;*

*import org.apache.http.conn.socket.ConnectionSocketFactory;*

*import org.apache.http.conn.ssl.SSLSocketFactory;*

*import org.apache.http.impl.client.HttpClientBuilder;*

*import org.apache.http.impl.conn.BasicHttpClientConnectionManager;*

*import javax.net.ssl.KeyManagerFactory;*

*import javax.net.ssl.SSLContext;*

*import javax.net.ssl.TrustManagerFactory;*

*import java.io.File;*

*import java.io.FileInputStream;*

*import java.io.FileNotFoundException;*

*import java.io.IOException;*

*import java.security.\*;*

*import java.security.cert.CertificateException;*

*/\*\**

*\* Created by qianqian on 13/09/2017.*

*\*/*

*public class SSLHttpClientConfigurer implements HttpClientConfigurer {*

*protected static final String KEYSTORE\_PATH = "src/test/resources/certs.ks";*

*protected static final File KEYSTORE = new File(KEYSTORE\_PATH);*

*protected static final String SECURE\_SOCKET\_PROTOCOL = "TLS";*

*protected static final String PASSWORD = "changeit";*

*@Override*

*public void configureHttpClient(HttpClientBuilder clientBuilder) {*

*try {*

*KeyStore trustStore = KeyStore.getInstance(KeyStore.getDefaultType());*

*trustStore.load(new FileInputStream(KEYSTORE), PASSWORD.toCharArray());*

*TrustManagerFactory trustManagerFactory = TrustManagerFactory.getInstance(KeyManagerFactory.getDefaultAlgorithm());*

*trustManagerFactory.init(trustStore);*

*SSLContext sslcontext = SSLContext.getInstance(SECURE\_SOCKET\_PROTOCOL);*

*sslcontext.init(null, trustManagerFactory.getTrustManagers(), null);*

*SSLSocketFactory factory = new SSLSocketFactory(sslcontext, SSLSocketFactory.ALLOW\_ALL\_HOSTNAME\_VERIFIER);*

*SchemeRegistry schemeReg = new SchemeRegistry();*

*final Scheme scheme = new Scheme("https4", 7043, factory);*

*schemeReg.register(scheme);*

*clientBuilder.setSSLContext(sslcontext);*

*clientBuilder.setSSLSocketFactory(factory);*

*Registry<ConnectionSocketFactory> registry = RegistryBuilder.<ConnectionSocketFactory>create()*

*.register("https", factory)*

*.build();*

*HttpClientConnectionManager connectionManager = new BasicHttpClientConnectionManager(registry);*

*clientBuilder.setConnectionManager(connectionManager);*

*} catch (CertificateException e) {*

*e.printStackTrace();*

*} catch (NoSuchAlgorithmException e) {*

*e.printStackTrace();*

*} catch (KeyStoreException e) {*

*e.printStackTrace();*

*} catch (KeyManagementException e) {*

*e.printStackTrace();*

*} catch (FileNotFoundException e) {*

*e.printStackTrace();*

*} catch (IOException e) {*

*e.printStackTrace();*

*}*

*}*

*}*

WmqConfig.java

@Bean(name = "sslHttpClientConfigurer")

public SSLHttpClientConfigurer sslHttpClientConfigurer() {

SSLHttpClientConfigurer sslHttpClientConfigurer = new SSLHttpClientConfigurer();

// KeyStore keystore = ...;

// KeyStore truststore = ...;

//

// SchemeRegistry registry = new SchemeRegistry();

// registry.register(new Scheme("https", 443, new SSLSocketFactory(keystore, "mypassword", truststore)));

return sslHttpClientConfigurer;

}

TestCase: HttpJsonRequestTest.java

Map<String, Object> headers = new HashMap<>();

headers.put(Exchange.HTTP\_METHOD, "GET");

String reply = producer.requestBodyAndHeaders("https4:sbybz2070.sby.ibm.com:7084/jsonrequest\*?httpClientConfigurer=sslHttpClientConfigurer", "", headers, String.class);

assertNotNull(reply);