Crides a serveis web segurs SSL
Mitjançant el Framework Canigó

A qui va dirigit

Aquest how-to va dirigit a tots aquells que hagin de desenvolupar una aplicació Canigó que:
• accedeixi a webservices segurs

Versió de Canigó

Els passos descrits en aquest document són d'aplicació a les versions de Canigó a partir de la 2.3.12

Introducció

Les aplicacions que volen accedir a serveis web sergurs, tenen el dubte d'on s'han d'indicar el certificats si al servidor d'aplicacions, als cacerts de la JVM, ... Des del Centre de Suport Canigó recomanem que els certificats estiguin a la pròpia aplicació per a no interferir a d'altres aplicacions desplegades sobre el mateix servidor d'aplicacions.

Aquest how-to presenta de quina manera han d'accedir les aplicacions a webservices segurs utilitzant Canigó. Per això presenta dues alternatives, per a aplicacions que cridin els webservices mitjançant **Axis** o **Xfire**.

D'aquesta manera volem evitar problemes que es troben les aplicacions al realitzar canvis d'entorns i unificar la manera d'accedir via SSL a serveis web.

Crides a Serveis Web mitjançant AXIS

Per tal de facilitar la gestió de les keyStores i trustStores, el servei de webservices de Canigó incorpora una classe abstracta **CanigoCustomSSLSocketFactory**.

L'aplicació s'haurà de crear una classe que extengui de CanigoCustomSSLSocketFactory, un exemple d'aquesta classe és:

```
package net.gencat.ctti.canigo.services.webservices.utils;

import java.util.Hashtable;

public class CanigoCustomAxisSSLSocketFactory extends CanigoCustomSSLSocketFactory {

    public CanigoCustomAxisSSLSocketFactory(Hashtable attributes) {
                super(attributes);
                this.RESOURCE_PATH_TO_KEYSTORE = "webservices/keys/CustomKeystore.jks";
                this.MY_KEYSTORE_PASSWORD = "customKeystorePassword";
                this.KEYSTORE_TYPE = null;
                this.RESOURCE_PATH_TO_TRUSTSTORE = null;
                this.MY_TRUSTSTORE_PASSWORD = null;
                this.TRUSTSTORE_TYPE = null;
        }
}
```

En aquesta classe s'informarà el path, password i tipus de keyStore i trustStore. Per a aquest exemple, hem generat una java keystore a partir del certificat mitjançant la keytool:

```
keytool -import -noprompt -trustcacerts -alias CustomKeystoreSifecat -file
certificat.crt -keystore CustomKeystore.jks -storepass customKeystorePassword
```

Un cop creada la classe custom per a informar on es troben els certificats, cal informar que les crides a webservices Axis utilitzin aquesta classe custom. Això es realitza de la següent manera:

```
AxisProperties.setProperty("axis.socketSecureFactory","net.gencat.ctti.canigo.service
s.webservices.utils.CanigoCustomAxisSSLSocketFactory");
```

Crides a serveis web segurs SSL
Mitjançant el Framework Canigó

Crides a Serveis Web mitjançant XFIRE

Per tal de facilitar la gestió de les keyStores i trustStores, el servei de webservices de Canigó incorpora una classe **AuthSSLProtocolSocketFactory**.

L'aplicació només haurà de crear una instància d'aquesta classe indicant la keystore i truststore i registrant que les crides que es realitzin pel protocol https s'interceptin per aquesta classe.
A continuació s'inclou un exemple de la crida:

```
URL keyStore =
Thread.currentThread().getContextClassLoader().getResource("webservices/keys/certific
at.p12");
URL trustStore =
Thread.currentThread().getContextClassLoader().getResource("webservices/keys/truststo
re.jks");

AuthSSLProtocolSocketFactory protocolSocketFactory = new
AuthSSLProtocolSocketFactory(keyStore, "pass", "PKCS12", trustStore, "pass", null);

Protocol authhttps = new Protocol("https", (ProtocolSocketFactory)
protocolSocketFactory, 443);
Protocol.registerProtocol("https", authhttps);
```

Un cop realitzat això, la crida al webservice es realitzarà utilitzant les claus indicades.

Per a utilitzar webservices amb Xfire a serveis centrals és necessari desplegar l'aplicació en un .EAR amb el següent contingut dintre del weblogic-application.xml

```
<?xml version="1.0" encoding="UTF-8"?>
  <weblogic-application xmlns="http://www.bea.com/ns/weblogic/90">
    <application-param>
      <param-name>webapp.encoding.default</param-name>
      <param-value>UTF-8</param-value>
    </application-param>
    <prefer-application-packages>
      <package-name>javax.jws.*</package-name>
      <package-name>antlr.*</package-name>
      <package-name>org.codehaus.xfire.*</package-name>
      <package-name>org.apache.axis2.*</package-name>
      <package-name>javax.xml.*</package-name>
    </prefer-application-packages>
</weblogic-application>
```