Secure Socket Layer în *jetty*

Transmisia utilizând *Secure Socket Layer* - SSL (mai nou *Transport Layer Security* - TLS) înseamnă criptarea datelor care circulă între client şi server.

Realizarea presupune

1. Generarea unui certificat de securitate cu utilitarul `keytool` din distribuţia Java, de exemplu

```
keytool -genkey -alias jetty -keyalg RSA
  -keystore {cale}/keystore
  -dname "cn=SE, ou=cs, o=unitbv, l=brasov, c=RO"
  -keypass 1q2w3e -storepass 1q2w3e
```

   Se definesc două parole

   - `keypass` parola certificatului de securitate;
   - `storepass` parola de protecţie a locaţiei certificatului de securitate.

   Parametrul *cale* desemnează calea către catalogul unde în care se crează fişierul *keystore* - certificatul de securitate. Certificatul de securitate se mută în catalogul `JETTY_HOME\etc`.

2. Criptarea parolelor

```
java -cp %JETTY_HOME%\lib\jetty-util-*.jar
  org.eclipse.jetty.util.security.Password <parola>
```

   Pentru exemplul certificatului generat mai sus, pentru *parola=1q2w3e* rezultatul este *OBF:1irv1lml1mii1mmc1lj51iur*

3. Completarea fişierului %JETTY_HOME%/start.ini cu secvenţa

```
# Module: https
--module=https

# Module: ssl
--module=ssl
jetty.ssl.host=0.0.0.0
jetty.sslContext.securePort=8443
jetty.sslContext.keyStorePath=etc/keystore
jetty.sslContext.trustStorePath=etc/keystore
jetty.sslContext.keyStorePassword=OBF:1irv1lml1mii1mmc1lj51iur
jetty.sslContext.trustStorePassword=OBF:1irv1lml1mii1mmc1lj51iur
jetty.sslContext.keyManagerPassword=OBF:1irv1lml1mii1mmc1lj51iur
jetty.sslContext.trustStoreType=JKS
```

4. După lansarea serverului Web apelarea poate fi

```
https://localhost:8443
http://localhost:8080
```