Fediz Jetty

Jetty Plugin

Apache CXF Fediz ships plugins for Jetty 8 and 9 instances. Previous versions of Fediz shipped plugins for Jetty 7. From release 1.4.5, the Jetty 8 and 9 plugins support both WS-Federation and SAML SSO.

This page describes how to enable Federation for a Jetty 7/8 instance hosting Relying Party (RP) applications. This configuration is not for a separate Tomcat instance hosting the Fediz IDP and IDP STS WARs, or hosts for third-party applications that use Fediz STS-generated SAML assertions for authentication. After this configuration is done, the Jetty-RP instance will validate the incoming SignInResponse created by the IDP server.

Prior to doing this configuration, make sure you've first deployed the Fediz IDP and STS on the Tomcat IDP instance as discussed here, and can view the STS WSDL at the URL given on that page. That page also provides some tips for running multiple Tomcat instances on your machine.

Installation

You can either build the Fediz plugin on your own or download the package here. If you have built the plugin on your own you'll find the required libraries in `plugins/jetty${version}/target/...zip-with-dependencies.zip`

1. Create sub-directory `fediz` in `${jetty.home}/lib/fediz`
2. Update start.ini in ${jetty.home}/start.ini by adding `fediz` to the OPTIONS

```
OPTIONS=Server,fediz
```

3. Deploy the libraries to the directory created in (1)

Configuration

HTTPS configuration

It's recommended to set up a dedicated (separate) Jetty instance for the Relying Party. The Fediz RP web applications use the following TCP ports:

- HTTP port: 8080
- HTTPS port: 8443 (where IDP and STS are accessed)

These are the default ports for a standard Jetty installation.

The Relying Party must be accessed over HTTPS to protect the security tokens issued by the IDP.

The Jetty HTTP(s) configuration is done in etc/jetty-ssl.xml.

The configuration is described in detail here

This page also describes how to create certificates. Sample Jetty keystores (not for production use, but useful for demoing Fediz and running the sample applications) are provided in the examples/samplekeys folder of the Fediz distribution. Note the Jetty keystore here is different from the one used to configure the Tomcat-IDP instance.

To establish trust, there are significant keystore/truststore requirements between the Servlet Container instances and the various web applications (IDP, STS, Relying party applications, third party web services, etc.) See this page for more details, it lists the trust requirements as well as sample scripts for creating your own (self-signed) keys.

Warning: All sample keystores provided with Fediz (including in the WAR files for its services and examples) are for development/prototyping use only. They'll need to be replaced for production use, at a minimum with your own self-signed keys but strongly recommended to use third-party signed keys.

If you are currently just trying to run the Fediz samples, the configuration above is all you need (the below configuration is already provided within the samples) so you can return now to the samples' READMEs for the next steps in running them.

Fediz Plugin configuration for Your Web Application

The Fediz related configuration is done in a Servlet Container independent configuration file which is described here.

The Fediz plugin requires configuring the FederationAuthenticator like any other authenticator in Jetty. Detailed information about the Authenticators and SecurityHandler is available here.

The Fediz configuration file allows to configure all servlet contexts in one file or choosing one file per Servlet Context.

You can configure the context in context configuration file located in <jetty.home>/contexts.

fedizhelloworld.xml

Hint: file name must be equal to war file name

```
  <Get name="securityHandler">
    <Set name="loginService">
      <New class="org.apache.cxf.fediz.jetty.FederationLoginService">
        <Set name="name">WSFED</Set>
      </New>
    </Set>
    <Set name="authenticator">
      <New class="org.apache.cxf.fediz.jetty.FederationAuthenticator">
        <Set name="configFile"><SystemProperty name="jetty.home" default="."/>
/etc/fediz_config.xml</Set>
      </New>
    </Set>
  </Get>
```

The Fediz configuration file is a Servlet container independent configuration file and described here

Web Application deployment

Deploy your Web Application to your Jetty installation (<jetty.home>/webapps). If you're running the Fediz examples, their README files will have instructions on how to do this.

Federation Metadata document

The Jetty Fediz plugin supports publishing the WS-Federation Metadata document which is described here.