Fediz Configuration

FEDIZ PLUGIN CONFIGURATION

This page describes the Fediz configuration file referenced by the security interceptor of the Servlet Container (eg. authenticator in Tomcat/Jetty).

The Fediz configuration information is used to publish the WS-Federation or SAML SSO Metadata document, which is described here.

WS-Federation Example

The following example shows the minimum configuration for Fediz.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<FedizConfig>
    <contextConfig name="/fedizhelloworld">
        <audienceUris>
            <audienceItem>https://localhost:8443/fedizhelloworld<
/audienceItem>
        </audienceUris>
        <certificateStores>
            <trustManager>
                <keyStore file="conf/stsstore.jks" password="stsspass" type="
JKS" />
            </trustManager>
        </certificateStores>
        <trustedIssuers>
            <issuer certificateValidation="PeerTrust" />
        </trustedIssuers>
        <protocol xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:
type="federationProtocolType" version="1.2">
            <issuer>https://localhost:9443/fediz-idp/federation/</issuer>
        </protocol>
    </contextConfig>
</FedizConfig>
```

The IDP issues a SAML token which must be validated by the plugin. The validation requires the certificate store of the Certificate Authority(ies) of the certificate which signed the SAML token. This is defined in `certificateStore`. The signing certificate itself is not required because `certificateValidation` is set to `ChainTrust`. The audience URI is validated against the audience restriction in the SAML token.

The protocol element declares that the WS-Federation protocol is being used. If SAML SSO was being used instead, then the "xsi:type" value would be "samlProtocolType". The configuration items outside of the "protocol" section are independent of whether WS-Federation or SAML SSO are being used.

The issuer element shows the URL to which authenticated requests will be redirected with a SignIn request.

Protocol-independent configuration reference

The configuration schema can be seen here.

XML element	Use	Description
audienceUris	Optional	The values of the list of audience URIs are verified against the element `AudienceRestriction` in the SAML token. If a SAML token contains a audience restriction which is not listed within this collection, the token will be refused.
certificateStores	Required	The list of keystores (JKS, PEM) includes at least the certificate of the Certificate Authorities (CA) which signed the certificate which is used to sign the SAML token. If the file location is not fully qualified it needs to be relative to the Container home directory
tokenExpirationValidation	Optional	Decision whether the token validation (e.g. lifetime) shall be performed on every

		request (true) or only once at initial authentication (false). The default is "false".
addAuthenticatedRole	Optional	Whether to add the "Authenticated" role to the list of roles associated with the "authenticated" user. This could be useful if you don't care about authorizing the user, only about authentication. A role is required to activate authentication, and it may be problematic to list all relevant roles in web.xml. Note that if the user has no roles, then the "Authenticated" role is added automatically. The default is "false".
maximumClockSkew	Optional	Maximum allowable time difference between the system clocks of the IDP and RP. Default 5 seconds.
tokenReplayCache	Optional	The ReplayCache implementation to use to cache tokens. The default is an implementation based on EHCache.
signingKey	Optional	If configured, the published (WS-Federation or SAML

		SSO) Metadata document is signed by this key. Otherwise, not signed.
tokenDecryptionKey	Optional	A Keystore used to decrypt an encrypted token.
trustedIssuers	Required	There are two ways to configure a trusted issuer (IDP). Either you configure the subject name and the CA(s) who signed the certificate of the IDP (`certificateValidation=ChainTrust`) or you configure the certificate of the IDP and the CA (s) who signed it (`certificateValidation=PeerTrust`)
protocol	Required	A protocolType instance that defines the SSO protocol that is supported. Currently supported protocols are "federationProtocolType" and "samlProtocolType". See below for protocol-specific configuration items.
logoutURL	Optional	User defined logout URL to trigger federated logout process.
logoutRedirectTo	Optional	

		URL to landing-page after successful logout.
logoutRedirectToConstraint	Optional	A regular expression constraint on the 'wreply' parameter, which is used to obtain the URL to navigate to after successful logout. Only applies to WS-Federation protocol.
requestStateValidation	Optional	Decision on whether the received state must match the state saved in the context. Set it to "false" to support IdP initiated SSO. Only supported for CXF + Spring plugins thus far. The default is "true".

WS-Federation protocol configuration reference

XML element	Use	Metadata	Description
applicationServiceURL	Optional	entityID	Used to set the "entityID" for the Metadata. If not specified, the context path of the application is used instead.
roleDelimiter	Optional	NA	There are different ways to encode multi value attributes in SAML:

			• Single attribute with multiple values • Several attributes with the same name but only one value • Single attribute with single value. Roles are delimited by `roleDelimiter`
roleURI	Optional	NA	Defines the attribute name of the SAML token which contains the roles. Required for Role Based Access Control. Typically this is configured with the value "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/role".
claimTypesRequested	Optional	ClaimTypesRequested (WS-Fed) / RequestedAttribute (SAML SSO)	The claims required by the Relying Party are listed here. Claims can be optional. If a mandatory claim can't be provided

			by the IDP the issuance of the token should fail.
issuer	Required	NA	This URL defines the location of the IDP to whom unauthenticated requests are redirected.
realm	Optional	NA	Security realm of the Relying Party / Application. For WS-Federation, this value is part of the SignIn request as the `wt realm` parameter. For SAML SSO, it is used as the Issuer of the AuthnRequest. Default: URL including the Servlet Context
tokenValidators	Optional	NA	Custom Token validator classes can be configured here. The SAML Token validator is enabled by default. See example here.
metadataURI	Optional	NA	The URI where Metadata is served. The default is

			"FederationMetadata/2007-06/FederationMetadata.xml" for WS-Federation and "SAML/Metadata.xml" for SAML SSO.
reply	Optional	NA	The value to send to the IdP in the "wreply" parameter.
authenticationType	Optional	NA	The authentication type defines what kind of authentication is required. This information is provided in the SignInRequest to the IDP (parameter `wauth`). The WS-Federation standard defines a list of predefined URIs for wauth here.
homeRealm	Optional	NA	Indicates the Resource IDP the home realm of the requestor. This may be an URL or an identifier like urn: or uuid: and depends on the Resource IDP

			implementation. This value is part of the SignIn request as the `whr` parameter
freshness	Optional	NA	The desired "freshness" of the token from the IdP. This information is provided in the SignInRequest to the IdP (parameter `wfresh`)
request	Optional	NA	This value is part of the SignIn request as the wreq parameter. It can be used to specify a desired TokenType from the IdP.
signInQuery	Optional	NA	Additional queries to be appended to the sign-in URL.
signOutQuery	Optional	NA	Additional queries to be appended to the sign-out URL.

SAML SSO protocol configuration reference

XML element	Use	Metadata	Description
applicationServiceURL	Optional	entityID	Used to set the "entityID" for the

			Metadata. If not specified, the context path of the application is used instead.
roleDelimiter	Optional	NA	There are different ways to encode multi value attributes in SAML: • Single attribute with multiple values • Several attributes with the same name but only one value • Single attribute with single value. Roles are delimited by `roleDelimiter`
roleURI	Optional	NA	Defines the attribute name of the SAML token which contains the roles. Required for Role Based Access Control. Typically this is configured with the value "http://schemas.xmlsoap.

			org/ws/2005/05/identity/claims/role".
claimTypesRequested	Optional	ClaimTypesRequested (WS-Fed) / RequestedAttribute (SAML SSO)	The claims required by the Relying Party are listed here. Claims can be optional. If a mandatory claim can't be provided by the IDP the issuance of the token should fail.
issuer	Required	NA	This URL defines the location of the IDP to whom unauthenticated requests are redirected.
realm	Optional	NA	Security realm of the Relying Party / Application. For WS-Federation, this value is part of the SignIn request as the `wtrealm` parameter. For SAML SSO, it is used as the Issuer of the AuthnRequest. Default: URL including the Servlet Context
tokenValidators	Optional	NA	Custom Token validator classes

			can be configured here. The SAML Token validator is enabled by default. See example here.
metadataURI	Optional	NA	The URI where Metadata is served. The default is "FederationMetadata/2007-06/FederationMetadata.xml" for WS-Federation and "SAML/Metadata.xml" for SAML SSO.
reply	Optional	NA	The value for the AssertionConsumerService URL in the AuthnRequest
signRequest	Optional	NA	Whether to sign the AuthnRequest or not. The default is false.
authnRequestBuilder	Optional	NA	A SAMLPRequestBuilder instance used to build the AuthnRequest/LogoutRequest. The default is here.
disableDeflateEncoding	Optional	NA	Whether to disable deflate

			encoding or not. The default is "false".
doNotEnforceKnownIssuer	Optional	NA	Whether to not enforce that the issuer of the SAML Response is a known value. The default it false (meaning that it is enforced).
issuerLogoutURL	Optional	NA	The logout URL to redirect to. If not specified it falls back to the Issuer URL.
checkClientAddress	Optional	NA	Whether to check the client address against the subject confirmation data address. The default is true.

Attributes resolved at runtime

The following attributes can be either configured statically at deployment time or dynamically when the initial request is received:

- authenticationType
- homeRealm
- issuer
- realm
- logoutRedirectToConstraint
- request
- freshness
- signInQuery
- signOutQuery
- reply

These configuration elements allows for configuring a CallbackHandler which gets a Callback object where the appropriate value must be set. The CallbackHandler implementation has access to the HttpServletRequest. The XML attribute `type` must be set to `Class`.

For more information see Fediz Extensions.

Advanced WS-Federation example

The following example defines the required claims and configures a custom callback handler to define some configuration values at runtime.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<FedizConfig>
    <contextConfig name="/fedizhelloworld">
        <audienceUris>
            <audienceItem>https://localhost:8443/fedizhelloworld<
/audienceItem>
        </audienceUris>
        <certificateStores>
            <keyStore file="conf/stsstore.jks" password="stsspass" type="
JKS" />
        </certificateStores>
        <maximumClockSkew>10</maximumClockSkew>
        <trustedIssuers>
            <issuer certificateValidation="PeerTrust" />
        </trustedIssuers>
        <signingKey keyPassword="tompass">
            <keyStore file="tomcatKeystore.jks" password="tompass" type="
JKS" />
        </signingKey>
        <protocol xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:
type="federationProtocolType" version="1.2">
            <issuer>https://localhost:9443/fediz-idp/federation/</issuer>
            <roleDelimiter>,</roleDelimiter>
            <roleURI>http://schemas.xmlsoap.org/ws/2005/05/identity/claims
/role</roleURI>
            <claimTypesRequested>
                <claimType type="http://schemas.xmlsoap.org/ws/2005/05
/identity/claims/role" optional="true" />
            </claimTypesRequested>
            <authenticationType type="String" value="http://docs.oasis-open.
org/wsfed/authorization/200706/authntypes/smartcard" />
            <homeRealm type="Class" value="example.HomeRealmCallbackHandler"
/>
            <tokenValidators>
                <validator>org.apache.cxf.fediz.core.CustomValidator<
/validator>
            </tokenValidators>
        </protocol>
    </contextConfig>
</FedizConfig>
```