ProfileInterceptConfiguration

Current File(s): *conf/intercept/profile-intercept.xml*, *conf/relying-party.xml*

Format: Native Spring

Overview

An interceptor is a Spring Web Flow that can be run as a subflow at specific points during the processing of a request to the IdP to do work. It allows behavior to be customized without replacing core code or changing system files.

There are currently three different "interception" points in the profile flows:

- Inbound message processing
- Post-authentication during SSO profiles
- Outbound message processing

The inbound and outbound hooks are primarily used by the system to plug in security handling code and to allow for very specialized kinds of customization, and are not commonly used by deployers. In contrast, the post-authentication hook is a very fruitful injection point for supporting many useful features. Several predefined examples come with the software, and can be used to help develop your own custom behaviors.

General Configuration

Defining Interceptors

The *intercept/profile-intercept.xml* file is where all supported interceptor flows are (very minimally) described to the system.

The file contains a Spring list bean named **shibboleth.AvailableInterceptFlows***.* Descriptors are of a specific class, and generally there aren't additional settings needed (at least not that are defined here).

The `id` property of each descriptor is **not** arbitrary. It MUST be prefixed by "intercept/" and it corresponds to a web flow definition. The predefined beans correspond to built-in flows. Creating a new flow involves not only describing the flow in this list, but ensuring the `id` matches a flow definition created inside *flows/intercept/*. Specifically, creating the custom flow "intercept/foo" requires that the flow definition file be named *flows/intercept/foo/foo-flow.xml*.

The following interceptor flows are provided with the software:

- Attribute Release Consent
- Terms of Use Consent
- Context Checking
- Expiring Password 3.3
- Impersonate 3.4

Enabling Interceptors

The three interception points above correspond to three properties that can be specified on the profile configuration beans in the RelyingPartyConfiguration. Each property is a list of intercept flow IDs (excluding the "intercept/" prefix) to run.

All profile configurations include a pair of properties, `inboundInterceptorFlows` and `outboundInterceptorFlows`, for specifying inbound and outbound interceptors. The profile beans typically auto-declare the right inbound interceptor flow to run to provide the appropriate security checks; these interception points should generally be left to their default values.

Authentication profile configurations (e.g. CAS, SAML Browser SSO and ECP) include a `postAuthenticationFlows` property for specifying the ordered list of interceptors to run after most of the work of the system is done but before any outbound message/response has been generated. They run after the user has logged in and after any user attributes have been resolved and filtered; essentially all that's left is the production of a response, so this is an opportunity to affect the result that will be produced (or prevent one altogether).

Example enabling intercepts for the SAML SSO profiles

```
    <bean id="shibboleth.DefaultRelyingParty" parent="RelyingParty">
        <property name="profileConfigurations">
            <list>
                <bean parent="Shibboleth.SSO" p:postAuthenticationFlows="#{{'attribute-release', 'terms-of-use'}}" />
                <ref bean="SAML1.AttributeQuery" />
                <ref bean="SAML1.ArtifactResolution" />
                <bean parent="SAML2.SSO" p:postAuthenticationFlows="#{{'attribute-release', 'terms-of-use'}}" />
                <ref bean="SAML2.ECP" />
                <ref bean="SAML2.Logout" />
                <ref bean="SAML2.AttributeQuery" />
                <ref bean="SAML2.ArtifactResolution" />
                <ref bean="Liberty.SSOS" />
            </list>
        </property>
    </bean>
```

Developing Interceptor Flows

Refer to the ProfileHandling developer material for more technical details on developing interceptor flows.

Reference

Beans

Bean ID	Type	Function
shibboleth.AvailableInterceptFlows	List<ProfileInterceptorFlowDescriptor>	List of flow descriptors enumerating the interceptor flows available to the system
shibboleth.DefaultInterceptFlows	List<ProfileInterceptorFlowDescriptor>	List of built-in interceptor flows, into which user-defined flows are merged
shibboleth.InterceptFlow	ProfileInterceptorFlowDescriptor	Abstract parent bean for defining new flow descriptor beans

Notes

TBD