OSTwoUsrManJavaLibIntro

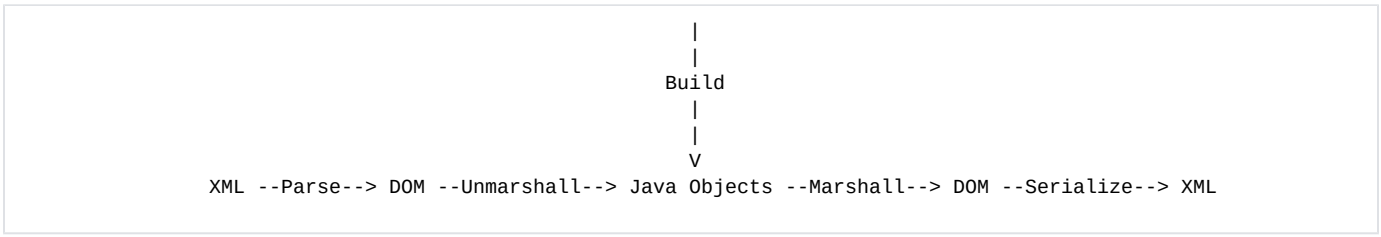
Introduction to the Library

The OpenSAML 2.0 library provides a mechanism for reading and writing SAML through Java objects. It supports SAML 1.0, 1.1, and 2.0. Version 2.0 is a complete rewrite of version 1.1 with a heavy focus on extensibility and usability. The library itself is not an implementation of SAML-enabled services such as an Identity or Service Provider or Liberty Alliance WSF services. Instead it is expected that such services could use this library in order to work with SAML messages.

At its core the library is schema type based; the code that works with AuthnStatement elements models the AuthnStatementType schema type defined by the SAML schema. This is analogous to how Java works; an element (Java object) is an instance of a schema type (Java class). This mapping means users of the library do not need to know much about XML Schema.

How it Works

The library is based on the following flow:



XML is **parsed** into its DOM (level 3) representation. The DOM goes through an **unmarshalling** process which produces the Java objects, known within the library as SAMLObjects or, more generically, XMLObjects. These objects are then **marshalled** into a DOM representation which can then be **serialized** out to XML. When creating a SAML document from scratch the SAMLObjects are **built** and then follow the marshalling/serialization process to get to XML. SAMLObjects may also be run through **validation** rules to verify they meet specific requirements.

The bold terms above represent the base functionality provided by OpenSAML.

Next refer to "Creating SAML Objects from an XML Source" for information on how to perform the parse/unmarshall step or "Creating SAML Objects from Scratch" for information on how to build SAMLObjects from scratch.

Configuring the Library

OpenSAML ships with a standard set of configuration files which should be sufficient for most use cases (the exception would be if you're using custom extensions to SAML messages). These files tell the library which classes to use for building, marshalling, and unmarshalling SAMLObjects. More information about these files can be found in the Configuration File section of the developer's guide but, for now, we'll just use the default files provided.

To bootstrap the library with these configuration files invoke the

```
org.opensaml.DefaultBootstrap#bootstrap()
```

method prior to using the library.

Default Object Provider

The default configuration file common-config.xml contains an entry that specifies a default object provider (group of building, marshaller, and unmarshaller). This provider will be used when no other provider can be located. In this file it's set to use the builder, marshaller, and unmarshaller for org. opensaml.xml.ElementProxy . This class provides a generic means for accessing data from an XML element but does not support every aspect of DOM elements (e.g. mixed content).

In most cases developers won't encounter this XMLObject. However, it's common that this object will be used as the container for AttributeValue elements. Since this particular element neither requires a schema type (so the object provider can't be looked up by it) nor has consistent content (so a provider can't be written to handle everything and then registered under the element name) the proxy gets used. Note that if there is a specified schema type and a provider is registered for it then that provider will be used.