

IdPDevExtDataCtr

Data Connector Extensions

This information is meant to be used in conjunction with the [Creating Custom IdP Extensions - General Overview](#) guide.

Classes, Schema Types, and Schema Files

- Class to extend:
`edu.internet2.middleware.shibboleth.common.attribute.resolver.provider.dataConnector.BaseDataConnector`
- Bean Definition Parser to extend:
`edu.internet2.middleware.shibboleth.common.config.attribute.resolver.dataConnector.BaseDataConnectorBeanDefinitionParser`
- Bean Factory to extend:
`edu.internet2.middleware.shibboleth.common.config.attribute.resolver.dataConnector.BaseDataConnectorFactoryBean`
- Location of schema file to import:
`/schema/shibboleth-2.0-attribute-resolver.xsd`
- Namespace containing schema type to extend:
`urn:mace:shibboleth:2.0:resolver`
- Complex type to extend:
`BaseDataConnectorType`

Example Extension

Extension Class

```
public class MyDataConnector extends BaseDataConnector {  
  
    private String lookupUrl; //This is an example custom field, yours may vary (if any)  
  
    public MyDataConnector(String url) {  
        this.lookupUrl = url;  
    }  
  
    public Map<String, BaseAttribute> resolve(ShibbolethResolutionContext resolutionContext)  
        throws AttributeResolutionException {  
        Map<String, BaseAttribute> result = new HashMap<String, BaseAttribute>();  
        String username = resolutionContext.getAttributeRequestContext().getPrincipalName();  
        // add BasicAttributes to the result here.  
        return result;  
    }  
}
```

Note that your extension class **must** return a non-null result, even if there are no attributes to return.

Extension Schema

Located in your JAR file as **schema/myConnectors.xsd**. (Note this name is not special in anyway, it must just match between the locations it is referenced)

```

<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="urn:example.org:shibboleth:2.0:resolver"
  xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:resolver="urn:mace:shibboleth:2.0:resolver"
  elementFormDefault="qualified">

  <import namespace="urn:mace:shibboleth:2.0:resolver"
    schemaLocation="classpath:/schema/shibboleth-2.0-attribute-resolver.xsd" />

  <complexType name="UserLookup">
    <annotation>
      <documentation>
        Description of your data connector.
      </documentation>
    </annotation>
    <complexContent>
      <extension base="resolver:BaseDataConnectorType">
        <attribute name="lookupUrl" type="string" use="required">
          <annotation>
            <documentation>
              This is an example of a custom attribute called "lookupUrl".
              Look in the shib-common.jar at schema/shibboleth-2.0-attribute-resolver-dc.xml
              for more examples of how to define custom attributes and elements.
            </documentation>
          </annotation>
        </attribute>
      </extension>
    </complexContent>
  </complexType>
</schema>

```

Bean Definition Parser

```

public class MyDataConnectorBeanDefinitionParser extends BaseDataConnectorBeanDefinitionParser {

  public static final QName SCHEMA_NAME = new QName(MyDataConnectorNamespaceHandler.NAMESPACE, "UserLookup");

  protected Class getBeanClass(Element element) {
    return MyDataConnectorFactoryBean.class;
  }

  protected void doParse(Element element, BeanDefinitionBuilder builder) {
    super.doParse(element, builder);
    String lookupUrl = pluginConfig.getAttributeNS(null, "lookupUrl");
    builder.addPropertyValue("lookupUrl", lookupUrl);
  }
}

```

Bean Factory

```

public class MyDataConnectorFactoryBean extends BaseDataConnectorFactoryBean {

    private String lookupUrl;

    public Class getObjectType() {
        return MyDataConnector.class;
    }

    protected Object createInstance() throws Exception {
        MyDataConnector connector = new MyDataConnector(getLookupUrl());
        populateDataConnector(connector);
        return connector;
    }

    public void setLookupUrl(String url) { this.lookupUrl = url; }
    public String getLookupUrl() { return this.lookupUrl; }

}

```

Namespace Handler

```

public class MyDataConnectorNamespaceHandler extends BaseSpringNamespaceHandler {

    public static String NAMESPACE = "urn:example.org:shibboleth:2.0:resolver";

    public void init() {
        registerBeanDefinitionParser(MyDataConnectorBeanDefinitionParser.SCHEMA_NAME,
            new MyDataConnectorBeanDefinitionParser());
    }

}

```

spring.schemas File

Located in your JAR file as **META-INF/spring.schemas**.

```

urn\:example.org\:shibboleth\:2.0\:resolver = schema/myConnectors.xsd

```

spring.handlers File

Located in your JAR file as **META-INF/spring.handlers**.

```

urn\:example.org\:shibboleth\:2.0\:resolver = com.example.shibboleth.MyDataConnectorNamespaceHandler

```

attribute-resolver.xml File

Now you can make use of your DataConnector via attribute-resolver.xml.

Declare your name space, add your XML schema file to the documents **schemaLocation** and add your DataConnector:

```

<ProfileHandlerGroup xmlns="urn:mace:shibboleth:2.0:idp:profile-handler"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:example="http://example.org/shibboleth/authn"
    xsi:schemaLocation="urn:mace:shibboleth:2.0:idp:profile-handler classpath:/schema
/shibboleth-2.0-idp-profile-handler.xsd
    urn:example.org:shibboleth:2.0:resolver classpath:/schema/myConnectors.xsd">
[.]
    <resolver:DataConnector id="userLookupAttributes" xsi:type="UserLookup" xmlns="urn:example.org:shibboleth:
2.0:resolver" lookupUrl="http://data.example.org"/>

```