

Grouper Shibboleth Integration

[Wiki Home](#)[Download Grouper](#)[Grouper Guides](#)[Community Contributions](#)[Developer Resources](#)[Grouper Website](#)

As of Grouper 2.1.0, the grouper-shib project (grouper-shib.jar) provides [Data Connector](#) extensions and [Attribute Definition](#) extensions for the Shibboleth Attribute Resolver. Previously as of version 1.5, the Grouper API distribution (grouper.jar) provided this functionality. [Read the Grouper Shib Integration documentation prior to Grouper 2.1](#) .

Please note: Using the Grouper-Shib connector is not the suggested way of integrating Grouper and Shib, unless you have very specific use cases. The primary use case for the Grouper-Shib connector is to support the [PSP](#) which uses the Shibboleth Attribute Resolver to assist with pushing groups to LDAP/AD. The more common way of integrating Grouper and Shibboleth it is to push the group information into either a LDAP server or a SQL database and then to consume the groups from there. This eliminates the need to also install the Grouper API and supporting jars into your IdP.



If you are interested in using Shibboleth as your Grouper log-in mechanism, then see this page on [Authentication to the Grouper UI](#)

- [Overview](#)
- [Group Data Connector](#)
- [GroupDataConnector - Attributes](#)
- [GroupDataConnector - Lists \(Memberships\)](#)
- [GroupDataConnector - Default List \(Members\)](#)
- [GroupDataConnector - List \(Membership\) Scope](#)
- [GroupDataConnector - Custom List \(Membership\)](#)
- [GroupDataConnector - Member Of List](#)
- [GroupDataConnector - Privileges](#)
- [Member Data Connector](#)
- [Member Data Connector - Attributes](#)
- [Member Data Connector - Lists](#)
- [Member Data Connector - Privileges](#)
- [Stem Data Connector](#)
- [Change Log Data Connector](#)
- [Filters](#)
- [Filter - GroupExactAttribute](#)
- [Filter - GroupInStem](#)
- [Filter - AND](#)
- [Filter - OR](#)
- [Filter - MINUS](#)
- [Filter - StemInStem](#)
- [Filter - StemNameExact](#)
- [Filter - ChangeLogAudit](#)
- [Filter - ChangeLogEntry](#)
- [Filter - ChangeLogExactAttribute](#)
- [Filter - ChangeLogAttributeAssignType](#)
- [Group Attribute Definition](#)
- [Member Attribute Definition](#)
- [Subject Attribute Definition](#)
- [See Also](#)

Overview

View Shib IdP and Grouper Data Connection in [the Grouper architectural diagram](#).

Source code is available [here](#).

Download from [Maven Central](#).

```
<dependency>
  <groupId>edu.internet2.middleware.grouper</groupId>
  <artifactId>grouper-shib</artifactId>
  <version>2.1.0</version>
</dependency>
```

Group Data Connector

The GroupDataConnector returns attributes which represent the Grouper Group whose name is the principal name of an attribute request.

GroupDataConnector - Attributes

The attributes returned for a group include built-in attributes such as *id*, *name*, *displayName*, *extension*, *displayExtension*, and *description*, as well as custom attributes and [attribute framework](#) attributes.

See the [Grouper Glossary](#) for more information on attributes.

The following example will return an attribute named "description" whose value is the description of a group :

```
<resolver:DataConnector id="GroupDataConnector" xsi:type="grouper:GroupDataConnector" />

<resolver:AttributeDefinition id="description" xsi:type="ad:Simple">
  <resolver:Dependency ref="GroupDataConnector" />
</resolver:AttributeDefinition>
```

GroupDataConnector - Lists (Memberships)

By default, no lists (memberships) are returned by the GroupDataConnector because they may be expensive to query. Lists which should be returned as attributes may be defined using the following naming convention :

```
<resolver:DataConnector id="GroupDataConnector" xsi:type="grouper:GroupDataConnector">
  <grouper:Attribute id="<members|group>[:<all|immediate|effective|composite>[:<list name>]]" />
</resolver:DataConnector>
```

GroupDataConnector - Default List (Members)

The following example will return an attribute named "member" whose values are the "name" of every member from the "jdbc" subject source of the default "members" list of a group :

```
<resolver:DataConnector id="GroupDataConnector" xsi:type="grouper:GroupDataConnector">
  <grouper:Attribute id="members" />
</resolver:DataConnector>

<resolver:AttributeDefinition id="member" xsi:type="grouper:Member" sourceAttributeID="members" >
  <resolver:Dependency ref="GroupDataConnector" />
  <grouper:Attribute id="name" source="jdbc" />
</resolver:AttributeDefinition>
```

GroupDataConnector - List (Membership) Scope

The following example will return an attribute named "immediateMembers" whose values are the "name" of every immediate member from the "jdbc" source of the default "members" list of a group :

```
<resolver:DataConnector id="GroupDataConnector" xsi:type="grouper:GroupDataConnector">
  <grouper:Attribute id="members:immediate" />
</resolver:DataConnector>

<resolver:AttributeDefinition id="immediateMembers" xsi:type="grouper:Member" sourceAttributeID="members:immediate" >
  <resolver:Dependency ref="GroupDataConnector" />
  <grouper:Attribute id="name" source="jdbc" />
</resolver:AttributeDefinition>
```

GroupDataConnector - Custom List (Membership)

The following example will return an attribute named "customMembers" whose values are the "name" of every member from the "jdbc" source of the "customList" list of a group :

```

<resolver:DataConnector id="GroupDataConnector" xsi:type="grouper:GroupDataConnector">
  <grouper:Attribute id="members:all:customList" />
</resolver:DataConnector>

<resolver:AttributeDefinition id="customMembers" xsi:type="grouper:Member" sourceAttributeID="members:all:
customList" >
  <resolver:Dependency ref="GroupDataConnector" />
  <grouper:Attribute id="name" source="jdbc" />
</resolver:AttributeDefinition>

```

GroupDataConnector - Member Of List

The following example will return an attribute named "isMemberOf" whose values are the "name" of every group of which the group is a member of :

```

<resolver:DataConnector id="GroupDataConnector" xsi:type="grouper:GroupDataConnector">
  <grouper:Attribute id="groups" />
</resolver:DataConnector>

<resolver:AttributeDefinition id="isMemberOf" xsi:type="grouper:Group" sourceAttributeID="groups" >
  <resolver:Dependency ref="GroupDataConnector" />
  <grouper:Attribute id="name" />
</resolver:AttributeDefinition>

```

GroupDataConnector - Privileges

Attributes representing Subjects which have Access Privileges to a group may be defined by privilege name as defined in the [Grouper Glossary](#).

```

<resolver:DataConnector id="GroupDataConnector" xsi:type="grouper:GroupDataConnector">
  <grouper:Attribute id="admins" />
  <grouper:Attribute id="optins" />
  <grouper:Attribute id="optouts" />
  <grouper:Attribute id="readers" />
  <grouper:Attribute id="updaters" />
  <grouper:Attribute id="viewers" />
</resolver:DataConnector>

```

The following example will return an attribute named "admin" whose values are the "name" of every Subject which has the ADMIN privilege on a group :

```

<resolver:DataConnector id="GroupDataConnector" xsi:type="grouper:GroupDataConnector">
  <grouper:Attribute id="admins" />
</resolver:DataConnector>

<resolver:AttributeDefinition id="admin" xsi:type="grouper:Subject" sourceAttributeID="admins" >
  <resolver:Dependency ref="GroupDataConnector" />
  <grouper:Attribute id="name" source="jdbc" />
</resolver:AttributeDefinition>

```

Member Data Connector

The MemberDataConnector returns attributes which represent a Grouper Member whose subject id or identifier is the principal name of an attribute request. Returned attributes, lists, and privileges must be specified to maximize retrieval performance.

```

<resolver:DataConnector id="MemberDataConnector" xsi:type="grouper:MemberDataConnector">
  <grouper:Attribute id="name" />
  <grouper:Attribute id="description" />
  <grouper:Attribute id="groups" />
  <grouper:Attribute id="admins" />
</resolver:DataConnector>

```

Member Data Connector - Attributes

The following example will return an attribute named "name" whose value is the name of a Member :

```

<resolver:DataConnector id="MemberDataConnector" xsi:type="grouper:MemberDataConnector" >
  <grouper:Attribute id="name" />
</resolver:DataConnector>

<resolver:AttributeDefinition id="name" xsi:type="ad:Simple">
  <resolver:Dependency ref="MemberDataConnector" />
</resolver:AttributeDefinition>

```

Member Data Connector - Lists

The following example will return an attribute named "isMemberOf" whose values are the "name" of every Group to which the Member is a member of the default "members" list :

```

<resolver:DataConnector id="MemberDataConnector" xsi:type="grouper:MemberDataConnector">
  <grouper:Attribute id="groups" />
</resolver:DataConnector>

<resolver:AttributeDefinition id="isMemberOf" xsi:type="grouper:Group" sourceAttributeID="groups" >
  <resolver:Dependency ref="MemberDataConnector" />
  <grouper:Attribute id="name" />
</resolver:AttributeDefinition>

```

Member Data Connector - Privileges

Attributes representing Groups to which a Member's subject has Access Privileges may be defined by privilege name as defined in the [Grouper Glossary](#).

```

<resolver:DataConnector id="MemberDataConnector" xsi:type="grouper:MemberDataConnector">
  <grouper:Attribute id="admins" />
  <grouper:Attribute id="optins" />
  <grouper:Attribute id="optouts" />
  <grouper:Attribute id="readers" />
  <grouper:Attribute id="updaters" />
  <grouper:Attribute id="viewers" />
</resolver:DataConnector>

```

The following example will return an attribute named "admin" whose values are the "name" of every Group to which the Member's subject has the ADMIN privilege :

```

<resolver:DataConnector id="MemberDataConnector" xsi:type="grouper:MemberDataConnector">
  <grouper:Attribute id="admins" />
</resolver:DataConnector>

<resolver:AttributeDefinition id="admin" xsi:type="grouper:Group" sourceAttributeID="admins" >
  <resolver:Dependency ref="MemberDataConnector" />
  <grouper:Attribute id="name" />
</resolver:AttributeDefinition>

```

Stem Data Connector

The StemDataConnector returns attributes which represent the Grouper stem whose name is the principal name of an attribute request. The attributes returned for a stem include built-in attributes such as *id*, *name*, *displayName*, *extension*, *displayExtension*, and *description*, as well as custom attributes and attribute framework attributes.

```

<resolver:DataConnector id="StemDataConnector" xsi:type="grouper:StemDataConnector" />

```

Change Log Data Connector

The ChangeLogDataConnector returns attributes representing the [Grouper change log](#) entry whose sequence number is the principal name of an attribute request.

The ChangeLogDataConnector returns an attribute for every change log field, plus *actionName*, and *changeLogCategory*.

For example, for a membership add change log entry, the built in *id*, *fieldName*, *subjectId*, *sourceId*, *membershipType*, *groupId*, *groupName*, *memberId*, and *fieldId* attributes are returned. The ChangeLogDataConnector also returns an *actionName* attribute, in this case with value "membership", as well as a *changeLogCategory* attribute, in this case with value "addMembership".

```
package edu.internet2.middleware.grouper.changeLog;

public enum ChangeLogTypeBuiltin implements ChangeLogTypeIdentifier {

    /**
     * add membership
     */
    MEMBERSHIP_ADD(new ChangeLogType("membership", "addMembership",
        ChangeLogLabels.MEMBERSHIP_ADD.id,
        ChangeLogLabels.MEMBERSHIP_ADD.fieldName,
        ChangeLogLabels.MEMBERSHIP_ADD.subjectId,
        ChangeLogLabels.MEMBERSHIP_ADD.sourceId,
        ChangeLogLabels.MEMBERSHIP_ADD.membershipType,
        ChangeLogLabels.MEMBERSHIP_ADD.groupId,
        ChangeLogLabels.MEMBERSHIP_ADD.groupName,
        ChangeLogLabels.MEMBERSHIP_ADD.memberId,
        ChangeLogLabels.MEMBERSHIP_ADD.fieldId)),
```

If a change log entry includes *subjectId* and *sourceId*, the subject name is returned via the *subjectName* attribute, the subject description is returned via the *subjectDescription* attribute, and any other subject attribute is returned via the *subject<attributeName>* attribute.

For an attribute framework attribute value assignment change log entry, the *attributeDefName* and *attributeAssignType* fields are returned. If the attribute value assign type is "group", then the name of the group is returned via the *name* attribute. If the attribute value assign type is "stem", then the name of the stem is returned via the *name* attribute. If the attribute value assign type is "member", then the member subject id is returned via the *name* attribute.

Source code for the change log data connector and filters are available from the [psp-grouper-changelog](#) project.

Filters

Objects returned by the data connectors may be filtered.

Filter - GroupExactAttribute

The GroupExactAttribute returns groups which have an exact attribute value :

```
<resolver:DataConnector id="testFilterExactAttribute" xsi:type="grouper:GroupDataConnector">
  <grouper:Filter xsi:type="grouper:GroupExactAttribute" name="name" value="stem:group" />
</resolver:DataConnector>
```

Filter - GroupInStem

The GroupInStem returns groups which are children of the named stem with the given scope :

```
<resolver:DataConnector id="StemNameFilterONE" xsi:type="grouper:GroupDataConnector">
  <grouper:Filter xsi:type="grouper:GroupInStem" name="parentStem" scope="ONE" />
</resolver:DataConnector>

<resolver:DataConnector id="StemNameFilterSUB" xsi:type="grouper:GroupDataConnector">
  <grouper:Filter xsi:type="grouper:GroupInStem" name="parentStem" scope="SUB" />
</resolver:DataConnector>
```

Filter - AND

The AND filter returns objects which match both child filters, in other words, an Intersection :

```
<grouper:Filter xsi:type="grouper:AND">
  <grouper:Filter xsi:type="grouper:ExactAttribute" name="name" value="parentStem:group_name" />
  <grouper:Filter xsi:type="grouper:StemName" name="parentStem" scope="ONE" />
</grouper:Filter>
```

Filter - OR

The OR filter returns objects which match either of two child filters, in other words, a Union :

```
<grouper:Filter xsi:type="grouper:OR">
  <grouper:Filter xsi:type="grouper:ExactAttribute" name="name" value="parentStem:group_name" />
  <grouper:Filter xsi:type="grouper:StemName" name="parentStem:childStem" scope="ONE" />
</grouper:Filter>
```

Filter - MINUS

The MINUS filter returns objects which match the result of the first child filter minus the result of the second child filter, in other words, the Complement :

```
<grouper:GroupFilter xsi:type="grouper:Minus">
  <grouper:GroupFilter xsi:type="grouper:StemName" name="parentStem" scope="ONE" />
  <grouper:GroupFilter xsi:type="grouper:ExactAttribute" name="name" value="parentStem:group_name" />
</grouper:GroupFilter>
```

Filter - StemInStem

The StemInStem filter returns stems which are children of the named stem with the given scope :

```
<resolver:DataConnector id="StemNameFilterONE" xsi:type="grouper:GroupDataConnector">
  <grouper:Filter xsi:type="grouper:StemInStem" name="parentStem" scope="ONE" />
</resolver:DataConnector>

<resolver:DataConnector id="StemNameFilterSUB" xsi:type="grouper:GroupDataConnector">
  <grouper:Filter xsi:type="grouper:StemInStem" name="parentStem" scope="SUB" />
</resolver:DataConnector>
```

Filter - StemNameExact

The StemNameExact filter returns stems with the given name :

```
<resolver:DataConnector id="testFilterStemNameExact" xsi:type="grouper:StemDataConnector">
  <grouper:Filter xsi:type="grouper:StemNameExact" name="parentStem" />
</resolver:DataConnector>
```

Filter - ChangeLogAudit

The ChangeLogAudit filter returns change log entries with the given audit category and or action.

```
<grouper:Filter xsi:type="psp-grouper-changelog:ChangeLogAudit" category="group" action="deleteGroup" />
```

Filter - ChangeLogEntry

The ChangeLogEntry filter returns change log entries with the given change log category and or action.

```
<grouper:Filter xsi:type="psp-grouper-changelog:ChangeLogEntry" category="membership" action="deleteMembership" />
```

Filter - ChangeLogExactAttribute

The ChangeLogExactAttribute filter returns change log entries with the given attribute name and value.

```
<grouper:Filter xsi:type="psp-grouper-changelog:ChangeLogExactAttribute" name="propertyChanged" value="description" />
```

Filter - ChangeLogAttributeAssignType

The `ChangeLogAttributeAssignType` filter returns change log entries with the given attribute value assign type.

```
<grouper:Filter xsi:type="psp-grouper-changelog:ChangeLogAttributeAssignType" attributeAssignType="group" />
```

Group Attribute Definition

The `GroupAttributeDefinition` returns Group attributes.

For example, the following "isMemberOf" attribute will have values consisting of the "name" of every Group :

```
<resolver:AttributeDefinition id="isMemberOf" xsi:type="grouper:Group" sourceAttributeID="groups" >
  <resolver:Dependency ref="GroupDataConnector" />
  <grouper:Attribute id="name" />
</resolver:AttributeDefinition>
```

Member Attribute Definition

The `MemberAttributeDefinition` returns Member attributes.

For example, the following "member" attribute will have values consisting of the "name" attribute of every Member whose subject is from the "jdbc" source :

```
<resolver:AttributeDefinition id="member" xsi:type="grouper:Member" sourceAttributeID="members" >
  <resolver:Dependency ref="GroupDataConnector" />
  <grouper:Attribute id="name" source="jdbc" />
</resolver:AttributeDefinition>
```

Subject Attribute Definition

The `SubjectAttributeDefinition` returns Subject attributes.

For example, the following "owner" attribute will have values consisting of the "name" attribute of every Subject from the "jdbc" source :

```
<resolver:AttributeDefinition id="owner" xsi:type="grouper:Subject" sourceAttributeID="members" >
  <resolver:Dependency ref="GroupDataConnector" />
  <grouper:Attribute id="name" source="jdbc" />
</resolver:AttributeDefinition>
```

See Also

[Exposing Groups Through Shibboleth](#)

For an overview of authenticating to Grouper using Shib, see also the [Grouper UI Training Video](#), around minute 7.30.

For Controlling Access to the Grouper UI with Shib [see the Newcastle University Contribution](#)