subject-0.3.1-doc

Subject API v0.3.1 Documentation

This version of the Subject API distribution contains a configurable JNDI source adapter and a configurable JDBC source adapter. In addition, the Grouper distribution contains a source adapter (the GrouperSourceAdapter) that presents Grouper groups as subjects. Third parties may write their own source adapters; however, in this version of the Subject API it may be necessary to modify Subject API source code beyond merely implementing the Source and Subject interfaces.

Changes from v0.2.1

The JDBCSourceAdapter and JNDISourceAdapter have been restructured to prevent parameter injection attacks, and the JDBCSourceAdapter has had some performance improvement.

The Subject and Source interfaces are unchanged from v0.2.1.

Configuration & Usage

Below is the structure of the Subject API v0.3.1 'sources.xml' configuration file. Following that, its elements and attributes are described in detail.

```
<sources>

<source adapterClass="aClassRef"/>
<id>sourceId</id>
<name>sourceDisplayName</name>
<type>subjectType</type>
<init-param>
<param-name>various</param-name>
<param-value>site-specific</param-value>
</init-param>
...
<attribute>attributeType</attribute>
...
<search>
<searchType>searchSubject, searchSubjectByIdentifier, or search</searchType>
<param>
<param-name>JNDI or JDBC specific params</param-name>
<param-value>back-end specific declarations</param-value>
</param>
...
</searchType>
</search>
...
</source>

...

</sources>
```

sources element

A sources.xml file contains a single **sources** element with one or more subordinate **source** elements.

source element

Each **source** element configures one instance of a source adapter. Its **adapterClass** attribute is the name of the java class configured by this element.

id element

A string identifying this identity source. This value is the value of the sourceId attribute of every subject resolved from this source.

NOTE: A Subject API caller need only persist the sourceId and subjectId of a subject in order to be able to resolve its attributes later. It is important that the sourceId values be stable over time so that the pair (sourceId, subjectId) continue to refer to the same subject. Hence, a wise deployer will spend some time to determine a good scheme for assigning sourceId values before making initial production use of the Subject API.

name element

A displayable name for this identity source.

type element

The type of subject presented by this source adapter instance. In v0.3.1 this is limited to one of 'person', 'group', and 'application'.

Note: The notion of type in the Subject API will be removed in v1.0. Grouper and Signet, in particular, either do not now or in upcoming releases will not make special use of a subject type as signaled by the Subject API. Subject API callers should instead identify any caller-specific special handling of subjects by the sourceId or by other attributes of subject objects.

init-param elements

The JDBC and JNDI source adapters each require distinct parameter declarations to set up connections to back-end stores. They share three other parameters that declare which back-end attributes or columns will be presented as the Subject object's distinguished attributes.

The form of these parameter declarations is

```
<init-param>
<param-name>parameter_name</param-name>
<param-value>parameter_value</param-value>
</init-param>
```

The parameters required for each source adapter class and descriptions of each follow. The GrouperSourceAdapter requires no parameters.

adapterClass	parameter name	parameter value
JDBCSourceAdapter	dbDriver	JDBC driver classname
JDBCSourceAdapter	dbURL	JDBC URL for the database
JDBCSourceAdapter	dbUser	database user
JDBCSourceAdapter	dbPwd	database user's password
JDBCSourceAdapter	maxActive	refer to Apache Commons DBCP documentation
JDBCSourceAdapter	maxIdle	refer to Apache Commons DBCP documentation
JDBCSourceAdapter	maxWait	refer to Apache Commons DBCP documentation
JNDISourceAdapter	INITIAL_CONTEXT_FACTORY	A string specific to the java you are using. For Sun's java it is "com.sun.jndi.ldap.LdapCtxFactory". See the JNDI documentation
JNDISourceAdapter	PROVIDER_URL	The LDAP URL of the LDAP server to connect to.
JNDISourceAdapter	SECURITY_AUTHENTICATION	See the list of allowable values
JNDISourceAdapter	SECURITY_PRINCIPAL	The DN to BIND as to the LDAP server specified in the PROVIDER_URL.
JNDISourceAdapter	SECURITY_CREDENTIALS	A hashed password, clear-text password, key, certificate, whatever you use to authenticate the SECURITY_PRINCIPAL to the LDAP server at the PROVIDER_URL.
Both	SubjectID_AttributeType	The name of the attribute or column whose value is the subjectId.
Both	Name_AttributeType	The name of the attribute or column whose value is the subject's name.
Both	Description_AttributeType	The name of the attribute or column whose value is the subject's description.

attribute element(s)

The JNDI source adapter will limit the set of attributes returned in an LDAP search to those listed in **attribute** elements. If there are no **attribute** elements, then all attributes visible to the SECURITY_PRINCIPAL will be presented to the calling program.

search element

The Subject API defines three methods used to select or search for subjects. There must be one **search** element for each of these three methods.

searchType element

Identifies the Source interface method configured by this **search** element, as given in the table below. The parameter set for each **search** element defines how the selection or search is to be carried out against the back-end identity store, and which columns or attributes will be used as attributes of the subject objects that are returned. The string "%TERM%" should be used in search filters and the string '?' in WHERE clauses - they are replaced by the selection criterion or search term presented to the corresponding method.

searchType value	Source interface method	? or %TERM% is ...	What the search should accomplish
searchSubject	getSubject	a subjectId value	Select the unique record or entry with subjectId=%TERM%, or none if %TERM% is no subject's subjectId.
searchSubjectByIdentifier	getSubjectByIdentifier	the value of an identifying attribute	Select the unique record or entry which has %TERM% for the value of one of its identifying columns or attributes, or none if %TERM% is not the value of any subject's identifying column or attribute.
search	search	a string	Select all records or entries in which the %TERM% causes a match.

The "getSubject" method is used to select a specific subject from the back-end identity store, for example, to show the name and department of a person belonging to a group.

The "getSubjectByIdentifier" method enables identifying a subject by means of a column or attribute different from that used as the subjectId. For example, if a UI user authenticates with a loginId, but the subjectId is an opaque registryId, this method is used to identify the subject given their loginId.

The "search" method is used to help a UI user select the subject they want from a list. It is typically implemented as a substring search on several non-identifying columns or attributes such as lastname, firstname, and department. The results of a search are displayed in a checkbox list to the UI user.

sql element

The value of this element is a (possibly compound) SQL statement. Before being executed, all occurrences of the '?' variable in the SQL statement are replaced with the corresponding method's argument. The SQL statement should return exactly one table with zero or more rows. Each row corresponds to exactly one subject, and the column names of the returned table are used as the attribute names of the subject objects created for each row. The set of rows is assumed to be the set of all subjects meeting the selection or search criterion of the containing **search** element. The **sql** element is only used for configuring the JDBCSourceAdapter.

filter, base, and scope elements

These elements are only used for configuring the JNDISourceAdapter. They correspond to the various parts of an LDAP URL of the same name. Thus, the **filter** element defines a boolean search filter, the **base** and **scope** specify the portion of the Directory Information Tree to be searched, and zero or more **attribute** element values form the list of attributes to be returned with each matching entry.

The **scope** element MUST contain one of the values "OBJECT_SCOPE", "ONELEVEL_SCOPE", or "SUBTREE_SCOPE", which corresponds to an RFC2255 scope parameter of "0", "1", or "2", respectively.

The **base** element is the DN (distinguished name) of an entry in the directory which is the root of the portion of the Directory Information Tree to be searched.

Example of a sources.xml file

```
<sources>

<!-- Group Subject Resolver -->
<source adapterClass="edu.internet2.middleware.grouper.GrouperSourceAdapter">
<id>g:gsa</id>
<name>Grouper: Group Source Adapter</name>
<type>group</type>
</source>

<!-- Example JDBC Person Resolver -->
<source adapterClass="edu.internet2.middleware.subject.provider.JDBCSourceAdapter">
<id>uc</id>
<name>Bogus UC People</name>
<type>person</type>

<init-param>
<param-name>maxActive</param-name>
<param-value>16</param-value>
</init-param>
```

```
<init-param>
<param-name>maxIdle</param-name>
<param-value>16</param-value>
</init-param>
<init-param>
<param-name>maxWait</param-name>
<param-value>-1</param-value>
</init-param>
<init-param>
<param-name>dbDriver</param-name>
<param-value>org.hsqldb.jdbcDriver</param-value>
</init-param>
<init-param>
<param-name>dbUrl</param-name>
<param-value>jdbc:hsqldb:hsql://localhost:9002/bogus-uc-people</param-value>
</init-param>
<init-param>
<param-name>dbUser</param-name>
<param-value>sa</param-value>
</init-param>
<init-param>
<param-name>dbPwd</param-name>
<param-value></param-value>
</init-param>
<init-param>
<param-name>SubjectID_AttributeType</param-name>
<param-value>id</param-value>
</init-param>
<init-param>
<param-name>Name_AttributeType</param-name>
<param-value>name</param-value>
</init-param>
<init-param>
<param-name>Description_AttributeType</param-name>
<param-value>name</param-value>
</init-param>

<search>
<searchType>searchSubject</searchType>
<param>
<param-name>sql</param-name>
<param-value>
select id,
concat(firstname, concat(' ', lastname)) as name,
concat(lastname, concat(', ', firstname)) as lfname,
lastname, firstname, middlename,
account.name as loginid,
department, curriculum, appointment
from
individual
left join account on (account.individualid = id)
left join faculty on (faculty.individualid = id)
left join staff on (staff.individualid = id)
left join student on (student.individualid = id)
where (id=?)
</param-value>
</param>
</search>
<search>
<searchType>searchSubjectByIdentifier</searchType>
<param>
<param-name>sql</param-name>
<param-value>
select id,
concat(firstname, concat(' ', lastname)) as name,
concat(lastname, concat(', ', firstname)) as lfname,
lastname, firstname, middlename,
account.name as loginid,
department, curriculum, appointment
from
individual
```

```
left join account on (account.individualid = id)
left join faculty on (faculty.individualid = id)
left join staff on (staff.individualid = id)
left join student on (student.individualid = id)
where (account.name=?)
</param-value>
</param>
</search>
<search>
<searchType>search</searchType>
<param>
<param-name>sql</param-name>
<param-value>
select id,
concat(firstname, concat(' ', lastname)) as name,
concat(lastname, concat(', ', firstname)) as lfname,
lastname, firstname, middlename,
account.name as loginid,
department, curriculum, appointment
from
individual
left join account on (account.individualid = id)
left join faculty on (faculty.individualid = id)
left join staff on (staff.individualid = id)
left join student on (student.individualid = id)
where (firstname like '%'||?||'%')
or (lastname like '%'||?||'%')
or (department like '%'||?||'%')
or (account.name like '%'||?||'%')
</param-value>
</param>
</search>
</source>

<!-- Example JNDI Person Resolver -->
<source adapterClass="edu.internet2.middleware.subject.provider.JNDISourceAdapter">
<id>kitn-person</id>
<name>KITN People</name>
<type>person</type>
<init-param>
<param-name>INITIAL_CONTEXT_FACTORY</param-name>
<param-value>com.sun.jndi.ldap.LdapCtxFactory</param-value>
</init-param>
<init-param>
<param-name>PROVIDER_URL</param-name>
<param-value>ldap://localhost:389</param-value>
</init-param>
<init-param>
<param-name>SECURITY_AUTHENTICATION</param-name>
<param-value>simple</param-value>
</init-param>
<init-param>
<param-name>SECURITY_PRINCIPAL</param-name>
<param-value>cn=Manager,dc=example,dc=edu</param-value>
</init-param>
<init-param>
<param-name>SECURITY_CREDENTIALS</param-name>
<param-value>secret</param-value>
</init-param>
<init-param>
<param-name>SubjectID_AttributeType</param-name>
<param-value>exampleEduRegID</param-value>
</init-param>
<init-param>
<param-name>Name_AttributeType</param-name>
<param-value>cn</param-value>
</init-param>
<init-param>
<param-name>Description_AttributeType</param-name>
<param-value>description</param-value>
</init-param>
```

```
<search>
<searchType>searchSubject</searchType>
<param>
<param-name>filter</param-name>
<param-value>
(&amp;amp; (exampleEduRegId=%TERM%)(objectclass=exampleEduPerson))
</param-value>
</param>
<param>
<param-name>scope</param-name>
<param-value>SUBTREE_SCOPE</param-value>
</param>
<param>
<param-name>base</param-name>
<param-value>ou=people,dc=kitn,dc=edu</param-value>
</param>
</search>
<search>
<searchType>searchSubjectByIdentifier</searchType>
<param>
<param-name>filter</param-name>
<param-value>
(&amp;amp; (uid=%TERM%)(objectclass=exampleEduPerson))
</param-value>
</param>
<param>
<param-name>scope</param-name>
<param-value>SUBTREE_SCOPE</param-value>
</param>
<param>
<param-name>base</param-name>
<param-value>ou=people,dc=example,dc=edu</param-value>
</param>
</search>
<search>
<searchType>search</searchType>
<param>
<param-name>filter</param-name>
<param-value>
(&amp;amp;(|(uid=%TERM%)(cn=*%TERM%*)(exampleEduRegId=%TERM%))(objectclass=exampleEduPerson))
</param-value>
</param>
<param>
<param-name>scope</param-name>
<param-value>SUBTREE_SCOPE</param-value>
</param>
<param>
<param-name>base</param-name>
<param-value>ou=people,dc=example,dc=edu</param-value>
</param>
</search>
</source>

</sources>
```