

Grouper web service - subject source - SQL table

Wiki Home	Grouper Release Announcements	Grouper Guides	Grouper Deployment Guide	Community Contributions	Internal Developer Resources
---------------------------	---	--------------------------------	--	---	--

Custom SQL table and source

Note, you can use the built in grouper sql source like the grouper demo server

Here is an example of custom sql table and source used at Penn with Oracle

Table:

```
CREATE TABLE SERVICE_PRINCIPALS
(
  PRINCIPAL_NAME      VARCHAR2(512 CHAR),
  ID                  INTEGER,
  LAST_UPDATED        TIMESTAMP(6),
  REASON              VARCHAR2(4000 CHAR),
  PENNKEY_WHO_ADDED   CHAR(20 CHAR),
  PENNID_WHO_ADDED    VARCHAR2(20 CHAR)
);
COMMENT ON COLUMN SERVICE_PRINCIPALS.REASON IS 'reason for adding this principal';
COMMENT ON COLUMN SERVICE_PRINCIPALS.PENNKEY_WHO_ADDED IS 'pennkey who added the record';
COMMENT ON COLUMN SERVICE_PRINCIPALS.PENNID_WHO_ADDED IS 'pennid who added the record';
CREATE UNIQUE INDEX SERVICE_PRINCIPALS_PK ON SERVICE_PRINCIPALS
(PRINCIPAL_NAME);

ALTER TABLE SERVICE_PRINCIPALS ADD (
  CONSTRAINT SERVICE_PRINCIPALS_PK
  PRIMARY KEY
  (PRINCIPAL_NAME)
  USING INDEX SERVICE_PRINCIPALS_PK);
```

Subject source in subject.properties

```
#####
## Configuration for source id: servPrinc
## Source configName: servPrinc
#####
subjectApi.source.servPrinc.id = servPrinc

# this is a friendly name for the source
subjectApi.source.servPrinc.name = Kerberos service principals

# type is not used all that much. Can have multiple types, comma separate. Can be person, group, application
subjectApi.source.servPrinc.types = application

# the adapter class implements the interface: edu.internet2.middleware.subject.Source
# adapter class must extend: edu.internet2.middleware.subject.provider.BaseSourceAdapter
# edu.internet2.middleware.grouper.subj.GrouperJdbcSourceAdapter2 : if doing JDBC this should be used if
possible. All subject data in one table/view.
# edu.internet2.middleware.grouper.subj.GrouperJdbcSourceAdapter : oldest JDBC source. Put freeform queries
in here
# edu.internet2.middleware.grouper.subj.GrouperJndiSourceAdapter : used for LDAP
subjectApi.source.servPrinc.adapterClass = edu.internet2.middleware.subject.provider.JDBCSourceAdapter

subjectApi.source.servPrinc.param.jdbcConnectionProvider.value = edu.internet2.middleware.grouper.subj.
GrouperJdbcConnectionProvider

subjectApi.source.servPrinc.param.maxPageSize.value = 100
```

```

# ldap attribute which is the subject id. e.g. exampleEduRegID Each subject has one and only one subject
id. Generally it is opaque and permanent.
subjectApi.source.servPrinc.param.SubjectID_AttributeType.value = loginid

# attribute which is the subject name
subjectApi.source.servPrinc.param.Name_AttributeType.value = name

# attribute which is the subject description
subjectApi.source.servPrinc.param.Description_AttributeType.value = description

# the 1st sort attribute for lists on screen that are derived from member table (e.g. search for member in
group)
# you can have up to 5 sort attributes
subjectApi.source.servPrinc.param.sortAttribute0.value = loginid

# the 1st search attribute for lists on screen that are derived from member table (e.g. search for member in
group)
# you can have up to 5 search attributes
subjectApi.source.servPrinc.param.searchAttribute0.value = loginid

subjectApi.source.servPrinc.param.useInClauseForIdAndIdentifier.value = true

subjectApi.source.servPrinc.param.identifierAttributes.value = loginid

#searchSubject: find a subject by ID. ID is generally an opaque and permanent identifier, e.g. 12345678.
# Each subject has one and only one ID. Returns one result when searching for one ID.
subjectApi.source.servPrinc.search.searchSubject.param.numParameters.value = 1

# sql is the sql to search for the subject by id should use an {inclause}
subjectApi.source.servPrinc.search.searchSubject.param.sql.value = select principal_name as name,
principal_name as loginid, principal_name as description from service_principals where {inclause}

# inclause allows searching by subject for multiple ids or identifiers in one query, must have {inclause} in
the sql query,
# this will be substituted to in clause with the following. Should use a question mark ? for bind variable
subjectApi.source.servPrinc.search.searchSubject.param.inclause.value = principal_name = ?

#searchSubjectByIdentifier: find a subject by identifier. Identifier is anything that uniquely
# identifies the user, e.g. jsmith or jsmith@institution.edu.
# Subjects can have multiple identifiers. Note: it is nice to have if identifiers are unique
# even across sources. Returns one result when searching for one identifier.
subjectApi.source.servPrinc.search.searchSubjectByIdentifier.param.numParameters.value = 1

# sql is the sql to search for the subject by identifier should use an {inclause}
subjectApi.source.servPrinc.search.searchSubjectByIdentifier.param.sql.value = select principal_name as
name, principal_name as loginid, principal_name as description from service_principals where
{inclause}

# inclause allows searching by subject for multiple ids or identifiers in one query, must have {inclause} in
the sql query,
# this will be substituted to in clause with the following. Should use a question mark ? for bind variable
subjectApi.source.servPrinc.search.searchSubjectByIdentifier.param.inclause.value = principal_name = ?

# search: find subjects by free form search. Returns multiple results.
subjectApi.source.servPrinc.search.search.param.numParameters.value = 1

# sql is the sql to search for the subject free-form search. user question marks for bind variables
subjectApi.source.servPrinc.search.search.param.sql.value = select principal_name as name, principal_name
as loginid, principal_name as description from service_principals where (lower(principal_name) like
lower(concat('%',concat(?, '%'))))

```

Script to add a user (including to the WS group)

```
grouperSession = GrouperSession.startRootSession();
kerb = "something/somewhere.institution.edu";
reason = "canvas grouper integration";
sqlRun("insert into service_principals (principal_name, id, last_updated, reason) values ('" + kerb + "',
hibernate_sequence.nextval, systimestamp, '" + reason + "')");
addMember("etc:ldapUsers", kerb);
addMember("etc:webServiceClientUsers", kerb);
```

Run the script

```
./gsh scriptName.gsh
```