LDAP Subject API example

Public LDAP example

CMU has a public LDAP server. We can hook up a subject source to it as an example

Server: ldap.andrew.cmu.edu
Base DN: dc=cmu,dc=edu
URL: ldap://ldap.andrew.cmu.edu:389/dc=cmu,dc=edu

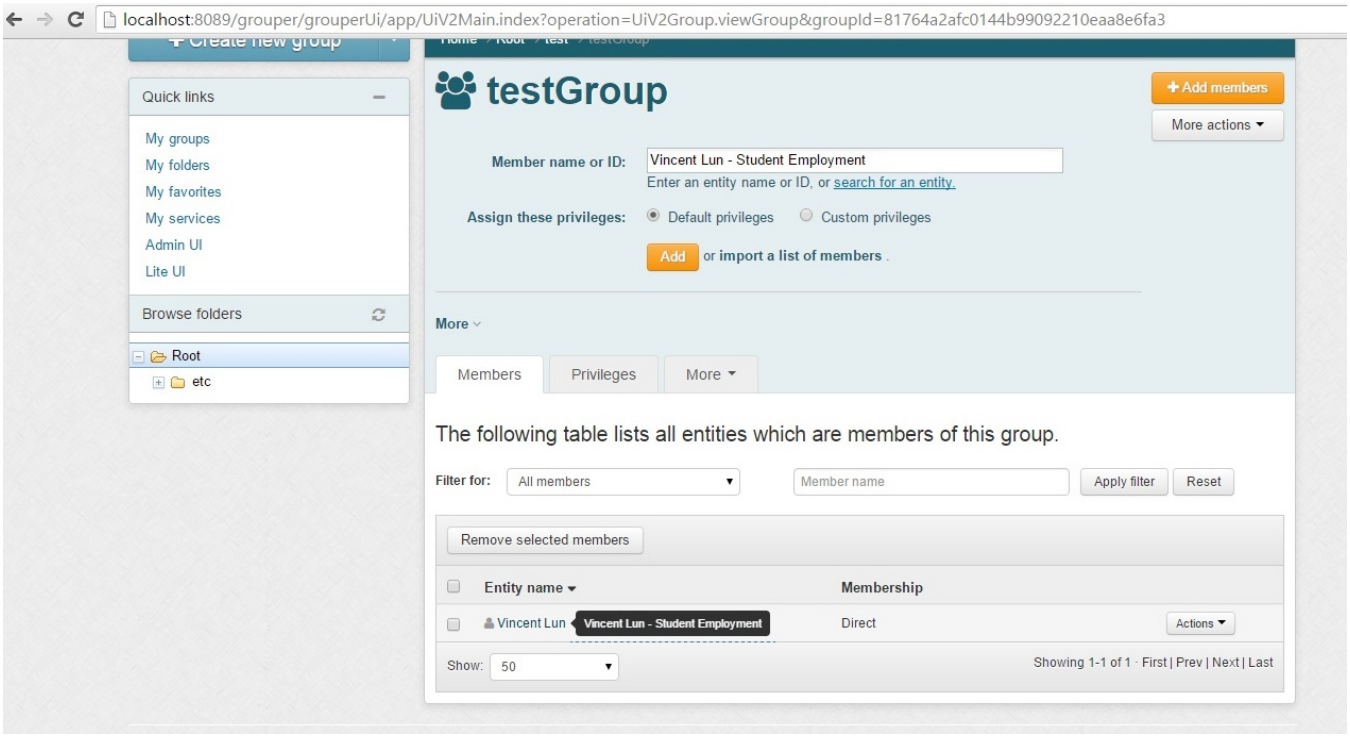
Top OU: ou=person

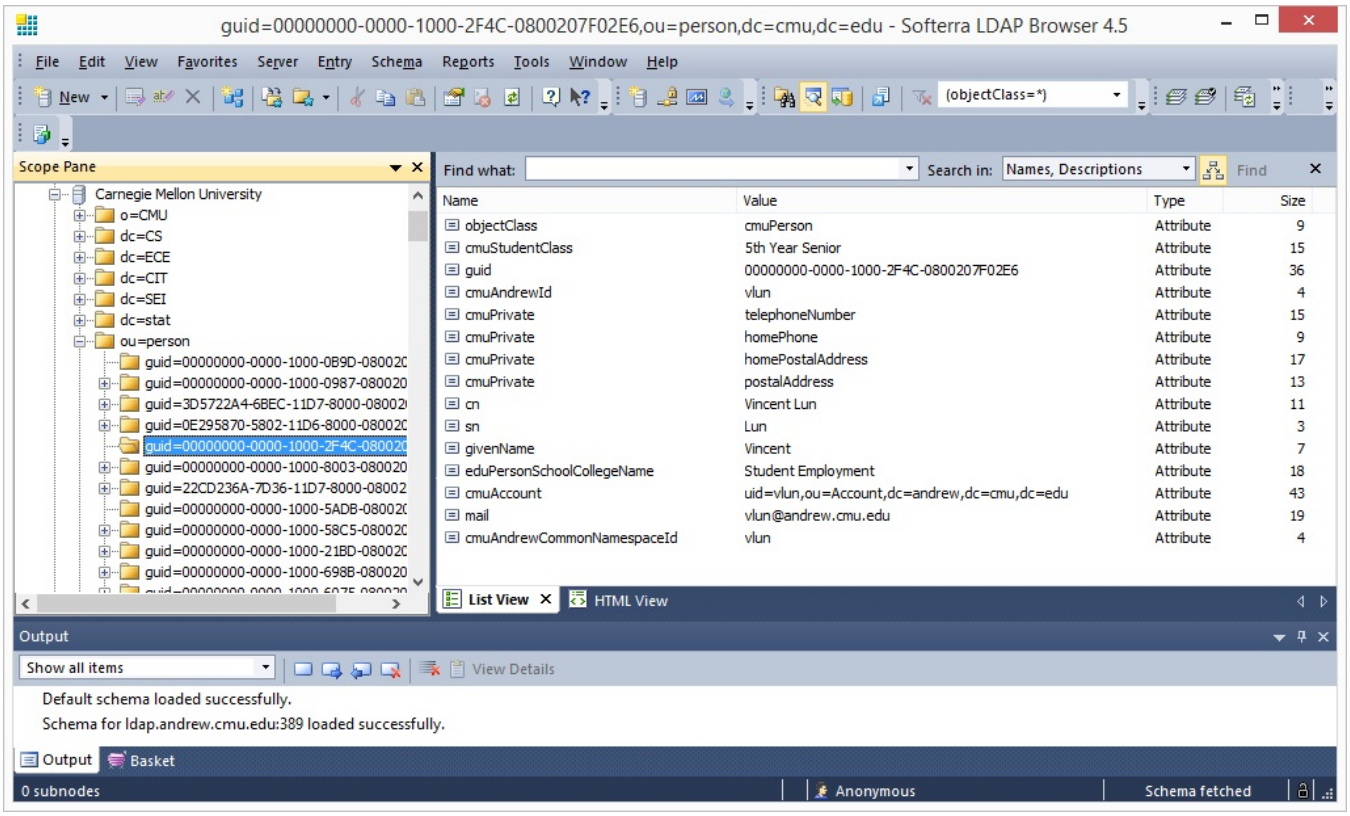
Users: guid=ABC123

Attributes:

- objectClass: cmuPerson
- cn (First Last)
- mail
- eduPersonSchoolCollegeName
- cmuAndrewId: netId

This example shows have a description that is the concatenation of the name and the school college name. It will concatenate if it exists or not if there is no school college name.





subject.properties

```
# Copyright 2016 Internet2
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#   http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

#
# Subject configuration
#

# The subject properties uses Grouper Configuration Overlays (documented on wiki)
# By default the configuration is read from subject.base.properties
# (which should not be edited), and the subject.properties overlays
# the base settings.  See the subject.base.properties for the possible
# settings that can be applied to the subject.properties

# enter the location of the sources.xml.  Must start with classpath: or file:
# blank means dont use sources.xml, use subject.properties
# default is: classpath:sources.xml
# e.g. file:/dir1/dir2/sources.xml
subject.sources.xml.location =


##########################################
## Configuration for source id: cmu
## Source configName: cmu
##########################################
```

```
subjectApi.source.cmu.id = cmu

# this is a friendly name for the source
subjectApi.source.cmu.name = cmu

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

# 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 da
ta 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.cmu.adapterClass = edu.internet2.middleware.grouper.subj.GrouperJndiSourceAdapter

# e.g. com.sun.jndi.ldap.LdapCtxFactory
subjectApi.source.cmu.param.INITIAL_CONTEXT_FACTORY.value = com.sun.jndi.ldap.LdapCtxFactory

# e.g. ldap://localhost:389
subjectApi.source.cmu.param.PROVIDER_URL.value = ldap://ldap.andrew.cmu.edu:389/dc=cmu,dc=edu

# e.g. simple, none, sasl_mech
subjectApi.source.cmu.param.SECURITY_AUTHENTICATION.value = none

# ldap attribute which is the subject id.  e.g. exampleEduRegID   Each subject has one and only one subject
id.  Generally it is opa
que and permanent.
subjectApi.source.cmu.param.SubjectID_AttributeType.value = guid

# if the subject id should be changed to lower case after reading from datastore.  true or false
subjectApi.source.cmu.param.SubjectID_formatToLowerCase.value = false

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

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

# when evaluating the virtual attribute EL expression, this variable can be used from this java class.
# subjectVirtualAttributeVariable_grouperUtilElSafe variable is the edu.internet2.middleware.grouper.util.
GrouperUtilElSafe class.
Call static methods
subjectApi.source.cmu.param.subjectVirtualAttributeVariable_grouperUtilElSafe.value = edu.internet2.middleware.
grouper.util.GrouperU
tilElSafe

# This virtual attribute index 0 is accessible via: subject.getAttributeValue("nameLong");
subjectApi.source.cmu.param.subjectVirtualAttribute_0_nameLong.value = ${grouperUtilElSafe.
appendIfNotBlankString(grouperUtilElSafe.
defaultIfBlank(subject.getAttributeValue('cn'), ''), ' - ', grouperUtilElSafe.defaultIfBlank(subject.
getAttributeValue('eduPersonSch
oolCollegeName'), ''))}

# 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.cmu.param.sortAttribute0.value = nameLong

# 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.cmu.param.searchAttribute0.value = nameLong

#searchSubject: find a subject by ID.  ID is generally an opaque and permanent identifier, e.g. 12345678.
#  Each subject has one and only on ID.  Returns one result when searching for one ID.

# sql is the sql to search for the subject by id.  %TERM% will be subsituted by the id searched for
subjectApi.source.cmu.search.searchSubject.param.filter.value = (& (guid=%TERM%) (objectclass=cmuPerson))
```

```
# Scope Values can be: OBJECT_SCOPE, ONELEVEL_SCOPE, SUBTREE_SCOPE
subjectApi.source.cmu.search.searchSubject.param.scope.value = ONELEVEL_SCOPE

# base dn to search in
subjectApi.source.cmu.search.searchSubject.param.base.value = ou=person

#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.

# sql is the sql to search for the subject by identifier.  %TERM% will be subsituted by the identifier searched
for
subjectApi.source.cmu.search.searchSubjectByIdentifier.param.filter.value = (& (cmuAndrewId=%TERM%)
(objectclass=cmuPerson))

# Scope Values can be: OBJECT_SCOPE, ONELEVEL_SCOPE, SUBTREE_SCOPE
subjectApi.source.cmu.search.searchSubjectByIdentifier.param.scope.value = ONELEVEL_SCOPE

# base dn to search in
subjectApi.source.cmu.search.searchSubjectByIdentifier.param.base.value = ou=person

#   search: find subjects by free form search.  Returns multiple results.

# sql is the sql to search for the subject by free form search.  %TERM% will be subsituted by the text searched
for
subjectApi.source.cmu.search.search.param.filter.value = (& (|(|(cmuAndrewId=%TERM%)(cn=*%TERM%*))(guid=%TERM%))
(objectclass=cmuPers
on))

# Scope Values can be: OBJECT_SCOPE, ONELEVEL_SCOPE, SUBTREE_SCOPE
subjectApi.source.cmu.search.search.param.scope.value = ONELEVEL_SCOPE

# base dn to search in
subjectApi.source.cmu.search.search.param.base.value = ou=person

# attributes from ldap object to become subject attributes.  comma separated
subjectApi.source.cmu.attributes = eduPersonSchoolCollegeName, sn, cmuStudentClass, givenName, mail

# internal attributes are used by grouper only not exposed to code that uses subjects.  comma separated
subjectApi.source.cmu.internalAttributes = searchAttribute0
```

sources.xml (pre v2.3)

```
  <source adapterClass="edu.internet2.middleware.grouper.subj.GrouperJndiSourceAdapter">
    <id>cmu</id>
    <name>cmu</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://ldap.andrew.cmu.edu:389/dc=cmu,dc=edu</param-value>
    </init-param>
    <init-param>
      <param-name>SECURITY_AUTHENTICATION</param-name>
      <param-value>none</param-value>
      <!-- param-value>simple</param-value -->
    </init-param>
    <!-- init-param>
      <param-name>SECURITY_PRINCIPAL</param-name>
      <param-value>CN=grouperad,OU=Service Accounts</param-value>
    </init-param>
```

```
<init-param>
  <param-name>SECURITY_CREDENTIALS</param-name>
  <param-value>/etc/grouper/ADSource.pass</param-value>
</init-param -->
 <init-param>
  <param-name>SubjectID_AttributeType</param-name>
  <param-value>guid</param-value>
</init-param>
 <init-param>
  <param-name>SubjectID_formatToLowerCase</param-name>
  <param-value>false</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>nameLong</param-value>
</init-param>

<!--  ///
      /// For filter use  -->

<search>
    <searchType>searchSubject</searchType>
    <param>
        <param-name>filter</param-name>
        <param-value>
            (&amp;amp; (guid=%TERM%) (objectclass=cmuPerson))
        </param-value>
    </param>
    <param>
        <param-name>scope</param-name>
        <!--  Scope Values can be: OBJECT_SCOPE, ONELEVEL_SCOPE, SUBTREE_SCOPE  -->
        <param-value>
            ONELEVEL_SCOPE
        </param-value>
    </param>
    <param>
        <param-name>base</param-name>
        <param-value>
            ou=person
        </param-value>
    </param>

</search>
<search>
    <searchType>searchSubjectByIdentifier</searchType>
    <param>
        <param-name>filter</param-name>
        <param-value>
            (&amp;amp; (cmuAndrewId=%TERM%) (objectclass=cmuPerson))
        </param-value>
    </param>
    <param>
        <param-name>scope</param-name>
        <param-value>
            ONELEVEL_SCOPE
        </param-value>
    </param>
    <param>
        <param-name>base</param-name>
        <param-value>
            ou=person
        </param-value>
    </param>
</search>

<search>
   <searchType>search</searchType>
```

```
         <param>
            <param-name>filter</param-name>
            <param-value>
                (&amp;amp; (|(|(cmuAndrewId=%TERM%)(cn=*%TERM%*))(guid=%TERM%))(objectclass=cmuPerson))
            </param-value>
        </param>
        <param>
            <param-name>scope</param-name>
            <param-value>
                ONELEVEL_SCOPE
            </param-value>
        </param>
         <param>
            <param-name>base</param-name>
            <param-value>
                ou=person
            </param-value>
        </param>
    </search>
    <!-- you need this to be able to reference GrouperUtilElSafe in scripts -->
    <init-param>
      <param-name>subjectVirtualAttributeVariable_grouperUtilElSafe</param-name>
      <param-value>edu.internet2.middleware.grouper.util.GrouperUtilElSafe</param-value>
    </init-param>
    <!-- make sure subjectVirtualAttributeVariable_grouperUtilElSafe is set above -->
    <init-param>
      <param-name>subjectVirtualAttribute_0_nameLong</param-name>
      <param-value>${grouperUtilElSafe.appendIfNotBlankString(grouperUtilElSafe.defaultIfBlank(subject.
getAttributeValue('cn'), ''), ' - ', grouperUtilElSafe.defaultIfBlank(subject.getAttributeValue
('eduPersonSchoolCollegeName'), ''))}</param-value>
    </init-param>

    <init-param>
      <param-name>sortAttribute0</param-name>
      <param-value>nameLong</param-value>
    </init-param>
    <init-param>
      <param-name>searchAttribute0</param-name>
      <param-value>nameLong</param-value>
    </init-param>
    <internal-attribute>searchAttribute0</internal-attribute>
    <!-- ///Attributes you would like to display when doing a search  -->
    <attribute>eduPersonSchoolCollegeName</attribute>
    <attribute>sn</attribute>
    <attribute>cmuStudentClass</attribute>
    <attribute>givenName</attribute>
    <attribute>mail</attribute>

  </source>
```

this is the Java test case: LdapSubjectTest.java in Grouper API

```
Subject subject = SubjectFinder.findByIdAndSource("00000000-0000-1000-2F4C-0800207F02E6", "cmu", true);

assertEquals("Vincent Lun", subject.getName());

assertEquals("vlun@andrew.cmu.edu", subject.getAttributeValue("mail"));

assertEquals("Vincent Lun - Student Employment", subject.getDescription());

assertEquals("Vincent Lun - Student Employment", subject.getAttributeValue("nameLong"));

//check the search and sort attributes
Member member = MemberFinder.findBySubject(GrouperSession.startRootSession(), subject, true);
assertEquals("Vincent Lun - Student Employment", member.getSortString0());
assertEquals("vincent lun - student employment", member.getSearchString0());

Subject subject2 = SubjectFinder.findByIdentifierAndSource("vlun", "cmu", true);

assertEquals(subject.getId(), subject2.getId());
Set<Subject> subjects = SubjectFinder.findAll("Vincent Lun", "cmu");

//hmmm, will this be one?  maybe
assertEquals(1, GrouperUtil.length(subjects));
assertEquals(subject.getId(), subjects.iterator().next().getId());
```