Grouper UI - subject API diagnostics

The Subject API Diagnostics screen is an administrative screen for Grouper admins to troubleshoot and verify subject API sources.

This is on by default in a 2.3.0 patch or 2.3.1+.

Configure

Set this in the grouper-ui.properties to customize this enhancement

```
# should show subject api diagnostics?
uiV2.admin.subjectApiDiagnostics.show = true

# put in a group here if you want to allow the subject API diagnostics to certin users.
# note, admins can always see the screen
uiV2.admin.subjectApiDiagnostics.must.be.in.group =
```

Run from GSH

If the UI doesnt start

grouper.properties (temporarily)

```
gsh.exitOnSubjectCheckConfigProblem = false
```

if gsh doesnt start, try:

1. take source out of subject.properties
2. start gsh and get prompt
3. add source back in to subject.properties
4. run this in GSH

```
GrouperSession.startRootSession();
edu.internet2.middleware.grouper.cache.GrouperCacheUtils.clearAllCaches();
GrouperUtil.assignField(edu.internet2.middleware.subject.provider.SourceManager.class, null, "manager",
null, null);
-- continue below --
```

gsh

```
GrouperSession.startRootSession();
new edu.internet2.middleware.grouper.grouperUi.serviceLogic.SubjectSourceDiagnostics().assignSourceId
("SMUPerson_DEV").assignSubjectId("empl1").assignSubjectIdentifier("netid@school.edu").assignSearchString("em").
subjectSourceDiagnosticsFromGsh()
===>
SUCCESS: Found subject by id in 37ms: 'empl1'
         with SubjectFinder.findByIdAndSource("empl1", "SMUPerson_DEV", false)
SUCCESS: Subject id in returned subject matches the subject id searched for: 'empl1'
WARNING: No subject found by identifier in 14ms: 'netid@school.edu'
         with SubjectFinder.findByIdentifierAndSource("netid@school.edu", "SMUPerson_DEV", false)
```

Run

Go to the New UI, click on Miscellaneous, then click on "Subject API diagnostics"

Try

If you have an account on the demo server you can see the diagnostics there:

https://grouperdemo.internet2.edu/grouper_v2_3/grouperUi/app/UiV2Main.index?operation=UiV2Admin.subjectApiDiagnostics

Diagnostics

Subject API diagnostics

Source ID: Example JDBC Source Adapter

The Source ID is configured in the subject.properties file. Some sources are internal for Grouper and one or more are custom for your institution. The fewer the better.

Subject ID: test.subject.0

Enter a Subject ID to search for. This should be an unchanging opaque identifier. This defaults to the subject.properties source param 'subjectIdToFindOnCheckConfig' or to a hardcoded default value if not configured, which is used in the **status servlet**.

Subject Identifier: id.test.subject.0

Enter a Subject Identifier to search for. This is any identifier that uniquely identifies this subject in this source. e.g. netId, eppn, etc. It is nice if identifiers are unique across sources, which defaults to the subject.properties source param 'subjectIdentifierToFindOnCheckConfig' or to a hardcoded default value if not configured. This is used in the **status servlet**.

Search String: te

Enter a Search String to search for. This usually searches based on name, id, identifier, description, etc. This defaults to the subject.properties source param 'stringToFindOnCheckConfig' or to a hardcoded default value if not configured, which is used in the **status servlet**.

Act as:

Act as another user. Some subject sources use the current user for privileges (e.g. searching for groups). Search for a user or enter a subject id or subject identifier (must be unique across sources if entering id or identifier).

Run diagnostics | Cancel

```
SUCCESS: Found subject by id in 0ms: 'test.subject.0'
         with SubjectFinder.findByIdAndSource("test.subject.0", "jdbc", false)
SUCCESS: Subject id in returned subject matches the subject id searched for: 'test.subject.0'
SUCCESS: Found subject by identifier in 4ms: 'id.test.subject.0'
         with SubjectFinder.findByIdentifierAndSource("id.test.subject.0", "jdbc", false)
SUCCESS: Found 10 subjects by search string in 0ms: 'te'
         with SubjectFinder.findAll("te", "jdbc")
SUCCESS: Found 10 subjects by paged search string in 0ms: 'te'
         with SubjectFinder.findPage("te", "jdbc")

######## SUBJECT ATTRIBUTES ########

Subject id: test.subject.0 with subject.getId()
  - the subject id should be an unchanging opaque identifier
  - the subject id is stored in the grouper_members table
Subject name: my name is test.subject.0 with subject.getName()
  - the subject name is generally first last
Subject description: description.test.subject.0 with subject.getDescription()
  - the subject description can have more info such as the id, name, dept, etc
Subject type: person with subject.getTypeName()
  - the subject type is not really used
```

```
Subject name: my name is test.subject.0 with subject.getName()
  - the subject name is generally first last
Subject description: description.test.subject.0 with subject.getDescription()
  - the subject description can have more info such as the id, name, dept, etc
Subject type: person with subject.getTypeName()
  - the subject type is not really used
Subject attribute 'subjectid' has 1 value: 'test.subject.0'
  - with subject.getAttributeValue("subjectid")
Subject attribute 'name' has 1 value: 'my name is test.subject.0'
  - with subject.getAttributeValue("name")
Subject attribute 'lfname' has 1 value: 'name.test.subject.0'
  - with subject.getAttributeValue("lfname")
Subject attribute 'loginid' has 1 value: 'id.test.subject.0'
  - with subject.getAttributeValue("loginid")
Subject attribute 'description' has 1 value: 'description.test.subject.0'
  - with subject.getAttributeValue("description")
Subject attribute 'email' has 1 value: 'test.subject.0@somewhere.someSchool.edu'
  - with subject.getAttributeValue("email")

######## SUBJECT IN UI ########

Short link with icon: my name is test.subject.0
  - This is configured in grouper.text.en.us.base.properties with guiSubjectShortLink
  - Also configured in grouper-ui.properties with grouperUi.screenLabel2.sourceId.X
  - By default this is the name of the subject with a tooltip for description
Long label with icon: description.test.subject.0
  - This is not used in the new UI
  - It is configured in grouper-ui.properties with grouperUi.subjectImg.screenEl.
  - By default this is the description of the subject

######## SUBJECT IN WS ########

Look in grouper-ws.properties to see how the WS uses this subject.  This is the default configuation:

# subject attribute names to send back when a WsSubjectResult is sent, comma separated
# e.g. name, netid
# default is none
ws.subject.result.attribute.names =

# subject result attribute names when extended data is requested (comma separated)
# default is name, description
# note, these will be in addition to ws.subject.result.attribute.names
ws.subject.result.detail.attribute.names =

######## SOURCE CONFIGURATION ########

Adapter class: 'edu.internet2.middleware.grouper.subj.GrouperJdbcSourceAdapter'
  - configured in subject.properties: subjectApi.source.jdbc.adapterClass
SUCCESS: Found adapter class
SUCCESS: Instantiated adapter class
Source id: 'jdbc'
  - configured in subject.properties: subjectApi.source.jdbc.id
Source name: 'Example JDBC Source Adapter'
  - configured in subject.properties: subjectApi.source.jdbc.name
Source types: 'person'
  - configured in subject.properties: subjectApi.source.jdbc.types
Source param name: 'Name_AttributeType' has value: 'name'
```

```
  - configured in subject.properties: subjectApi.source.jdbc.search.searchSubjectByIdentifier.param.inclause.value
Search 'search' param name: 'sql' has value: 'select    s.subjectid as id, s.name as name,    (select sa2.value from s
ubjectattribute sa2 where name='name' and sa2.SUBJECTID = s.subjectid) as lfname,    (select sa3.value from subjectatt
ribute sa3 where name='loginid' and sa3.SUBJECTID = s.subjectid) as loginid,    (select sa4.value from subjectattribut
e sa4 where name='description' and sa4.SUBJECTID = s.subjectid) as description,    (select sa5.value from subjectattri
bute sa5 where name='email' and sa5.SUBJECTID = s.subjectid) as email from    subject s where    s.subjectid in (
  select subjectid from subject where lower(name) like concat('%',concat(?,'%')) union       select subjectid from sub
jectattribute where searchvalue like concat('%',concat(?,'%'))    )'
  - configured in subject.properties: subjectApi.source.jdbc.search.search.param.sql.value

######## SUBJECT SEARCH RESULTS ########

Subject 0: id: test.subject.0, name: my name is test.subject.0
  - description: description.test.subject.0
Subject 1: id: test.subject.1, name: my name is test.subject.1
  - description: description.test.subject.1
Subject 2: id: test.subject.2, name: my name is test.subject.2
  - description: description.test.subject.2
Subject 3: id: test.subject.3, name: my name is test.subject.3
  - description: description.test.subject.3
Subject 4: id: test.subject.4, name: my name is test.subject.4
  - description: description.test.subject.4
Subject 5: id: test.subject.5, name: my name is test.subject.5
  - description: description.test.subject.5
Subject 6: id: test.subject.6, name: my name is test.subject.6
  - description: description.test.subject.6
Subject 7: id: test.subject.7, name: my name is test.subject.7
  - description: description.test.subject.7
Subject 8: id: test.subject.8, name: my name is test.subject.8
  - description: description.test.subject.8
Subject 9: id: test.subject.9, name: my name is test.subject.9
  - description: description.test.subject.9

######## SUBJECT PAGE RESULTS ########

Subject 0: id: test.subject.0, name: my name is test.subject.0
  - description: description.test.subject.0
Subject 1: id: test.subject.1, name: my name is test.subject.1
  - description: description.test.subject.1
Subject 2: id: test.subject.2, name: my name is test.subject.2
  - description: description.test.subject.2
Subject 3: id: test.subject.3, name: my name is test.subject.3
  - description: description.test.subject.3
Subject 4: id: test.subject.4, name: my name is test.subject.4
  - description: description.test.subject.4
Subject 5: id: test.subject.5, name: my name is test.subject.5
  - description: description.test.subject.5
Subject 6: id: test.subject.6, name: my name is test.subject.6
  - description: description.test.subject.6
Subject 7: id: test.subject.7, name: my name is test.subject.7
  - description: description.test.subject.7
Subject 8: id: test.subject.8, name: my name is test.subject.8
  - description: description.test.subject.8
Subject 9: id: test.subject.9, name: my name is test.subject.9
  - description: description.test.subject.9
```

Note: email attribute is not validated

```
SUCCESS: The emailAttributeName is configured to be: 'mail'
SUCCESS: The email address 'whatever@someplace.edu' was found and has a valid format
```

See Also

Subject API