Grouper Web Services FAQ

Grouper Web Services

- Subject attributes can be exposed via web service
  - First off, you need to have the subject api configuration to get attribute from the DB / LDAP
    - Here is an example from a JDBC2 subject source

```
# now you can count up from 0 to N of attributes for various cols.  The name is how to
reference in subject.getAttribute()
subjectApi.source.pennperson.param.subjectAttributeCol0.value = pennname

# you can count up from 0 to N of attributes for various cols.  The name is how to
reference in subject.getAttribute()
subjectApi.source.pennperson.param.subjectAttributeName0.value = PENNNAME
```

    - You can test this from GSH

```
groovy:000> grouperSession = GrouperSession.startRootSession();
===> 4a29e140d3b346558c7c23ed996d1d8a,'GrouperSystem','application'
groovy:000> subject = SubjectFinder.findByIdOrIdentifier("mchyzer", true);
===> Subject id: 10021368, sourceId: pennperson, name: Chris Hyzer
groovy:000> subject.getAttributeValue("pennname")
===> mchyzer
```

  - Subject attributes are exposed via web service in the grouper-ws.properties

```
# 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 = PENNNAME
```

  - Then when a subject is returned, since ws.subject.result.attribute.names is blank, no custom attribute are returned by default

```
[mchyzer@flash pennGroupsClient-2.4.0]$ java -jar grouperClient.jar --operation=getSubjectsWs --
subjectIdentifiers=mchyzer --debug=true
Reading resource: grouper.client.properties, from: /home/mchyzer/grouper/pennGroupsClient-2.4.0
/grouper.client.properties
WebService: connecting as user: 'someuser'
WebService: connecting to URL: 'https://server.school.edu/grouperWs/servicesRest/v2_4_000/subjects'

################ REQUEST START (indented) ###############
```

```
POST /grouperWs/servicesRest/v2_4_000/subjects HTTP/1.1
Connection: close
Authorization: Basic xxxxxxxxxxxxxxx
User-Agent: Jakarta Commons-HttpClient/3.1
Host: somehost.school.edu:-1
Content-Length: 171
Content-Type: text/xml; charset=UTF-8

<WsRestGetSubjectsRequest>
  <wsSubjectLookups>
    <WsSubjectLookup>
      <subjectIdentifier>mchyzer</subjectIdentifier>
    </WsSubjectLookup>
  </wsSubjectLookups>
</WsRestGetSubjectsRequest>

################ REQUEST END ###############



################ RESPONSE START (indented) ###############

HTTP/1.1 200 OK
Date: Wed, 17 Apr 2019 18:12:37 GMT
Set-Cookie: JSESSIONID=xxxxxxxxxxx; HttpOnly
X-Grouper-resultCode: SUCCESS
X-Grouper-success: T
X-Grouper-resultCode2: NONE
Content-Type: application/xml;charset=UTF-8
Vary: Accept-Encoding
Connection: close
Transfer-Encoding: chunked

<WsGetSubjectsResults>
  <wsSubjects>
    <WsSubject>
      <identifierLookup>mchyzer</identifierLookup>
      <resultCode>SUCCESS</resultCode>
      <success>T</success>
      <id>10021368</id>
      <name>Chris Hyzer</name>
      <sourceId>pennperson</sourceId>
    </WsSubject>
  </wsSubjects>
  <resultMetadata>
    <resultCode>SUCCESS</resultCode>
    <resultMessage>Queried 1 subjects</resultMessage>
    <success>T</success>
  </resultMetadata>
  <responseMetadata>
    <resultWarnings></resultWarnings>
    <millis>11</millis>
    <serverVersion>2.4.0</serverVersion>
  </responseMetadata>
</WsGetSubjectsResults>

################ RESPONSE END ###############


Output template: Index: ${index}: success: ${success}, code: ${wsSubject.resultCode}, subject:
${wsSubject.id}, available variables: wsGetSubjectsResults, grouperClientUtils, index, wsSubject,
wsGroup, success
Index: 0: success: T, code: SUCCESS, subject: 10021368
Elapsed time: 957ms
[mchyzer@flash pennGroupsClient-2.4.0]$
```

- If you request the subject detail (attrs in ws.subject.result.detail.attribute.name), then they will be returned

```
[mchyzer@flash pennGroupsClient-2.4.0]$ java -jar grouperClient.jar --operation=getSubjectsWs --
subjectIdentifiers=mchyzer --debug=true --includeSubjectDetail=true
Reading resource: grouper.client.properties, from: /home/mchyzer/grouper/pennGroupsClient-2.4.0
/grouper.client.properties
WebService: connecting as user: 'user'
WebService: connecting to URL: 'https://server.school.edu/grouperWs/servicesRest/v2_4_000/subjects'

################ REQUEST START (indented) ###############

POST /grouperWs/servicesRest/v2_4_000/subjects HTTP/1.1
Connection: close
Authorization: Basic xxxxxxxxxxxxxxx
User-Agent: Jakarta Commons-HttpClient/3.1
Host: fastprod-medium-a-01.apps.upenn.edu:-1
Content-Length: 217
Content-Type: text/xml; charset=UTF-8

<WsRestGetSubjectsRequest>
  <includeSubjectDetail>T</includeSubjectDetail>
  <wsSubjectLookups>
    <WsSubjectLookup>
      <subjectIdentifier>mchyzer</subjectIdentifier>
    </WsSubjectLookup>
  </wsSubjectLookups>
</WsRestGetSubjectsRequest>

################ REQUEST END ###############



################ RESPONSE START (indented) ###############

HTTP/1.1 200 OK
Date: Wed, 17 Apr 2019 18:16:55 GMT
Set-Cookie: JSESSIONID=xxxxxxxxxxx; HttpOnly
X-Grouper-resultCode: SUCCESS
X-Grouper-success: T
X-Grouper-resultCode2: NONE
Content-Type: application/xml;charset=UTF-8
Vary: Accept-Encoding
Connection: close
Transfer-Encoding: chunked

<WsGetSubjectsResults>
  <subjectAttributeNames>
    <string>name</string>
    <string>description</string>
    <string>PENNNAME</string>
    <string>EMAIL</string>
  </subjectAttributeNames>
  <wsSubjects>
    <WsSubject>
      <identifierLookup>mchyzer</identifierLookup>
      <resultCode>SUCCESS</resultCode>
      <success>T</success>
      <id>10021368</id>
      <name>Chris Hyzer</name>
      <sourceId>pennperson</sourceId>
      <attributeValues>
        <string>Chris Hyzer</string>
        <string>Chris Hyzer (mchyzer, 10021368) (active) Staff - Isc-applications & Information
Services - Application Architect (also: Alumni)</string>
        <string>mchyzer</string>
        <string>mchyzer@upenn.edu</string>
      </attributeValues>
    </WsSubject>
  </wsSubjects>
  <resultMetadata>
    <resultCode>SUCCESS</resultCode>
    <resultMessage>Queried 1 subjects</resultMessage>
    <success>T</success>
```

```
  </resultMetadata>
  <responseMetadata>
    <resultWarnings></resultWarnings>
    <millis>121</millis>
    <serverVersion>2.4.0</serverVersion>
  </responseMetadata>
</WsGetSubjectsResults>

################ RESPONSE END ###############


Output template: Index: ${index}: success: ${success}, code: ${wsSubject.resultCode}, subject:
${wsSubject.id}, available variables: wsGetSubjectsResults, grouperClientUtils, index, wsSubject,
wsGroup, success
Index: 0: success: T, code: SUCCESS, subject: 10021368
Elapsed time: 1106ms
[mchyzer@flash pennGroupsClient-2.4.0]$
```

- Or you can request individual attributes

```
[mchyzer@flash pennGroupsClient-2.4.0]$ java -jar grouperClient.jar --operation=getSubjectsWs --
subjectIdentifiers=mchyzer --debug=true --subjectAttributeNames=PENNNAME
Reading resource: grouper.client.properties, from: /home/mchyzer/grouper/pennGroupsClient-2.4.0
/grouper.client.properties
WebService: connecting as user: 'user'
WebService: connecting to URL: 'https://server.school.edu/grouperWs/servicesRest/v2_4_000/subjects'

################ REQUEST START (indented) ###############

POST /grouperWs/servicesRest/v2_4_000/subjects HTTP/1.1
Connection: close
Authorization: Basic xxxxxxxxxxxxxxx
User-Agent: Jakarta Commons-HttpClient/3.1
Host: fastprod-medium-a-01.apps.upenn.edu:-1
Content-Length: 243
Content-Type: text/xml; charset=UTF-8

<WsRestGetSubjectsRequest>
  <subjectAttributeNames>
    <string>PENNNAME</string>
  </subjectAttributeNames>
  <wsSubjectLookups>
    <WsSubjectLookup>
      <subjectIdentifier>mchyzer</subjectIdentifier>
    </WsSubjectLookup>
  </wsSubjectLookups>
</WsRestGetSubjectsRequest>

################ REQUEST END ###############



################ RESPONSE START (indented) ###############

HTTP/1.1 200 OK
Date: Wed, 17 Apr 2019 18:26:29 GMT
Set-Cookie: JSESSIONID=xxxxxxxxxxxx; HttpOnly
X-Grouper-resultCode: SUCCESS
X-Grouper-success: T
X-Grouper-resultCode2: NONE
Content-Type: application/xml;charset=UTF-8
Vary: Accept-Encoding
Connection: close
Transfer-Encoding: chunked

<WsGetSubjectsResults>
  <subjectAttributeNames>
    <string>PENNNAME</string>
```

```
  </subjectAttributeNames>
  <wsSubjects>
    <WsSubject>
      <identifierLookup>mchyzer</identifierLookup>
      <resultCode>SUCCESS</resultCode>
      <success>T</success>
      <id>10021368</id>
      <name>Chris Hyzer</name>
      <sourceId>pennperson</sourceId>
      <attributeValues>
        <string>mchyzer</string>
      </attributeValues>
    </WsSubject>
  </wsSubjects>
  <resultMetadata>
    <resultCode>SUCCESS</resultCode>
    <resultMessage>Queried 1 subjects</resultMessage>
    <success>T</success>
  </resultMetadata>
  <responseMetadata>
    <resultWarnings></resultWarnings>
    <millis>207</millis>
    <serverVersion>2.4.0</serverVersion>
  </responseMetadata>
</WsGetSubjectsResults>

################ RESPONSE END ###############


Output template: Index: ${index}: success: ${success}, code: ${wsSubject.resultCode}, subject:
${wsSubject.id}, available variables: wsGetSubjectsResults, grouperClientUtils, index, wsSubject,
wsGroup, success
Index: 0: success: T, code: SUCCESS, subject: 10021368
Elapsed time: 1180ms
[mchyzer@flash pennGroupsClient-2.4.0]$
```

    - dsf

- Which SOAP call should I use for getting the list of groups that group A is a member of?

> *To get a list of groups that A is a member of, call getGroups, there is a soap example there. Note: just use g:gsa as the subject source, and the group name as the subjectIdentifier, or the group uuid as the subjectId. You can change the query to ask about immediate, effective, nonimmediate, etc. I think you can get similar information from getMemberships (using the same subject call as described above)*

- Can I run grouper web services against any version of Grouper?

> *No, you should use the version of Grouper bundled in the web services. This makes it a little complicated if you have a UI / GSH / etc running against your Grouper DB. You will need to keep them in sync...*

- Why are there three flavors or web services (SOAP, XML-HTTP, REST-Lite)?

> *SOAP is supported since many schools required it. REST-Lite is supported because many schools required it. XML-HTTP is supported because Axis2 gives it for free when building a SOAP web service. This way the same SOAP interface can be accessed by clients who only want to talk HTTP and XML and not SOAP.*

- Why is the rampart .aar file named GrouperServiceWssec.aar, but the URL is still /services/GrouperService?

> *The URL for ramprt or not is the same. Inside the services.xml in the .aar files, it configures the app name. grouper-ws will not work if you change this (unless to do other build activities also)*

- Why is there a "simple" operation for every non simple operation in SOAP and XML-HTTP?

The non-simple operations are batchable if the client wants to do one operation multiple times with one request. The simple operations are for if the client only needs to do one thing (e.g. assign a member to a group and not assign multiple members to a group). Also, there is a valuable side effect that for the XML-HTTP, if the operation only has scaler input params (and not complex types or arrays), that the input does not need XML, it can be in the query string for GET or in the http form param pairs in the body for POST.

It is confusing that there are so many different ways to call grouper via web service. The documentation will be improved to make it easier to find the best strategy.

- Why element named "return" (by axis)

This is an unfortunate "feature" by axis, the default element is named "return" which is a keyword in many programming languages, so if the language automatically converts the xml to an object graph, then it will be broken. Chris will followup with Axis to see if there is a fix for this

- Why returning error codes and messages and not just use SOAP faults?

For two of three of the flavors of web services SOAP faults are not an option since they are not SOAP. Also, for SOAP batched, the status of each line item needs to be returned for the client to process, and a SOAP fault would preclude that. Also, the fewer SOAP specific features that are used, the more widespread the compatibility will be. All three flavors of web service use the same underlying logic, so the more consistent the better.

- Can we add a service for subject search?

Yes, this will be added