Group Save

Grouper Web Services

Description

Group save will insert or update a group's uuid, extension, display name, or description (with restrictions). Note: the group displayName and extension are not used in a groupSave. That information is used from the group name, and displayExtension. Note, when you call a groupSave, You need to send in all the attributes. Might want to get the group, edit what you want, and send back.

Features

- Can pass SaveMode which is INSERT, UPDATE, or INSERT_OR_UPDATE (default)
- If the stem doesnt exist, the call will fail
- Lookup group to edit by group lookup (by name or uuid)
- Returns group, can be detailed or not
- Can actAs another user
- In version 2.1 and later, you can pass in typeOfGroup to create roles or entities
- In version 2.2.1.patch+ you can move or copy groups (see below)
- In version 2.3.0 with patches, you have the option to not set the alternateName during a group rename (see below)

Group save Lite service

- Accepts one group to save
- Documentation: SOAP (click on groupSaveLite), REST (click on groupSaveLite)
- For REST, the request can put data in query string (in URL or request body)
- REST request (colon is escaped to %3A): PUT /grouper-ws/servicesRest/v1_3_000/groups/aStem%3AaGroup
- (see documentation above for details): Request object, response object
- Response codes
- Samples (all files with "Lite" in them, click on "download" to see file)

Group save service

- Accepts multiple groups to save
- This will persist (insert/update/delete) types, attributes, composites from detail
- Documentation: SOAP (click on groupSave), REST (click on groupSave)
- REST request (colon is escaped to %3A): PUT /grouper-ws/servicesRest/v1_3_000/groups
- (see documentation above for details): Request object, response object
- Response codes overall, response codes for each assignment
- Returns an overall status, and a status for each assignment
- Samples (all files without "Lite" in them, click on "download" to see files)

FAQ

- **How can I make a group which has a manual membership list and requires users to be faculty student or staff?**
  First off, you need permission to view the facultyStudentStaff group, if it is not public. Note, the composite arguments shouldnt be necessary, but until it is fixed, use them and it will work. This makes a group, a system of record group (where the manual entries go), and the overall group is a composite intersection of the manual group and the facultyStudentStaff group.What does that look like in a soap request? (note, fields which arent used need to be there, due to axis bug. Note you need to enable "requireGroups" in your grouper.properties

```
<?xml version='1.0' encoding='UTF-8'?>
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope">
  <soapenv:Body>
    <ns1:groupSave xmlns:ns1="http://soap.ws.grouper.middleware.internet2.edu/xsd">
      <ns1:clientVersion>v1_4_002</ns1:clientVersion>
      <ns1:wsGroupToSaves>
        <ns1:wsGroup>
          <ns1:description>
            test group with requiring active facultyStudentStaff
          </ns1:description>
          <ns1:detail>
            <ns1:attributeNames>requireAlsoInGroups</ns1:attributeNames>
            <ns1:attributeValues>penn:community:facultyStudentStaff</ns1:attributeValues>
            <ns1:compositeType>intersection</ns1:compositeType>
            <ns1:hasComposite>T</ns1:hasComposite>
            <ns1:leftGroup>
              <ns1:description></ns1:description>
              <ns1:displayExtension></ns1:displayExtension>
              <ns1:displayName></ns1:displayName>
              <ns1:extension></ns1:extension>
              <ns1:name>penn:community:facultyStudentStaff</ns1:name>
              <ns1:uuid></ns1:uuid>
            </ns1:leftGroup>
            <ns1:rightGroup>
              <ns1:description></ns1:description>
              <ns1:displayExtension></ns1:displayExtension>
              <ns1:displayName></ns1:displayName>
              <ns1:extension></ns1:extension>
              <ns1:name>test:isc:astt:chris:myGroup_systemOfRecord</ns1:name>
              <ns1:uuid></ns1:uuid>
            </ns1:rightGroup>
            <ns1:typeNames>requireInGroups</ns1:typeNames>
          </ns1:detail>
          <ns1:displayExtension>My test group</ns1:displayExtension>
          <ns1:extension>myGroup</ns1:extension>
          <ns1:name>test:isc:astt:chris:myGroup</ns1:name>
        </ns1:wsGroup>
        <ns1:wsGroupLookup>
          <ns1:groupName>test:isc:astt:chris:myGroup</ns1:groupName>
        </ns1:wsGroupLookup>
      </ns1:wsGroupToSaves>
      <ns1:actAsSubjectLookup>
        <ns1:subjectId></ns1:subjectId>
      </ns1:actAsSubjectLookup>
      <ns1:txType></ns1:txType>
      <ns1:includeGroupDetail>T</ns1:includeGroupDetail>
    </ns1:groupSave>
  </soapenv:Body>
</soapenv:Envelope>
```

Move or copy group

Send in a wsGroupLookup, and no wsGroup. This will return SUCCESS_INSERTED if successful

Note, the documentation below is copied from the move/copy documentation which is most of up to date

Params (note param indexes passed in should be sequential starting with 0)

This will return the destination stem with the extension of the Group (which may or may not be the group that was copied)

Param	Value	Notes
[--paramName0=moveOrCopy] [--paramValue0=copy]	move|copy	
[--paramName1=moveOrCopyToStemUuid] [--paramValue1=abc123]	uuid of a stem to move or copy to	mutually exclusive with moveOrCopyStemName and moveOrCopyToStemIdIndex

[--paramName2=moveOrCopyToStemName] [--paramValue2=a:b:c]	name of stem to move or copy to	mutually exclusive with moveOrCopyStemUuid and moveOrCopyToStemIdIndex
[--paramName3=moveOrCopyToStemIdIndex] [--paramValue3=19]	id index of stem to move or copy to	mutually exclusive with moveOrCopyStemUuid and moveOrCopyStemName
[--paramName4=copyPrivilegesOfGroup] [--paramValue4=true]	true|false	Whether to copy the access privileges of the group. If this option is selected, you must have READ access to all privileges.
[--paramName5=copyGroupAsPrivilege] [--paramValue5=true]	true|false	Whether to copy access and naming privileges where the group is a member. For instance, if you are copying Group X and Group X has admin privileges to Group Y, then if this option is enabled, after Group X is copied, the new group will also have admin privileges to Group Y. If this option is selected, you must have access to add privileges to the other groups and folders.
[--paramName6=copyListMembersOfGroup] [--paramValue6=true]	true|false	Whether to copy list memberships of the group. If this option is selected and this group has custom lists, you must have read access to them.
[--paramName7=copyListGroupAsMember] [--paramValue7=true]	true|false	Whether to copy list memberships where the group is a member. For instance, if you are copying Group X and Group X is a member of Group Y, then if this option is enabled, after Group X is copied, the new group will also be a member of Group Y. If this option is selected, you must have access to add memberships to the other groups.
[--paramName8=copyAttributes] [--paramValue8=true]	true|false	Whether to copy attributes. If this option is selected, you must have READ access to all attributes. These are attributes that are added to GroupTypes. This does not include attributes in the new attribute framework in v1.5.0.
[--paramName9=moveAssignAlternateName] [--paramValue9=true]	true|false	Whether to assign the old name of the group as an alternate name of the group after the move. This allows API methods like GroupFinder.findByName() to find the group using the old and new names and can make it easier to transition from the old name to the new name.

Example

```
<WsRestGroupSaveRequest>
  <wsGroupToSaves>
    <WsGroupToSave>
      <wsGroupLookup>
        <groupName>test:testGroupToCopy</groupName>
      </wsGroupLookup>
      <wsGroup>
        <displayExtension>testGroupToCopy</displayExtension>
        <name>test:testGroupToCopy</name>
      </wsGroup>
    </WsGroupToSave>
  </wsGroupToSaves>
  <params>
    <WsParam>
      <paramName>moveOrCopy</paramName>
      <paramValue>copy</paramValue>
    </WsParam>
    <WsParam>
      <paramName>moveOrCopyToStemName</paramName>
      <paramValue>test:stemDestination</paramValue>
    </WsParam>
    <WsParam>
      <paramName>copyPrivilegesOfGroup</paramName>
      <paramValue>true</paramValue>
    </WsParam>
    <WsParam>
      <paramName>copyGroupAsPrivilege</paramName>
      <paramValue>true</paramValue>
    </WsParam>
    <WsParam>
      <paramName>copyListMembersOfGroup</paramName>
      <paramValue>true</paramValue>
    </WsParam>
    <WsParam>
      <paramName>copyListGroupAsMember</paramName>
      <paramValue>true</paramValue>
    </WsParam>
    <WsParam>
      <paramName>copyAttributes</paramName>
      <paramValue>true</paramValue>
    </WsParam>
  </params>
</WsRestGroupSaveRequest>
```

Rename group

You have the option to not set the alternateName during a group rename by using the param renameAssignAlternateName (true/false)

```
<WsRestGroupSaveRequest>
   <wsGroupToSaves>
      <WsGroupToSave>
         <wsGroupLookup>
            <groupName>d:oldName</groupName>
         </wsGroupLookup>
         <wsGroup>
            <name>d:newName</name>
         </wsGroup>
      </WsGroupToSave>
   </wsGroupToSaves>
   <params>
      <WsParam>
         <paramName>renameAssignAlternateName</paramName>
         <paramValue>false</paramValue>
      </WsParam>
   </params>
</WsRestGroupSaveRequest>
```

sdf