Include exclude and require groups

This topic is discussed in the "How to Design Groups" training video.

Grouper features optional support for types, attributes, and hooks which facilitate easily setting up an include and exclude membership list, and being able to require memberships in other groups to enable a membership in a group.

Using

To enable this feature, set the two settings to true in the grouper.properties config:

```java
# if the addIncludeExclude and requireInGroups should be enabled, and if the type(s) should be
# auto-created, and used to auto create groups to facilitate include and exclude lists, and require lists
grouperIncludeExclude.use = false
#requireInGroups.use = false
```

Decide which groups should be built-in types or attributes for "require groups". If it is a type, it should probably be global so that most users can benefit from it (e.g. activeEmployee). If it is an attribute, it is a little more hidden. If not neither, users can still enter in group names in the requireGroups freeform attribute:

```java
# for requireGroups (groups that the members must be to be in the overall group). name is the name of the
# attribute or type
#attributeOrType is either attribute for an attribute underneath the requireInGroups type, or type to be a top
#level type
#group is the group to be anded in. note attributes are a global namespace, so you might want to use a naming
#convention, 
#e.g. prefix with "require". description is the tooltip. add as many as you like.
#grouperIncludeExclude.requireGroup.name.0 = requireActiveEmployee
#grouperIncludeExclude.requireGroup.attributeOrType.0 = type
#grouperIncludeExclude.requireGroup.group.0 = school:community:activeEmployee
#grouperIncludeExclude.requireGroup.description.0 = If value is true, members of the overall group must be an
#active employee (in the school:community:activeEmployee group). Otherwise, leave this value not filled in.
#grouperIncludeExclude.requireGroup.name.1 = requireActiveStudent
#grouperIncludeExclude.requireGroup.attributeOrType.1 = attribute
#grouperIncludeExclude.requireGroup.group.1 = school:community:activeStudent
#grouperIncludeExclude.requireGroup.description.1 = If value is true, members of the overall group must be an
#active student (in the school:community:activeStudent group). Otherwise leave this value not filled in.
```

Screenshots

Here is a stem:
You can look for groups throughout the hierarchy. (You might not be able to see some groups if you lack appropriate privileges.)

Current location is:

![Root: aStem](image)

Shoving 1-5 of 5 items

- activeEmployee
- activeStudent
- aGroup
- anotherGroup
- yetAnotherGroup

Search from Root

Search groups

Include and exclude tooltip

Current location is:

![Root: aStem: aGroup](image)

Name: aGroup

ID: aGroup

Description: Group containing list of aGroup after adding the include

Assign privileges to everyone

Select group types

- addInclude
- exclude
- requireActiveStudent
- requireInGroups

Save

Back to group summary
Custom require group type tooltip (configured in grouper.properties)

Require in groups tooltip (this group type holds custom attributes for require groups, also a freeform attribute):

Groups created when include exclude is selected
Selected a require group would look like this:

(You might not be able to see some groups if you lack appropriate privileges.)

This is the composite membership of the overall group when a require group is selected.
This is how the require in groups attributes appear:

**Group summary**

Note: The group [aGroup] was successfully saved

<table>
<thead>
<tr>
<th>Name</th>
<th>aGroup</th>
</tr>
</thead>
<tbody>
<tr>
<td>Path</td>
<td>aStem:aGroup</td>
</tr>
<tr>
<td>Description</td>
<td>Group containing list of aGroup after adding the includes and subtracting the excludes</td>
</tr>
<tr>
<td>ID</td>
<td>aGroup</td>
</tr>
<tr>
<td>TID Path</td>
<td>aStem:aGroup</td>
</tr>
<tr>
<td>UUID</td>
<td>088801f38-d76f-4d6f-b9eb-8301544f704</td>
</tr>
<tr>
<td>Types</td>
<td>requireActiveStudent, requireInGroups, requireActiveEmployee, requireAlsoInGroups, aStem:anotherGroup, aStem:yetAnotherGroup</td>
</tr>
</tbody>
</table>

Here are the groups created with multiple require groups:
Here is a require group membership, based on another require group

Members

Current location is:
Root: aStem

Membership list
- Show DIRECT members of this group
- Show INDIRECT members of this group
- Show ALL members of this group (direct and indirect)

Change display

Showing 1-1 of 1 items
Click an entity name to view entity details, or click a membership descriptor:

- This is a composite group
  - aStem:aGroup requireGroups 3 intersection
  - aStem:activeStudent
Here is a group with an include and exclude list, and a bunch of require groups, all chained together.

You can look for groups throughout the hierarchy. (You might not be able to see some groups if you lack appropriate privileges.)

**Browse or list groups**

**Current location is:**

- **Root:** aStem

Showing 1-13 of 13 items

- activeEmployee
- activeStudent
- aGroup
- aGroup excludes
- aGroup includes
- aGroup includes minus excludes
- aGroup requireGroups 1
- aGroup requireGroups 2
- aGroup requireGroups 3
- aGroup system of record
- aGroup system of record and includes
- anotherGroup
- yetAnotherGroup

---

### Configuration

#### Grouper include / exclude and requireGroups

- **use** = false

- **requireGroups**
  - **use** = false

- **requireGroup**
  - name = requireActiveEmployee
  - attributeOrType = type
  - group = school:community:activeEmployee
  - description = If value is true, members of the overall group must be an active employee (in the school:community:activeEmployee group). Otherwise, leave this value not filled in.
# grouperIncludeExclude.requireGroup.name.1 = requireActiveStudent
# grouperIncludeExclude.requireGroup.attributeOrType.1 = attribute
# grouperIncludeExclude.requireGroup.group.1 = school:community:activeStudent
# grouperIncludeExclude.requireGroup.description.1 = If value is true, members of the overall group must be an active student (in the school:community:activeStudent group). Otherwise leave this value not filled in.

# set some names and tooltips
grouperIncludeExclude.type.name = addIncludeExclude
grouperIncludeExclude.tooltip = Select this type to auto-create other groups which facilitate having include and exclude list

# grouperIncludeExclude.requireGroups.type.name = requireInGroups
grouperIncludeExclude.requireGroups.tooltip = Select this type to auto-create other groups which set up group math so that other groups can be required for membership (e.g. activeEmployee)

# leave grouperIncludeExclude.andGroups.attributeName blank if you dont want to use this attribute...
# though if you were using it, it wont remove already configured groups
grouperIncludeExclude.requireGroups.attributeName = requireAlsoInGroups
grouperIncludeExclude.requireGroups.attribute.tooltip = Enter in comma separated group path(s). An entity must be in these groups for it to be in the overall group. e.g. stem1:stem2:group1, stem1:stem3:group2

# suffixes for various include/exclude groups (can use "space" for space).
# these should uniquely identify various parts of the include/exclude.
# i.e. if the grouperIncludeExclude type is applied to a group with a suffix of the include suffix, the other groups will not be created...
grouperIncludeExclude.systemOfRecord.extension.suffix = _systemOfRecord
grouperIncludeExclude.include.extension.suffix = _includes
grouperIncludeExclude.exclude.extension.suffix = _excludes
grouperIncludeExclude.systemOfRecordAndIncludes.extension.suffix = _systemOfRecordAndIncludes
grouperIncludeExclude.includesMinusExcludes.extension.suffix = _includesMinusExcludes

# suffixes for various include/exclude groups (can use "space" for space).
# these should uniquely identify various parts of the include/exclude.
# i.e. if the grouperIncludeExclude type is applied to a group with a suffix of the include suffix, the other groups will not be created...
grouperIncludeExclude.systemOfRecord.displayExtension.suffix = ${space}system of record
grouperIncludeExclude.include.displayExtension.suffix = ${space}includes
grouperIncludeExclude.exclude.displayExtension.suffix = ${space}excludes
grouperIncludeExclude.systemOfRecordAndIncludes.displayExtension.suffix = ${space}system of record and includes
grouperIncludeExclude.includesMinusExcludes.displayExtension.suffix = ${space}includes minus excludes

# suffixes for various include/exclude groups (can use "space" for space).
# these should uniquely identify various parts of the include/exclude.
# i.e. if the grouperIncludeExclude type is applied to a group with a suffix of the include suffix, the other groups will not be created...
grouperIncludeExclude.requireGroups.displayExtension.suffix = ${space}requireGroups ${i}

# suffixes for various include/exclude groups (can use "space" for space).
# these should uniquely identify various parts of the include/exclude.
# i.e. if the grouperIncludeExclude type is applied to a group with a suffix of the include suffix, the other groups will not be created...
grouperIncludeExclude.requireGroups.extension.suffix = _requireGroups${i}

# can use ${extension} as the group extension, or ${displayExtension} for group display extension
grouperIncludeExclude.overall.description = Group containing list of ${displayExtension} after adding the includes and subtracting the excludes
grouperIncludeExclude.systemOfRecord.description = Group containing list of ${displayExtension} (generally straight from the system of record) without yet considering manual include or exclude lists
grouperIncludeExclude.include.description = Group containing manual list of includes for group ${displayExtension} which will be added to the system of record list (unless the subject is also in the excludes group)
grouperIncludeExclude.exclude.description = Group containing manual list of excludes for group ${displayExtension} which will not be in the overall group
grouperIncludeExclude.systemOfRecordAndIncludes.description = Internal utility group for group ${displayExtension} which facilitates the group math for the include and exclude lists
grouperIncludeExclude.includesMinusExclude.description = Internal utility group for group ${displayExtension} which facilitates includes, excludes, and required groups (e.g. activeEmployee)

Features
• There is a GSH command to do this manually without the types and attributes enabled

```gsh
group.manageIncludesExcludes(grouperSession, isIncludeExclude)
group.manageIncludesExcludes(grouperSession, isIncludeExclude, groupThatMembersMustAlsoBeIn)
group.manageIncludesExcludes(grouperSession, isIncludeExclude, setOfGroupsThatMembersMustAlsoBeIn)
```

• All groups above are auto-created or in some cases auto-fixed when the types or attributes are selected
• Config file entries are validated with descriptive error messages when not filled in
• Built-in hook is installed automatically if enabled
• If overall group originally has list members, then move them to the system of record group so the overall can be converted into a composite group
• Since multiple require groups can be configured, variable substitution occurs in names and descriptions
• See which state we are in (on sor or overall). If the types or attributes are selected on a group with a suffix which is the system of record configured suffix, then the overall group will be created without that suffix. This is especially useful for loader related groups where the system of record group is created first
• Transactions are tested, if a failure occurs in the hook, all actions are rolled back
  • note that since this will run in one transaction, for huge groups (tens of thousands of members) it might cause memory problems...
• If an include exclude group has that type removed, nothing will change. This is because it would be confusing if the include and exclude groups existed and weren't being used. And if they contain members that are useful they should not be deleted. To delete these groups, first take out exclude out of the composite group, then delete the exclude and include groups, then remove the include exclude type and the other groups will be cleaned up.
• If include exclude are selected or require groups, then removed (if legal, see above point about include and exclude), then the overall group will remain with the system of record group as its only member
• Add debug logging to Group class and GroupTypeTupleIncludeExcludeHook class. To see debug info on include exclude and require groups, add this to the log4j.properties

```prop
log4j.logger.edu.internet2.middleware.grouper.hooks.examples.GroupTypeTupleIncludeExcludeHook = DEBUG
log4j.logger.edu.internet2.middleware.grouper.Group = DEBUG
```

• Many configurations and changes of state are unit tested
• Auto-create system groups (well, only the wheel group) if configured in grouper.properties, also leave entries to auto create other groups, and assign memberships (e.g. the uiUsers group)

See Also

Example from University of Pennsylvania with includes and excludes