

Grouper Loader example include exclude and privileges

Wiki Home	Grouper Release Announcements	Grouper Guides	Grouper Deployment Guide	Community Contributions	Internal Developer Resources
---------------------------	---	--------------------------------	--	---	--

[Main Grouper Loader page](#)

Simple list of groups

You can make a loader group that returns a list of groups:

```
grouperLoaderQuery:  select group_name, subject_id, subject_source_id from my_course_grouper_v
```

GROUP_NAME	SUBJECT_ID	SUBJECT_SOURCE_ID
courses:math101:math101	jsmith	person
courses:math101:math101	mjones	person
courses:english101:english101	pdavis	person

If you want to have the groups have some privileges, then make a group query for this loader job

```
grouperLoaderGroupQuery: select group_name, 'courses:etc:courseReaders' as readers from my_course_grouper_v
```

GROUP_NAME	READERS
courses:math101:math101	courses:etc:courseReaders
courses:english101:english101	courses:etc:courseReaders

List of groups with include exclude

One way to do include/exclude with a list of loader groups is to add that attribute to all the groups. However, if you do this, then managing privileges in 2.1.5- is a little challenging. Note the group name has _systemOfRecord at the end

```
grouperLoaderQuery:  select group_name || '_systemOfRecord', subject_id, subject_source_id from my_course_grouper_v
```

GROUP_NAME	SUBJECT_ID	SUBJECT_SOURCE_ID
courses:math101:math101_systemOfRecord	jsmith	person
courses:math101:math101_systemOfRecord	mjones	person
courses:english101:english101_systemOfRecord	pdavis	person

If you want to have the groups have some privileges, then make a group query for this loader job

```
grouperLoaderGroupQuery: select group_name || '_systemOfRecord', 'courses:etc:courseReaders' as readers from my_course_grouper_v
grouperLoaderGroupTypes: addIncludeExclude
```

Then there would be each of these for each group:

```
courses:math101:math101
courses:math101:math101_includes
courses:math101:math101_excludes
courses:math101:math101_systemOfRecord
```

For privileges on that, you should need: courses:etc:courseReaders to be able to read all groups, and courses:etc:courseUpdaters to be able to only update the include/exclude group. [You should be able to apply a rule](#) on the "courses" stem which has an "if" on the group name, if it ends in _includes or _excludes, then courseUpdaters should be able to update the group.

Multiple lists of groups with include and privileges

This is a common use case where you have the students, you have an includes group, you have an instructors group, and includes for instructors, and the overall group. Also, the instructors should be able to read/update the course, and the overall course admins should too

Here is the overall list of groups for a course

```
courses:math101:math101
courses:math101:math101_students
courses:math101:math101_students_includes
courses:math101:math101_students_systemOfRecord
courses:math101:math101_instructors
courses:math101:math101_instructors_includes
courses:math101:math101_instructors_systemOfRecord
```

You can make a loader job for the students and instructors. You can have a UNION query, or you could have two loader jobs.

```
grouperLoaderQuery: select group_name || '_systemOfRecord', subject_id, subject_source_id from
my_course_instr_grouper_v
```

GROUP_NAME	SUBJECT_ID	SUBJECT_SOURCE_ID
courses:math101:math101_students_systemOfRecord	jsmith	person
courses:math101:math101_instructors_systemOfRecord	mjones	person
courses:english101: english101_students_systemOfRecord	pdavis	person

Maybe you want course readers and instructors to be able to read the system of record groups. Note, if you can join to the grouper_groups table and use the group UUIDs, it would be faster. You can do that like this:

```
grouperLoaderGroupQuery: select group_name || '_systemOfRecord' as group_name, 'courses:etc:courseReaders,' ||
group_base_name || '_instructors' as readers from my_course_instr_grouper_v
```

GROUP_NAME	READERS
courses:math101:math101_students_systemOfRecord	courses:etc:courseReaders,courses:math101:math101_instructors
courses:math101:math101_instructors_systemOfRecord	courses:etc:courseReaders,courses:math101:math101_instructors
courses:english101: english101_students_systemOfRecord	courses:etc:courseReaders,courses:english101: english101_instructors

Now you need to flex your SQL muscles and make a query that will create all the other groups for you (use UNION?). Or this could be multiple loader jobs. This is in a separate loader job that runs after (?) the first one. You can join to the grouper_groups table so that it only does groups that exist and wont have unresolvable subject errors. Note if you join to the grouper_groups table, the you will get better performance too. However you might want to run this 3 times in a row (or however many times it needs) so that all can be managed in one day.

```
grouperLoaderQuery: select group_name, subject_identifier, subject_source_id from course_group_structure_v
```

GROUP_NAME	SUBJECT_IDENTIFIER	SUBJECT_SOURCE_ID
courses:math101:math101	courses:math101:math101_students	g:gsa
courses:math101:math101	courses:math101:math101_instructors	g:gsa
courses:math101:math101_students	courses:math101:math101_students_includes	g:gsa
courses:math101:math101_students	courses:math101:math101_students_systemOfRecord	g:gsa
courses:math101:math101_instructors	courses:math101:math101_instructors_includes	g:gsa
courses:math101:math101_instructors	courses:math101:math101_instructors_systemOfRecord	g:gsa
courses:english101:english101	courses:english101:english101_students	g:gsa
courses:english101:english101	courses:english101:english101_instructors	g:gsa

Now we can have privileges on these groups... again, you can do this with views, and unions, etc Note, if you can join to the grouper_groups table and use the group UUIDs, it would be faster.

```
grouperLoaderGroupQuery: select group_name, readers, updaters from course_group_query_v
```

GROUP_NAME	READERS	UPDATERS
courses:math101:math101	courses:etc:courseReaders,courses:math101:math101_instructors	
courses:math101:math101_students	courses:etc:courseReaders,courses:math101:math101_instructors	
courses:math101:math101_students_includes	courses:etc:courseReaders,courses:math101:math101_instructors	courses:etc:courseUpdaters,courses:math101:math101_instructors
courses:math101:math101_instructors	courses:etc:courseReaders,courses:math101:math101_instructors	
courses:math101:math101_instructors_includes	courses:etc:courseReaders,courses:math101:math101_instructors	courses:etc:courseUpdaters,courses:math101:math101_instructors
courses:english101:english101	courses:etc:courseReaders,courses:english101:english101_instructors	