

# Grouper Loader add new job

<a href="#">Wiki Home</a>	<a href="#">Grouper Release Announcements</a>	<a href="#">Grouper Guides</a>	<a href="#">Grouper Deployment Guide</a>	<a href="#">Community Contributions</a>	<a href="#">Internal Developer Resources</a>
---------------------------	---	--------------------------------	--	---	--

[Main Grouper Loader page](#)

As a Grouper administrator, you might get a request to add a job to the loader. This is how to do this:

The request should have two parts (at least) for a simple job (simple means one group one query):

1. The query
2. The name of the group (hopefully the requestor has created that group and already has admin rights, or you need to grant it, or READ or whatever)

Note, first decide which envs you are going to do this in. Just prod? Test and prod? At Penn sometimes we do just prod, sometimes we do both.

To configure the group, first you can make a view from the query that returns at least the subjectId (maybe the numeric ID at your institution), or subjectIdentifier (maybe the netId at your institution). Note, it is assumed that the query is intended for a database that has a connection configured in the grouper-loader.properties. At Penn this is our people database (where Grouper resides), or the data warehouse. For the Grouper database, the database name is configured as "grouper", for the other, it is a custom name. At Penn we use "warehouse" for our data warehouse.

Note, generally all loader jobs have members as people, if this is the case, you can also select the subject source id to make the job more efficient. Test the resulting query of the view, and note the columns must be correct (in this case the subject id)

```
select penn_id as subject_id, 'pennperson' as SUBJECT_SOURCE_ID from some_descriptive_name_v
```

Now we can go to the group in the UI that the requester specified. "Edit" the group. Mark this as a loader group. Then edit the attributes:

EXPLORE

Edit attributes

Current location: Root: penn: [redacted] ms

Group type	Attribute	Value
base		
	members	List field
<u>grouperLoader</u>		
	<u>grouperLoaderAndGroups</u>	
	<u>*grouperLoaderDbName</u>	warehouse
	<u>grouperLoaderGroupQuery</u>	
	<u>grouperLoaderGroupTypes</u>	
	<u>grouperLoaderGroupsLike</u>	
	<u>grouperLoaderIntervalSeconds</u>	
	<u>grouperLoaderPriority</u>	
	<u>grouperLoaderQuartzCron</u>	0 52 7 * * ?
	<u>*grouperLoaderQuery</u>	select penn_id as subject_id, 'pennperson' as SUBJECT_
	<u>*grouperLoaderScheduleType</u>	CRON
	<u>*grouperLoaderType</u>	SQL_SIMPLE
	<u>*Required attribute</u>	

Save attributes and finish

Save attributes and add members

Note, for cron, generally I set the job for once per day, and I pick a random time in the morning so all the jobs don't run at the same time.

At this point, the job is there, and the WS status (if your institution runs this with nagios or something) will start throwing errors since that job has never run successfully. So lets run it.

Log in to the daemon server where you run GSH:

```
[appadmin@someServer1 ~]$ ssh someServer2
[appadmin@someServer2 ~]$ locate gsh
/opt/appserv/tomcat_3b/webapps/grouperWs/WEB-INF/bin/gsh
/opt/appserv/tomcat_3b/webapps/grouperWs/WEB-INF/bin/gsh.bat
/opt/appserv/tomcat_3b/webapps/grouperWs/WEB-INF/bin/gsh.sh
[appadmin@someServer2 ~]$ cd /opt/appserv/tomcat_3b/webapps/grouperWs/WEB-INF/bin
[appadmin@someServer2 bin]$ ./gsh
Type help() for instructions
gsh 0% grouperSession = GrouperSession.startRootSession();
edu.internet2.middleware.grouper.GrouperSession: 96f3d78c030b45f1b43f5d5d4a598448,'GrouperSystem','application'
gsh 1% loaderGroup = GroupFinder.findByName(grouperSession, "some:group");
gsh 2% loaderRunOneJob(loaderGroup);
loader ran successfully, inserted 713 memberships, deleted 0 memberships, total membership count: 713
gsh 3% exit
```

In order for the loader to pick up the new job on the next day, it needs to be bounced. Note: generally it is good to not do this in the morning when most jobs are scheduled. Here is how to do that if the loader is running as a unix service

```
[appadmin@someServer2 bin]$ /sbin/service grouperLoader restart
Restarting Grouper loader: Grouper loader is running with processId: 2505, waiting for exit...
Waiting for exit...
Grouper loader is stopped
Starting Grouper loader service...
Grouper loader is running...
done.
[appadmin@lorenzo bin]$ ps -ef | grep gsh
appadmin 30212      1  0 00:52 pts/0      00:00:00 /bin/sh /opt/appserv/tomcat_3b/webapps/grouperWs/WEB-INF/bin
/gsh -loader
appadmin 30213 30212 99 00:52 pts/0      00:00:12 /opt/appserv/java6/bin/java -Xms64m -Xmx750m -Dgrouper.home=/opt
/appserv/tomcat_3b/webapps/grouperWs/WEB-INF/bin/.. -classpath /opt/appserv/tomcat_3b/webapps/grouperWs/WEB-INF
/bin/./classes:/opt/appserv/tomcat_3b/webapps/grouperWs/WEB-INF/bin/./lib/*: edu.internet2.middleware.grouper.
app.gsh.GrouperShellWrapper -loader
appadmin 30232 29763  0 00:52 pts/0      00:00:00 grep gsh
[appadmin@lorenzo bin]$
```

Note that as of Grouper 2.3, if you create a new loader job, you can have it scheduled without restarting the daemon by using an option in the New UI. If you're an admin of the group, under "More actions", there's an option named "Schedule loader process". This option can also be used if you change the loader's schedule.