v0.1.0_gsh

⚠ This is an archived document. For the current version, please see GrouperShell (gsh)

GrouperShell v0.1.0

GrouperShell (gsh) is a shell for administering and interacting with the Grouper API. It can be used in both a batch and interactive manner.

API Compability

This version of of **GrouperShell** is compatible with Grouper v1.2.0.

Build GrouperShell

```
% cd grouper/contrib/gsh % ant
```

Test GrouperShell's scripting capabilities

```
% ant test
```

Grouper must have a JDBCSourceAdapter for subjects for the test suite to complete successfully. This need is adequately met by testing GrouperShell using the same grouper/conf directory contents used to test the Grouper API.

Note: In some environments the tests may fail due to a bug exposed by the testing procedure which might not indicate any actual error with the gsh utility itself. This is known to occur, for example, under cygwin on Windows. An alternative testing process is to run each of the test suites, which are gsh scripts, individually as follows:

```
% bin/gsh.sh src/test/test.gsh % bin/gsh.sh src/test/groups.gsh % bin/gsh.sh src/test/stems.gsh % bin/gsh.sh src
/test/composites.gsh % bin/gsh.sh src/test/privs.gsh
```

Build `gsh.jar` file in the `dist` subdirectory

```
% ant jar
```

Build javadoc in `doc/html`

```
% ant html
```

Use GrouperShell

- Run **GrouperShell** in an interactive manner from a Unix-like environment:

  ```
  % ./bin/gsh.sh
  ```

- Run **GrouperShell** (crudely) from Ant:

```
% ant shell
```

- Read **GrouperShell** commands from STDIN:

```
% ./bin/gsh.sh -
```

- Read **GrouperShell** commands from a script file:

```
% ./bin/gsh.sh /path/to/your/script.gsh
```

GrouperShell Commands

Command	Description
addComposite(group, type, left group, right group.	Add composite membership.
addGroup(parent, extension, displayExtension)	Add group beneath parent stem with the specified extension and displayExtension.
addMember(group, subject id)	Add subject as a member to the group.
addRootStem(extension, displayExtension)	Add root stem with the specified extension and displayExtension.
addStem(parent, extension, displayExtension)	Add stem beneath parent stem with the specified extension and displayExtension.
addSubject(id, type, name)	Add a HibernateSubject to the Groups Registry.
delComposite(group)	Delete composite membership from the specified group.
delGroup(name)	Delete group with the specified name.
delMember(group, subject id)	Remove subject as a member of the group.
delStem(name)	Delete stem with the specified name.
exit	Terminate shell.
findSubject(id)	Find a subject.
findSubject(id, type)	Find a subject.
findSubject(id, type, source)	Find a subject.
getGroupAttr(stem, attr)	Get value of group's attr attribute.
getGroups(name)	Find all groups with name in any naming attribute value.
getMembers(group)	Get members of the group.
getSources()	Find all Subject sources.
getStemAttr(stem, attr)	Get value of stem's attr attribute.
getStems(name)	Find all stems with name in any naming attribute value.
grantPriv(name, subject id, privilege)	Grant privilege to subject id on name. *privilege* must be an *AccessPrivilege* (e.g. `AccessPrivilege.ADMIN`) or *NamingPrivilege* (e.g. `NamingPrivilege.STEM`) constant.
groupAddType(group, type)	Add group type *type* to *group*.
groupDelType(group, type)	Delete group type *type* from *group*.
groupGetTypes(group)	Get *group*'s group types.
groupHasType(group, type)	Check whether *group* has group type *type*.
hasMember(group, subject id)	Is subject a member of this group.
hasPriv(name, subject id, privilege)	Does subject id have privilege on name? *privilege* must be a n_AccessPrivilege_ (e.g. `AccessPrivilege.ADMIN`) or *NamingPrivilege* (e.g. `NamingPrivilege.STEM`) constant.
help()	Display usage information.

history()	Print commands that have been run.
history(N)	Print the last N commands that have been run.
last()	Run the last command executed.
last(N)	Execute command number N.
p(command)	Pretty print results. This command is more useful when GSH_DEVEL is enabled.
quit	Terminate shell.
resetRegistry()	Restore the Groups Registry to a default state.
revokePriv(name, subject id, privilege)	Revoke privilege from subject id on name. *privilege* must be an *AccessPrivilege* (e.g. `AccessPrivilege.ADMIN`) or *NamingPrivilege* (e.g. `NamingPrivilege.STEM`) constant.
setGroupAttr(group, attr, value)	Set value of *group*'s *attr* attribute.
setStemAttr(stem, attr, value)	Set value of *stem*'s *attr* attribute.
typeAdd(name)	Create custom group named *name*.
typeAddAttr(type, name, read, write, required)	Create a custom group attribute named *name* on group type *type*. *read* and *write* must be an `AccessPrivilege` (e.g. `AccessPrivilege.ADMIN`).
typeAddList(type, name, read, write)	Create a custom membership list named *name* on group type *type*. *read* and *write* must be an `AccessPrivilege` (e.g. `AccessPrivilege.ADMIN`).
typeDel(name)	Delete the group type named *name*.
typeDelField(type, name)	Delete the custom field named *name* from group type *type*.
typeFind(name)	Find the group type named *name*
typeGetFields(name)	Get the fields associated with the group type named *name*.
version()	Return version information.
xmlFromFile(filename)	Load Groups Registry with XML contained in *filename*.
xmlFromString(xml)	Load Groups Registry with XML in the *xml* string.
xmlFromURL(url)	Load Groups Registry with XML located at *url*.
xmlToFile(filename)	Exports Groups Registry to *filename*.
xmlToString()	Exports Groups Registry to string.
xmlUpdateFromFile(filename)	Update Groups Registry with XML contained in *filename*.
xmlUpdateFromString(xml)	Update Groups Registry with XML in the *xml* string.
xmlUpdateFromURL(url)	Update Groups Registry with XML located at *url*.

In addition, any Grouper API method can be directly invoked just by referencing it, inclusive of the class in which it is defined. And methods return a java object which can be stored in a variable. For example, the following gsh session determines all of the groups to which a given subject belongs:

```
gsh-0.0.1 0% subj = findSubject("SD00125") subject: id='SD00125' type='person' source='kitn-person'
name='Barton, Tom'  gsh-0.0.1 1% sess = GrouperSession.start(subj) edu.internet2.middleware.grouper.
GrouperSession: 29c40f97-9fb0-4e45-88bc-a14877a6c9b5,'SD00125','person'  gsh-0.0.1 2% member = MemberFinder.
findBySubject(sess, subj) member: id='SD00125' type='person' source='kitn-person' uuid='d0fa765e-1439-4701-89b1-
9b08b4ce9daa'  gsh-0.0.1 3% member.getGroups() group: name='etc:wheel' displayName='Grouper Administration:
Wheel Group' uuid='6f77fb36-b466-481a-84a7-7af609f1ad09'
```

GrouperShell Variables

Variable	Description
GSH_DEBUG	If set to true, stack traces will be printed upon failure.
GSH_DEVEL	If set to true, summaries of returned objects are not automatically printed.
GSH_TIMER	If set to true, the time taken to evaluate each command will be displayed.

Example:

```
gsh-0.0.1 4% GSH_DEVEL = true  gsh-0.0.1 5% subj = findSubject("SD00125")  gsh-0.0.1 6% sess = GrouperSession.
start(subj)  gsh-0.0.1 7% member = MemberFinder.findBySubject(sess, subj)  gsh-0.0.1 8% p(member.getGroups())
group: name='etc:wheel' displayName='Grouper Administration:Wheel Group' uuid='6f77fb36-b466-481a-84a7-
7af609f1ad09'
```