

# Grouper external users on demo server

<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>
---------------------------	---	--------------------------------	--	---	--

Grouper 2.0, 2.1, and 2.2 are on the demo server, with external users support.

Click here to the jump page for the demo server: <https://grouperdemo.internet2.edu>

Next to one of the envs, click self-register. it ends with /grouperExternal/appHtml/grouper.html?operation=ExternalSubjectSelfRegister.  
externalSubjectSelfRegister

Note: if you do not have an InCommon ID, you can use facebook or google.

After registering, click here to go to the UI from the jump page. Note, you might need to email grouper list and ask for a folder to be able to manage.

To be added to the library group,  
go to this URL: [https://grouperdemo.internet2.edu/grouper\\_v2\\_1\\_0/grouperExternal/appHtml/grouper.html?operation=ExternalSubjectSelfRegister.  
externalSubjectSelfRegister&externalSubjectInviteName=library](https://grouperdemo.internet2.edu/grouper_v2_1_0/grouperExternal/appHtml/grouper.html?operation=ExternalSubjectSelfRegister.externalSubjectSelfRegister&externalSubjectInviteName=library)

When you registered you were added to etc:externalSubjectInviters, which allows you to invite other people to register to this server.

To invite others,  
go to this URL: [https://grouperdemo.internet2.edu/grouper\\_v2\\_1\\_0/grouperUi/appHtml/grouper.html?operation=InviteExternalSubjects.  
inviteExternalSubject](https://grouperdemo.internet2.edu/grouper_v2_1_0/grouperUi/appHtml/grouper.html?operation=InviteExternalSubjects.inviteExternalSubject)

These groups are public if you want to assign people to them (e.g. through the invite process): test:testGroup0, test:testTestGroup1, test:testGroup2, test:testGroup3

## Auto-provision groups

When registering, users will be automatically added to etc:uiUsers and etc:externalSubjectInviters, which is the group which is required to use the UI, and the group which lets you invite others. When registering in the library URL, the user will be added to the aStem:library group

## Configuration

grouper.properties: (everything is default except)

```

configuration.autocreate.group.name.2 = etc:uiGroup
configuration.autocreate.group.description.2 = user interface users
configuration.autocreate.group.subjects.2 = mchyzer

configuration.autocreate.group.name.3 = aStem:library
configuration.autocreate.group.description.3 = access to the library application
configuration.autocreate.group.subjects.3 =

externalSubjects.autoCreateSource = true

#put some group names comma separated for groups to auto add subjects to
externalSubjects.autoaddGroups=etc:uiGroup,etc:externalSubjectInviters
#should be insert, update, or insert,update
externalSubjects.autoaddGroupActions=insert,update
#if a number is here, expire the group assignment after a certain number of days
externalSubjects.autoaddGroupExpireAfterDays=

#add multiple group assignment actions by URL param: externalSubjectInviteName
externalSubjects.autoadd.testingLibrary.externalSubjectInviteName=library
#comma separated groups to add for this type of invite
externalSubjects.autoadd.testingLibrary.groups=aStem:library
#should be insert, update, or insert,update
externalSubjects.autoadd.testingLibrary.actions=insert,update
#should be insert, update, or insert,update
externalSubjects.autoadd.testingLibrary.expireAfterDays=

#####
## mail settings (optional, e.g. for daily report form loader)
#####

#smtp server is a domain name or dns name, must be simple clear text stmp with no authentication
mail.smtp.server = smtp.gmail.com

#leave blank if unauthenticated
mail.smtp.user = grouperstystem@gmail.com

#leave blank if unauthenticated
mail.smtp.pass = *****

#leave blank or false for no ssl, true for ssl
mail.smtp.ssl = true

#leave blank for default (probably 25), if ssl is true, default is 465, else specify
#mail.smtp.port =

#this is the default email address where mail from grouper will come from
mail.from.address = grouperstystem@gmail.com

#this is the subject prefix of emails, which will help differentiate prod vs test vs dev etc
mail.subject.prefix = GROUPERDEMO_2_1_0:

#when running junit tests, this is the address that will be used
mail.test.address = mchyzer@yahoo.com

```

htpd.conf (note, in this case the admin UI and external UI part are protected by shib, but in reality, maybe the admin UI would not let external users in, just the external UI)

```
#match anything that is not grouperExternal
<LocationMatch ^/grouper_v2_0[^/]*(?!grouperExternal/)>
  AuthType shibboleth
  ShibRequestSetting requireSession 1
  require valid-user

</LocationMatch>

#match anything that is grouperExternal, but not public
<LocationMatch ^/grouper_v2_0[^/]*grouperExternal/(?!public/)>
  AuthType shibboleth
  ShibRequestSetting requireSession 1
  require valid-user

</LocationMatch>
```

Note, shib might loop if someone goes in with non SSL, and the SP requires SSL, so you could add something like this to the httpd.conf (depending on which url you are mapped, to, in this case "grouper"):

```
RewriteCond %{HTTPS} off
RewriteCond %{REQUEST_URI} ^/grouper/
RewriteRule (.*?) https://%{HTTP_HOST}%{REQUEST_URI}
```

media.properties:

- restrict UI users to users in this group, or else they get an error, enabled registration, and invites, and invites by identifier, etc

```

require.group.for.logins=etc:uiUsers

#users must be in this group to invite external users to grouper
require.group.for.inviteExternalSubjects.logins=etc:externalSubjectInviters

#####
## External subjects invitation
#####

# if the registration screen is enabled
externalMembers.enabledRegistration = true

#if admins should be emailed after each action, put comma separated addresses here
externalMembers.emailAdminsAddressesAfterActions = me@myschool.edu

#####
## Invite external members
#####

# if the invitation screen is enabled
inviteExternalMembers.enableInvitation = true

#if link from admin UI
inviteExternalPeople.link-from-admin-ui = true

#if link from lite UI
inviteExternalPeople.link-from-lite-ui = true

#if admins should be emailed after each action, put comma separated addresses here
inviteExternalMembers.emailAdminsAddressesAfterActions = me@myschool.edu

#if we should allow invite by identifier
inviteExternalMembers.allowInviteByIdentifier = true

```

sd

## Setup on the demo server

1. Turn off the UI default basic auth in the web.xml by commenting out the security sections in web.core.xml and web.ajax.xml
2. Shibbolize the app, except for the external part (per above config)
1. See that you are not prompted for shib authn here (well, in our case, we want shib for the UI too):[https://grouperdemo.internet2.edu/grouper\\_v2\\_1\\_0/](https://grouperdemo.internet2.edu/grouper_v2_1_0/)
2. But you are prompted for shib authn here:[https://grouperdemo.internet2.edu/grouper\\_v2\\_1\\_0/grouperExternal/appHtml/grouper.html?operation=ExternalSubjectSelfRegister.index](https://grouperdemo.internet2.edu/grouper_v2_1_0/grouperExternal/appHtml/grouper.html?operation=ExternalSubjectSelfRegister.index)

Run these commands to rebuild from svn

GrouperApi:

```
[mchyzer@i2mibuild bin]$ buildGrouper.sh trunk
[mchyzer@i2midev1 grouper2.0]$ cd /tmp/grouper2.0/
[mchyzer@i2midev1 grouper2.0]$ sftp i2mibuild
sftp> get /home/mchyzer/tmp/grouper/build_mchyzer/grouper.apiBinary-2.0.0.tar.gz
[appadmin@i2midev1 2.0.0]$ cd /opt/grouper/2.0.0
[appadmin@i2midev1 2.0.0]$ rm -rf grouper.apiBinary-2.0.0*
[appadmin@i2midev1 2.0.0]$ cp /tmp/grouper2.0/grouper.apiBinary-2.0.0.tar.gz .
[appadmin@i2midev1 2.0.0]$ tar xzvf grouper.apiBinary-2.0.0.tar.gz
[appadmin@i2midev1 2.0.0]$ cp -Rv filesGrouper/* grouper.apiBinary-2.0.0/
[appadmin@i2midev1 2.0.0]$ cd grouper.apiBinary-2.0.0/bin
[appadmin@i2midev1 bin]$ ./gsh.sh
gsh 0% grouperSession = GrouperSession.startRootSession();
gsh 1% addStem(null, "test", "test");
gsh 2% addGroup("test", "testGroup0", "testGroup0");
gsh 4% addGroup("test", "testGroup1", "testGroup1");
gsh 5% addGroup("test", "testGroup2", "testGroup2");
gsh 6% addGroup("test", "testGroup3", "testGroup3");
gsh 7% grantPriv("test:testGroup0", "GrouperAll", AccessPrivilege.UPDATE);
gsh 8% grantPriv("test:testGroup1", "GrouperAll", AccessPrivilege.UPDATE);
gsh 9% grantPriv("test:testGroup2", "GrouperAll", AccessPrivilege.UPDATE);
gsh 10% grantPriv("test:testGroup3", "GrouperAll", AccessPrivilege.UPDATE);
gsh 11% grantPriv("etc:externalSubjectInviters", "GrouperAll", AccessPrivilege.UPDATE);
```

GrouperUi:

```
[mchyzer@i2mibuild bin]$ buildGrouperUi.sh trunk
[mchyzer@i2midev1 grouper2.0]$ cd /tmp/grouper2.0/
[mchyzer@i2midev1 grouper2.0]$ sftp i2mibuild
sftp> get /home/mchyzer/tmp/grouperUi/build_mchyzer/grouper.ui-2.0.0.tar.gz
[mchyzer@i2midev1 grouper2.0]$ sudo su - appadmin
[appadmin@i2midev1 ~]$ cd /opt/grouper/2.0.0/
[appadmin@i2midev1 2.0.0]$ rm -rf grouper.ui-2.0.0*
[appadmin@i2midev1 2.0.0]$ cp /tmp/grouper2.0/grouper.ui-2.0.0.tar.gz .
[appadmin@i2midev1 2.0.0]$ tar xzf grouper.ui-2.0.0.tar.gz
[appadmin@i2midev1 2.0.0]$ cp -Rv filesGrouperUi/* grouper.ui-2.0.0/
[appadmin@i2midev1 2.0.0]$ cd grouper.ui-2.0.0
[appadmin@i2midev1 grouper.ui-2.0.0]$ ant clean
[appadmin@i2midev1 2.0.0]$ cd /opt/grouper/2.0.0/
[appadmin@i2midev1 2.0.0]$ cp -Rv filesGrouperUiTomcat/* /opt/tomcats/tomcat_g/webapps/grouper_v2_0_0/

[appadmin@i2midev1 2.0.0]$ /sbin/service tomcat_e restart
```