Grouper reporting

In patches in Grouper 2.4 (api #61, ui #37), Grouper has a reporting capability. This will start simple and we can add more features later.

> You may want to check out the blog on Grouper Reporting from November 2019.

- Grouper reporting use cases

High level description

- Configure a report on a group or folder
- This report will have a cron that will run like loader jobs run
- The SQL report type consists of a SQL to run in a database, generating a CSV file
  - Note: it is a best practice to put the SQL in a view and call the view from grouper. Keep the source in the view in source control for versioning
  - The SQL report type is available in 2.5.0, 2.6.0, and 2.4.0 api #60 and ui #37
- The GSH report type uses a GSH script to either output directly to the report file, or to build data rows that are converted to a CSV file
  - The GSH report type is available in 2.5.53 and 2.6.0
- The output of the report will be encrypted and stored to storage
- Users can be notified by email that the report exists
- When the login they can download the most recent report
  - This will have Grouper reverse proxy the report from storage, unencrypt it, and deliver it to the user
- Reports will be automatically deleted after 30 days or if there are more than 100 instances of a report

Reporting Configuration

You need to use a file system (if you have a shared filesystem among all grouper component JVMs), or Amazon AWS S3.

grouper.properties

```
######################################
## Grouper Reporting
######################################

# folder where system objects are for reporting config
# {valueType: "stem"}
reportConfig.systemFolder = $$grouper.rootStemForBuiltinObjects$$:reportConfig

# if grouper reporting should be enabled
# {valueType: "boolean", required: true}
grouperReporting.enable = true

# grouper reporting storage# grouper reporting storage option. valid values are database, fileSystem or S3
# {valueType: "string", required: true}
reporting.storage.option = database

# grouper reporting file system path where reports will be stored, e.g. /opt/grouper/reports
# {valueType: "string", required: false}
reporting.file.system.path =

# grouper reporting s3 bucket name where the reports will be uploaded
# {valueType: "string", required: false}
reporting.s3.bucket.name =

# grouper reporting s3 bucket name where the reports will be uploaded, e.g. us-west-2
# {valueType: "string", required: false}
reporting.s3.region =

# grouper reporting s3 access key
# {valueType: "string", required: false}
reporting.s3.access.key =

# grouper reporting s3 secret key
# {valueType: "string", required: false}
reporting.s3.secret.key =

#grouper reporting email subject
# {valueType: "string"}
reporting.email.subject = Report $$reportConfigName$$ generated

#grouper reporting email body.  Can use variables
# {valueType: "string"}
reporting.email.body = Hello $$subjectName$$, \n\n Report $$reportConfigName$$ has been generated. Download the
report: $$reportLink$$ \n\n Thanks
```

For this example lets use the file system. Configure in grouper.properties

```
# grouper reporting file system path where reports will be stored, e.g. /opt/grouper/reports
# {valueType: "string", required: false}
reporting.file.system.path = d:/temp/temp/grouperReports
```

Make sure you have mail setup in the SMTP external system

```
#smtp server is a domain name or dns name.  set to "testing" if you want to log instead of send (e.g. for
testing)
# {valueType: "string"}
mail.smtp.server = localhost
mail.smtp.from.address = noreply@whatever.edu
```

Make sure you have a mailAttributeName in your person subject source

Make sure you have grouper.ui.url set in grouper.properties

```
#put the URL which will be used e.g. in emails to users.  include the webappname at the end, and nothing after
that.
#e.g. https://server.school.edu/grouper/
# {valueType: "string"}
grouper.ui.url = http://localhost:8097/grouper/
```

Make sure you have an encrypt.key in morphString.properties

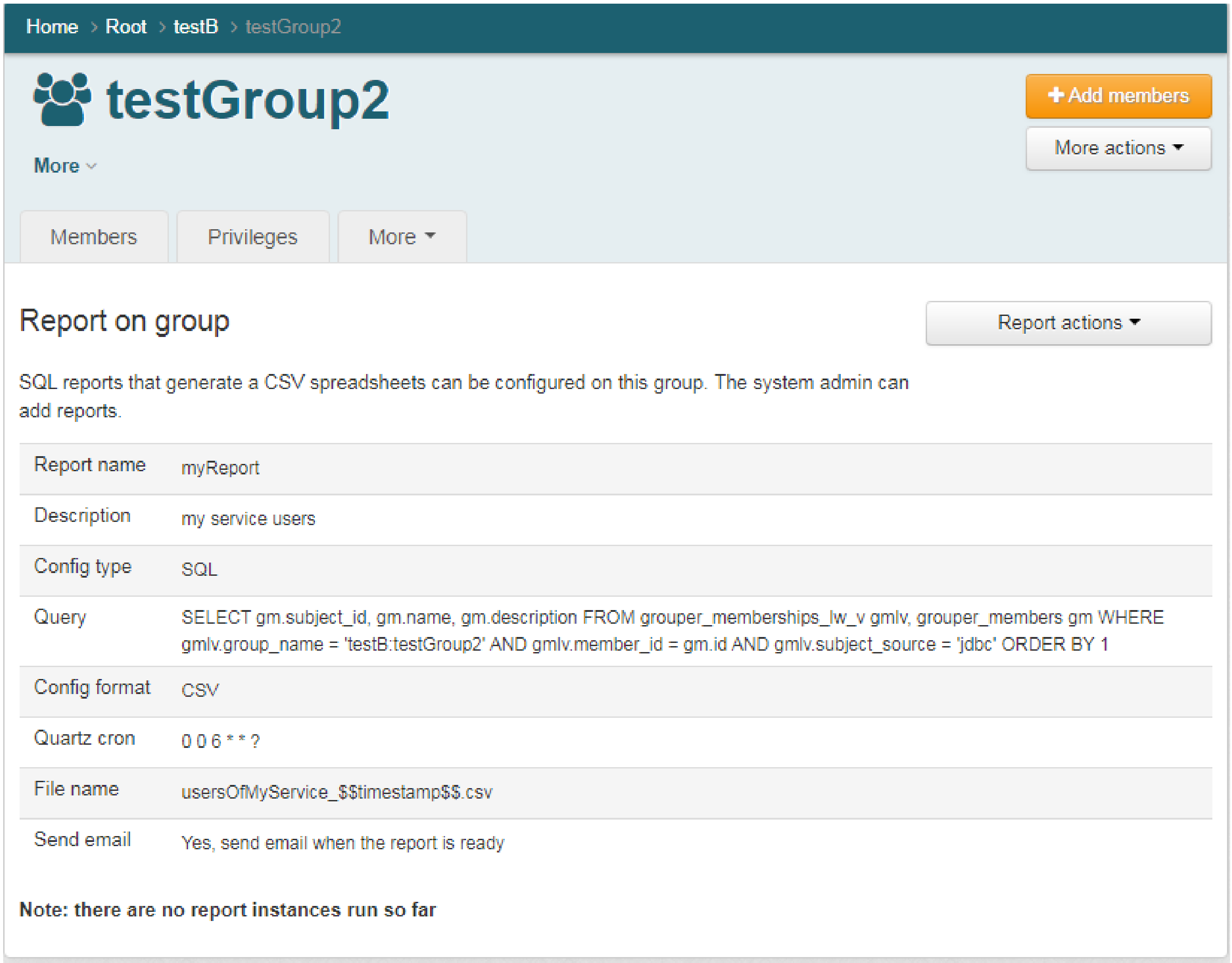
```
subjectApi.source.jdbc.param.emailAttributeName.value = email
```

If you are using the built in subject source, you can add a user for yourself with an email address (yours), this is GSH

```
grouperSession = GrouperSession.startRootSession();
RegistrySubject.addOrUpdate(grouperSession, "mchyzer", "person", "Chris Hyzer", "Chris Hyzer", "mchyzer",
"Chris Hyzer - IAM architect", "your@email.address");
```

SQL Report Example

Open a group, add a new report



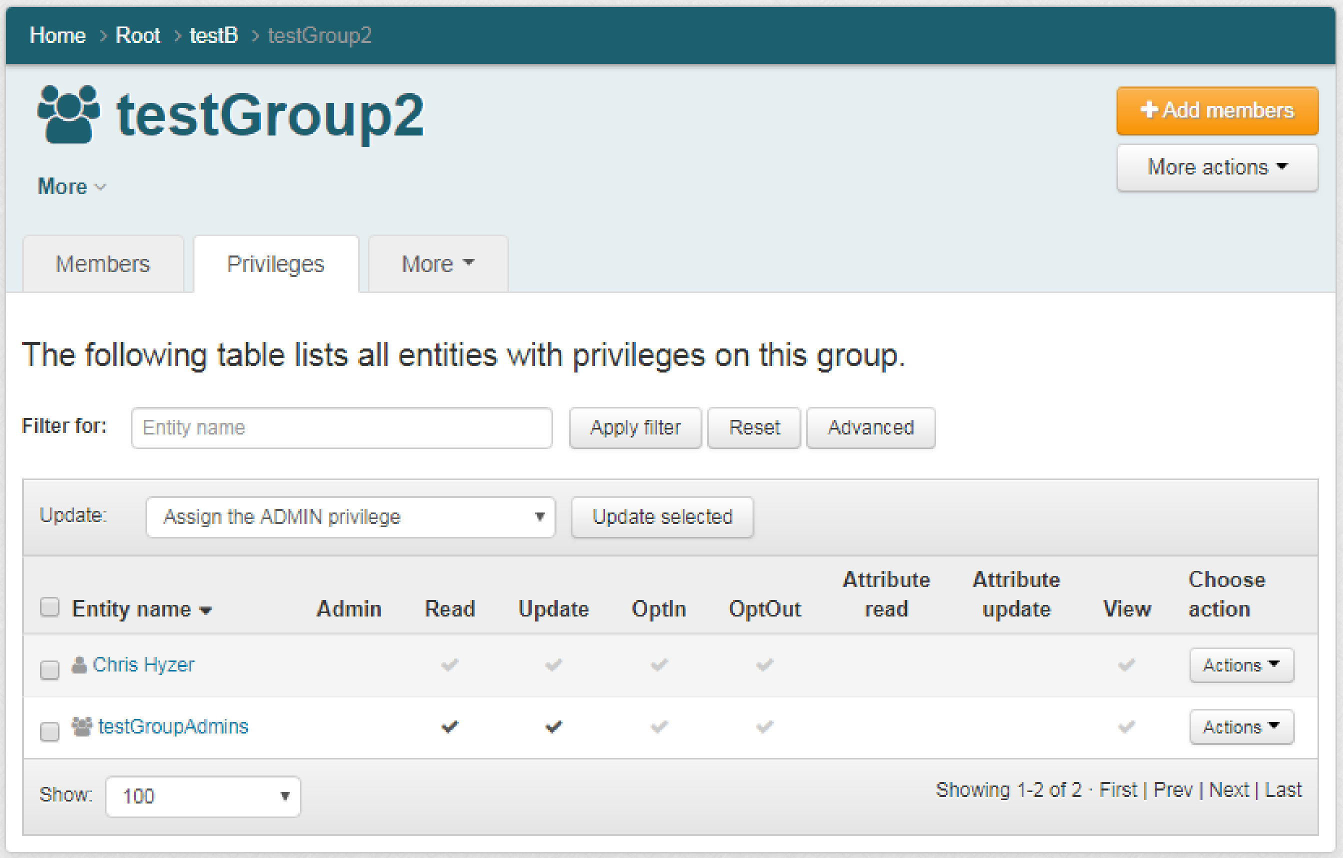
Report on group

SQL reports that generate a CSV spreadsheets can be configured on this group. The system admin can add reports.

Report name	myReport
Description	my service users
Config type	SQL
Query	SELECT gm.subject_id, gm.name, gm.description FROM grouper_memberships_lw_v gmlv, grouper_members gm WHERE gmlv.group_name = 'testB:testGroup2' AND gmlv.member_id = gm.id AND gmlv.subject_source = 'jdbc' ORDER BY 1
Config format	CSV
Quartz cron	0 0 6 * * ?
File name	usersOfMyService_$$timestamp$$.csv
Send email	Yes, send email when the report is ready

Note: there are no report instances run so far

Report name	myReport
Description	my service users
Config type	SQL
Query	SELECT gm.subject_id as SUBJECT_ID, gm.name as NAME, gm.description as DESCRIPTION FROM grouper_memberships_lw_v gmlv, grouper_members gm WHERE gmlv.group_name = 'testB:testGroup2' AND gmlv.member_id = gm.id AND gmlv.subject_source = 'jdbc' ORDER BY 1
Config format	CSV
Quartz cron	0 0 6 * * ? (run daily at 6am)
File name	usersOfMyService_$$timestamp$$.csv
Send email	Yes, send email when the report is ready
Allowed group id	A group with your user in it



The following table lists all entities with privileges on this group.

Filter for: Entity name

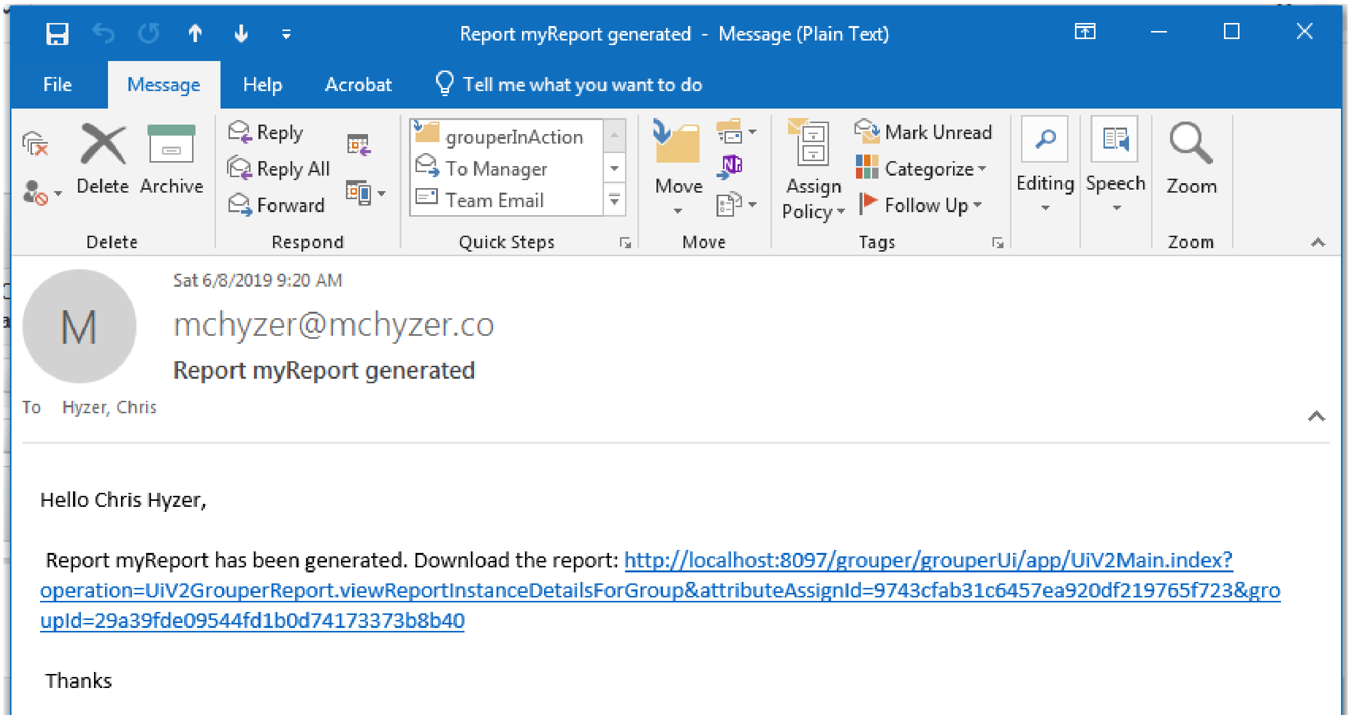
Update: Assign the ADMIN privilege

□ Entity name	Admin	Read	Update	OptIn	OptOut	Attribute read	Attribute update	View	Choose action
□ Chris Hyzer		✓	✓	✓	✓			✓	Actions
□ testGroupAdmins		✓	✓	✓	✓			✓	Actions

Show: 100

Showing 1-2 of 2 · First \| Prev \| Next \| Last

Now get an email about the report



Sat 6/8/2019 9:20 AM

mchyzer@mchyzer.co

Report myReport generated

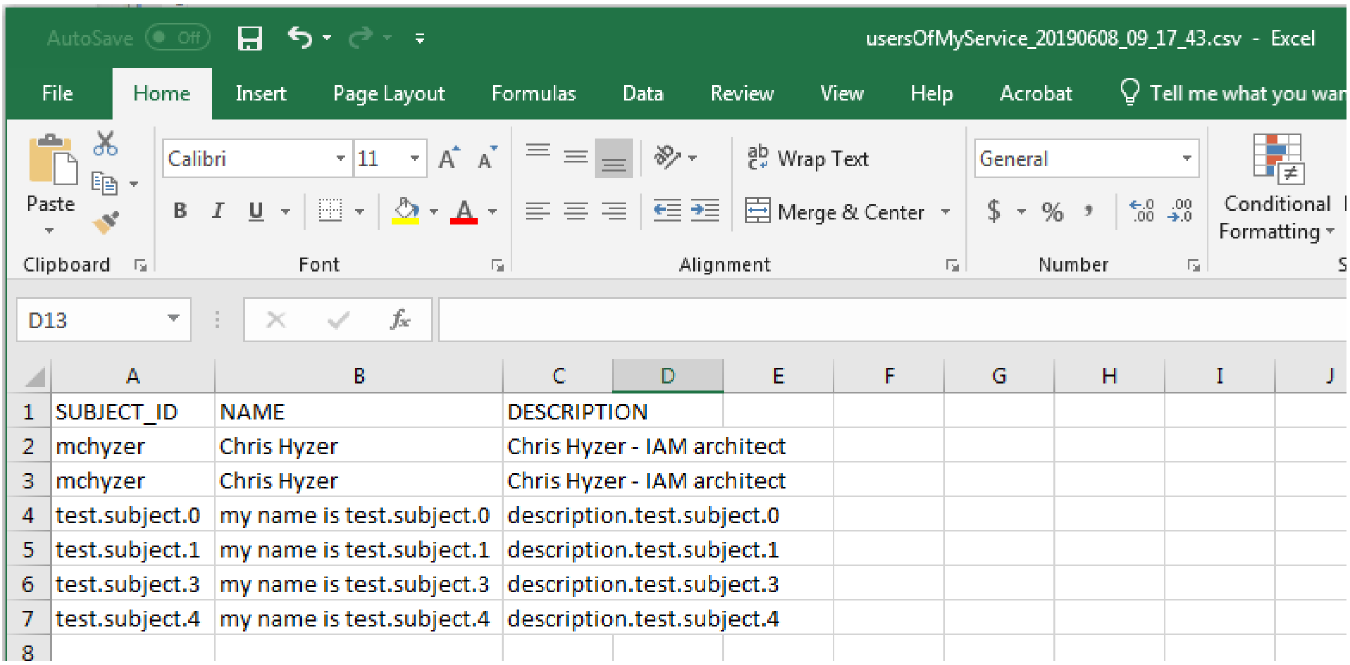
To Hyzer, Chris

Hello Chris Hyzer,

Report myReport has been generated. Download the report: http://localhost:8097/grouper/grouperUi/app/UiV2Main.index?operation=UiV2GrouperReport.viewReportInstanceDetailsForGroup&attributeAssignId=9743cfab31c6457ea920df219765f723&groupId=29a39fde09544fd1b0d74173373b8b40

Thanks

See the report



	A	B	C	D	E	F	G	H	I	J
1	SUBJECT_ID	NAME	DESCRIPTION							
2	mchyzer	Chris Hyzer	Chris Hyzer - IAM architect							
3	mchyzer	Chris Hyzer	Chris Hyzer - IAM architect							
4	test.subject.0	my name is test.subject.0	description.test.subject.0							
5	test.subject.1	my name is test.subject.1	description.test.subject.1							
6	test.subject.3	my name is test.subject.3	description.test.subject.3							
7	test.subject.4	my name is test.subject.4	description.test.subject.4							
8										

GSH Report Example

The GSH report type can have an output type of either CSV or FILE. For both types, the script will use fields from the available gsh_builtin_gshReportRuntime variable to add data to the output. For CSV output, the script will set a header array and a list of data arrays. For FILE output, the script will open a Writer and write arbitrary data to the character stream.

Variables available to the GSH script

Variable	Java class	Description
gsh_builtin_grouperSession	GrouperSession	session the script runs as
gsh_builtin_ownerStemName	String	owner stem name where template was called
gsh_builtin_ownerGroupName	String	owner group name where template was called
gsh_builtin_gshReportRuntime	GshReportRuntime	container to hold important information about the run
gsh_builtin_gshReportRuntime.getOwnerGroup()	Group	owner group where template was called
gsh_builtin_gshReportRuntime.getOwnerStem()	Stem	owner stem where template was called
gsh_builtin_gshReportRuntime.getOwnerGroupName()	String	same as gsh_builtin_ownerGroupName
gsh_builtin_gshReportRuntime.getOwnerStemName()	String	same as gsh_builtin_ownerStemName
gsh_builtin_gshReportRuntime.getGrouperReportData()	GrouperReportData	container for the output file (FILE) or csv rows (CSV)
gsh_builtin_gshReportRuntime.getGrouperReportData().getFile()	File	(FILE) file object to be written to
gsh_builtin_gshReportRuntime.getGrouperReportData().getHeaders()	List<String>	(CSV) column names to appear in csv header row
gsh_builtin_gshReportRuntime.getGrouperReportData().getData()	List<String[]>	(CSV) rows of data to appear in the csv; not set by default, so must be initialized with at least an empty list

Example script writing to FILE format

```
Group g = gsh_builtin_gshReportRuntime.ownerGroup
File file = gsh_builtin_gshReportRuntime.grouperReportData.file

file.withWriter('utf-8') { writer ->
    writer << ['Row', 'ID', 'UID', 'Name', 'Email'].join(",") << "\n"
    g.members.eachWithIndex { it, i ->
        writer << i+1 << ","
        writer << it.subject.getAttributeValue('employeenumber') << ","
        writer << it.subject.getAttributeValue('uid') << ","
        writer << it.subject.getAttributeValue('cn') << ","
        writer << it.subject.getAttributeValue('mail') << "\n"
    }
}
```

Example script writing to CSV format

```
Group g = gsh_builtin_gshReportRuntime.ownerGroup
GrouperReportData grouperReportData = gsh_builtin_gshReportRuntime.grouperReportData

grouperReportData.headers = ['Row', 'ID', 'UID', 'Name', 'Email']
grouperReportData.data = new ArrayList<String[]>()

g.members.eachWithIndex { it, i ->
    String[] row = [
            i+1,
            it.subject.getAttributeValue('employeenumber'),
            it.subject.getAttributeValue('uid'),
            it.subject.getAttributeValue('cn'),
            it.subject.getAttributeValue('mail'),
    ]

    grouperReportData.data << row
}
```

Mock setup of objects for development of a GSH reporting script

```
import edu.internet2.middleware.grouper.app.reports.GshReportRuntime
import edu.internet2.middleware.grouper.app.reports.GrouperReportData

def gs = GrouperSession.startRootSessionIfNotStarted().grouperSession

def g = GroupFinder.findByName(gs, "test:vpn:vpn_legacy_exceptions", true)
GshReportRuntime gshReportRuntime = new GshReportRuntime()
gshReportRuntime.ownerGroup = g
gshReportRuntime.ownerGroupName = g.name

GrouperReportData grouperReportData = new GrouperReportData()
gshReportRuntime.grouperReportData = grouperReportData

// (next line is for FILE output only, set to an arbitrary file instead of the autogenerated one)
grouperReportData.file = new File('/tmp/legacy_exceptions.csv')

// simulate the built-in variables
GrouperSession gsh_builtin_grouperSession = gs
GshReportRuntime gsh_builtin_gshReportRuntime = gshReportRuntime
String gsh_builtin_ownerStemName = gsh_builtin_gshReportRuntime.ownerStemName
String gsh_builtin_ownerGroupName = gsh_builtin_gshReportRuntime.ownerGroupName

/** continue from here with reporting script */
```

Internal attributes

The configuration will follow the same attribute structure as other Grouper modules like attestation and deprovisioning

Attribute definitions for config

Definition	Assigned To	Purpose	Value	Cardinality
reportConfigDef	folder, group	identify a report config	marker	Multi assign
reportConfigValueDef	folder assignment, group assignment	name/value pairs	string	Single assign, single valued

Attribute names for config

Name	Definition	Required?	Value
reportConfig Marker	reportConfig Def		<none>

reportConfigType	reportConfigValueDef	required (SQL and blank available)	Currently only SQL is available
reportConfigFormat	reportConfigValueDef	required (CSV and blank available)	Currently only CSV is available
reportConfigName	reportConfigValueDef	required	Name of report. No two reports in the same owner should have the same name
reportConfigFilename	reportConfigValueDef	required and shown for CSV type	e.g. usersOfMyService_$$timestamp$$.csv $$timestamp$$ translates to current time in this format: yyyy_mm_dd_hh24_mi_ss
reportConfigDescription	reportConfigValueDef	required	Textarea which describes the information in the report. Must be less than 4k
reportConfigViewersGroupId	reportConfigValueDef	optional	GroupId of people who can view this report. Grouper admins can view any report (blank means admin only), check if EveryEntity is in the group, then public
reportConfigQuartzCron	reportConfigValueDef	required	Quartz cron-like schedule
reportConfigSendEmail	reportConfigValueDef	required (default to true, no blank option available)	true/false if email should be sent
reportConfigEmailSubject	reportConfigValueDef	optional (default to generated subject, blank means use generated)	subject for email (optional, will be generated from report name if blank)
reportConfigEmailBody	reportConfigValueDef	optional (default to generated body, blank means use default, this should be a textarea, on submit, convert the newlines (/r/n, or /r, to standard \n)	optional, will be generated by a grouper default if blank body for email, support \n for newlines, and substitute in: $$reportConfigName$$, $$reportConfigDescription$$, $$subjectName$$ and $$reportLink$$ The link will go to the report instance screen for this report note: the $$reportLink$$ must be in the email template if it is not blank
reportConfigSendEmailToViewers	reportConfigValueDef	required if reportConfigSendEmail=true, default to true, no blank option	true/false if report viewers should get email (if reportConfigSendEmail is true)
reportConfigSendEmailToGroupId	reportConfigValueDef	required if reportConfigSendEmail=true and reportConfigSendEmailToViewers=false	if reportConfigSendEmail is true, and reportConfigSendEmailToViewers is false), this is the groupId where members are retrieved from, and the subject email attribute, if not null then send
reportConfigQuery	reportConfigValueDef	required and shown for CSV type	SQL for the report. The columns must be named in the SQL (e.g. not select *) and generally this comes from a view
reportConfigEnabled	reportConfigValueDef	default to true (required, no blank option)	Use logic from loader enabled, either enable or disabled this job

Attribute definitions for instance (a report that was run)

This attribute is assigned to the same owner as the config attribute (e.g. the same group/folder)

Definition	Assigned To	Purpose	Value	Cardinality
reportInstanceDef	folder, group	identify a report that was run	marker	Multi assign
reportInstanceValueDef	folder assignment, group assignment	name/value pairs	string	Single assign, single valued

Attribute names for instance

Note: the ID is the attribute assign id of the marker (this is passed in URLs/emails etc)

Name	Definition	Value
reportInstanceMarker	reportInstanceDef	<none>
reportInstanceStatus	reportInstanceValueDef	SUCCESS means link to the report from screen, ERROR means didnt execute successfully
reportElapsedMillis	reportInstanceValueDef	number of millis it took to generate this report
reportInstanceConfigMarkerAssignmentId	reportInstanceValueDef	Attribute assign ID of the marker attribute of the config (same owner as this attribute, but there could be many reports configured on one owner)

reportInstanceMillisSince1970	reportInstanceValueDef	millis since 1970 that this report was run. This must match the timestamp in the report name and storage
reportInstanceSizeBytes	reportInstanceValueDef	number of bytes of the unencrypted report
reportInstanceFilename	reportInstanceValueDef	filename of report
reportInstanceFilePointer	reportInstanceValueDef	depending on storage type, this is a pointer to the report in storage, e.g. the S3 address. note the S3 address is .csv suffix, but change to __metadata.json for instance metadata
reportInstanceDownloadCount	reportInstanceValueDef	number of times this report was downloaded (note update this in try/catch and a for loop so concurrency doesnt cause problems)
reportInstanceEncryptionKey	reportInstanceValueDef	randomly generated 16 char alphanumeric encryption key (never allow display or edit of this)
reportInstanceRows	reportInstanceValueDef	number of rows returned in report
reportInstanceEmailToSubjects	reportInstanceValueDef	source::::subjectId1, source2::::subjectId2 list for subjects who were were emailed successfully (cant be more than 4k chars)
reportInstanceEmailToSubjectsError	reportInstanceValueDef	source::::subjectId1, source2::::subjectId2 list for subjects who were were NOT emailed successfully, dont include g:gsa groups (cant be more than 4k chars)

Changes to group or folder drop down

Under folders or groups, in the more actions, should be "Reports", which goes to View reports screen. Note we need to harmonize this with Shilen's group and folder reports. Should they share a menu item?

View reports screen

This is the default screen.   Drop down with the following options:

1. View reports
2. Edit reports

Screen shows

1. For all the configured reports, see if the current user can view them (wheel group or in the reportViewers group for the report), and if so, list the reports there, one line per report, with a link to the report page, and a link to the latest report download
2. If there are no reports available, display a message "There are no reports you are allowed to view"
3. Column for report name (clickable to report screen)
4. Column for if enabled or not
5. Column for last timestamp it was run (from most recent report instance attribute)
6. Column for status (SUCCESS?) (from most recent report instance attribute)
7. Column for number of rows in report (from most recent report instance attribute)
8. Column for cron schedule (from most recent report config attribute)
9. Column with drop down to download most recent report, view most recent report instance, report (report screen), report logs, enable /disable (group admins or wheel only)

Edit reports screen

1. Only for wheel group
2. Can pick a report to edit or can add a new. Like the deprovisioning edit
3. Drop down same as view reports screen

Report screen

1. Can see if wheel user or in the reportViewers group for the report
2. Show the report name and description
3. Show table with most recent 100 entries with columns (from report instance attributes)
    a. Report name (same for each row)
    b. Timestamp (sorted descending)
    c. Download report link
    d. View report details link (goes to report instance screen)
    e. Status
    f. When the report run
    g. How many rows in report
4. Show the settings in read only mode

5. In right of screen have one actions drop down: download most recent report, view reports (report screen), report logs, enable/disable (group admins or wheel only), delete report (group admins or wheel only)

Report instance screen

1. Clickable from Report screen or drop down in some of the report screens or from email to user
2. Dropdown on right of screen: Download most recent report, view reports (report screen), report logs, enable/disable (group admins or wheel only)
3. Show timestamp
4. Show the report name and description
5. Download link (if status SUCCESS)
6. Show report instance attributes
    a. Friendly size of unencrypted report (e.g. 150kb or 1.5mb, there is a commons file utils method to generate this)
    b. Filename
    c. Row count
    d. Download count
    e. List of subject names that were emailed successfully (comma separated)
    f. List of subject names that were emailed unsuccessfully (comma separated)
    g. (only to wheel users) First 3 chars of encryption key (mask with commons util method with asterisks: b4W****************)

Report logs screen

1. Clickable from Report screen or drop down in some of the report screens
2. Can see screen if wheel user or in the reportViewers group for the report
3. Show list of most recent 100 report logs from grouper loader log table
4. Should be a table that looks like the grouper loader log screen
5. Should have exception stack if there was an error

Report

The report will take the SQL and columns and make a CSV with all the results. Chris has this logic and will commit it in the branch. This will be delivered as a download from browser

Report emails

If reports are being configured to be emailed, then the configured or default email will be sent. Note, the actual report will not be attached in the email for security reasons. A link to the report instance screen will be in the email.

Report storage

In 2.4 we dont want to add a new table to store files, so for people who want to use this feature the only option will be AWS S3 buckets or filesystem with the report encrypted. We can add more storage options later

In 2.5.34+ this is stored by default in the database.

Database

Stores in grouper_file table

AWS S3

The deployer will need an AWS account, the free level might suffice

Need to configure the AWS creds in grouper.properties

- You might create an IAM user or role, or have best practices, but to get started quickly do this
- In the upper right click on your aws console username, "my security credentials"
- Under access keys, create new access key

Configure the AWS S3 bucket location

- Note the region
- You can keep all the settings as the default since we dont need versions and Grouper does its own encryption
- Do you not allow public access to bucket

Filesystem

Configure the path where report files will be stored

Generic storage info

Inside there Grouper will create "folders"

$base$/reports/YYYY/MM/DD/$someUniqueId$/$reportFilename$.csv.encrypt

- $groupName$ is the group system name
- $groupId$ is the group UUID
- $reportInstanceId$ is the attribute assign id of the marker for the instance of the report
- This is the encrypted CSV report

$base$/reports/YYYY/MM/DD/$someUniqueId$/$reportFilename$__metadata.json

- This is the JSON of the report instance attribute values
- Do NOT include the encryption key
- This exists so we can see metadata about each report, e.g. size, without downloading it

Report encryption

- 16 char alphanumeric encryption key will be generated for each report
- Use the rijndael algorithm like password encryption

To delete a report instance, delete the metadata and report data from storage. If not it will be deleted eventually with a clean up daemon

When a report is deleted, delete all the metadata and report data from storage.   If not it will be deleted eventually with a clean up daemon

There are no direct links to reports, and they are encrypted anyways. The only way to download reports is through the Grouper UI (or API), by authorized users. This is a reverse proxy to the report storage.

Overall report daemon

The overall report daemon should go through storage, and

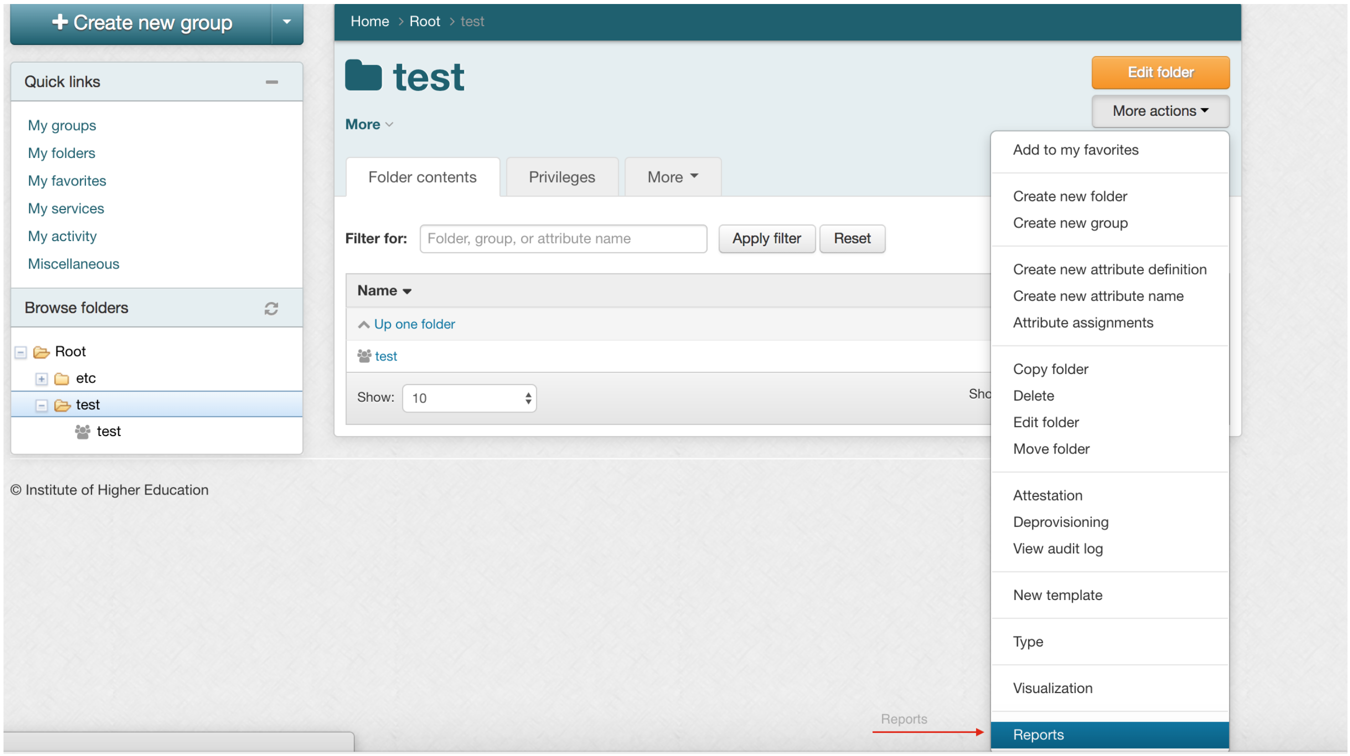
1. Look for reports that have more than 100 instances, and delete the older ones (reports and metadata)
2. Look for reports older than 30 days and delete (reports and metadata)

Auditing

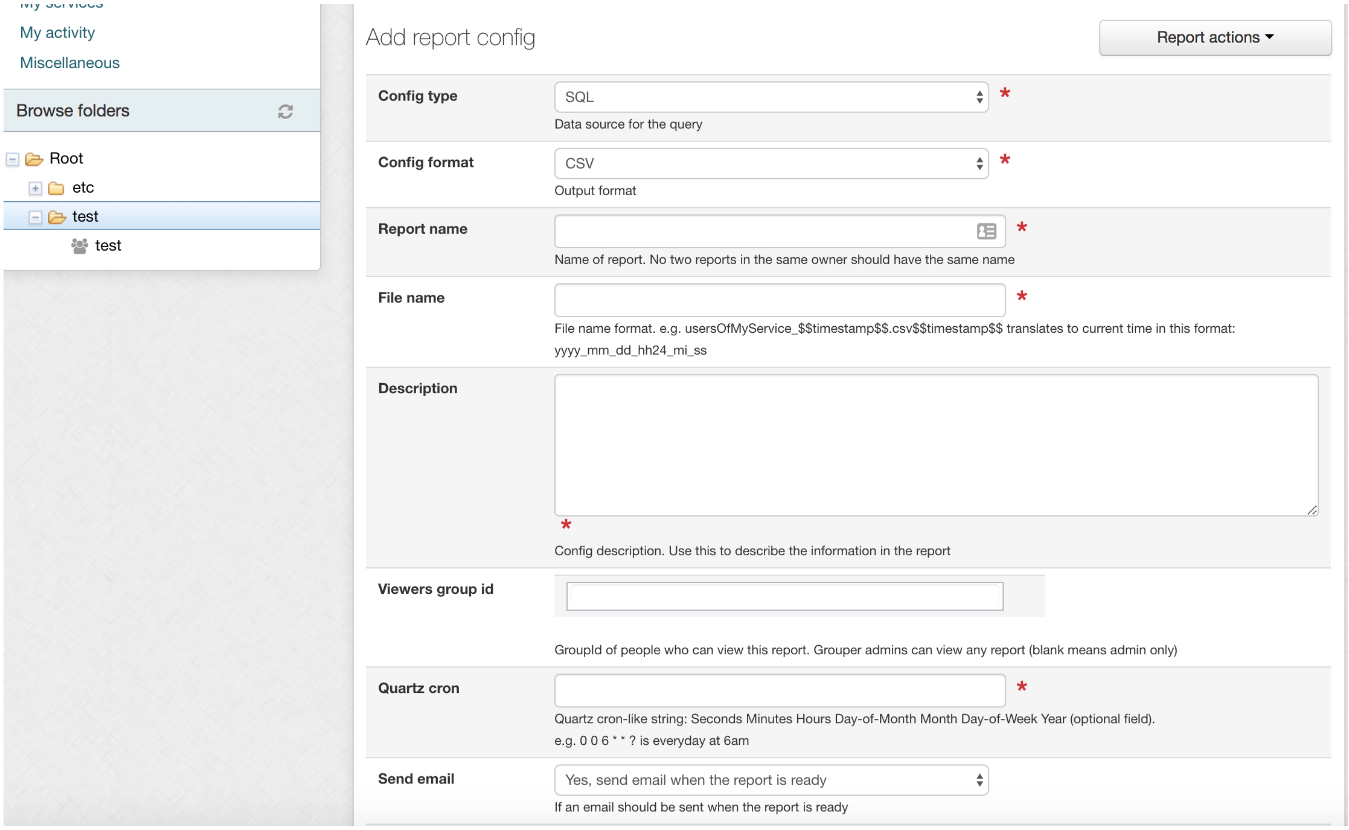
Audits should be added for reports creation/editing/downloading. No audits for emails sent. These audits should be linked to the group or folder where the report is configured

Screenshots

A new item "Reports" is available in More actions dropdown.



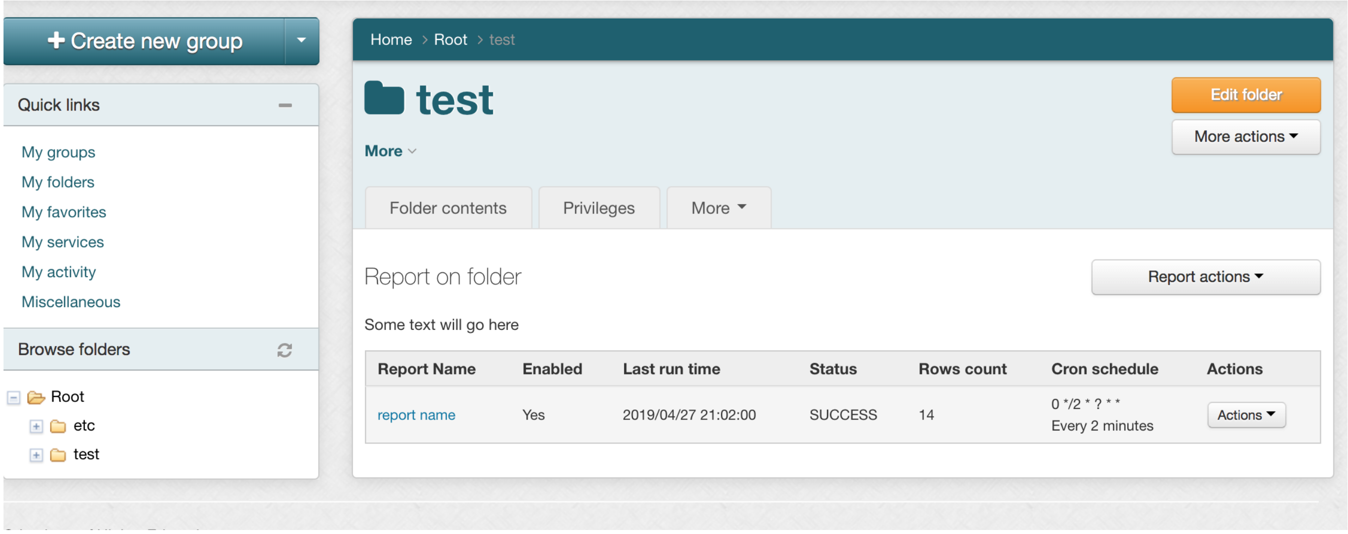
Grouper admins can add new reports as shown in the screenshot below



Add report config

Config type	SQL * Data source for the query
Config format	CSV * Output format
Report name	* Name of report. No two reports in the same owner should have the same name
File name	* File name format. e.g. usersOfMyService_$$timestamp$$.csv$$timestamp$$ translates to current time in this format: yyyy_mm_dd_hh24_mi_ss
Description	* Config description. Use this to describe the information in the report
Viewers group id	GroupId of people who can view this report. Grouper admins can view any report (blank means admin only)
Quartz cron	* Quartz cron-like string: Seconds Minutes Hours Day-of-Month Month Day-of-Week Year (optional field). e.g. 0 0 6 * * ? is everyday at 6am
Send email	Yes, send email when the report is ready If an email should be sent when the report is ready

The screenshot below shows the existing reports.

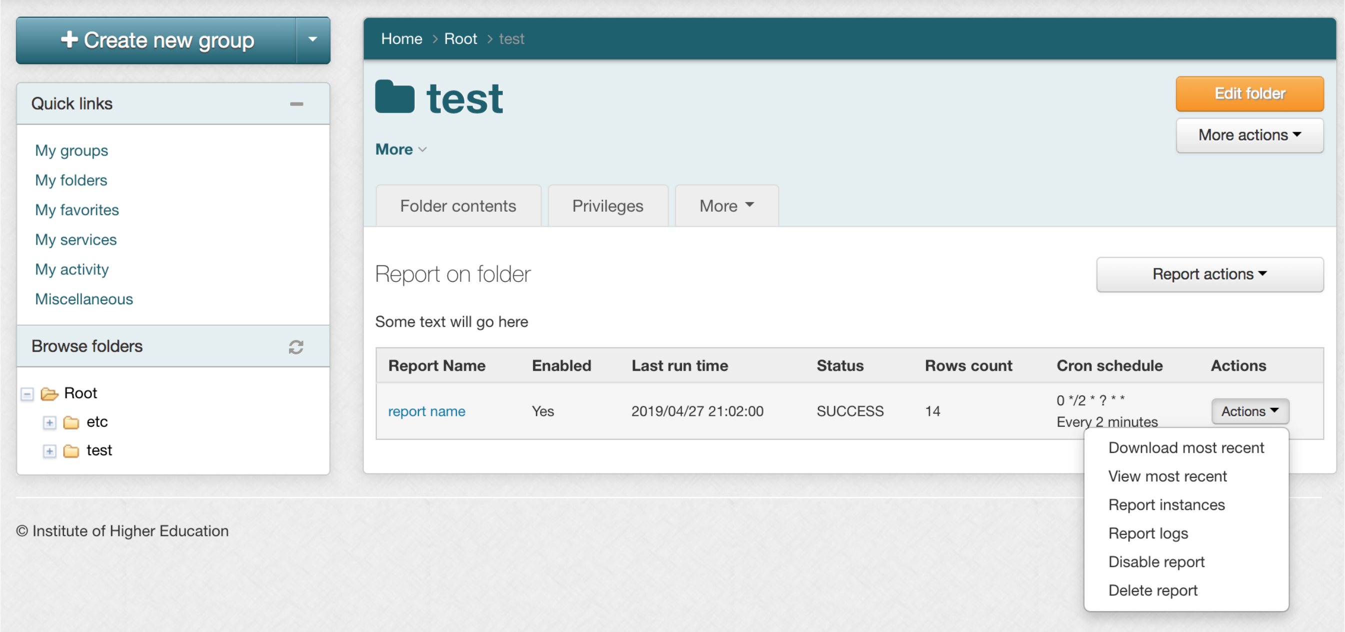


Report on folder

Some text will go here

Report Name	Enabled	Last run time	Status	Rows count	Cron schedule	Actions
report name	Yes	2019/04/27 21:02:00	SUCCESS	14	0 */2 * ? * * Every 2 minutes	Actions

For each report config, a few actions are available as shown in the screenshot below



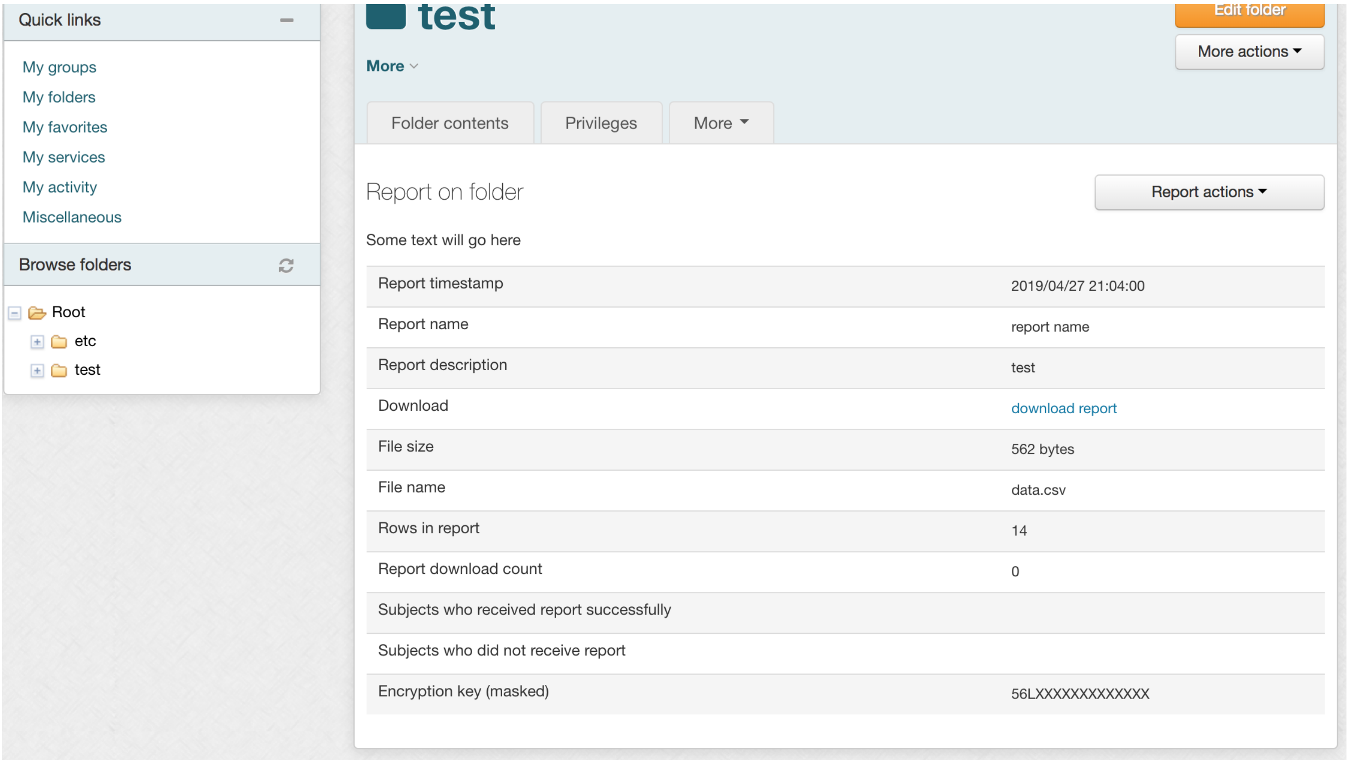
Report on folder

Some text will go here

Report Name	Enabled	Last run time	Status	Rows count	Cron schedule	Actions
report name	Yes	2019/04/27 21:02:00	SUCCESS	14	0 */2 * ? * * Every 2 minutes	Actions

© Institute of Higher Education

A report can be downloaded from the report instance page as shown in the screenshot below



Report on folder

Some text will go here

Report timestamp	2019/04/27 21:04:00
Report name	report name
Report description	test
Download	download report
File size	562 bytes
File name	data.csv
Rows in report	14
Report download count	0
Subjects who received report successfully	
Subjects who did not receive report	
Encryption key (masked)	56LXXXXXXXXXXXXX

FAQ

1. As a grouper admin, how do I get the correct SQL?
    a. We will publish a lot of examples and increase the number of views Grouper has
2. Can I join to external person tables to get extra attributes
    a. Absolutely, if you can ETL the data to your grouper database or maybe join over DB link then you will have extra attributes. If people need LDAP attributes we can discuss that
    b. Note: columns in the report could also be group memberships (e.g. enrolled in MFA? enrolled in Duo push? required to be in MFA? etc)

To Do later

- Add entity relationship diagram on wiki to help with making queries
- Add ability to use SCP to store reports
- Add ability to use SCP to send reports
- Add ability to store reports in box folder
- Errors in report should be logged and throw error but maybe also store error in txt report (not sent out or available except to admins)
- Add diagnostics to test that a report is setup correctly
- Add paging to report instance list
- Configure how long reports are stored
- Screen in a user's subject screen that shows all the reports they have access to
- Centralized report dashboard
- Have a config option to "run now" (allows report viewers to run now)
    - This would send a message to a daemon to run so it doesnt run in the UI
    - Like Loader "run now"
- Allow another report type which runs off membership list (not straight SQL)
    - Allow non admins to configure?
    - Allow more columns to be added (join other database tables if allowed)
- Allow reports from GSH / java?
- Allow reports from WS as user
- Add another output type for JASPER report (PDF, etc)
- Support excel
- Add ability to display a CSV in the JSP in an HTML table
    - Add metadata to make it clickable?
- Allow fields to be added from an LDAP filter
- Add the daily Grouper report to run like this (dont email)
- Email batching per user (user gets a weekly digest about their reports)?
- In another pass we could create a report based on loading/provisioning.
- On the Daemons job screen, show user friendly names for jobs (pull grouper object name from id and show under the report job id)

See Also

Grouper Report showing summary of your installation

Grouper Reporting Blog