

Sharing and Permissions

Overview

One of the most important features in Catalog is the ability to decide which users will have access to the data. This can be easily achieved thanks to the new Access Control List (ACLs) that have been defined, providing a fully customizable authorization mechanism.

Each member, either a user or a group, can have associated a list of predefined permissions that are needed in order to have granted access to the data. Permissions can be assigned to almost any entry level except for *User* and *Project*. The permissions that can be assigned can be divided into three different main groups:

Owner

The owner of a study will always be able to perform any action over the data contained in the study. The kind of actions that will only be possible to be performed by the owner are **deleting the study** and **assign or remove users to/from the admins groups** (next subsection).

Administrative groups

OpenCGA defines two reserved groups that will have some special behaviour.

Admins

Every *Study* in OpenCGA contains a special group called **admins**. This group will contain a list of users that will be able to do most of the administrative work the owner might want other users to do. Users belonging to this group will be able to perform almost any action except for the two ones that are only allowed for the owner of the study. Special operations that only these users will be able to perform are **create/update/delete groups of users**, **create/update/delete variable sets** and **assign/remove permissions to other users/groups**.

Members

Apart from *admins*, there is also a special group called *members*. Any user with any kind of granted access to the study will automatically belong to this group. The main aim of this group is to keep track of the users with any access to the study, but it also has other advantages such as:

- The *admin* users might want to predefine some permissions any *member* of a study will have. In such a case, *admin* users will just add new users to that group and those users will automatically be granted the permissions the group has.
- If an *admin* user wants to completely revoke any permission to one user, by removing that user from the *members* group, OpenCGA will automatically search for any permissions set for that user in any entity and remove it.

Other groups

Any other user belonging to other groups will be able to be granted the rest of permissions that are not for *admins* or *owners*.

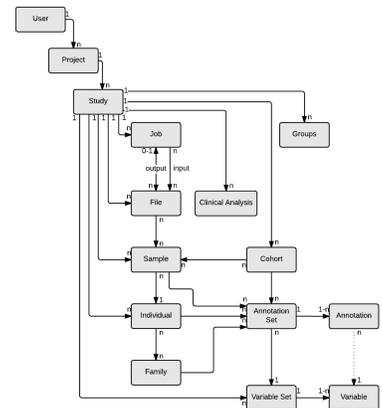
Summary Table

Category	Permission	Users and Groups	Admins (users in admins group)	Owner
Study	Delete	✗	✗	✓
	Create new admins	✗	✗	✓
	Groups	✗	✓	✓
	Variable Sets	✗	✓	✓
	Set permissions (<i>share</i>)	✗	✓	✓
	Run analyses (Execution)	✓	✓	✓
Job	View, Write and Delete	✓	✓	✓
File	View, Write and Delete	✓	✓	✓

Table of Contents:

- Overview
 - Owner
 - Administrative groups
 - Admins
 - Members
 - Other groups
 - Summary Table
- How it works
 - Templates
- Special cases
 - Files
 - Individuals/Samples
- Use cases
 - Give public access to non-existing users

Remember Catalog Data Models:



	View Header	✓	✓	✓
	Download	✓	✓	✓
Sample	View, Write and Delete	✓	✓	✓
	Annotations (<i>clinical data</i>)	✓	✓	✓
Individual	View, Write and Delete	✓	✓	✓
	Annotations (<i>clinical data</i>)	✓	✓	✓
Family	View, Write and Delete	✓	✓	✓
	Annotations (<i>clinical data</i>)	✓	✓	✓
Cohort	View, Write and Delete	✓	✓	✓
	Annotations (<i>clinical data</i>)	✓	✓	✓
Clinical Analysis	View, Write and Delete	✓	✓	✓

How it works

Permissions can be associated to almost any entry level except for *User* or *Project*. Entries that can have permissions associated are *Study*, *File*, *Sample*, *Job*, *Individual*, *Cohort*, and *Family*.

A list of the basic permissions and their explanations can be found in the list below:

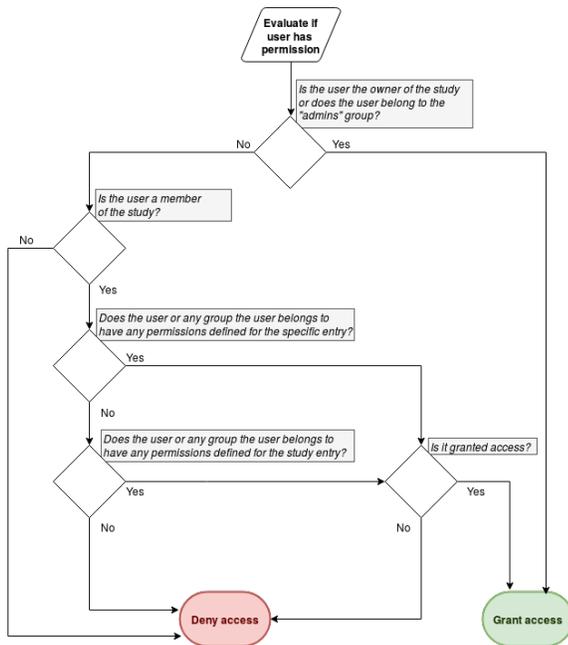
- **VIEW_***: Give permission to access in read-only mode to the entry (study, file, sample...).
- **WRITE_***: Give permission to create and update that kind of entries within the study. This do not include permissions to modify annotation and/or annotation sets. Those actions will need additional permissions.
- **DELETE_***: Give permission to delete that kind of entries.
- ***_ANNOTATIONS**: In *Sample*, *Individual*, *Family* and *Cohort* we have three additional permissions to *create*, *write* or *delete* annotations in the same way it was described before.

Files deserve a special treatment as they not only exist in the database, but also physically in the file system. The special permissions added for files are the following:

- **VIEW_FILE_HEADERS** or **VIEW_HEADER**: Give permission to retrieve just the header of a file.
- **DOWNLOAD_FILES** or **DOWNLOAD**: Give permission to download the whole file.

The permission names might differ depending the entry they are related to. For instance, if we would like one user to have read-only permissions to all the files, samples, cohorts... the permissions **VIEW_FILES**, **VIEW_SAMPLES**, **VIEW_COHORTS**, etc should be set for the user. However, if we only want that user to have permission in just one file, sample or cohort, only the **VIEW** permission will be needed to that concrete file, sample or cohort.

Taking into account that permissions can be defined at many different levels and given for users and groups, we need to establish an algorithm to decide if permissions will be granted or not. A summary of that decision algorithm can be found below:



There are some special circumstances under which the algorithm behaves as follows:

- If the user and any of the groups where the user belongs to have permissions defined for one entry, the permissions that will be actually used will be the user's.
- In case the user belongs to more than one group and those groups are assigned different permissions for one concrete entry, the effective permissions that will be used will be the union of the permissions found in all those groups.

All the permissions OpenCGA defines are "positive" permissions, designed specifically to grant access. However, study owners or admins can deny access to users or groups by explicitly defining "null" permissions to them. This can be better understood in the following table:

Study permission	Sample permission	Description
X	VIEW	The user has permissions at the most specific entry, so the user will be granted permissions.
VIEW_SAMPLES	X	The user doesn't have any permission specified at the sample entry, so we would check the permissions defined at the study entry. In this case, the user have the generic "VIEW_SAMPLES" permission, so the user will be granted permission.
VIEW_SAMPLES	NONE	The user have permissions defined at the sample entry level, but it lacks the VIEW permission, so the user will directly have revoked permissions for that specific sample.
NONE	X	The user doesn't have any permission specified at the sample entry, so we would check the permissions defined at the study entry. In this case, the user has permissions defined at the study level but it lacks the VIEW_SAMPLES permission, so the user will have revoked permission for that sample.
X	X	The user doesn't have any permission specified neither at the <i>Sample</i> nor at the <i>Study</i> level, so it will have revoked permissions for that sample (default behaviour).

Templates

We have created two sets of predefined *roles* to assign some generic *Study* permissions to users/groups, **analyst** and **view_only**:

- **analyst**: The user or group will be given full READ and WRITE (not DELETE) permissions for all the entries related to the study. These users will be able to view and do modifications on all the data that is related to the study.

- **view_only**: The user or group will be given full READ permissions.

Special cases

Permissions can be given to any concrete entity (file, sample, cohort...) to deny or grant access to just one concrete entry. This is always true except for a few exceptions in which we might propagate those same permissions to other entries:

Files

File entry might be of type file or folder. Permissions assigned in folders are propagated to all the children (files and folders) recursively.

⚠ All permissions that might have had files and folders under the folder being given permissions will be modified according to the action being performed in the parent folder. In other words, if we are setting new permissions for the folder, any possible permissions the files and folders under the parent folder might have had will be completely replaced by the parent folder's permissions. However, if the action being performed is just adding a new permission to the parent folder, children files and folders will keep their old permissions plus the new one(s) added to the parent folder.

Individuals/Samples

Individuals are really strong related with samples. So every time permissions are given to an individual, the same permissions can be applied to all the related samples if the user sets the 'propagate' field to True, and vice-versa.

Use cases

Give public access to non-existing users

Catalog has one special user for this purpose represented with * symbol. Anytime a user tries to fetch anything and no session id is provided, Catalog will treat that user as *. By default, only authorised users will have access to data. However, study managers can still define permissions for non-authenticated users assigning permissions to the "user" *.