# Server 2.0

## Models resource overview:

We used the following schema:

Corpus

\_id

name

description

history

ACL

Layer

\_id

id\_corpus

name

description

fragment\_type

data\_type

history

ACL

Media

\_id

id\_corpus

name

description

url

history

Annotation

\_id

id\_layer

id\_media

fragment

data

history

∞

∞

∞

∞

1

1

1

1

* Corpus: a data collection with specific rights (ACL) for each user/group
  + name: string
  + description: json
  + history: automatic
  + ACL: json
* Media: media belonging a corpus, rights are inherited from corpus level
  + id\_corpus: string
  + name: string
  + description: json
  + history: automatic
* Layer: layer (speaker id, speech transcription, etc.) for a corpus, with specific rights (ACL) for each user/group
  + id\_corpus: string
  + name: string
  + description: json
  + fragment\_type: json
  + data\_type: json
  + history: automatic
  + ACL: json
* Annotation: data associated to a segment of a layer for a media, rights are inherited from layer level
  + id\_layer: string
  + id\_media: string
  + fragment: json
  + data: json
  + history: automatic

## Resources access:

Short list :

* Add/update/delete a corpus/layer/media/annotation
* Get whole list of corpus/layer/media/annotation
* Get list of medias/layers of a corpus
* Get list of annotation of a layer (optionally belong to a media)
* Get information on a corpus/layer/media/annotation
* Update/remove ACL for corpus/layers of users/groups
* Add a layer with several annotations in only one request
* Read video stream of a media
* …

The whole list can be found at: <http://camomile-project.github.io/camomile-server>

## Management of users and groups:

The user role can be “user” or “admin”, an admin can:

* Create/update user and group
* Get all group belong to a user
* Add/remove user to a group

Only the “root” user can delete user and group.

## Rights management:

Access rights are specified for corpus and layer, media inherit corpus rights and annotation inherit layer rights. Rights are specified for a each user or group. User rights overpass group rights.

Right type:

* R: reader of a resource:
  + Can get information
  + Read media stream
* W: writer of a resource:
  + Have R rights
  + Can add and delete sub-resources into this resource
* O: owner of a resource:
  + Have W rights
  + Can update and delete it (for corpus only ‘root’ is allowed to delete)
  + Can update, delete sub-resources into this resource
  + Read and update user / group rights

Group

\_id

name

description

users\_list

User

\_id

username

description

role

Corpus/Layer

\_id

…

ACL

∞

∞

∞

1

1

1

## History:

After each update, the resource history are completed with a json containing:

* date: server date
* id\_user: id of the user that do the update
* modification: json of the new state of the ressource

## Queue:

List of resources to be treated

* name: queue name
* description: json
* list: free

## Access right of Queue:

* Only ‘root’ user can get all queues
* All logged in user can (with \_id of the queue):
  + Get info on a queue
  + Update a queue
  + Push into a queue
  + Pop a queue
* Only admin users can delete a queue

## Tools:

* Get server date

## Resources:

* Camomile server2.0

<https://github.com/camomile-project/camomile-server>

(branch server-develop-2.0)

* Python client:

<https://github.com/camomile-project/camomile-client-python>

(branch server2.0)

Some example scripts are in the tutorial folder

* Documentation for python and shell client (javascript coming soon):

<http://camomile-project.github.io/camomile-server>