tomolog-cli Documentation

Release 0.1

Argonne National Laboratory

Nov 03, 2022

CONTENTS

1 Installation 3

2 Usage 15

3 API reference 17

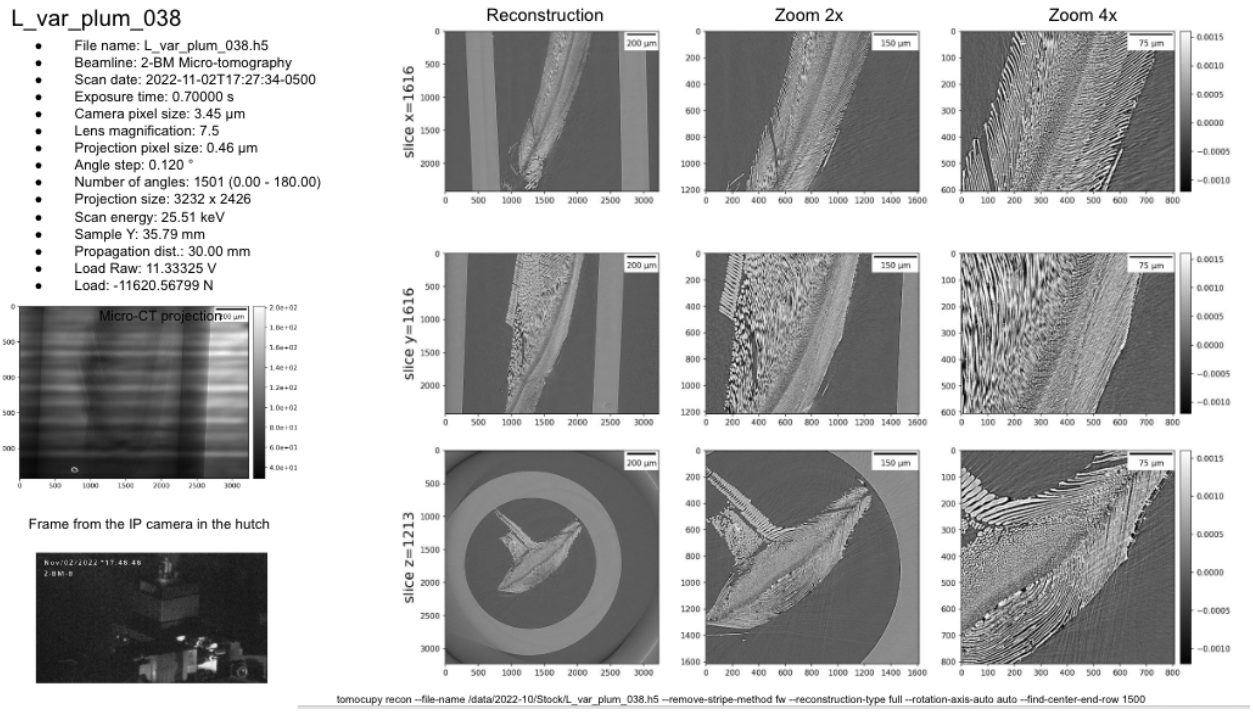
4 Credits 21

Bibliography 23

Python Module Index 25

Index 27

tomolog-cli is a command-line-interface for publishing tomography experiment data and meta data in a stack of Google slides



CHAPTER

ONE

INSTALLATION

First, you must have Conda installed and create a dedicated conda environment:

```
(base)$ conda create -n tomolog python=3.9
```

and:

```
(base)$ conda activate tomolog
(tomolog)$
```

then install all requirements with:

```
(tomolog)$ conda install  -c conda-forge python=3.9 dropbox google-api-python-client␣
↪matplotlib dxchange dxfile python-dotenv opencv matplotlib-scalebar
```

install meta

```
(tomolog)$ git clone https://github.com/xray-imaging/meta.git
(tomolog)$ cd meta
(tomolog)$ python setup.py install
```

and install tomolog

```
(tomolog)$ git clone https://github.com/xray-imaging/tomolog-cli.git
(tomolog)$ cd tomolog
(tomolog)$ python setup.py install
```

1.1 Requirements

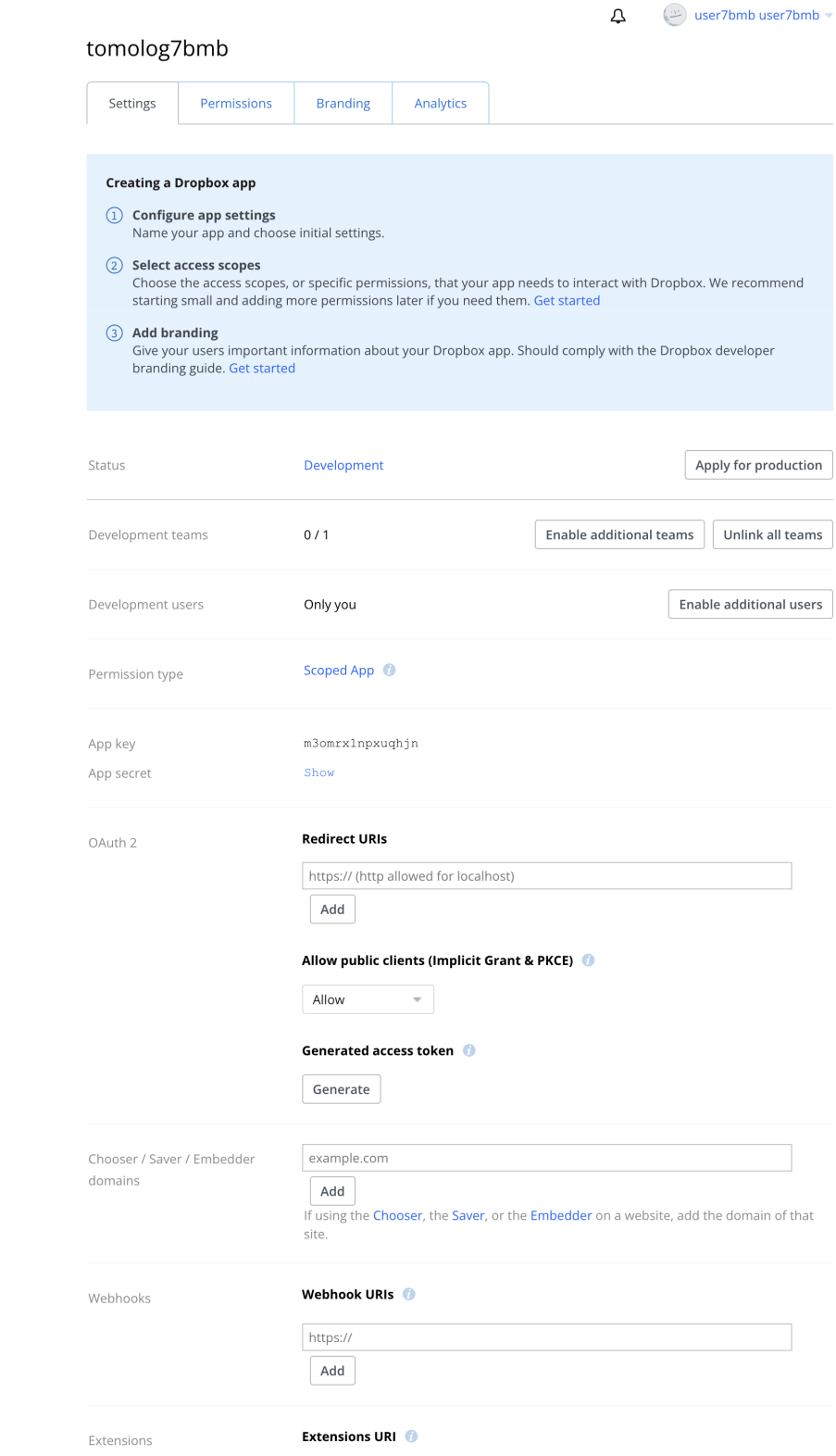
Please install all the packages listed in requirements file.

tomolog also requires access tokens from dropbox and google services.

1.1.1 Dropbox

Go to dropbox developer site , login using your google credentials and select “Create an App”:

Take the App key and App secret from the Settings tab:

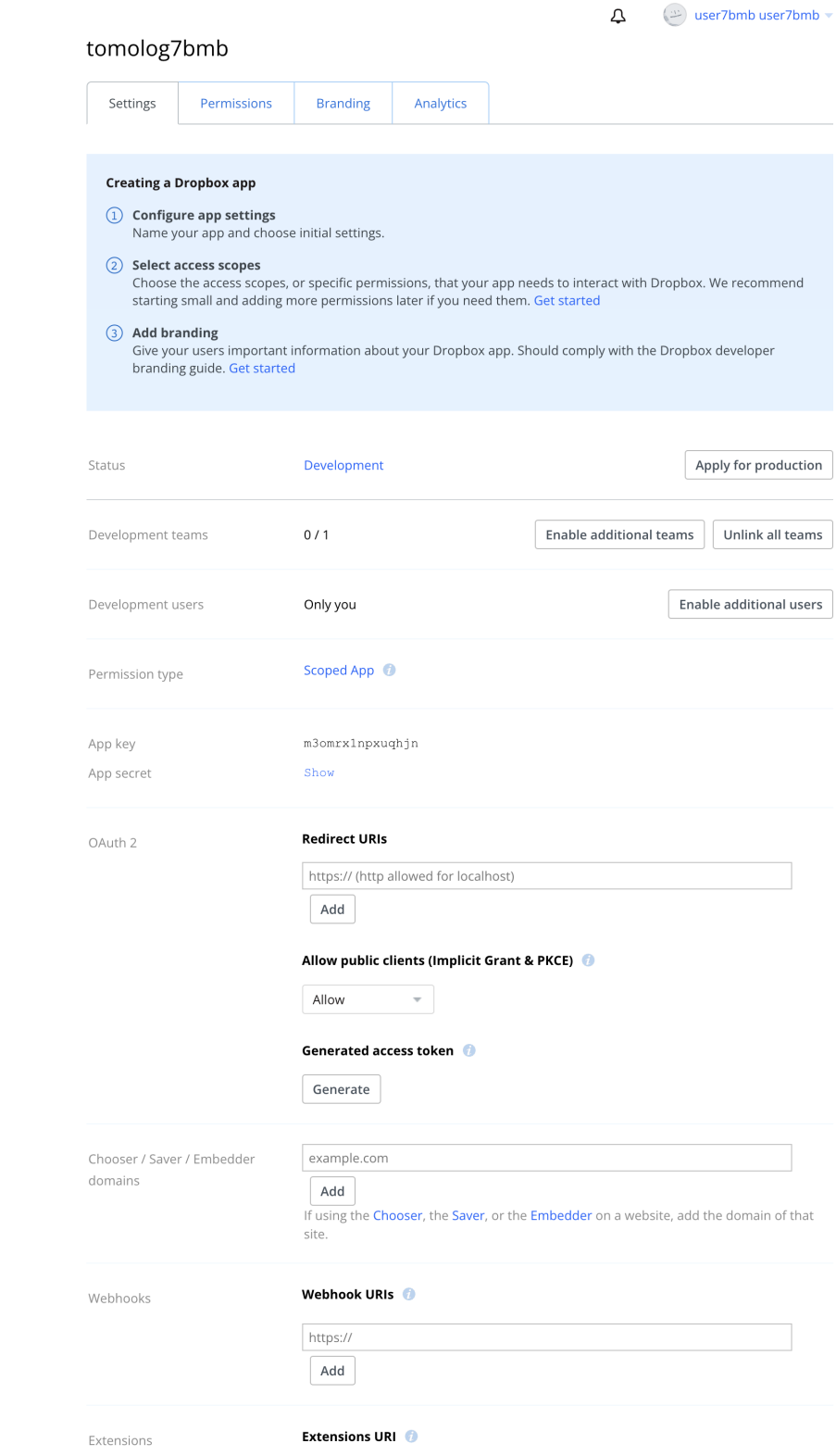


an copy them in a file in your home directory called:

```
$ ~/.tomologenv

APP_KEY=....
APP_SECRET=....
```

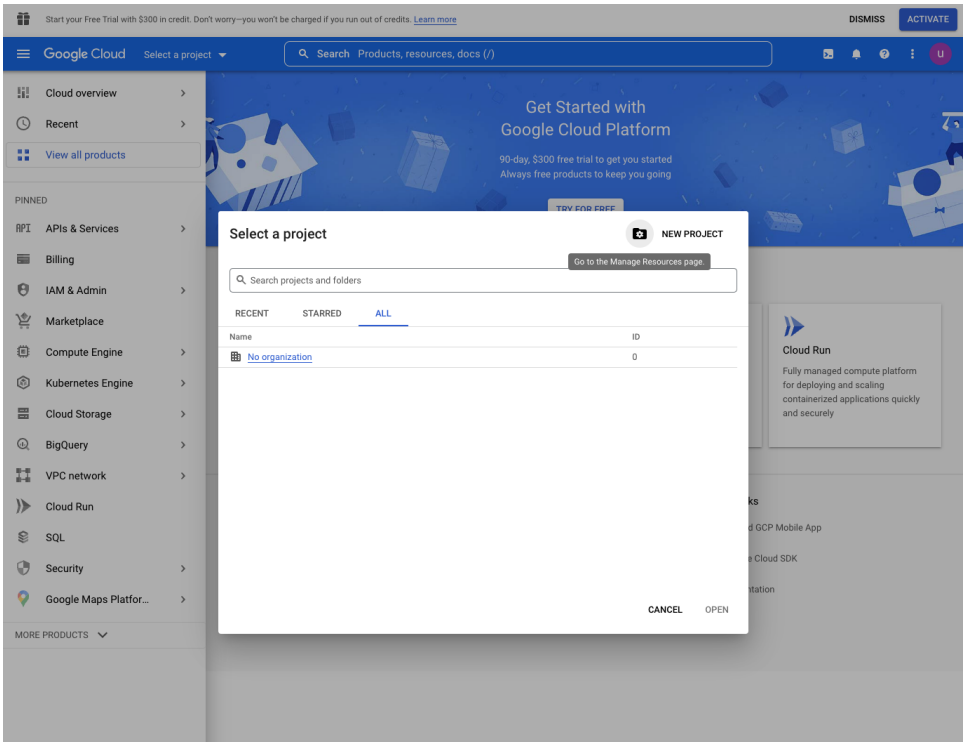
Set the following permissions:

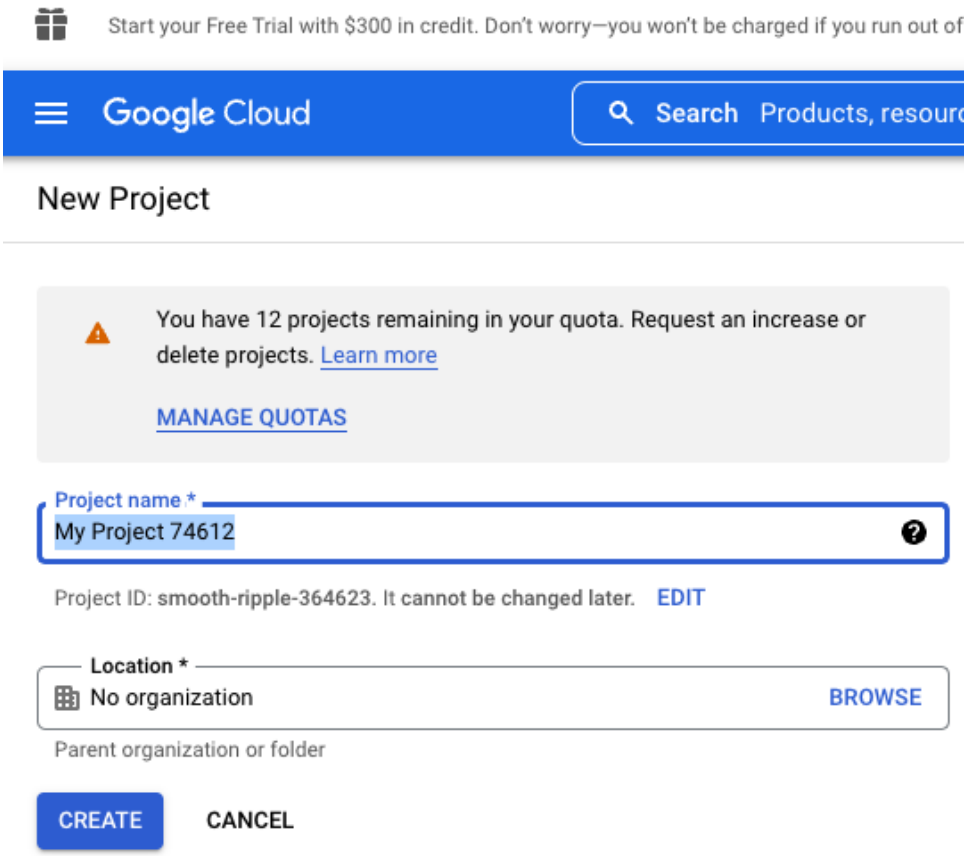

user7bmb user7bmb
tomolog7bmb
Settings
Permissions
Branding
Analytics
Creating a Dropbox app
1 Configure app settings
Name your app and choose initial settings.
2 Select access scopes
Choose the access scopes, or specific permissions, that your app needs to interact with Dropbox. We recommend starting small and adding more permissions later if you need them. Get started
3 Add branding
Give your users important information about your Dropbox app. Should comply with the Dropbox developer branding guide. Get started
Status
Development
Apply for production
Development teams
0 / 1
Enable additional teams
Unlink all teams
Development users
Only you
Enable additional users
Permission type
Scoped App
App key
m3omrx1npxuqhjn
App secret
Show
OAuth 2
Redirect URIs
https:// (http allowed for localhost)
Add
Allow public clients (Implicit Grant & PKCE)
Allow
Generated access token
Generate
Chooser / Saver / Embedder domains
example.com
Add
If using the Chooser, the Saver, or the Embedder on a website, add the domain of that site.
Webhooks
Webhook URIs
https://
Add
Extensions
Extensions URI

1.1.2 Google

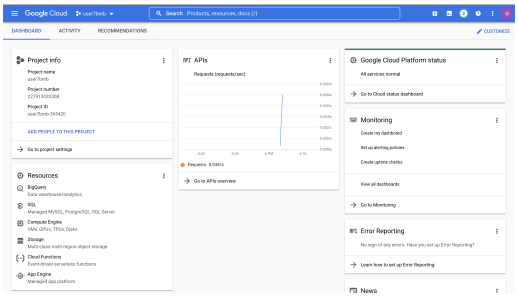
Next step is to authorize tomolog to create slides on the . . . @gmail.com gmail account.

Open a web browser and login as . . . @gmail.com then go to google developer site and press "Select a project" to create a new project

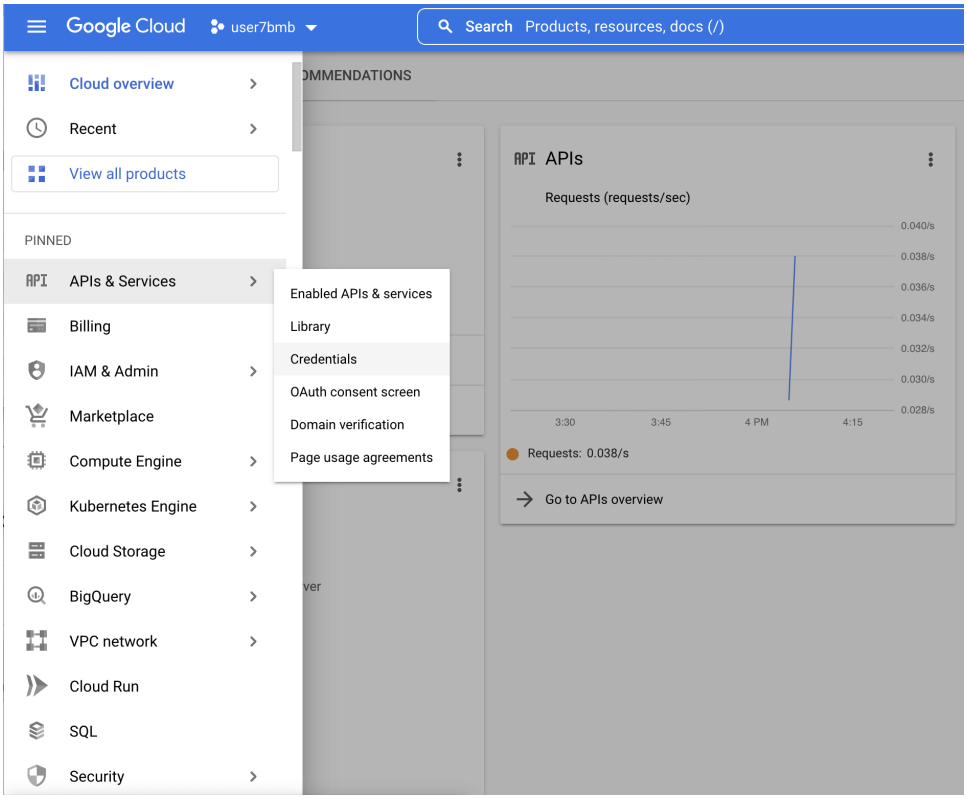




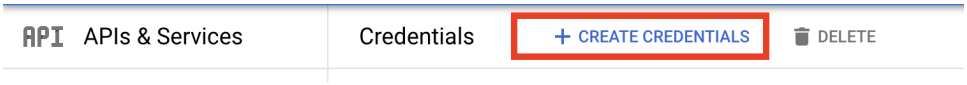
Once the project is selected go to the project dashboard:



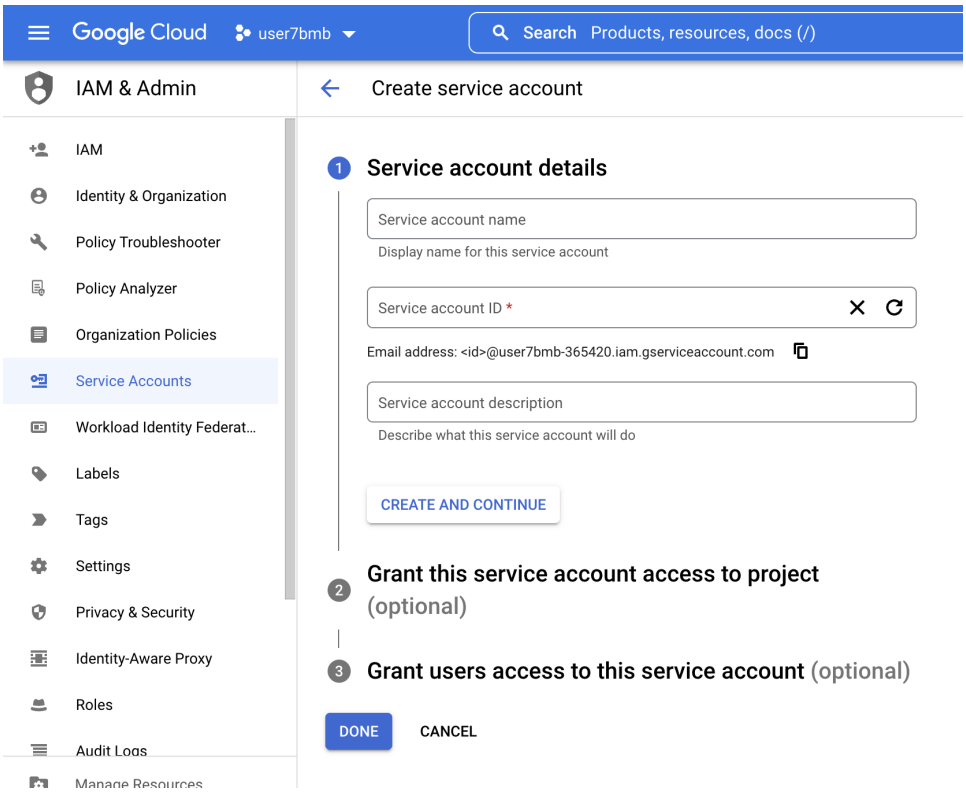
and select API & Services / Credentials



select it and then go to Create Credentials / Service account



where you enter the service account name, i.e. tomolog

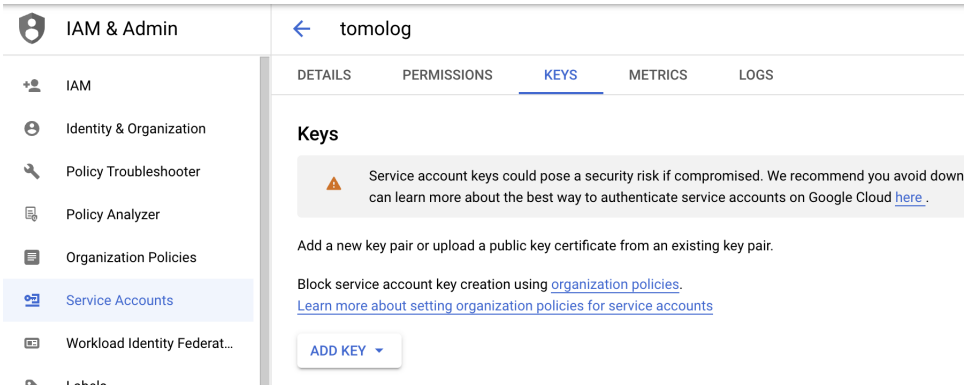


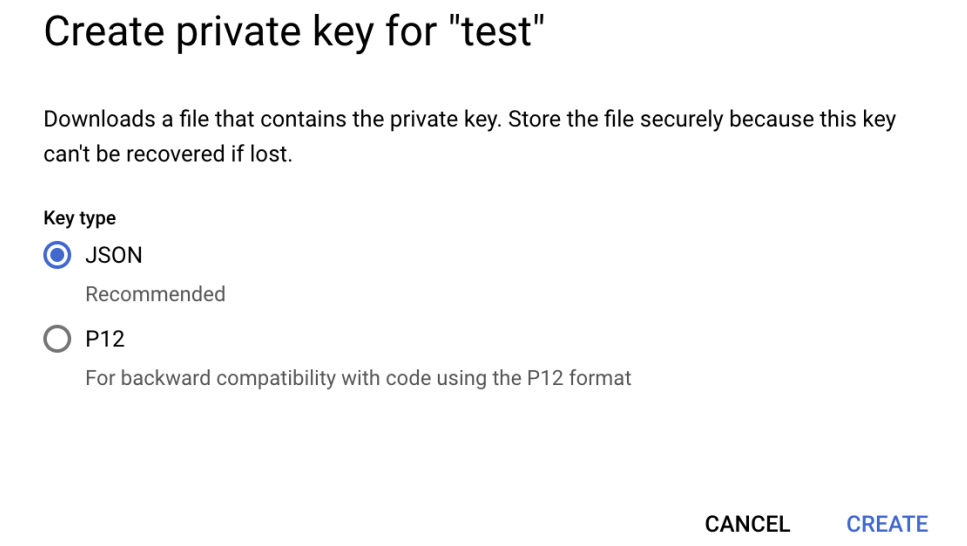
Grant this service account access to project: Owner

Grant users access to this service account:

```
service account user role: Google Account email: ....@gmail.com
service account admin role: Google Account email: ....@gmail.com
```

Once the service account is selected you need to create the authorization key:

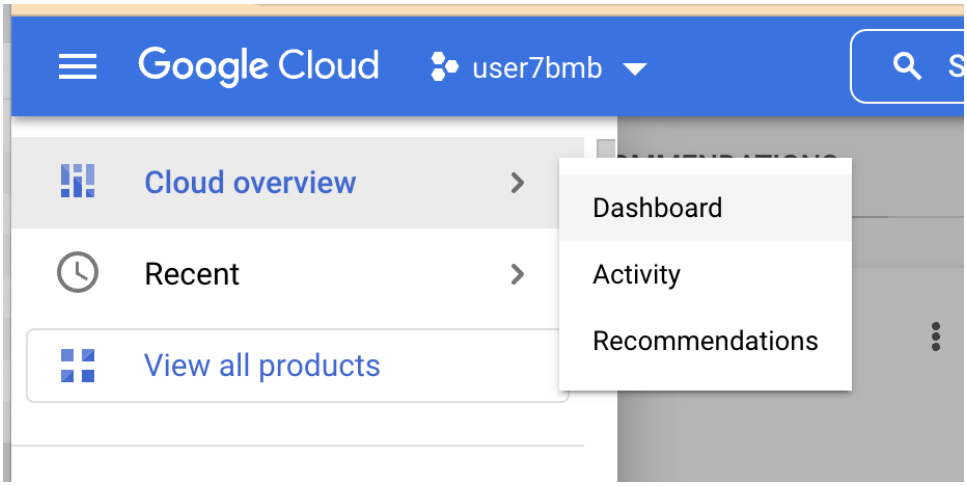




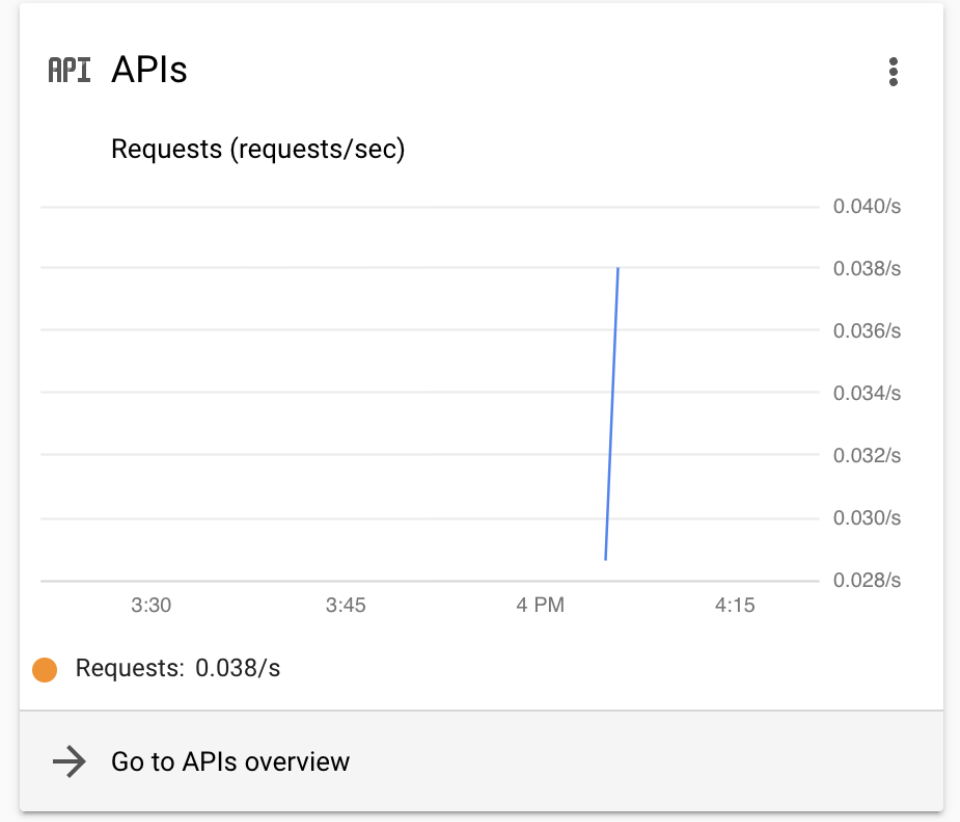
open the json file and copy the authorization token in a file called:

```
$ ~/tokens/google_token.json
```

Go back to the project dashboard

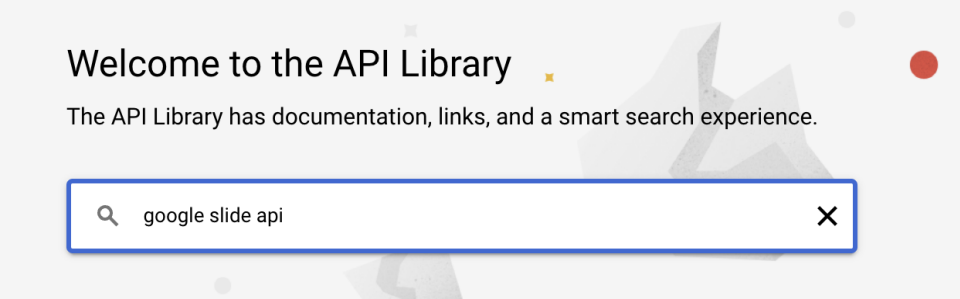


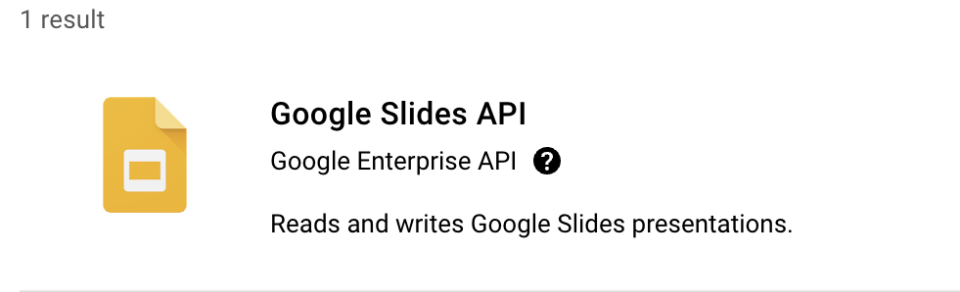
and select “Go to the API overview”



and select "Enable API and Services"

search and add the "google slide api"





Now tomolog will be able to create slides in the . . . @gmail.com user account.

CHAPTER

TWO

USAGE

To publish experiment log information to a google page:

```
$ tomolog run --file-name /local/data/2022-03/Peters/B4_Pb_03_c_10keV_892.h5 --
↪presentation-url https://docs.google.com/presentation/d/
↪128c8JYiJ5EjbQhAtegYYetwDUVZILQjZ5fUIoWuR_aI/edit#slide=id.p
```

For help:

```
$ tomolog run -h
usage: tomolog run [-h] [--file-name PATH] [--PV-prefix PV_PREFIX] [--beamline {None,2-
↪bm,7-bm,32-id}] [--idx IDX] [--idy IDY] [--idz IDZ] [--max MAX] [--min MIN]
                   [--presentation-url PRESENTATION_URL] [--rec-type {recgpu,rec}] [--
↪config FILE] [--config-update] [--double-fov] [--logs-home FILE] [--token-home FILE]
                   [--verbose]

optional arguments:
  -h, --help            show this help message and exit
  --file-name PATH      Name of the hdf file (default: .)
  --PV-prefix PV_PREFIX
                        PV prefix for camera (default: 32idcSP1:)
  --beamline {None,2-bm,7-bm,32-id}
                        Customized the goodle slide to the beamline selected (default:␣
↪32-id)
  --idx IDX             Id of x slice for reconstruction visualization (default: -1)
  --idy IDY             Id of y slice for reconstruction visualization (default: -1)
  --idz IDZ             Id of z slice for reconstruction visualization (default: -1)
  --max MAX             Maximum threshold value for reconstruction visualization␣
↪(default: 0.0)
  --min MIN             Minimum threshold value for reconstruction visualization␣
↪(default: 0.0)
  --presentation-url PRESENTATION_URL
                        Google presention url (default: None)
  --rec-type {recgpu,rec}
                        Specify the prefix of the recon folder (default: recgpu)
  --config FILE         File name of configuration file (default: /home/beams/FAST/logs/
↪tomolog.conf)
  --config-update       When set, the content of the config file is updated using the␣
↪current params values (default: False)
  --double-fov          Set to true for 0-360 data sets (default: False)
  --logs-home FILE      Log file directory (default: /home/beams/FAST/logs)
```

(continues on next page)

(continued from previous page)

```
  --token-home FILE     Token file directory (default: /home/beams/FAST/tokens)
  --verbose             Verbose output (default: False)
```

For other options:

```
$ tomolog -h
usage: tomolog [-h] [--config FILE]  ...

optional arguments:
  -h, --help     show this help message and exit
  --config FILE  File name of configuration file

Commands:

    init         Create configuration file
    run          Run data logging to google slides
    status       Show the tomolog status
```

CHAPTER

THREE

API REFERENCE

tomolog-cli Modules:

3.1 tomolog_cli.tomolog

Functions:

TomoLog(args)	Class to publish experiment meta data, tomography projection and reconstruction on a google slide document.

class tomolog_cli.tomolog.**TomoLog**(*args*)

Bases: object

Class to publish experiment meta data, tomography projection and reconstruction on a google slide document.

init_slide()

plot_projection(*proj*, *fname*)

plot_recon(*recon*, *fname*)

publish_descr(*presentation_id*, *page_id*)

publish_proj(*presentation_id*, *page_id*, *proj*, *resolution=1*)

publish_recon(*presentation_id*, *page_id*, *recon*, *resolution=1*)

read_meta_item(*template*)

read_raw()

read_recon()

run_log()

3.2 tomolog_cli.tomolog_2bm

Functions:

TomoLog2BM(args)	Class to publish experiment meta data, tomography projection and reconstruction on a google slide document.

class tomolog_cli.tomolog_2bm.**TomoLog2BM**(*args*)

Bases: *TomoLog*

Class to publish experiment meta data, tomography projection and reconstruction on a google slide document.

plot_projection(*proj*, *fname*)

plot_recon(*recon*, *fname*)

publish_descr(*presentation_id*, *page_id*)

publish_proj(*presentation_id*, *page_id*, *proj*)

publish_recon(*presentation_id*, *page_id*, *recon*)

read_raw()

read_rec_line()

read_recon()

read_tiff_part(*fname*, *x*, *y*, *z_start*, *z0_start*, *lchunk*)

run_log()

3.3 tomolog_cli.tomolog_32id

Functions:

TomoLog32ID(args)	Class to publish experiment meta data, tomography projection and reconstruction on a google slide document.

class tomolog_cli.tomolog_32id.**TomoLog32ID**(*args*)

Bases: *TomoLog*

Class to publish experiment meta data, tomography projection and reconstruction on a google slide document.

plot_projection(*proj*, *fname*, *scalebar='nano'*)

plot_recon(*recon*, *fname*)

publish_descr(*presentation_id*, *page_id*)

publish_proj(*presentation_id*, *page_id*, *proj*)

publish_recon(*presentation_id*, *page_id*, *recon*)

read_raw()

read_rec_line()

read_recon()

run_log()

CHAPTER

FOUR

CREDITS

4.1 Citations

GPU based tomographic reconstruction is available at tomocupy

Cite [A1] if you use tomobank

Cite [A2] if you use tomopy/tomopy cli

Cite [A3] if you use DataExchange

4.2 References

BIBLIOGRAPHY

[A1] Francesco De Carlo, Doga Gursoy, Daniel Jackson Ching, Kees Joost Batenburg, Wolfgang Ludwig, Lucia Mancini, Federica Marone, Rajmund Mokso, Daniel M. Pelt, Jan Sijbers, and Mark Rivers. Tomobank: a tomographic data repository for computational x-ray science. *Measurement Science and Technology*, 2017. URL: https://doi.org/10.1088/1361-6501/aa9c19.

[A2] Gürsoy D, De Carlo F, Xiao X, and Jacobsen C. Tomopy: a framework for the analysis of synchrotron tomographic data. *Journal of Synchrotron Radiation*, 21(5):1188–1193, 2014.

[A3] De Carlo F, Gursoy D, Marone F, Rivers M, Parkinson YD, Khan F, Schwarz N, Vine DJ, Vogt S, Gleber SC, Narayanan S, Newville M, Lanzirotti T, Sun Y, Hong YP, and Jacobsen C. Scientific data exchange: a schema for hdf5-based storage of raw and analyzed data. *Journal of Synchrotron Radiation*, 21(6):1224–1230, 2014.

[B1] Nghia T. Vo, Robert C. Atwood, Michael Drakopoulos, and Thomas Connolley. Data processing methods and data acquisition for samples larger than the field of view in parallel-beam tomography. *Opt. Express*, 29(12):17849–17874, Jun 2021. URL: http://www.opticsexpress.org/abstract.cfm?URI=oe-29-12-17849, doi:10.1364/OE.418448.

PYTHON MODULE INDEX

t

tomolog_cli.tomolog, 17
tomolog_cli.tomolog_2bm, 18
tomolog_cli.tomolog_32id, 18

INDEX

I

init_slide() (*tomolog_cli.tomolog.TomoLog method*), 17

M

module
    tomolog_cli.tomolog, 17
    tomolog_cli.tomolog_2bm, 18
    tomolog_cli.tomolog_32id, 18

P

plot_projection() (*tomolog_cli.tomolog.TomoLog method*), 17
plot_projection() (*tomolog_cli.tomolog_2bm.TomoLog2BM method*), 18
plot_projection() (*tomolog_cli.tomolog_32id.TomoLog32ID method*), 18
plot_recon() (*tomolog_cli.tomolog.TomoLog method*), 17
plot_recon() (*tomolog_cli.tomolog_2bm.TomoLog2BM method*), 18
plot_recon() (*tomolog_cli.tomolog_32id.TomoLog32ID method*), 18
publish_descr() (*tomolog_cli.tomolog.TomoLog method*), 17
publish_descr() (*tomolog_cli.tomolog_2bm.TomoLog2BM method*), 18
publish_descr() (*tomolog_cli.tomolog_32id.TomoLog32ID method*), 18
publish_proj() (*tomolog_cli.tomolog.TomoLog method*), 17
publish_proj() (*tomolog_cli.tomolog_2bm.TomoLog2BM method*), 18
publish_proj() (*tomolog_cli.tomolog_32id.TomoLog32ID method*), 18
publish_recon() (*tomolog_cli.tomolog.TomoLog method*), 17
publish_recon() (*tomolog_cli.tomolog_2bm.TomoLog2BM method*), 18
publish_recon() (*tomolog_cli.tomolog_32id.TomoLog32ID method*), 18

R

read_meta_item() (*tomolog_cli.tomolog.TomoLog method*), 17
read_raw() (*tomolog_cli.tomolog.TomoLog method*), 17
read_raw() (*tomolog_cli.tomolog_2bm.TomoLog2BM method*), 18
read_raw() (*tomolog_cli.tomolog_32id.TomoLog32ID method*), 18
read_rec_line() (*tomolog_cli.tomolog_2bm.TomoLog2BM method*), 18
read_rec_line() (*tomolog_cli.tomolog_32id.TomoLog32ID method*), 19
read_recon() (*tomolog_cli.tomolog.TomoLog method*), 17
read_recon() (*tomolog_cli.tomolog_2bm.TomoLog2BM method*), 18
read_recon() (*tomolog_cli.tomolog_32id.TomoLog32ID method*), 19
read_tiff_part() (*tomolog_cli.tomolog_2bm.TomoLog2BM method*), 18
run_log() (*tomolog_cli.tomolog.TomoLog method*), 17
run_log() (*tomolog_cli.tomolog_2bm.TomoLog2BM method*), 18
run_log() (*tomolog_cli.tomolog_32id.TomoLog32ID method*), 19

T

TomoLog (*class in tomolog_cli.tomolog*), 17
TomoLog2BM (*class in tomolog_cli.tomolog_2bm*), 18
TomoLog32ID (*class in tomolog_cli.tomolog_32id*), 18
tomolog_cli.tomolog
    module, 17

tomolog_cli.tomolog_2bm
    module, 18
tomolog_cli.tomolog_32id
    module, 18