
tomolog-cli Documentation

Release 0.1

Argonne National Laboratory

Nov 03, 2022

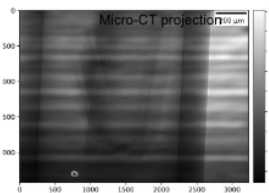
CONTENTS

1	Installation	3
2	Usage	15
3	API reference	17
4	Credits	21
	Bibliography	23
	Python Module Index	25
	Index	27

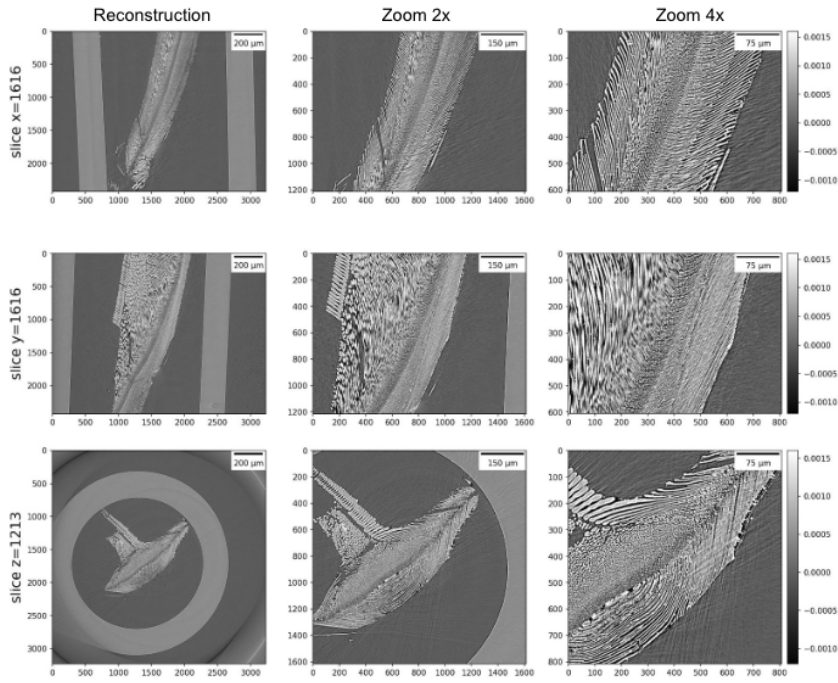
tomolog-cli is a command-line-interface for publishing tomography experiment data and meta data in a stack of Google slides

L_var_plum_038

- File name: L_var_plum_038.h5
- Beamline: 2-BM Micro-tomography
- Scan date: 2022-11-02T17:27:34-0500
- Exposure time: 0.70000 s
- Camera pixel size: 3.45 μm
- Lens magnification: 7.5
- Projection pixel size: 0.46 μm
- Angle step: 0.120 $^\circ$
- Number of angles: 1501 (0.00 - 180.00)
- Projection size: 3232 x 2426
- Scan energy: 25.51 keV
- Sample Y: 35.79 mm
- Propagation dist.: 30.00 mm
- Load Raw: 11.33325 V
- Load: -11620.56799 N



Frame from the IP camera in the hutch



tomocopy recon --file-name /data/2022-10/Stock/L_var_plum_038.h5 --remove-stripe-method fw --reconstruction-type full --rotation-axis-auto auto --find-center-end-row 1500

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:



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 scopesChoose 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

Add

Allow public clients (Implicit Grant & PKCE) ⓘ**Generated access token ⓘ**

Generate

Chooser / Saver / Embedder domains

Add

If using the [Chooser](#), the [Saver](#), or the [Embedder](#) on a website, add the domain of that site.

Webhooks

Webhook URIs ⓘ

Add

Extensions

Extensions URI ⓘ

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

```
$ ~/.tomologenv  
  
APP_KEY=....  
APP_SECRET=....
```

Set the following permissions:



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

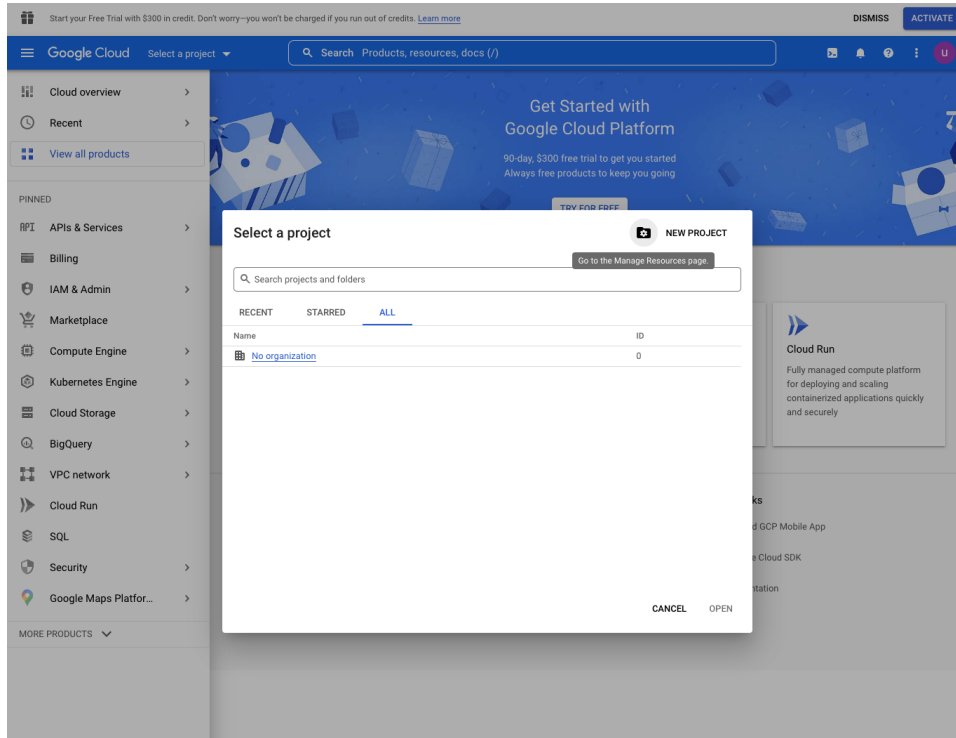
Extensions

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





Start your Free Trial with \$300 in credit. Don't worry—you won't be charged if you run out of



Google Cloud



Search Products, resource

New Project



You have 12 projects remaining in your quota. Request an increase or delete projects. [Learn more](#)

[MANAGE QUOTAS](#)

Project name *

My Project 74612



Project ID: smooth-ripple-364623. It cannot be changed later. [EDIT](#)

Location *



No organization

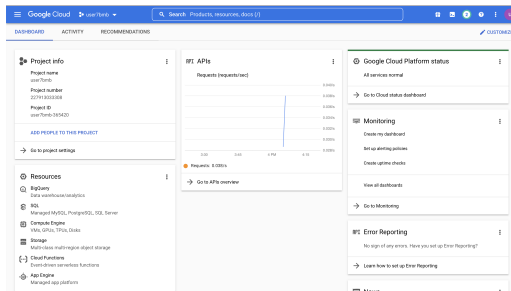
[BROWSE](#)

Parent organization or folder

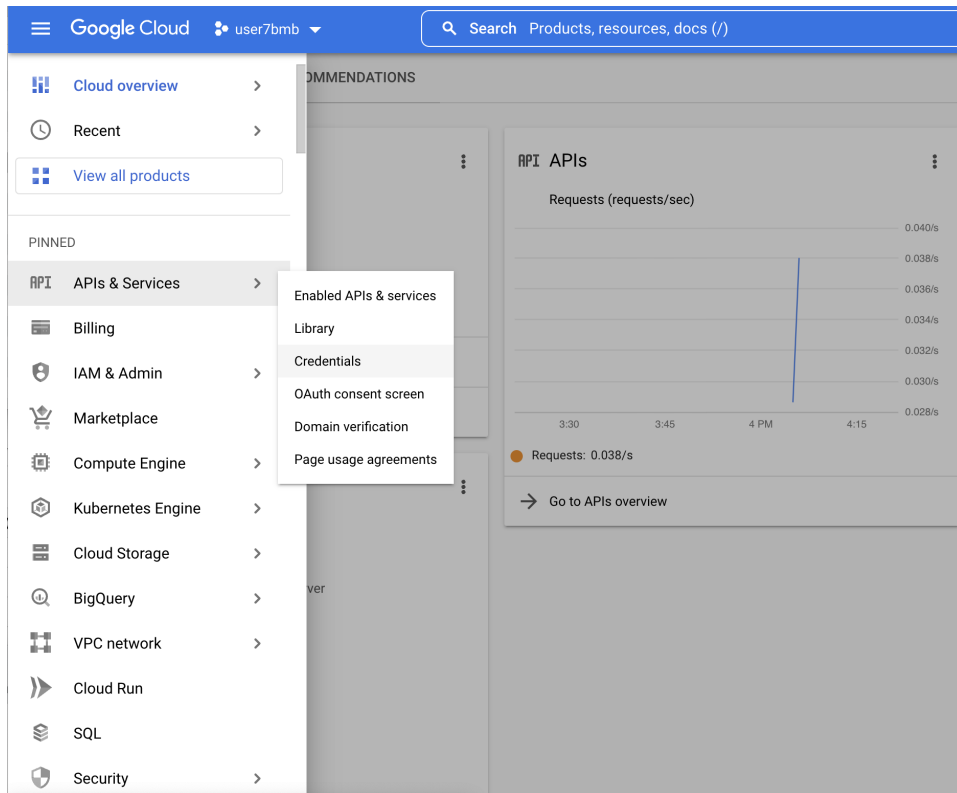
[CREATE](#)

[CANCEL](#)

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

Google Cloud user7bmb

Search Products, resources, docs (/)

IAM & Admin

- IAM
- Identity & Organization
- Policy Troubleshooter
- Policy Analyzer
- Organization Policies
- Service Accounts**
- Workload Identity Federat...
- Labels
- Tags
- Settings
- Privacy & Security
- Identity-Aware Proxy
- Roles
- Audit Logs
- Manage Resources

← Create service account

1 Service account details

Service account name
Display name for this service account

Service account ID * ✕ ↺

Email address: <id>@user7bmb-365420.iam.gserviceaccount.com 📧

Service account description
Describe what this service account will do

CREATE AND CONTINUE

2 Grant this service account access to project (optional)

3 Grant users access to this service account (optional)

DONE **CANCEL**

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:

IAM & Admin

- IAM
- Identity & Organization
- Policy Troubleshooter
- Policy Analyzer
- Organization Policies
- Service Accounts**
- Workload Identity Federat...
- Labels
- Tags
- Settings
- Privacy & Security
- Identity-Aware Proxy
- Roles
- Audit Logs
- Manage Resources

← tomolog

DETAILS PERMISSIONS **KEYS** METRICS LOGS

Keys

⚠ Service account keys could pose a security risk if compromised. We recommend you avoid down can learn more about the best way to authenticate service accounts on Google Cloud [here](#).

Add a new key pair or upload a public key certificate from an existing key pair.

Block service account key creation using [organization policies](#).
[Learn more about setting organization policies for service accounts](#)

ADD KEY ▾

Create private key for "test"

Downloads a file that contains the private key. Store the file securely because this key can't be recovered if lost.

Key type

☒ JSON

Recommended

☐ P12

For backward compatibility with code using the P12 format

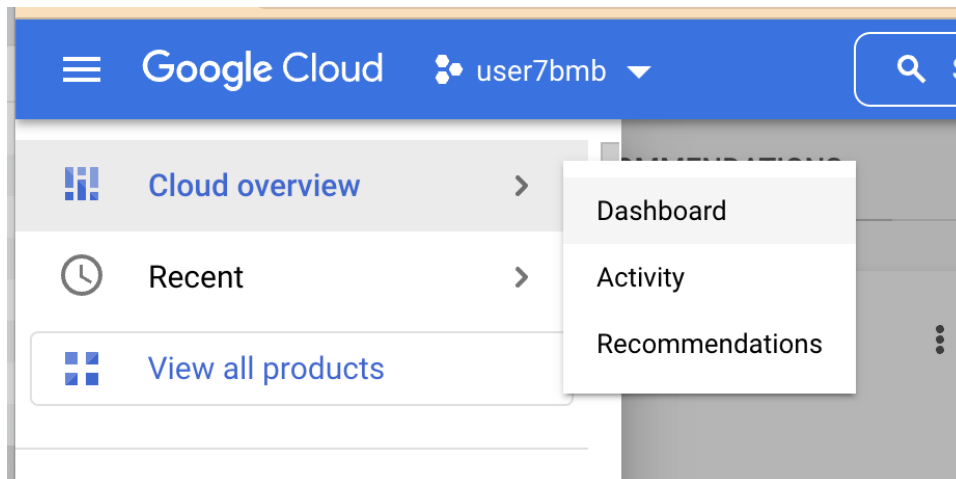
CANCEL

CREATE

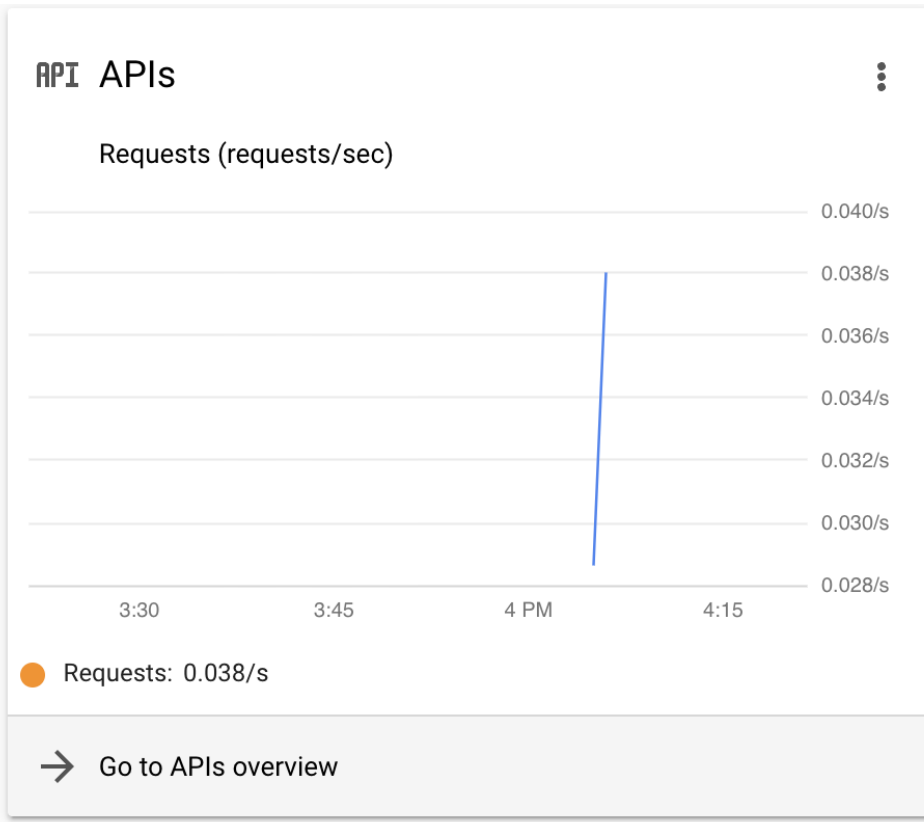
open the json file and copy the authorization token in a file called:

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

Go back to the project dashboard



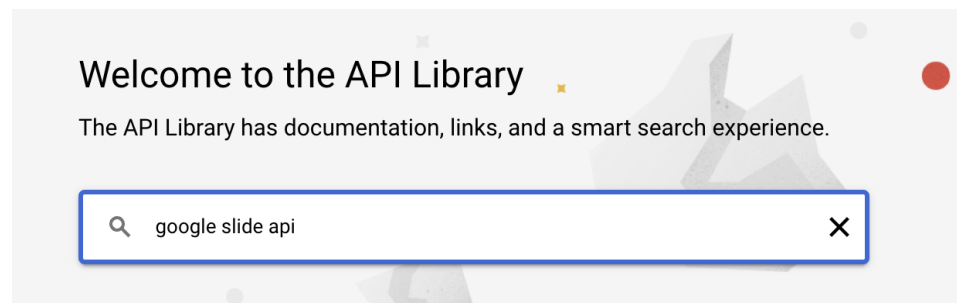
and select "Go to the API overview"



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.

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)  
→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 presentation 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)

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

For other options:

<pre>\$ 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</pre>	
--	--

API REFERENCE

tomolog-cli Modules:

3.1 tomolog_cli.tomolog

Functions:

<i>Tomolog</i> (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:

<i>Tomolog2BM</i> (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:

<i>Tomolog32ID</i> (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()`

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](#)