
tag*images_for_google_drive*Documentation

Release 1.1.dev1+g7ee06fe.d20220413

tagimagesforgdrive

Apr 13, 2022

Contents

1	Tag images for google drive documentation	3
1.1	README	3
1.2	Make commands	6
1.3	Contribute	7
1.4	Changelog	7
2	Indices and tables	9
	Index	11

Manage tags and descriptions in image files to be indexed by Google Drive

Warning: WARNING: It's not a released version !
--

Tag images for google drive documentation

1.1 README

1.1.1 Motivation

Synchronize a CSV database and PNG/JPEG files to add #hashtag in image description. Then, you can synchronize all files with Google drive and search image with *tags*. Google set the metadata of the file from the 'description' metadata in graphic file.

1.1.2 Synopsis

Google drive use only the description meta-data to index an image. After this synchronisation it's possible to search an image with "type:image a_hashtag".

```
type:image apple
```

This tool use [Exiftool](#)

```
$ sudo apt-get install exiftool          # Debian
$ sudo brew install exiftool            # Mac
$ sudo yum install perl-Image-ExifTool  # CentOS
...
...
```

You can update the tags inside the description in your CSV file, or use some others tools like [XnView](#) and extract tags to CSV and descriptions.

By default, this tool merge the tags from CSV and files.

```
$ # Merge tags from descriptions.csv and selected files, and save all tags in tags.
→txt
$ tag_images_for_google_drive -v --db descriptions.csv '**/*.png' '**/*.jpg' \
--tagfile tags.txt
```

But it's possible to apply tags from database or files only

```
$ tag_images_for_google_drive -v --from-db --db descriptions.csv '**/*.png' '**/
↳ *.jpg'
$ tag_images_for_google_drive -v --from-file --db descriptions.csv '**/*.png' '**/
↳ *.jpg'
```

To add a specific tag for all images in a directory, add it in command line.

```
$ tag_images_for_google_drive -v --db descriptions.csv -t myimages '**/*.png' '**/
↳ *.jpg'
```

The, you can filter theses specifics images with `type:image myimages`

For more informations

```
$ tag_images_for_google_drive --help
```

or [read the documentation](#)

To synchronize the google files, you can use different tools. In the proposed Docker image, we use the `google-drive-ocamlfuse`.

1.1.3 The latest version

Clone the git repository (see upper button)

1.1.4 Installation

Different solutions is possible.

For windows

Use `chcp 16001 (utf-8)`, before use this tools.

Installation from one executable

- Copy the file `'tag_images_for_google_drive.${OS}'` to local directory
- Rename this file to `'tag_images_for_google_drive'`
- And run-it

```
$ tag_images_for_google_drive --help
```

Installation from PIP

- In virtualenv or conda env, use

```
$ pip install tag_images_for_google_drive
```

- Then, run-it

```
$ tag_images_for_google_drive --help
```


Installation in Docker

- From the source code, use `make Dockerfile`
- WARNING, this image have the credential for manipulate all yours Google files
- Eventually, create a dedicated volume for the GDrive cache

```
> docker volume create --name tag_image_for_google_drive
```

- Create the container with custom parameters

```
$ docker build \
-f Dockerfile \
--build-arg OS_VERSION="latest" \
--build-arg GDRIVE_ROOT_FOLDER="/Images" \
--build-arg GDRIVE_TEAM_DRIVE_ID="" \
--build-arg PARAMS="'**/*.png' '**/*.jpg'" \
--build-arg CRON_FREQUENCY="* */12 * * *" \
-t "$(USER)/tag_image_for_google_drive:latest" .
```

- Start the container

```
$ docker run --detach --cpus=0.5 \
--privileged \
-v tag_image_for_google_drive:/cache
-i "$(USER)/tag_image_for_google_drive:latest"
```

Inside the container, a `google-drive-ocamlfuse` is installed to synchronize the google files from `GDRIVE_ROOT_FOLDER` in the cache, and a crontab is periodically executed (see `CRON_FREQUENCY`) to invoke `tag_image_for_google_drive` with `PARAMS`.

Installation from source

Go inside the directory and

```
$ make configure
$ conda activate tag_images_for_google_drive
$ make install
```

1.1.5 Tests

To test the project

```
$ make test
```

To validate the typing

```
$ make typing
```

To validate all the project

```
$ make validate
```

1.1.6 Project Organization

```
|— Makefile                <- Makefile with commands like `make data` or `make_
↪train`
|— README.md              <- The top-level README for developers using this_
↪project.
|— docs                  <- A default Sphinx project; see sphinx-doc.org for_
↪details
|— setup.py              <- makes project pip installable (pip install -e .
↪[tests])
|                          so sources can be imported and dependencies installed
|— tag_images_for_google_drive <- Source code for use in this_
↪project
|— tests                 <- Unit and integrations tests ((Mark directory as a_
↪sources root)).
```

1.2 Make commands

The Makefile contains the central entry points for common tasks related to this project.

- `make help` will print all majors target
- `make configure` will prepare the environment (conda venv, kernel, ...)
- `make lint` will lint the code
- `make test` will run all unit-tests
- `make typing` will check the typing
- `make add-typing` will add annotation for typing
- `make validate` will validate the version before commit
- `make clean` will clean current environment
- `make docs` will create and show a HTML documentation in 'build/'
- `make dist` will create a full wheel distribution

1.2.1 Twine commands

- `make check-twine` will check the packaging before publication
- `make test-twine` will publish the package in test.pypi.org <<https://test.pypi.org>>_)
- `make twine` will publish the package in pypi.org <<https://pypi.org>>_)

1.2.2 Docker commands

- `make docker-build` will build the Dockerfile and container
- `make docker-run` will start the container in background and attach the console
- `make docker-start` will start the container in background
- `make docker-stop` will stop the container
- `make docker-attach` will attach the console to the container
- `make docker-bash` will attach a shell in the container

1.3 Contribute

The code use [Conda](#) , because we want some dependencies other than Python.

Download sources

```
$ git clone |giturl| tag_images_for_google_drive
```

then:

On Windows, use the Ubuntu subsystem, then install miniconda:

```
$ wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
$ ./Miniconda3-latest-Linux-x86_64.sh
```

then: .. code-block:: bash

```
$ cd tag_images_for_google_drive $ conda install -y -c anaconda make $ make configure $ conda
activate tag_images_for_google_drive $ make installer
```

- A make validate is executed before a git push in branch master. It possible to force the push with `FORCE=y git push`.

1.4 Changelog

All notable changes to this project will be documented in this file.

The format is based on [Keep a Changelog](#), and this project adheres to [Semantic Versioning](#).

1.4.1 [Unreleased]

Added

- First release TODO

Changed

Deprecated

Removed

Fixed

Security

1.4.2 [0.0.1] - 2020-12-05

- First release

CHAPTER 2

Indices and tables

- `genindex`
- `modindex`
- `search`

C

clone, 6