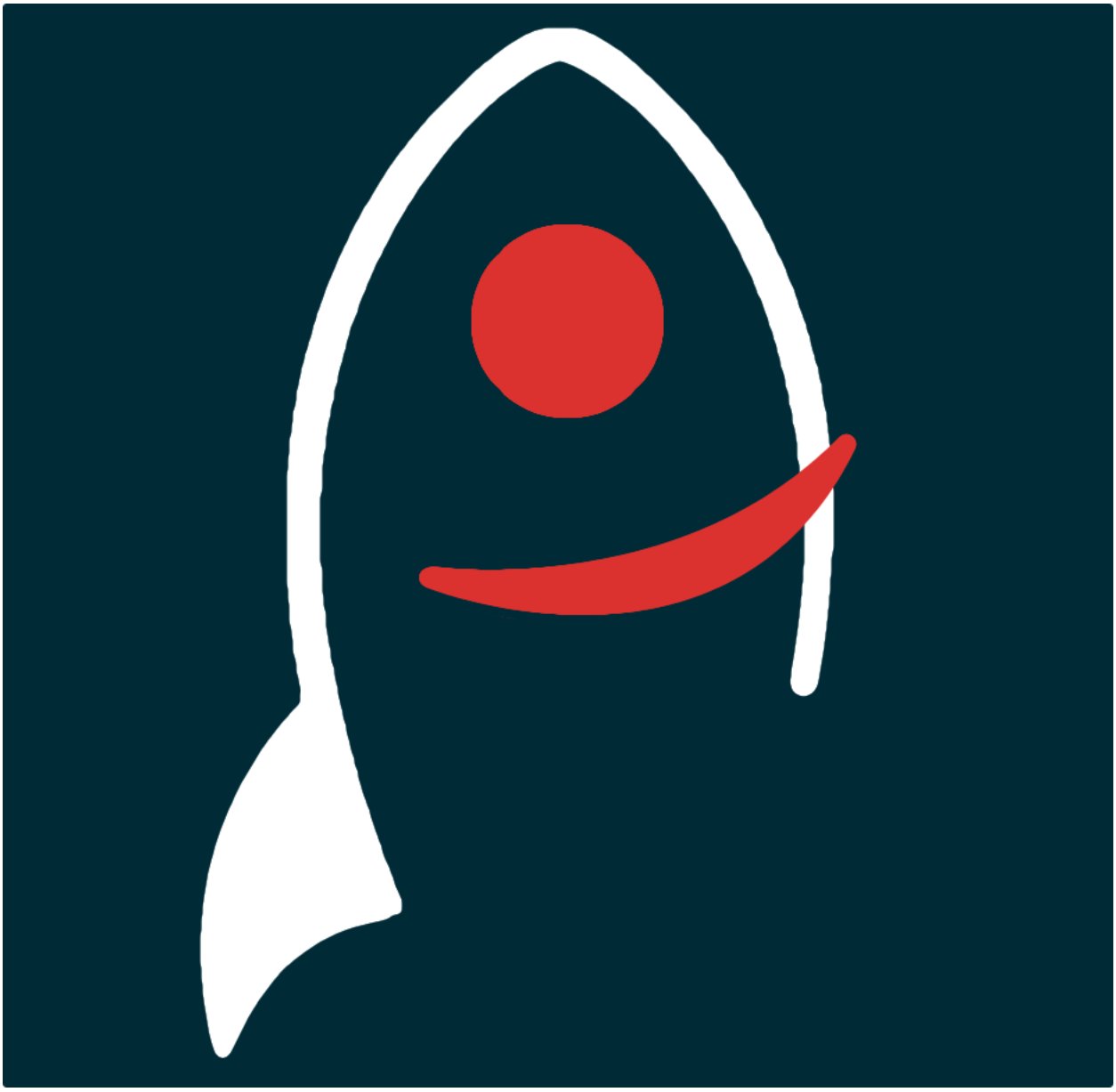

picaxe Documentation

Release 0.4.1

Dave Young

2017

A python package and command-line tools for work with the Flickr API to upload images, sort images, generate MD image reference links etc.

Here's a summary of what's included in the python package:

Classes

`picaxe.picaxe`	*work with the Flickr API to upload images, sort images, generate MD image reference links etc*

Functions

CHAPTER 1

Installation

The easiest way to install picaxe is to use `pip`:

```
pip install picaxe
```

Or you can clone the github repo and install from a local version of the code:

```
git clone git@github.com:thespacedoctor/picaxe.git
cd picaxe
python setup.py install
```

To upgrade to the latest version of picaxe use the command:

```
pip install picaxe --upgrade
```

Development

If you want to tinker with the code, then install in development mode. This means you can modify the code from your cloned repo:

```
git clone git@github.com:thespacedoctor/picaxe.git
cd picaxe
python setup.py develop
```

Pull requests are welcomed!

Sublime Snippets

If you use Sublime Text as your code editor, and you're planning to develop your own python code with picaxe, you might find my Sublime Snippets useful.

Issues

Please report any issues here.

CHAPTER 2

Command-Line Usage

```
Documentation for picaxe can be found here: http://picaxe.readthedocs.org/en/stable

Usage:
    picaxe init
    picaxe auth [-s <pathToSettingsFile>]
    picaxe md <urlOrPhotoid> [<width>] [-s <pathToSettingsFile>]
    picaxe albums [-s <pathToSettingsFile>]
    picaxe [-giop] upload <imagePath> [--title=<title> --tags=<tags> --desc=<desc> --album=<album>]
    picaxe [-op] grab [--title=<title> --tags=<tags> --desc=<desc> --album=<album> --delay=<sec>]

Options:
    init                  setup the polygot settings file for the first time
    auth                  authenticate picaxe against your flickr account
    md                    generate the MD reference link for the image in the given flickr URL
    albums                list all the albums in the flickr account
    upload                upload a local image to flickr

    <pathToSettingsFile>  path to the picaxe settings file
    <urlOrPhotoid>        the flickr URL or photoid
    <width>               pixel width resolution of the linked image. Default *original*. [75|100|150
    <imagePath>           path to the local image to upload to flickr

    --title=<title>       the image title
    --tags=<tags>         quoted, comma-sepatated tags
    --desc=<desc>         image description
    --delay=<sec>         the delay time before screen-grab selection tool appears

    -p, --public          make the image public (private by default)
    -o, --open            open the image in the flickr web-app once uploaded
    -i, --image           "photo" is an image
    -g, --screenGrab      "photo" is a screengrab
    -h, --help            show this help message
    -v, --version         show version
    -s, --settings        the settings file
```

CHAPTER 3

Documentation

Documentation for picaxe is hosted by Read the Docs (last stable version and latest version).

CHAPTER 4

Command-Line Tutorial

Before you begin using picaxe you'll need to populate some custom settings within the picaxe settings file.

To setup the default settings file at `~/.config/picaxe/picaxe.yaml` run the command:

```
picaxe init
```

This should create and open the settings file; follow the instructions in the file to populate the missing settings values (usually given an `XXX` place-holder).

Authenticating Picaxe Against Your Flickr Account

In order to use picaxe for the first time you'll need to first authenticate it against your Flickr account. This is to give picaxe permission to read your private photo metadata to generate markdown image links etc and also the ability to upload images and screengrabs to your various albums.

The `picaxe init` command should initiate the authentication process if you're running picaxe for the first time, but if you need to run the authentication process again for any reason use:

```
picaxe auth
```

You should see something like this, and then your default browser *should* open at the URL presented (if not just copy and paste the URL into your browser):

```
Now to authorize picaxe against your personal flickr account
Navigate to https://www.flickr.com/services/oauth/authorize?perms=write&oauth_token=72157678178240312
What is the oauth_verifier value in URL you where redirected to?
  >
```

You'll be presented with an authentication request like the one below. Click 'OK, I'LL AUTHOURIZE IT'.

You'll then be redirected to *thespacedoctor* website and in the URL you'll notice there are `oauth_token` and `oauth_verifier` parameters.

Copy the `oauth_verifier` value, paste it into the terminal and hit return. That's it. Simples. Your credentials are now written into the picaxe settings file which can be found at `~/.config/picaxe/picaxe.yaml`.

Listing Albums in Flickr Account

To list all of the albums in your Flickr account run the command:

```
picaxe albums
```

This prints the titles of all the albums you have created in your Flickr account to stdout:

```
Auto Upload
home movies
projects: thespacedoctor
notes: images and screengrabs
blog: workflow tags
family photos
```

Generating a Multi-Markdown Image Link from Any Flickr Image

To generate a MMD image link for any image in your Flickr account (private or public), or any other public Flickr image, run the command:

```
picaxe md <urlOrPhotoid>
```

Take this image for example. To generate the MMD image link run:

```
picaxe md https://www.flickr.com/photos/92344016@N06/30588492355/in/album-72157675576126786/
```

or just quote the photo-id (*30588492355* in this case):

```
picaxe md 30588492355
```

Here's the MMD link dumped to stdout:

```
![][Photoelectric effect 30588492355]

[Photoelectric effect 30588492355]: https://farm6.staticflickr.com/5722/30588492355_147111fcd3_o.png
```

Note the image reference is generated from the image title (if there is one) and photo-id so should always be unique (i.e. no reference name clashes in your MMD documents).

Uploading Local Images to Flickr

It's possible to upload images to Flickr via the command-line with options to set tags, album, titles, descriptions and privacy levels with picaxe. To do so use the command:

```
picaxe [-giop] upload <imagePath> [--title=<title> --tags=<tags> --desc=<desc> --album=<album>]
```

So in its simplest form you could upload an image with picaxe like:

```
picaxe upload "/path/to/image.png"
```

as *title*, *description*, *album* and *tags* are optional arguments. The `-g` flag indicates that the uploaded image is a screengrab, `-i` that it is an image (as opposed to a photo), `-p` requests that the image be made public and `-o` that the image be opened in the Flickr web-app in your default browser once upload has completed.

Taking Screenshots with picaxe

The command for taking a screenshot with picaxe is similar to the command for uploading local images:

```
picaxe [-op] grab [--title=<title> --tags=<tags> --desc=<desc> --album=<album> --delay=<sec>]
```

By default picaxe will upload screenshots to a 'screengrabs' album unless a specific album is specified. All I need to do to trigger a screenshot selection cursor is run the following:

```
picaxe grab
```

I can now select the section of the screen I want to clip, or press space-bar to change to a window-selection cursor, and picaxe will upload the resulting image to flickr and dump the multi-markdown image link to stdout.

As this command is run from the terminal you will probably want a little time to navigate to the correct desktop/application you wish to take a screenshot of before the screen-capture cursor is activated. To do this pass in a delay in seconds via the `--delay` flag; so for a 3 sec delay run:

```
picaxe grab --delay=3
```

CHAPTER 5

Installation

The easiest way to install picaxe is to use `pip`:

```
pip install picaxe
```

Or you can clone the github repo and install from a local version of the code:

```
git clone git@github.com:thespacedoctor/picaxe.git
cd picaxe
python setup.py install
```

To upgrade to the latest version of picaxe use the command:

```
pip install picaxe --upgrade
```

Development

If you want to tinker with the code, then install in development mode. This means you can modify the code from your cloned repo:

```
git clone git@github.com:thespacedoctor/picaxe.git
cd picaxe
python setup.py develop
```

Pull requests are welcomed!

Sublime Snippets

If you use Sublime Text as your code editor, and you're planning to develop your own python code with picaxe, you might find my Sublime Snippets useful.

Issues

Please report any issues here.

CHAPTER 6

Command-Line Usage

```
Documentation for picaxe can be found here: http://picaxe.readthedocs.org/en/stable

Usage:
    picaxe init
    picaxe auth [-s <pathToSettingsFile>]
    picaxe md <urlOrPhotoid> [<width>] [-s <pathToSettingsFile>]
    picaxe albums [-s <pathToSettingsFile>]
    picaxe [-giop] upload <imagePath> [--title=<title> --tags=<tags> --desc=<desc> --album=<album>]
    picaxe [-op] grab [--title=<title> --tags=<tags> --desc=<desc> --album=<album> --delay=<sec>]

Options:
    init                  setup the polygot settings file for the first time
    auth                  authenticate picaxe against your flickr account
    md                    generate the MD reference link for the image in the given flickr URL
    albums                list all the albums in the flickr account
    upload                upload a local image to flickr

    <pathToSettingsFile>  path to the picaxe settings file
    <urlOrPhotoid>        the flickr URL or photoid
    <width>               pixel width resolution of the linked image. Default *original*. [75|100|150
    <imagePath>           path to the local image to upload to flickr

    --title=<title>       the image title
    --tags=<tags>         quoted, comma-sepatated tags
    --desc=<desc>         image description
    --delay=<sec>         the delay time before screen-grab selection tool appears

    -p, --public          make the image public (private by default)
    -o, --open            open the image in the flickr web-app once uploaded
    -i, --image           "photo" is an image
    -g, --screenGrab      "photo" is a screengrab
    -h, --help            show this help message
    -v, --version         show version
    -s, --settings        the settings file
```

CHAPTER 7

Documentation

Documentation for picaxe is hosted by Read the Docs (last stable version and latest version).

CHAPTER 8

Command-Line Tutorial

Before you begin using picaxe you'll need to populate some custom settings within the picaxe settings file.

To setup the default settings file at `~/.config/picaxe/picaxe.yaml` run the command:

```
picaxe init
```

This should create and open the settings file; follow the instructions in the file to populate the missing settings values (usually given an `XXX` place-holder).

Authenticating Picaxe Against Your Flickr Account

In order to use picaxe for the first time you'll need to first authenticate it against your Flickr account. This is to give picaxe permission to read your private photo metadata to generate markdown image links etc and also the ability to upload images and screengrabs to your various albums.

The `picaxe init` command should initiate the authentication process if you're running picaxe for the first time, but if you need to run the authentication process again for any reason use:

```
picaxe auth
```

You should see something like this, and then your default browser *should* open at the URL presented (if not just copy and paste the URL into your browser):

```
Now to authorize picaxe against your personal flickr account
Navigate to https://www.flickr.com/services/oauth/authorize?perms=write&oauth_token=72157678178240312
What is the oauth_verifier value in URL you where redirected to?
  >
```

You'll be presented with an authentication request like the one below. Click 'OK, I'LL AUTHOURIZE IT'.

You'll then be redirected to *thespacedoctor* website and in the URL you'll notice there are `oauth_token` and `oauth_verifier` parameters.

Copy the `oauth_verifier` value, paste it into the terminal and hit return. That's it. Simples. Your credentials are now written into the picaxe settings file which can be found at `~/.config/picaxe/picaxe.yaml`.

Listing Albums in Flickr Account

To list all of the albums in your Flickr account run the command:

```
picaxe albums
```

This prints the titles of all the albums you have created in your Flickr account to stdout:

```
Auto Upload
home movies
projects: thespacedoctor
notes: images and screengrabs
blog: workflow tags
family photos
```

Generating a Multi-Markdown Image Link from Any Flickr Image

To generate a MMD image link for any image in your Flickr account (private or public), or any other public Flickr image, run the command:

```
picaxe md <urlOrPhotoid>
```

Take this image for example. To generate the MMD image link run:

```
picaxe md https://www.flickr.com/photos/92344016@N06/30588492355/in/album-72157675576126786/
```

or just quote the photo-id (*30588492355* in this case):

```
picaxe md 30588492355
```

Here's the MMD link dumped to stdout:

```
![][Photoelectric effect 30588492355]

[Photoelectric effect 30588492355]: https://farm6.staticflickr.com/5722/30588492355_147111fcd3_o.png
```

Note the image reference is generated from the image title (if there is one) and photo-id so should always be unique (i.e. no reference name clashes in your MMD documents).

Uploading Local Images to Flickr

It's possible to upload images to Flickr via the command-line with options to set tags, album, titles, descriptions and privacy levels with picaxe. To do so use the command:

```
picaxe [-giop] upload <imagePath> [--title=<title> --tags=<tags> --desc=<desc> --album=<album>]
```

So in its simplest form you could upload an image with picaxe like:

```
picaxe upload "/path/to/image.png"
```

as *title*, *description*, *album* and *tags* are optional arguments. The `-g` flag indicates that the uploaded image is a screengrab, `-i` that it is an image (as opposed to a photo), `-p` requests that the image be made public and `-o` that the image be opened in the Flickr web-app in your default browser once upload has completed.

Taking Screenshots with picaxe

The command for taking a screenshot with picaxe is similar to the command for uploading local images:

```
picaxe [-op] grab [--title=<title> --tags=<tags> --desc=<desc> --album=<album> --delay=<sec>]
```

By default picaxe will upload screenshots to a 'screengrabs' album unless a specific album is specified. All I need to do to trigger a screenshot selection cursor is run the following:

```
picaxe grab
```

I can now select the section of the screen I want to clip, or press space-bar to change to a window-selection cursor, and picaxe will upload the resulting image to flickr and dump the multi-markdown image link to stdout.

As this command is run from the terminal you will probably want a little time to navigate to the correct desktop/application you wish to take a screenshot of before the screen-capture cursor is activated. To do this pass in a delay in seconds via the `--delay` flag; so for a 3 sec delay run:

```
picaxe grab --delay=3
```

Subpackages

picaxe	
picaxe.commonutils	*common tools used throughout package*

picaxe (*subpackage*)

picaxe.commonutils (*subpackage*)

common tools used throughout package

Modules

picaxe.cl_utils	Documentation for picaxe can be found here: http://picaxe.readthedocs.org/en/stable
picaxe.utKit	*Unit testing tools*

picaxe.cl_utils (*module*)

Documentation for picaxe can be found here: http://picaxe.readthedocs.org/en/stable

Usage: picaxe init picaxe auth [-s <pathToSettingsFile>] picaxe md <urlOrPhotoid> [<width>] [-s <pathToSettingsFile>] picaxe albums [-s <pathToSettingsFile>] picaxe [-giop] upload <imagePath> [–title=<title> –tags=<tags> –desc=<desc> –album=<album>] picaxe [-op] grab [–title=<title> –tags=<tags> –desc=<desc> –album=<album> –delay=<sec>]

Options: init setup the polygot settings file for the first time auth authenticate picaxe against your flickr account md generate the MD reference link for the image in the given flickr URL albums list all the albums in the flickr account upload upload a local image to flickr

<pathToSettingsFile> path to the picaxe settings file <urlOrPhotoid> the flickr URL or photoid <width> pixel width resolution of the linked image. Default *original*. [75|100|150|240|320|500|640|800|1024|1600|2048] <imagePath> path to the local image to upload to flickr

--title=<title> the image title

--tags=<tags> quoted, comma-sepatated tags

--desc=<desc>	image description
--delay=<sec>	the delay time before screen-grab selection tool appears
-p, --public	make the image public (private by default)
-o, --open	open the image in the flickr web-app once uploaded
-i, --image	"photo" is an image
-g, --screenGrab	"photo" is a screengrab
-h, --help	show this help message
-v, --version	show version
-s, --settings	the settings file

`picaxe.cl_utils.`**`main`**(*arguments=None*)
The main function used when ''cl_utils.py'' is run as a single script from the cl, or when installed as a cl command

picaxe.utKit (*module*)

Unit testing tools

class `picaxe.utKit.`**`utKit`**(*moduleDirectory*)
Override dryx utKit

Classes

picaxe.picaxe	*work with the Flickr API to upload images, sort images, generate MD image reference links etc*
picaxe.utKit.utKit	*Override dryx utKit*

picaxe.picaxe (*class*)

class `picaxe.`**`picaxe`**(*log*, *settings=False*, *pathToSettingsFile=False*)
work with the Flickr API to upload images, sort images, generate MD image reference links etc

Key Arguments:

- `log` – logger
- `settings` – the settings dictionary
- `pathToSettingsFile` – path to the settings file

`__init__`(*log*, *settings=False*, *pathToSettingsFile=False*)

Methods

`__init__`(log[, settings, pathToSettingsFile])	
`authenticate`()	*setup a Flickr API key to access a Flickr user account so picaxe can work on privat*
`get_photo_metadata`(url)	*get useful image metadata for the image found at a give Flickr share URL*
`list_album_titles`()	*list all of the albums (photosets) in the Flickr account*
`md`(url[, width])	*generate a multimarkdown image link viewable anywhere (no sign-in needed for pr*

Continued o

Table 8.4 – continued from previous page

upload(imagePath[, title, private, tags, ...])	*upload*

picaxe.utKit.utKit (*class*)

class picaxe.utKit.**utKit**(*moduleDirectory*)
Override dryx utKit

__init__(*moduleDirectory*)

Methods

__init__(moduleDirectory)	
setupModule()	*The setupModule method*
tearDownModule()	*The tearDownModule method*

Functions

Indexes

- Module Index
- Full Index

Todo

- Todolist

p

Symbols

M

P

U