[aps]

{abertay /*PROGRAMMING*/ society;}

Image of the day: w/ Reddit + Python

Abertay Programming Society
21-10-2018

Andrew

NEED LINUX FOR TODAY
(use netlab machines)

*sorry*

1. Create a reddit account
2. Register reddit app
3. Boot to Ubuntu!

   (Password = netlab)

   a. sudo apt-get install python3-pip
   b. pip3 install requests
   c. pip3 install praw

Connecting to reddit with PRAW

- Reddit requires OAuth authentication

```
reddit = praw.Reddit(client_id='your_client_id',
                     client_secret='your_client_secret',
                     password='your_password',
                     user_agent='script_name by /u/your_username',
                     username='your_username')
```

- Can test success of OAuth by checking username

```
If 'your_username' = reddit.user.me():
    print('Authenticated')
```

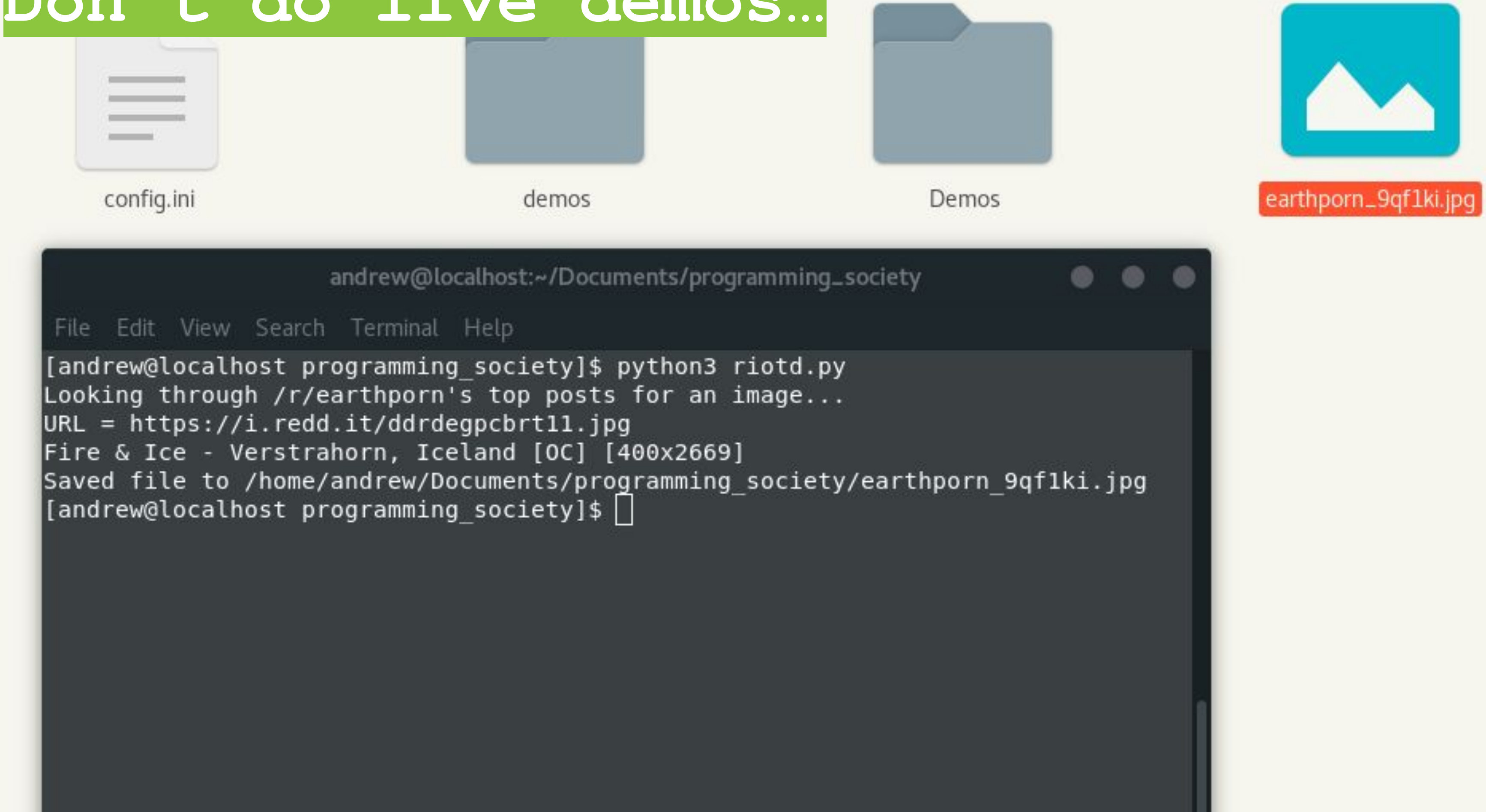
Todays sample

Reddit Image of the day:

- Uses praw to look through top 10 ‘hot’ posts of a given subreddit *any subreddit*
- Finds the first image post hosted on i.redd.it (easy to download from) *TODO: download from imgur, photos.google, etc…*
- Uses Requests to download that image, saves it to a file based on subreddit name + post id.

Do other fun stuff with https://praw.readthedocs.io

Don’t do live demos…



```
[andrew@localhost programming_society]$ python3 riotd.py
Looking through /r/earthporn's top posts for an image...
URL = https://i.redd.it/ddrdegpcbrt11.jpg
Fire & Ice - Verstrahorn, Iceland [OC] [400x2669]
Saved file to /home/andrew/Documents/programming_society/earthporn_9qf1ki.jpg
[andrew@localhost programming_society]$ 
```

Don't do live demos...

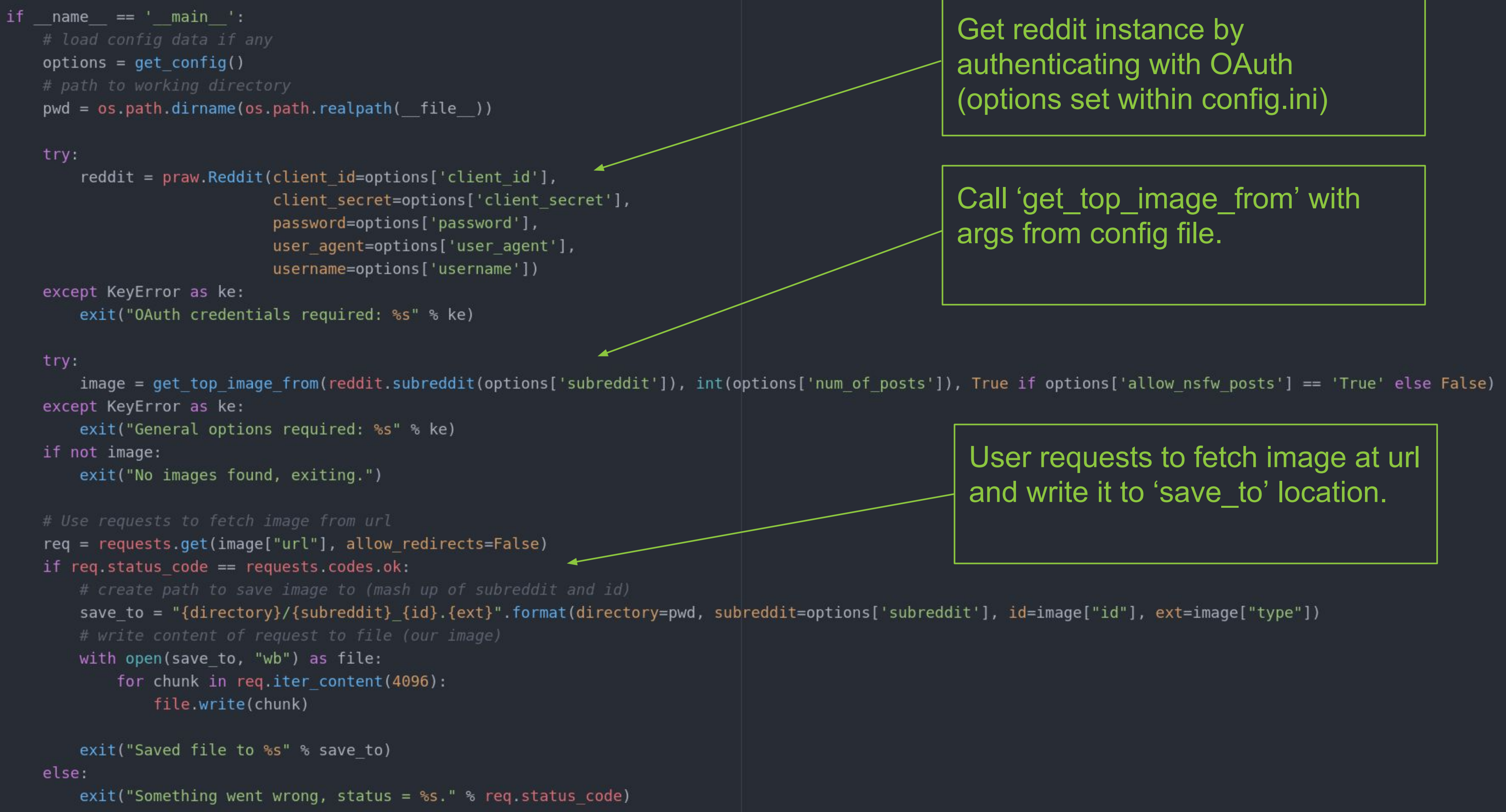
earthporn_9qf1ki.jpg

Properties

Size 4000 x 2669 pixels
Type JPEG image
File Size: 1.1 MB
Folder programming_society

Aperture
Exposure
Focal Length
ISO
Metering
Camera

Date
Time

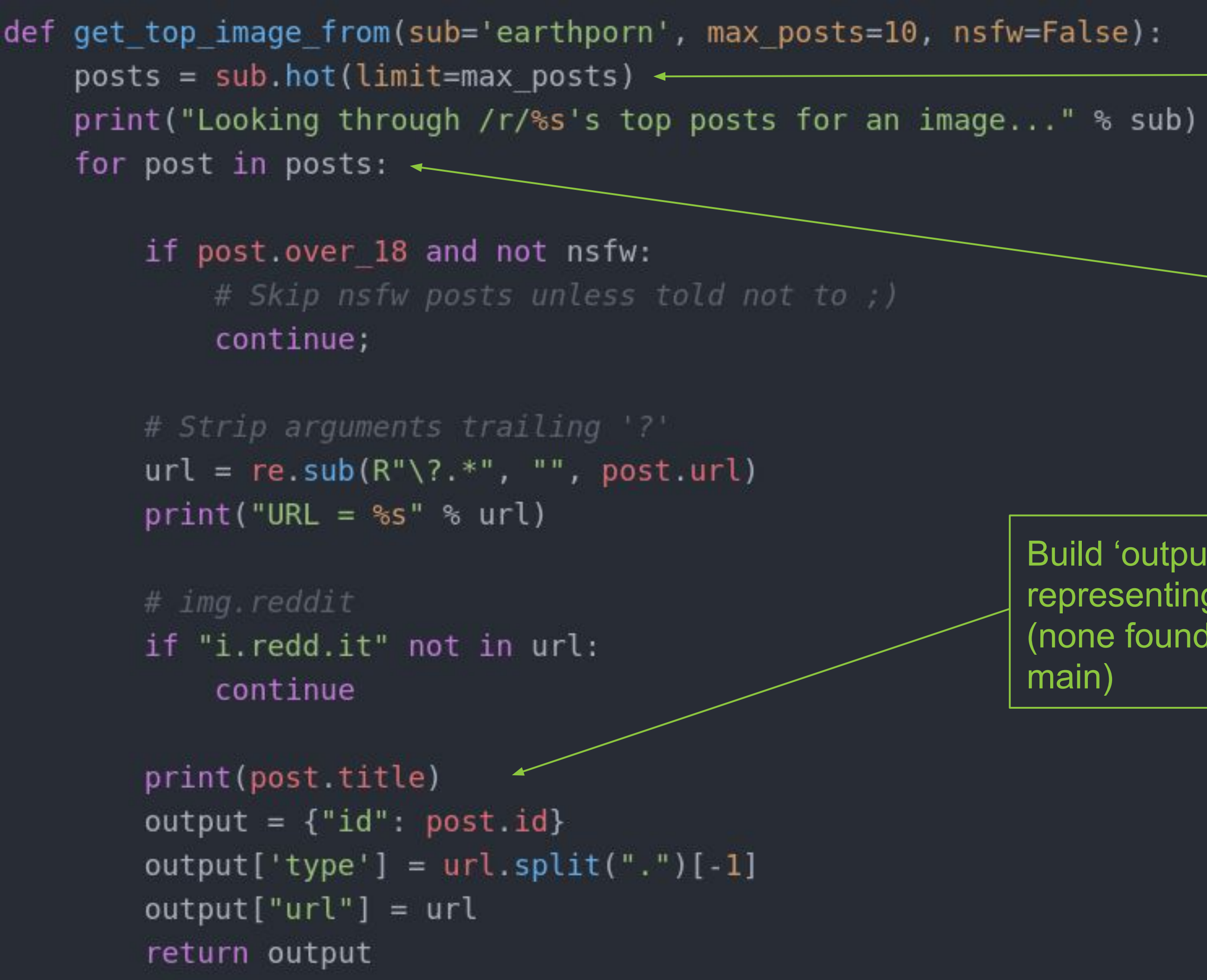


```
if __name__ == '__main__':
    # load config data if any
    options = get_config()
    # path to working directory
    pwd = os.path.dirname(os.path.realpath(__file__))

    try:
        reddit = praw.Reddit(client_id=options['client_id'],
                             client_secret=options['client_secret'],
                             password=options['password'],
                             user_agent=options['user_agent'],
                             username=options['username'])
    except KeyError as ke:
        exit("OAuth credentials required: %s" % ke)

    try:
        image = get_top_image_from(reddit.subreddit(options['subreddit']), int(options['num_of_posts']), True if options['allow_nsfw_posts'] == 'True' else False)
    except KeyError as ke:
        exit("General options required: %s" % ke)
    if not image:
        exit("No images found, exiting.")

    # Use requests to fetch image from url
    req = requests.get(image["url"], allow_redirects=False)
    if req.status_code == requests.codes.ok:
        # create path to save image to (mash up of subreddit and id)
        save_to = "{directory}/{subreddit}_{id}.{ext}".format(directory=pwd, subreddit=options['subreddit'], id=image["id"], ext=image["type"])
        # write content of request to file (our image)
        with open(save_to, "wb") as file:
            for chunk in req.iter_content(4096):
                file.write(chunk)

        exit("Saved file to %s" % save_to)
    else:
        exit("Something went wrong, status = %s." % req.status_code)
```

Get reddit instance by authenticating with OAuth (options set within config.ini)

Call ‘get_top_image_from’ with args from config file.

User requests to fetch image at url and write it to ‘save_to’ location.



```python
def get_top_image_from(sub='earthporn', max_posts=10, nsfw=False):
    posts = sub.hot(limit=max_posts)
    print("Looking through /r/%s's top posts for an image..." % sub)
    for post in posts:

        if post.over_18 and not nsfw:
            # Skip nsfw posts unless told not to ;)
            continue;

        # Strip arguments trailing '?'
        url = re.sub(R"\?.*", "", post.url)
        print("URL = %s" % url)

        # img.reddit
        if "i.redd.it" not in url:
            continue

        print(post.title)
        output = {"id": post.id}
        output['type'] = url.split(".")[-1]
        output["url"] = url
        return output
```

Get max_posts posts from subreddit object

Loop through each post in subreddit posts

Build ‘output’ dictionary representing our image (none found handled in main)

Challenges for today:

SAMPLE:

```
git clone https://github.com/AR-Calder/iotd
```

- Print hot 10 posts of a given subreddit
- Print top 10 comments of a given post
- Print commentForest of a given comment
- Reply with “Hello there!” to a given comment (Use one of your own to avoid getting suspended)

- Make something cool!

Thanks for listening!

See you next week 📆