reddit-photo-aggregator

hamolicious

May 31, 2023

CONTENTS

reddit-photo-aggregator's is a library to scrape subreddits for images.

Check out the *Usage* section for further information, including

Note: This project is under active development.

CHAPTER

ONE

CONTENTS

1.1 Installation

` pip install reddit-photo-aggregator `

1.2 Usage

To extract only the parts that you will need, use:

```
from reddit_photo_aggregator import SubReddit, ImageURLValidator, SortBy
```

Alternatively, import just the package:

```
import reddit_photo_aggregator
```

Set the image host whitelist using *.set_image_host_whitelist()*

```
ImageURLValidator.set_image_host_whitelist([
        'i.redd.it',
        'i.imgur.com',
])
```

Set the file extension whitelist using *.set_image_extension_whitelist()*

```
ImageURLValidator.set_image_extension_whitelist([
        'png',
        'jpeg',
        'jpg'
])
```

Create a *SubReddit* object with the sub-reddit name that you want to aggregate images from

```
sub = SubReddit('r/LandscapePhotography')
```

You can then discover images, this can be sorted using the *SortBy* class, this will also give you the amount of images found and the `sub.loaded_images` becomes filled out

```
amount = sub.discover_images(SortBy.hot)
print(amount)
print(sub.loaded_images)
```

Save the images to an /images directory.

```
for image in sub.loaded_images:
        print(f'Saving to {image.filename}')
        image.save(to='./images')
```

1.2.1 Full Example

```
from src import SubReddit, ImageURLValidator, SortBy


ImageURLValidator.set_image_host_whitelist([
        'i.redd.it',
        'i.imgur.com',
])
ImageURLValidator.set_image_extension_whitelist([
        'png',
        'jpeg',
        'jpg'
])

sub = SubReddit('r/LandscapePhotography')
amount = sub.discover_images(SortBy.hot)
print(amount)

for image in sub.loaded_images:
        print(f'Saving to {image.filename}')
        image.save(to='./images')
```

1.3 Reference

library to scrape subreddits for images.

class reddit_photo_aggregator.**ImageURLValidator**

Image URL validation class to validate image URLs before attempting to download them

classmethod set_image_extension_whitelist(*extension: list[str]*) → None

Set the image extensions that are allowed to be downloaded

Parameters
host (*list[str]*) – list of image extensions

classmethod set_image_host_whitelist(*host: list[str]*) → None

Set the domain hosts that are allowed to be downloaded from

Parameters
host (*list[str]*) – list of host domains

class reddit_photo_aggregator.**SubReddit**(*name: str*)

discover_images(*sorted_by='/hot'*) → int

Aggregates images from the subreddit feed

Parameters
sorted_by (*str, optional*) – *SortBy* value to sort the results by. Defaults to *SortBy.hot*.

Returns
the amount of images found

Return type
int

property loaded_images: list[*reddit_photo_aggregator.image.Image*]

List of *Image* loaded when running *.discover_images*

Returns
List of discovered image

Return type
list[*Image*]

property url: str

The URL of the subreddit based on *name*

class reddit_photo_aggregator.**Image**(*url: str*)

Image object used to store and load the image content from URL

property filename: str

the filename that the image will be saved using

load() → None

Retrieves the image content from the URL

Raises
ImageRetrieveError – When resource returns any status code other than 200

save(*to: str = './'*)

Saves the image to disk

Parameters
to (*str, optional*) – Additional path to append to cwd. Defaults to './'.

class reddit_photo_aggregator.**SortBy**

Used to sort sub-reddit results

controversial = '/controversial'

hot = '/hot'

new = '/new'

rising = '/rising'

top = '/top'

PYTHON MODULE INDEX

r

INDEX