
URL2Image

Feb 05, 2020

Contents:

1	url2_image package	1
1.1	Submodules	1
1.2	url2_image.app module	1
1.3	Module contents	2
2	wsgi module	3
3	Indices and tables	5
	Python Module Index	7
	HTTP Routing Table	9
	Index	11

1.1 Submodules

1.2 url2_image.app module

GET /get Image

Main API endpoint. This takes in an URL and returns an image.

Args: url (str): The URL of the target website to be downloaded.

width (int): Width of the target image (default=1920)

height (int): Height of the target image (default=1080)

format (str): The format of the target image. Either png or jpg (default=png)

quality (int): JPEG Quality from 0 to 100 (default=60)

Returns: A bytestream containing the downloaded website as image

GET /version

API endpoint to retrieve version information of the service.

Example: The Api endpoint can be queried as follows:

```
$ curl "localhost:5000/version"  
$ curl "localhost:5000/version?format=json"
```

Args: format (str): (Optional) when format=json a json object is returned

Returns: Version information of the service either as plaintext or json. Information contains version number, git hash (one commit behind) and git branch.

POST /login

Login the user using flask_jwt_extended. Accepts json as input and returns an access token. The configuration can be set via environment variables:

- `JWT_SECRET_KEY`: The secret key for JWT
- `JWT_USER`: The username of the JWT login. Default: `user`
- `JWT_PASSWORD`: The password for the JWT login. Default: `url2image`
- `USE_LOGIN`: Enables/Disables the requirement for login via JWT. Default: `True`
- `JWT_ACCESS_TOKEN_EXPIRES`: The expiration time (in seconds) of `False` for no expiration of the JWT. Default: `False`

A basic login can be achieved via:

```
curl -H "Content-Type: application/json" -X POST -d '{"username":"user", "password": "url2image"}' "http://localhost:5000/login"
{
  "access_token": "TOKEN"
}
```

The authorization is then done in the header:

```
curl -H "Authorization: Bearer TOKEN" "http://localhost:5000/getImage?url=google.de"
```

Args:

- username: The username to login
- password: The users password

Returns: The generated access token.

GET /

Return “Hello World” as a default for the “/” route

Main file for the `url2_image` app

1.3 Module contents

CHAPTER 2

wsgi module

WSGI config for the running of the app with gunicorn

CHAPTER 3

Indices and tables

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

U

url2_image, 2
url2_image.app, 2

W

wsgi, 3

HTTP Routing Table

/

GET /, 2

/getImage

GET /getImage, 1

/login

POST /login, 1

/version

GET /version, 1

U

`url2_image` (*module*), 2

`url2_image.app` (*module*), 2

W

`wsgi` (*module*), 3