
flask-github-proxy Documentation

Release

Thibault Clérice

January 19, 2017

1	What ?	1
2	Installation	3
3	Example	5
4	Funding and original development	7
4.1	Flask Github Proxy API Documentation	7
4.2	Flask Github Proxy Workflow	12

What ?

Flask Github proxy is a flask extension to provide saving resources to GitHub. To use this service, you will need a github token (Generate token : <https://github.com/settings/tokens>)

Installation

To install it, simply do : `pip3 install flask_github_proxy` or

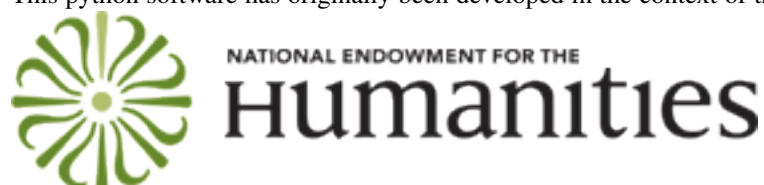
```
git clone https://github.com/ponteineptique/flask-github-proxy.git
cd flask-github-proxy
python3 setup.py install
```

Example

```
1 from flask import Flask
2 from flask_github_proxy import GithubProxy
3 from flask_github_proxy.models import Author
4
5 app = Flask("name")
6 proxy = GithubProxy(
7     "/proxy",
8     "ponteineptique/dummy1",
9     "alpheios-project/dummy1",
10    secret="something",
11    token="Github Token",
12    app=app,
13    origin_branch="master",
14    default_author=Author(
15        "Github Proxy",
16        "anonymous@github.com"
17    )
18 )
19 app.run()
```

Funding and original development

This python software has originally been developed in the context of the [Syriaca](#) Project, under the funding of NEH.



Index :

4.1 Flask Github Proxy API Documentation

4.1.1 Extension

```
class flask_github_proxy.GithubProxy(prefix, origin, upstream, secret, token, de-  
fault_branch=None, master_upstream='master', mas-  
ter_fork='master', app=None, default_author=None,  
logger=None, json_log_formatting=True)
```

Provides routes to push files to github and open pull request as a service

Generate token : <https://github.com/settings/tokens>

You can use GithubProxy.DEFAULT_BRANCH.NO and GithubProxy.DEFAULT_BRANCH.AUTO_SHA to build branch name automatically.

Default user is GithubProxy.DEFAULT_AUTHOR

Parameters

- **prefix** – URI Prefix
- **origin** – Origin Repository (Repository to Pull Request From)
- **upstream** – Upstream Repository (Repository to Pull Request To)
- **secret** – Secret Key. Used to check provenance of data
- **token** – Github Authentication User Token
- **default_branch** (*str*) – Default Branch to push to
- **pull_req** – Origin Branch to build on
- **master_fork** – Origin Repository Master Branch Name (Branch to update on the fork using the finale repositoru)

- **master_upstream** – Upstream Repository Master Branch Name (Branch to Pull Request to)
- **app** – Flask Application to connect to
- **default_author** (*Author*) – Default Author for Commit and Modification

Variables

- **URLS** – URLs routes of the proxy
- **DEFAULT_AUTHOR** – Default Author
- **blueprint** – Flask Blueprint Instance for the Extension
- **prefix** – Prefix of the Blueprint
- **name** – Name of the Blueprint
- **origin** – Git Repository to pull request from
- **upstream** – Git Repository to pull request to
- **default_author** – Default Author
- **secret** – Secret / Salt used to check provenance of data to be pushed

class **DEFAULT_BRANCH**

Parameter Constant for the default_branch parameter

Variables

- **NO** – Default Branch is equal to origin branch
- **AUTO_SHA** – Generate a sha based on the file to create the branch

class **GithubProxy.ProxyError** (*code, message, step=None, context=None*)

Carries information for errors

Parameters

- **code** (*int*) – HTTP Code Error
- **message** (*str or tuple*) – Message to display or a dict and its key

Variables

- **code** – HTTP Code Error
- **message** – Message to display

static **AdvancedJsonify** (*data, status_code*)

Advanced Jsonify Response Maker

Parameters

- **data** – Data
- **status_code** – Status_code

Returns Response

response (*callback=None*)

View representation of the object

Parameters **callback** (*function*) – Function to represent the error in view. Default : flask.jsonify

Returns View

GithubProxy.**check_sha** (*sha, content*)

Check sent sha against the salted hash of the content

Parameters

- **sha** – SHA sent through fproxy-secure-hash header
- **content** – Base 64 encoded Content

Returns Boolean indicating equality

`GithubProxy.default_branch(file)`

Decide the name of the default branch given the file and the configuration

Parameters **file** – File with informations about it

Returns Branch Name

`GithubProxy.get(file)`

Check on github if a file exists

Parameters **file** – File to check status of

Returns File with new information, including blob, or Error

Return type File or self.ProxyError

`GithubProxy.get_ref(branch, origin=None)`

Check if a reference exists

Parameters **branch** – The branch to check if it exists

Returns Sha of the branch if it exists, False if it does not exist, self.ProxyError if it went wrong

`GithubProxy.init_app(app)`

Initialize the application and register the blueprint

Parameters **app** – Flask Application

Returns Blueprint of the current nemo app

Return type flask.Blueprint

`GithubProxy.make_ref(branch)`

Make a branch on github

Parameters **branch** – Name of the branch to create

Returns Sha of the branch or self.ProxyError

`GithubProxy.patch_ref(sha)`

Patch reference on the origin master branch

Parameters **sha** – Sha to use for the branch

Returns Status of success

Return type str or self.ProxyError

`GithubProxy.pull_request(file)`

Create a pull request

Parameters **file** – File to push through pull request

Returns URL of the PullRequest or Proxy Error

`GithubProxy.put(file)`

Create a new file on github

Parameters **file** – File to create

Returns File or self.ProxyError

`GithubProxy.r_main()`

Main Route of the API

Returns Response

`GithubProxy.r_receive(filename)`

Function which receives the data from Perseids

- Check the branch does not exist
- Make the branch if needed
- Receive PUT from Perseids
- Check if content exist
- Update/Create content
- Open Pull Request
- Return PR link to Perseids

It can take a “branch” URI parameter for the name of the branch

Parameters `filename` – Path for the file

Returns JSON Response with status_code 201 if successful.

`GithubProxy.r_update()`

Updates a fork Master

- Check the ref of the origin repository
- Patch reference of fork repository
- Return status to Perseids

Returns JSON Response with status_code 201 if successful.

`GithubProxy.request(method, url, **kwargs)`

Unified method to make request to the Github API

Parameters

- **method** – HTTP Method to use
- **url** – URL to reach
- **kwargs** – dictionary of arguments (params for URL parameters, data for post/put data)

Returns Response

`GithubProxy.update(file)`

Make an update query on Github API for given file

Parameters `file` – File to update, with its content

Returns File with new information, including success (or Error)

4.1.2 Models

`class flask_github_proxy.models.File(path, content, author, date, logs)`

File Representation

Parameters

- **path** (*str*) – Path of the file on the repository
- **content** (*byte*) – Base64 encoded content of the file
- **author** (*Author*) – Author of the file
- **date** (*str*) – Date of the modification
- **logs** (*str*) – Message about the modification (Usually for commit message)

Variables

- **path** – Path of the file on the repository
- **content** – Content of the file
- **base64** – Base64 Encoded Content of the file
- **author** – Author of the file
- **date** – Date of the modification
- **sha** – Sha hash of the content

dict ()

Builds a dictionary representation of the object (eg: for JSON)

Returns Dictionary representation of the object

class flask_github_proxy.models.**Author** (*name, email*)

The author models carries information about committer and data modifiers.

Note: It behaves as a “static” object : its variables are private one and only getters are registered.

Parameters

- **name** (*str*) – Name of the user
- **email** (*str*) – Email of the user

Variables

- **name** – Name of the user
- **email** – Email of the user

dict ()

Builds a dictionary representation of the object (eg: for JSON)

Returns Dictionary representation of the object

class flask_github_proxy.models.**ProxyError** (*code, message, step=None, context=None*)

Carries information for errors

Parameters

- **code** (*int*) – HTTP Code Error
- **message** (*str or tuple*) – Message to display or a dict and its key

Variables

- **code** – HTTP Code Error
- **message** – Message to display

static AdvancedJsonify (*data*, *status_code*)

Advanced Jsonify Response Maker

Parameters

- **data** – Data
- **status_code** – Status_code

Returns Response

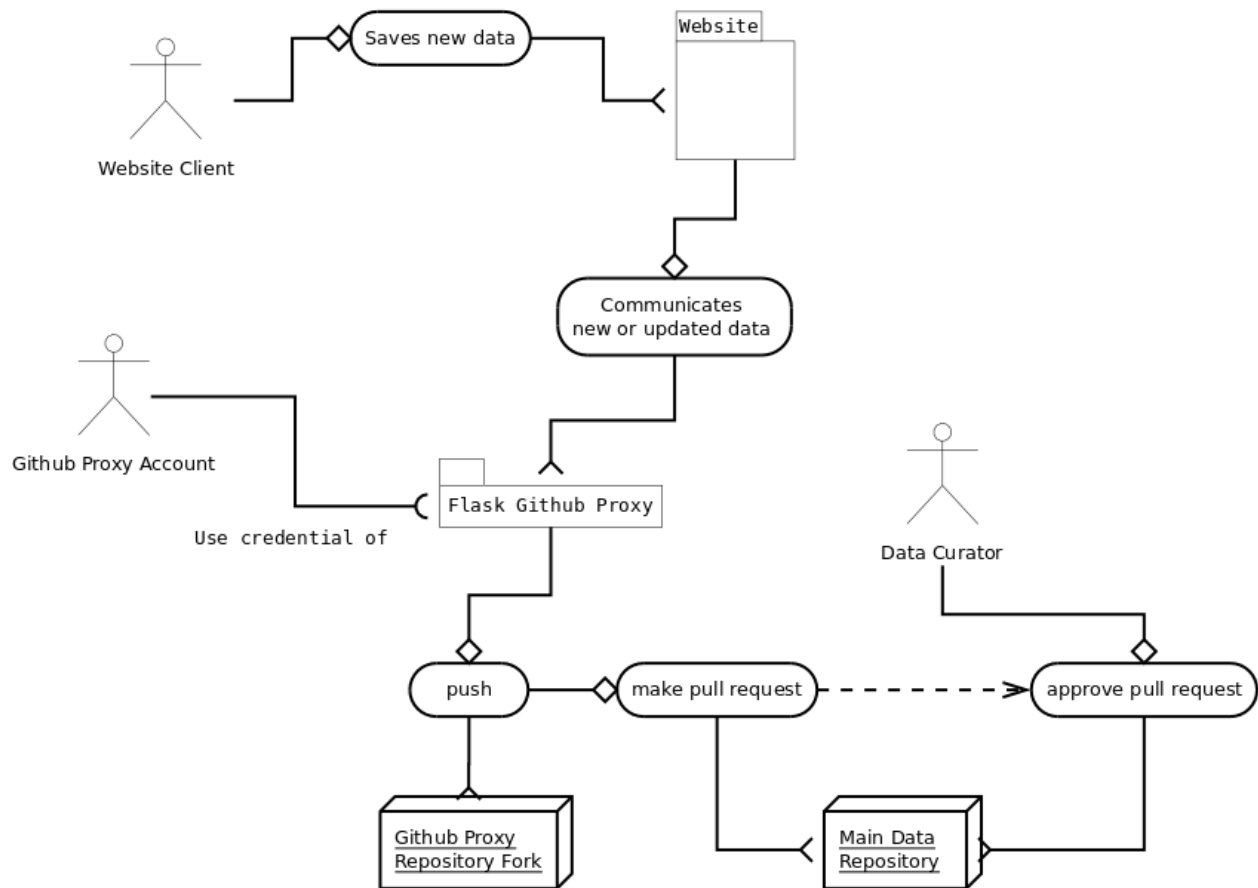
response (*callback=None*)

View representation of the object

Parameters **callback** (*function*) – Function to represent the error in view. Default : flask jsonify

Returns View

4.2 Flask Github Proxy Workflow



A

AdvancedJsonify() (flask_github_proxy.GithubProxy.ProxyError static method), 8
AdvancedJsonify() (flask_github_proxy.models.ProxyError static method), 11
Author (class in flask_github_proxy.models), 11

C

check_sha() (flask_github_proxy.GithubProxy method), 8

D

default_branch() (flask_github_proxy.GithubProxy method), 9
dict() (flask_github_proxy.models.Author method), 11
dict() (flask_github_proxy.models.File method), 11

F

File (class in flask_github_proxy.models), 10

G

get() (flask_github_proxy.GithubProxy method), 9
get_ref() (flask_github_proxy.GithubProxy method), 9
GithubProxy (class in flask_github_proxy), 7
GithubProxy.DEFAULT_BRANCH (class attribute in flask_github_proxy), 8
GithubProxy.ProxyError (class in flask_github_proxy), 8

I

init_app() (flask_github_proxy.GithubProxy method), 9

M

make_ref() (flask_github_proxy.GithubProxy method), 9

P

patch_ref() (flask_github_proxy.GithubProxy method), 9
ProxyError (class in flask_github_proxy.models), 11
pull_request() (flask_github_proxy.GithubProxy method), 9
put() (flask_github_proxy.GithubProxy method), 9

R

run() (flask_github_proxy.GithubProxy method), 9
r_receive() (flask_github_proxy.GithubProxy method), 10
r_update() (flask_github_proxy.GithubProxy method), 10
request() (flask_github_proxy.GithubProxy method), 10
response() (flask_github_proxy.GithubProxy.ProxyError method), 8
response() (flask_github_proxy.models.ProxyError method), 12

U

update() (flask_github_proxy.GithubProxy method), 10