Crappyspider Documentation

Release 0.1

Peopledoc

Sep 27, 2017

Contents

- source: https://github.com/novapost/crappyspider
- ticketing: https://github.com/novapost/crappyspider/issues
- documentation: http://crappyspider.readthedocs.org/en/latest/

CHAPTER 1

Sample configuration file

```
{
    "start_urls": ["http://example.com/dashboard"],
    "allowed_domains": ["example.com"],
    "patterns": [
        "/article/[0-9]+",
        "/user/.+",
        "/search/\\?"
    ],
    "excluded_patterns": [
        "/logout/"
    ],
    "credential": {
        "email": "myemail",
        "password": "mypassword"
    },
    "login_error_selector": ".errorlist"
}
```

CHAPTER 2

Parameters

Start urls

A set of urls the crawler will start.

Allowed domains

A set of domains to which the crawl will be restricted.

Patterns

A set of regex representing url patterns to visit only once.

Excluded patterns

A set of regex representing url patterns to ignore.

Credential

The credential hash is optional since it is used for sites requiring login. The underlying hashes are used as names for the login form inputs, so you might have to change them.

Example:

```
{
    "credential": {
        "email": "myemail",
        "password": "mypassword"
    },
}
```

There’s another credential scheme, take a list, with the name of the field.

Example:

```
{
    "credential": ["email", "password"]
}
```

Then add in your environnement two variables.

Example:

```
export CRAPPYSPIDER_EMAIL=test@test.com
export CRAPPYSPIDER_PASSWORD=my_password
```

Login error selector

The selector where the crawler should look for errors after a login attempt.

CHAPTER 3

Documentation of the command line

A few options are avaible via the command line.

Each option is to be preceeded by the *-a* argument. See examples below.

Config

This option is used to set the path of the configuration file. It is not required, but if not specified, you will need to enter the start_urls and allowed_domains parameters via the command line. Usage example:

```
scrapy crawl crappyspider -a config=my_rule.json
```

Output format

This option is used to set the formating of the generated report. Default value is json. Supported formats are json and yaml. Usage example:

```
scrapy crawl crappyspider -a config=my_rule.json -a output_format=json
```

Output filename

This option is used to set the name of the report file. Default value is *output* (and is thus saved in the current directory). Usage example:

```
scrapy crawl crappyspider -a config=my_rule.json output_filename=$HOME/visited_url.
↪json
```

Start urls

This option is used to set the page where the crawler will start. Usage example:

```
scrapy crawl crappyspider -a  start_urls=http://test.com -a allowed_domains=test.com
scrapy crawl crappyspider -a  start_urls="http://foo.com http://bar.com" -a allowed_
↪domains="foo.com bar.com baz.com"
```

Allowed domains

This option is used to set the allowed domains, to which the crawl will be restricted. Usage example:

```
scrapy crawl crappyspider -a  start_urls=http://test.com -a allowed_domains=test.com
```

CHAPTER 4

How to contribute

Choice to use a fairly close process to what is in us in other Free and Open Source Software projects.

It is recommended for a new developer or contributor to read this document.

Create a ticket

For all new features or bug fixes, fill a new ticket in the issue tracker.

Try to be specific in the ticket, with a proper title and description, if you can provide steps to help the developper to understand the process or the feature request. Without enough context, it can be difficult to understand.

The ticket number is then used as a reference for the branch and commits.

Create a branch

Except in exceptional cases, a developer must not directly commits in master.

The rule also applies to project core developers.

The branch must be prefixed by the ticket number, for example:

```
git pull origin master
git branch 425-assign_nobody
```

Don't forget to be update master before creating your new branch from it.

Unit test

Except in exceptional cases, a pull request must come with tests.

It is important to write tests, it's annoying but it pays in the long run, not regression, refactoring opportunities, and so on.

If this is a bug fix, write a test for this case.

A branch will not merged if there is no test. Learn from the many existing tests, ask questions if needed.

Code review

To err is human, several pair of eyes are better than one.

A developer should not merge in the master.

The code must be reviewed by another developer and merged by an experienced core developer.

It is possible to discuss and point out any mistakes or errors.

If we detect a mistake after the branch as been merged, we will all work together to fix it as soon as possible.

Hook to exec flake8

We want our beautiful code to be PEP8 compliant so it is strongly recommended to add this pre-commit hook in git (peopleask/.git/hooks/pre-commit):

```
#!/usr/bin/env bash
flake8 ./peopleask/
```

If not pep8 it will not be commiting

Conclusion

It is important to follow up the process. It is not to be annoying, but it opens the discussion, helps to progress and to have a better quality of code as well as letting every developer sees and valid project progress.

Do not take the critic for you, each developer make mistakes and it is fine.

Despite this process of functional bugs and errors happen and it is fine as well, never forget it.

CHAPTER 5

What is it?

crappyspider is a generic scrapy spider:

- it crawls a website
- logs on the standard output the visited urls and their http codes
- generates a log file with the visited urls (more data coming soon, such as http codes ;))

CHAPTER 6

Features

- a login step
- restric crawling through urls patterns (see ‘patterns’ in the ‘config’ doc page)
- exclude patterns of urls (see ‘excluded_patterns’ in the ‘config’ doc page)

CHAPTER 7

Usage

```
scrapy crawl crappyspider -a start_urls=http://example.com -a allowed_domains=example.
↪com
scrapy crawl crappyspider -a conf=conf.json -a output=outputfile -a output_format=json
```

CHAPTER 8

Use cases

After deployment:

You will be able to read through the standard input to pick up any HTTP error such as 500's (pink highlight <3)

```
scrapy crawl crappyspider -a start_urls=http://deployed-site.com -a allowed_
↪domains=deployed-site.com
```

After code change:

Feed any functional testing script with this crawler output file.

```
scrapy crawl crappyspider -a config=config.json -a output_filename=log.json
```

CHAPTER 9

Acknowledgements

Thanks to Peopledoc for freeing the project.