Crawler Documentation

Release 1.0

Eric Holscher

Feb 10, 2018

User Guide

Our Crawler will make your life as a web developer easier. You can learn more about it in our documentation.

CHAPTER 1

Installation

At the command line:

```
easy_install crawler
```

Or, if you have pip installed:

```
pip install crawler
```

CHAPTER 2

Support

The easiest way to get help with the project is to join the `#crawler` channel on Freenode. We hang out there and you can get real-time help with your projects. The other good way is to open an issue on Github.

The mailing list at https://groups.google.com/forum/#!forum/crawler is also available for support.

CHAPTER 3

Cookbook

3.1 Crawl a web page

The most simple way to use our program is with no arguments. Simply run:

```
python main.py -u <url>
```

to crawl a webpage.

3.2 Crawl a page slowly

To add a delay to your crawler, use *-d*:

```
python main.py -d 10 -u <url>
```

This will wait 10 seconds between page fetches.

3.3 Crawl only your blog

You will want to use the *-i* flag, which while ignore URLs matching the passed regex:

```
python main.py -i "^blog" -u <url>
```

This will only crawl pages that contain your blog URL.

CHAPTER 4

Command Line Options

These flags allow you to change the behavior of **`Crawler`**. Check out how to use them in the *Cookbook*.

`-d` `<sec>`, **`--delay`** `<sec>`
: Use a delay in between page fetchs so we don't overwhelm the remote server. Value in seconds.

  Default: 1 second

`-i` `<regex>`, **`--ignore`** `<regex>`
: Ignore pages that match a specific pattern.

  Default: None

CHAPTER 5

Crawler Python API

Getting started with Crawler is easy. The main class you need to care about is *Crawler*

5.1 crawler.main

Main Module

class crawler.main.**Crawler**(*url*, *delay*, *ignore*)

Main Crawler object.

Example:

```
c = Crawler('http://example.com')
c.crawl()
```

Parameters

- **delay** – Number of seconds to wait between searches
- **ignore** – Paths to ignore

crawl()

Crawl the URL set up in the crawler.

This is the main entry point, and will block while it runs.

get(*url*)

Get a specific URL, log its response, and return its content.

Parameters **url** – The fully qualified URL to retrieve

crawler.main.**run_main**()

A small wrapper that is used for running as a CLI Script.

5.2 crawler.utils

utils.**should_ignore**(*ignore_list*, *url*)
Returns True if the URL should be ignored

Parameters

- **ignore_list** – The list of regexs to ignore.
- **url** – The fully qualified URL to compare against.

```
>>> should_ignore(['blog/$'], 'http://ericholscher.com/blog/')
True
```

```
# This test should fail
>>> should_ignore(['home'], 'http://ericholscher.com/blog/')
True
```

utils.**log**(*url*, *status*)
Log information about a response to the console.

Parameters

- **url** – The URL that was retrieved.
- **status** – A status code for the *Response*.

```
>>> log('http://ericholscher.com/blog/', 200)
OK: 200 http://ericholscher.com/blog/
```

```
>>> log('http://ericholscher.com/blog/', 500)
ERR: 500 http://ericholscher.com/blog/
```

```
# This test should fail
>>> log('http://ericholscher.com/blog/', 500)
OK: 500 http://ericholscher.com/blog/
```

Python Module Index

C

Index

Symbols

C

G

L

R

S