spyda Documentation

Release 0.0.3.dev

James Mills

April 27, 2014

Contents

Release 0.0.3.dev

Date April 27, 2014

CHAPTER 1

About

spyda is a simple tool and library written in the Python Programming Language to crawl a given url whilst allowing you to restrict results to a specified domain and optionally also perform pattern matching against URLs crawled. spyda will report on any URLs it was unable to crawl along with their status code and store successfully crawled links and their content in a directory structure that matches the domain and URLs searched.

spyda was developed at Griffith University as a tool and library to assist with web crawling tasks and data extraction and has been used to help match researcher names against publications as well as extract data and links from external sources of data.

- Visit the Project Website
- Read the Docs
- Download it from the Downloads Page

1.1 Requirements

- restclient
- cssselect
- lxml
- url
- nltk
- calais
- BeautifulSoup

spyda also comes basic documentation and a full comprehensive unit test suite which require the following:

To build the docs:

- sphinx
- sphinxcontrib-bitbucket

To run the unit tests:

- pytest
- circuits

1.2 Installation

The simplest and recommended way to install spyda is with pip. You may install the latest stable release from PyPI with pip:

```
> pip install spyda
```

If you do not have pip, you may use easy_install:

```
> easy_install spyda
```

Alternatively, you may download the source package from the PyPI Page or the Downloads page on the Project Website; extract it and install using:

```
> python setup.py install
```

You can also install the latest-development version by using `pip` or `easy_install`:

```
> pip install spyda==dev
```

or:

```
> easy_install spyda==dev
```

For further information see the spyda documentation.

1.3 Supported Platforms

- Linux, FreeBSD, Mac OS X
- Python 2.6, 2.7

Windows: We acknowledge that Windows exists and make reasonable efforts to maintain compatibility. Unfortunately we cannot guarantee support at this time.

CHAPTER 2

Documentation

2.1 API Documentation

2.1.1 spyda

spyda Package

spyda Package

Spyda - Python Spider Tool and Library

spyda is a set of tools and a library written in the Python Programming Language for web crawling, article extraction entity matching and rdf graog geberatuib.

copyright CopyRight (C) 2012-2013 by James Mills

crawler Module

Crawler

spyda.crawler.**crawl**(*root_url, blacklist=None, content_types=['text/html', 'text/xml'], max_depth=0, patterns=None, verbose=False, whitelist=None*)

Crawl a given url recursively for urls.

Parameters

- **root_url** (*str*) – Root URL to start crawling from.
- **blacklist** (*list or None*) – A list of blacklisted urls (matched by regex) to not traverse.
- **content_types** (*list or CONTENT_TYPES*) – A list of allowable content types to follow.
- **max_depth** – Maximum depth to follow, 0 for unlimited depth.
- **max_depth** – int
- **patterns** (*list or None or False*) – A list of regex patterns to match urls against. If evaluates to `False`, matches all urls.
- **verbose** – If `True` will print verbose logging
- **verbose** – bool
- **whitelist** (*list or None*) – A list of whitelisted urls (matched by regex) to traverse.

Returns A dict in the form: {"error": set(...), "urls": set(...)} The errors set contains 2-item tuples of (status, url) The urls set contains 2-item tuples of (rel_url,abs_url)

Return type dict

In verbose mode the following single-character letters are used to denonate meaning for URLs being processed:

- 1.(I)nvalid URL
- 3.Did not match allowed (C)ontent Type(s).
- 6.(F)ound a valid URL
- 19.(S)een this URL before
- 5.(E)rror fetching URL
- 16.Did not match supplied (P)attern(s).
- 22.URL already (V)isitied
- 2.URL blacklisted
- 23.URL whitelisted

Also in verbose mode each followed URL is printed in the form: <status> <reason> <type> <length> <link> <url>

spyda.crawler.**parse_options**()

spyda.crawler.**main**()

extractor Module

Web Extraction Tool

spyda.extractor.**calais_options**(*parser*)

spyda.extractor.**parse_options**()

spyda.extractor.**extract**(*source*, *filters*)

spyda.extractor.**job**(*opts*, *source*)

spyda.extractor.**main**()

matcher Module

Entity Matching Tool

spyda.matcher.**parse_options**()

spyda.matcher.**build_datasets**(*opts*, *source*)

spyda.matcher.**job**(*opts*, *datasets*, *source*)

spyda.matcher.**main**()

processors Module

utils Module

Utilities

spyda.utils.**is_url**(*s*)

spyda.utils.**dict_to_text**(*d*)

spyda.utils.**unescape**(*text*)
Removes HTML or XML character references and entities from a text string.

Parameters **text** – The HTML (or XML) source text.

Returns The plain text, as a Unicode string, if necessary.

spyda.utils.**unichar_to_text**(*text*)
Convert some common unicode characters to their plain text equivilent.

This includes for example left and right double quotes, left and right single quotes, etc.

spyda.utils.**get_close_matches**(*word*, *possibilities*, *n=3*, *cutoff=0.6*)
Use SequenceMatcher to return list of close matches.

word is a sequence for which close matches are desired (typically a string).

possibilities is a list of sequences against which to match word (typically a list of strings).

Optional arg n (default 3) is the maximum number of close matches to return. n must be > 0.

Optional arg cutoff (default 0.6) is a float in [0.0, 1.0]. Possibilities that don't score at least that similar to word are ignored.

The best (no more than n) matches among the possibilities are returned in a list, sorted by similarity score, most similar first.

spyda.utils.**fetch_url**(*url*)

spyda.utils.**log**(*msg*, **args*, ***kwargs*)

spyda.utils.**error**(*e*)

spyda.utils.**status**(*msg*, **args*)

spyda.utils.**parse_html**(*html*)

spyda.utils.**doc_to_text**(*doc*)

spyda.utils.**get_links**(*html*, *badchars='"\' \x0b\x0c\t\n\r'*)

version Module

Version Module

So we only have to maintain version information in one place!

2.2 TODO

- Nothing planned at this stage.

2.3 Road Map

- No immediate future plans at this stage... Feel free to Create an Issue.

2.4 Changes

2.4.1 spyda 0.0.3.dev

- Fixed long option for Content-Type restriction to `--content-type`
- Properly removed deprecated option `--allowed_url`.
- Restrict Content Types of followed URIs). Adds `-t/--content-type` options to the `crawl` CLI and keyword argument to the `crawl()` function.
- Ported tests to circuits 3.x
- Added support for Python 2.6
- Added Continuous Integration

2.4.2 spyda 0.0.2 (2013-11-19)

- Updated the README
- Added build status
- Added install instructions

2.4.3 spyda 0.0.1 (2013-11-19)

- Initiali Public Release

CHAPTER 3

Indices and tables

- *genindex*
- *modindex*
- *search*

S