yc-lexisnexis-scraper Documentation

Release 0.1

Yanchuan Sim

Mar 25, 2017

Contents

This small Python module is intended to be a demonstration of how to use PhantomJS and Selenium WebDriver in Python to scrape websites. This code is not intended for use in anyway that contravenes LexisNexis Academic Terms of Use.

Disclaimer: We accept no liability for the content of this code, or for the consequences of any actions taken on the basis of the information provided.

CHAPTER 1

LexisNexis Scraper

class lexisnexis.**LexisNexisScraper**(*wait_timeouts=(15, 180)*, *documents_per_download=(250, 500)*, *user_agent_string=u'Mozilla/5.0 (Macintosh; Intel Mac OS X 10.9; rv:25.0) Gecko/20100101 Firefox/25.0'*, *mass_download_mode=False*)

Class for downloading documents given a query string to Lexis Nexis academic (http://www.lexisnexis.com/hottopics/lnacademic/).

Example:

```
downloader = LexisNexisScraper(mass_download_mode=True)
for (content, (doc_index, doc_count)) in downloader.iter_search_results(6318,
↪'DATE(=1987)'):
  print doc_id
```

This code uses PhantomJS and Selenium Webdriver to scrape LexisNexis pages.

__init__(*wait_timeouts=(15, 180)*, *documents_per_download=(250, 500)*, *user_agent_string=u'Mozilla/5.0 (Macintosh; Intel Mac OS X 10.9; rv:25.0) Gecko/20100101 Firefox/25.0'*, *mass_download_mode=False*)

Constructs a downloader object.

Parameters

- **wait_timeouts** (*float*, *float*) – tuple of *(short, long)* where *short* and *long* are the no. of seconds to wait while page elements are loaded (for Webdriver). *long* timeout is used when waiting for LexisNexis to format documents for mass downloads.
- **documents_per_download** (*int*, *int*) – a range specifying the number of documents to download each time when using `mass_download_mode`.
- **user_agent_string** (*str*) – the user agent string that PhantomJS should declare itself to be.
- **mass_download_mode** (*bool*) – whether to mass download articles using the download link or page through each document one by one and download.

iter_search_results(*csi*, *search_query*, *start_from=1*)

A generator function that executes LexisNexis search query on source data CSI (`csi`), with query `search_query` and downloads all documents returned by search.

Parameters

- **csi** (*str*) – LexisNexis CSI (see http://amdev.net/rpt_download.php for full list).
- **search_query** (*str*) – execute search query string.
- **start_from** (*int*) – document index to start downloading from.

Returns a tuple *(doc_content, (index, results_count))*, where *doc_content* is the HTML content of the *index'th document, and 'results_count* is the number of documents returned by specified search query.

exception lexisnexis.**SwitchFrameException**(*frame_name*)

Exception class when we are unable to load the require page properly. This is usually due to #. Page taking too long to load. This happens sometimes when loading LexisNexis for the first time. #. Improper page loading.

__weakref__

list of weak references to the object (if defined)

Python Module Index

l

Index

Symbols

I

L

S