

COURSE TITLE: CEF 506 - PYTHON/PERL PROGRAMMING DEVELOPMENT

NAME	MATRICULATION NUMBER
FRU ISIDORE CHE	FE12A078
THEOPHILUS WABA NASALI	FE12A183
NKENG NEWTON	FE12A140
MOBA MELVIS RINGNYU	FE12A107
ENOMBE THIERRY EWANE	FE12A053
ALANGI DERRICK	FE12A113

Title: Creation of a multi-threaded web spider

Hardware Specification

This application was developed and run on a machine with the following hardware requirement. It could be run on a machine with better specification

RAM: 3.00GB

Processor: AMD Phenom™ II N620 Dual-Core Processor

Processor speed: 2.80GHz

Hard drive: 1TB

Mark: Hewlett Packard (HP) ProBook 6455b

Software Specification

The following software used:

Operating System: Ubuntu 14.04

Editor: Sublime text editor 3

Command-line: Ubuntu terminal

Libraries:

Web crawler libraries needed

- **Scrapy** - a library used for web crawling.
 - The following are imported from the Scrapy library:
Item, Selector, Field and BaseSpider

FLOWCHART

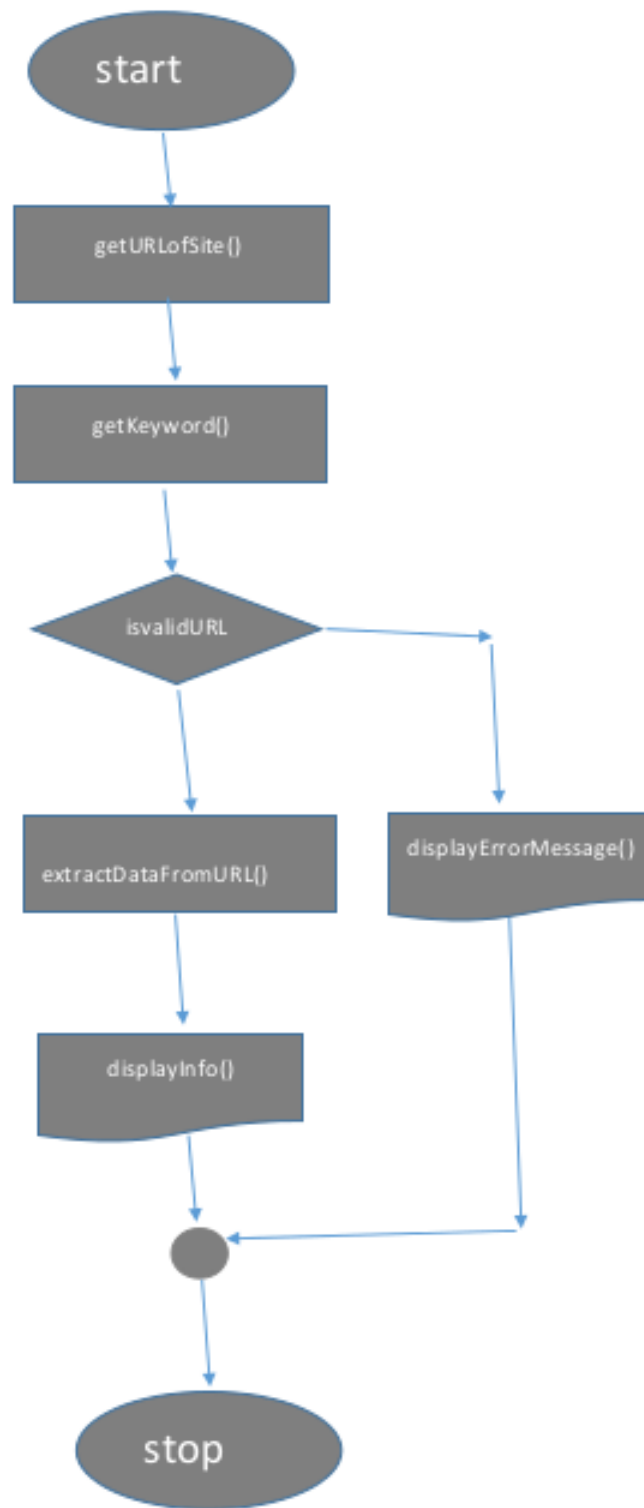


Figure 1: FlowChart of the spider webcrawler

CODE

```
from scrapy.item import Item, Field
from scrapy.selector import Selector
from scrapy.spider import BaseSpider

class Item(Item):
    # Items are defined in a declarative style. If you attempt to store a field
    # not defined here, an exception will be raised.
    title = Field()
    content = Field()
    links = Field()
    url = Field()

class SpiderMultithreaded(BaseSpider):
    """This spider crawls the website example.com."""
    # The name is the unique identifier for this spider.
    name = 'SpiderMultithreaded'
    # These URLs are the initial requests performed by the spider.
    start_urls = [
        'http://efarm.dev',
    ]

    # The default callback for the start urls is `parse`.
    # This method must return either items or requests.
    def parse(self, response):
        # Instance selector in order to query the html document.
        sel = Selector(response)
        # Instance our item. The item class have a dict-like interface.
        item = Item()
        # The method `extract()` always returns a list. So we extract the
        # first value with [0]. This is not needed when using the item loaders.
        # We can use a XPath rule to extract information from the html.
        item['title'] = sel.xpath('//h1/text()').extract()[0:]
        # Or we can use a CSS expression as well.
        item['content'] = sel.css('p::text').extract()[0:]
        item['links'] = sel.xpath('//a/text()').extract()[0:]
        item['url'] = response.url
```

Spaces:

Figure 2: Code snippet for spider webcrawler

OUTPUT

```
2016-06-07 16:51:59 [scrapy] INFO: Spider opened
2016-06-07 16:51:59 [scrapy] INFO: Crawled 0 pages (at 0 pages/min), scraped 0 items (at 0 items/min)
2016-06-07 16:51:59 [scrapy] DEBUG: Telnet console listening on 127.0.0.1:6023
2016-06-07 16:51:59 [scrapy] DEBUG: Crawled (200) <GET http://efarm.dev> (referer: None)
2016-06-07 16:52:00 [scrapy] DEBUG: Scraped from <200 http://efarm.dev>
{'content': [u'All in one online market place for agricultural products, you can try us. ',
             u'At eFarm.cm, it's natural, it's Green. Just search from the varieties of products from all over Africa. ',
             u'Selling at eFarm.cm is quiet easy. 4 Simple steps: register, login, post products and wait for buyers... ',
             u'We guarantee quantity, varieties to bid, available to resolve issues, we are simply the best. ',
             u'Naturally growing tomatoes from one eFarm.cm user's farm. Place order while it's still early. ',
             u'Pepper',
             u'Pepper',
             u'Carbage',
             u'Carbage',
             u'Dried Fish',
             u'Dried Fish',
             u'Easy Polo Black Edition',
             u'Easy Polo Black Edition',
             u'Easy Polo Black Edition',
             u'Easy Polo Black Edition',
             u'Easy Polo Black Edition',
             u'Easy Polo Black Edition',
             u'You can subscribe here if you want to be notified about out latest products',
             u'Copyright \xa9 2016 eFarm.cm, powered by GericomGroup. All rights reserved.',
             u'Designed by '],
 'links': [u' (+237) 650-000-100',
           u' store@efarm.cm',
           u'XAF',
           u'USD',
           u'English',
           u'French',
           u' ',
           u'Fixed Products',
           u'Best Offer Products',
           u'Auction Products',
           u'Promotion Products',
           u'Products',
           u'Category',
           u'Price',
           u'Payment ']
```

Figure 3: Snapshot of a section (content and links) of the web crawler output

Importance of Web crawling

Some of the importance of web crawling include:

- *Used to search for any important information on the web e.g jobs*
- *Improve the speed of a search engine*