Introduction to Scraping and scrapy

Getting started guide.

Tapasweni Pathak

- Software Developer @SAP Labs.
- GSOC 2015 Mentor for Systers Org.
- OWASP Summer Code Sprint 2015 Student.
- FOSS Enthusiast.
- <3es to code.

Tapasweni Pathak

Scrape?

- Extracting the information you want from any web page.

Why ScraPe?

- Not everyone has a API.
- Not everyone expose there API.
- Not all APIs are understandable.
- You have the control.
- Data Collection for some analysis.

Why SCRaPy?

- Portable, open source, 100% Python.
- Simple
- Productive (Very actually)
- Comes with lot of functionalities included.
- Extensively documented.
- JSON, CSV, XML output.

- Once you are into Scrapy, you can write a spider in less than 5 minutes that download images, creates thumbnails and export the extracted data directly to csv or json.

Let’s do it?

Installations

- Open your Cloud9 account.
- Create custom workspace.
- Create new project.
- Open terminal.
- Sudo pip install Scrapy

scrapy startproject ScrapScrapy

New Scrapy Project

```
ScrapScrapy/
        scrapy.cfg
        ScrapScrapy/
                __init__.py
                items.py
                pipelines.py
                settings.py
                spiders/
                        __init__.py
                        ...
```

Directory Structure of Scrapy Project

Auto generated Python Files?

- Items.py
  This file has the fields that you want to scrap and store from a website.
- setting.py
  Some settings, like allowing redirection, 404 while scraping
- spiders (folder/directory)
  In this you will store your spider. You can name your spider anything.
- pipelines.py
  Post Processing.
  You filter your data by adding your code in this files. Your drop duplicates by adding some lines of code in this file.

Let’s Write some code Now?

Spiders are classes which define..

- How to perform the crawling?
- How to extract structured data from the web pages.
- Spiders are the place where you define the custom behavior for crawling and parsing pages for a particular site.

BaseSpider

- Simplest Spider
- The one from which every other spider must inherit from.
- No special functionality.
- Requests given start_urls and calls the spider method parse for each resulting response.

- Parse

  Default callback used by Scrapy to process downloaded response, when their request don't specify a callback.

  Processes the data and returns scraped data.
- Start URL
- Allowed domains.
- ...and everything else.

Understanding the code

CrawlSpider

- Most commonly used spider for websites.
- Let's you define rules and will follow them for recursive crawling.
- Generic enough. Override it according to your needs.
- Rules

  Defines certain behavior / pattern for crawling a website.

- XMLFeedSpider
  Specially for XML feeds.

- CSVFeedSpider
  For CSV feed.

What else?

scrapy crawl spider-name -o output_file_name.csv -t csv

Saving as json/csv

XPath

- In simple terms it’s the path to the data which you need to extract.
- Xpath can be confusing sometimes.
- Then use cssselect. Very close in speed.

How to get the xpath?

- Go to the developer tools in your browser.
- Move to the Elements tab, and right click on the data and copy the xpath.
- You can use it in your spider with some minor changes. Check in the console tab before using or for debugging purpose.

- Beautiful Soup
- LXML
- Scrapy
- Many More.....

Python Libraries to scrape data

- Top libraries for scraping.
- As long as page is not broken, both are very accurate.

BeautifulSoup AND Lxml

LXML Over beautifulsoup?

- LXML uses xpath to identify elements.
- LXML uses cssselect to identify elements.
- The power.

Beautifulsoup over lxml

- Beautiful Soup supports the HTML parser included in Python’s standard library, but it also supports a number of third-party Python parsers.
- BS4 supports LXML Parser!
- So it comes up with advanced features and interface of BeautifulSoup without most of the performance hit.

Questions?

- Ask now?

  Write it.
- Ask Later?

  Tweet to TapasweniPathak.
  Mail me tapaswenipathak@gmail.com

- Thanks!