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

- Extracting the information you want from any web page.

SCRAPE?

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

WHY SCRAPE?

- Uses LXML with XPath to find elements.
- Uses Twisted for asynchronous crawling.
- Best to crawl or spidering the web.
- What about speed?
- Scrapy crawling is fastest than mechanize because uses asynchronous operations (on top of Twisted).

WHY SCRAPY?

- 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?

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

INSTALLATIONS

```
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

- 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
  You filter your data by adding your code in this files. Your drop duplicates by adding some lines of code in this file.

AUTO GENERATED PYTHON FILES?

LET’S WRITE SOME CODE NOW?

SAVING AS JSON/CSV

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

- Recursively crawling pages.
- JS controlled elements can be scraped using Selenium with Scrapy.

WHAT ELSE WITH SCRAPY?

- Submit a PR to fill this slide. ☺

WHAT NOT SCRAPY?

- 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.

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.

HOW TO GET THE XPATH?

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

LXML OVER BEAUTIFULSOUP?

- 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.

BEAUTIFULSOUP OVER LXML

- Ask now?

  Write it.
- Ask Later?

  Tweet to TapasweniPathak.
  Mail me tapaswenipathak@gmail.com

- Thanks!

QUESTIONS?