Google Summer Of Code 2021

Project Proposal

Scrapy: Creating a MIME Sniffing Library

- Name : Akshay Sharma
- School : University Of Florida, Gainesville, United States
- Degree Program : Senior Certificate Program in Computer & Information Science & Engineering
- Time Zone : Indian Standard Time (+5.30 GMT)
- Github : akshaysharmajs

Project Information:

Title: Creating a MIME Sniffing Library

Parent organization - PYTHON SOFTWARE FOUNDATION

Sub-org name - Zyte/Scrapy

Project Abstract:

This project is about creating a python library that implements the complete MIME Sniffing Standards.
Project Issue: https://github.com/scrapy/scrapy/issues/4240

Detailed Description:

MIME stands for “Multipurpose Internet Mail Extensions.” This method is used to examine the type of content in a web request or response to determining the content’s format.

```
HTTP/1.1 200 OK
Date: Wed, 23 Jan 2019 17:37:54 GMT
Expires: -1
Cache-Control: private, max-age=0
Content-Type: text/html; charset=UTF-8
Strict-Transport-Security: max-age=31536000
Server: gws
X-XSS-Protection: 1; mode=block
X-Frame-Options: SAMEORIGIN
Set-Cookie: 1P_JAR=2019-01-23-17; expires=Fri, 22-Feb-2019
17:37:54 GMT; path=/; domain=.google.com
Alt-Svc: quic=":443"; ma=2592000; v="44,43,39"
Connection: close
Content-Length: 246589

<!doctype html><html itemscope=""
itemtype="http://schema.org/WebPage" lang="en"><head><meta
charset="UTF-8"><meta content="origin" name="referrer"><meta
```

Content-Type response header for an HTML page from google.com

Usually, we can determine a resource’s MIME type by looking at the **Content-Type** response header in an HTTP response but some developers set values for *Content-Type* headers that are not suitable for the response’s content. This is where MIME Sniffing comes into play to determine the actual MIME type by analyzing the response’s content.

At present, Scrapy uses the basic inbuilt python library ‘mimetypes’ implemented inside the file ‘responsetypes.py’ to determine the MIME type of file and handle it accordingly. However, MIME types currently being recognized are very limited and some of the responses are misinterpreted. For instance, the above-mentioned issue describes that Scrapy recognizing a pdf file as a TextResponse whereas it should be an application/pdf.

Misinterpreting MIME types can cause bugs while using Scrapy frameworks in projects. Therefore, we should implement the python library that can handle all the MIME sniffing standards.

This library will follow the standards described in https://mimesniff.spec.whatwg.org/. We can implement the library from scratch or on top of some of the libraries that support MIME sniffing like 'python-magic', 'mimesniff', 'sniffpy' but in the end, we should have the best possible implementation that works the same way as any browser uses the sniffing algorithm.

Deliverables and Implementation:

The implementation language will be Python and all codes will accompany detailed documentation and testing.

By the end of this project, I propose to deliver the following:

1. Following MIME sniffing standards

- Standards will be followed to imitate the behavior of web-browsers' sniffing algorithms
- The library will follow the methods described in section 5 to parse and handle the MIME types are described in section 4.
- The library will also follow the pattern matching just as mentioned in section 6 to estimate the response content.
- The library will allow computing any unknown MIME type which is crucial to implement as Scrapy can encounter many unknown types. Different methods to identify the unknown type are described in section 7.
- The library will implement context-specific sniffing algorithms in a particular context method of which are described in section 8.

2. Reusing existing libraries.

- The reason to choose an external library is the ease of debugging and extensibility
- Most of the MIME types can be identified using the python-magic library. python -magic will mainly be used for the **content body sniffing**. These can be easily implemented using the grammar proposed by *python-magic*, otherwise, it will be hard-pressed to reinvent a similarly extensible parser in our library.
- We can also use *python-magic* for the cases not supported by standards.
- We can allow users to use a more customizable framework rather than a rigid custom parser like mimesniff by wrapping the *python-magic* for Scrapy.

3. Sniffing under ‘X-Content-Type-Options=nosniff’

- The X-Content-Type-Options is an HTTP header that is used by servers to instruct browsers not to perform MIME sniffing. With this header, all the algorithms required to compute the MIME type described in the standard will be aborted and the MIME type given by the server will be the computed MIME type.
- Still, we can try to estimate the MIME type through the information mentioned in headers or by analyzing the file endings. This needs more discussion with mentors.

4. Performance issues

- The new implementation of the sniffing library can affect the performance of Scrapy making it a little bit slower. So to deal with this, we can allow a setting like “SET_SNIFFING” which can allow users to enable or disable the use of the sniffing library while scraping as per the requirement.

5. If time permits, integrating the sniffing library with Scrapy.

Code Contribution:

Issue	PR	Description	Status
#4332	#4338	Small Documentation Fix	Merged
#4393	#4403	enable ANSI color (instead of ANSI color codes) in the Windows terminal	Merged
#4643	#4646	allowing to run .pyw files	Merged
#4715	#4736	Windows pip installation guide	Merged
#4946	#4953	adding variable type values support to 'scrapy.FormRequest()'	Changes Approved
#4892		Update all links in the installation guide	Solved and closed

Timeline :

Before Community Bonding (April 14, 2021 - May 17, 2021)

- Contribute more pull requests to Scrapy and Keep in touch with mentors.
- Read more about MIME Sniffing and analyze existing implementations

Community Bonding (May 18, 2021 - June 7, 2021)

- Actively participate in all discussions related to the issue
- Setup development environment
- Contribute to more bug fixes/patches
- Analyze the existing implementation of sniffing algorithm by popular open-source web browsers like Mozilla
- Setup blog for GSOC 2021

Week 1 (June 8, 2021 - June 14, 2021): Implementation of parsing and handling MIME standards mentioned in sections 4 and 5.

Week 2 (June 15, 2021 - June 21, 2021): Getting code reviewed for phase I evaluation. Perform fixes after getting code reviewed.

Week 3 (June 22, 2021 - June 28, 2021): Complete any leftover implementation of sections 4 and 5. Writing relevant documentation and full proof testing of the implemented code.

Week 4 (June 29, 2021 - July 5, 2021): Implement section 6 using python-magic

Week 5 (July 6, 2021 - July 12, 2021): Implement section 7 using python-magic

Week 6 (July 13, 2021 - July 19, 2021): Testing and documenting the implemented code

Week 7 (July 20, 2021 - July 26, 2021): Implementing context-specific sniffing.

Week 8 (July 27, 2021 - August 2, 2021): Discussion with mentors for the cases not supported by standards and implementing it.

Week 9 (August 2, 2021 - August 8, 2021): If time permits, integrating the MIME library into the Scrapy framework

Week 10 (August 9, 2021 - August 15, 2021): Complete any leftover tests or documentation

Final Week (August 16, 2021 - August 23, 2021): Get code reviewed and perform fixes. Wrap up, complete leftover work, submit the final report and code.

Benefits to Scrapy Community:

- Using MIME sniffing with complete implementation of standards will give Scrapy framework an edge over other popular web-scraping frameworks.
- Scrapy can allow its users a more diverse and easier analysis of the content they are scraping.
- Many issues currently on the list can be resolved

About me:

I am an undergrad student pursuing a certificate program in Computer & Information Science & Engineering at the University Of Florida, Gainesville, United States. I have four years of programming experience, particularly in Python, JavaScript and C++.

I have studied courses on Algorithms and Data Structures, Computer Networks, Artificial Intelligence, Information Retrieval, Web Semantics, and Development during my freshmen, sophomore, and junior years at Jaypee Institute of Information Technology, Noida, India. Currently, in my senior year at the University of Florida, I am studying courses on Analysis Of Algorithms, Computer Network Fundamentals, Intro to Data Science, and Operating Systems.

I have also done a variety of projects in areas like web development, machine learning, etc. Some of my major projects are:

- Unix based Korn shell using flex and bison
- Fake news detection using Scrapy and machine learning
- Price Monitoring Tool using Scrapy
- Subjective Answer Evaluation using Machine learning

Outside of academics, I've been participating in hackathons and love to hone my coding skills on platforms like Leetcode, Codechef, HackerRank.

Why me ?:

I have been using Scrapy for scraping web data for the past few years as it's easier to use and has a better performance compared to other frameworks like Selenium, Pyspider. I have used Scrapy to scrape a large news dataset and for scraping prices of products from e-commerce websites. Over the last year, the Scrapy community has helped me a lot to grow as a passionate open-source contributor.

I believe I aptly suit this project as I have the required skills needed to be successful in this project. Also, I have been contributing to Scrapy for a while now which has made me quite familiar with the workflow of Scrapy's codebase and architecture, so I won't need to spend any more time to familiarize myself with the codebase, which would be beneficial as we have a shorter timeline this year.

Furthermore, this project would be an amazing opportunity for me to gain an experience of working on a high-impact real world project improving and learning new skills through continuous code-reviews from such highly experienced and passionate developers in the community.

Availability & Other Commitments:

- I would be able to devote approx 35-40 hours per week to GSOC and during this period, I intend to stay in touch with mentors keeping them up to date with my daily progress. If required, I am also willing to put in a few more hours in order to meet the deadlines.
- I don't have any other commitments and I have only applied to Scrapy in GSOC 2021.