Midterm Project Report

Advanced Computer Programming

Student Name : Abiansyah Gymnastiar
Student ID : 112021201
Teacher : DINH-TRUNG VU

2024-04

Chapter 1 Introduction

1.1 Github

1) **Personal Github Account**: https://github.com/abi-gymnastiar
2) **Group Project Repository**: https://github.com/HanifiSetiawan/ACP-Group-06

1.2 Overview

This project utilizes Python's framework 'Scrapy' to extract repositories from a GitHub account. Some key features implemented in this project includes:

- Structured repository data storage
- URL validation and pattern matching using regular expressions
- API calls with Asynchronous requests
- Exporting XML file using Scrapy's built-in exporters feature

The program successfully extracts:

- Repository URLs
- Descriptions (falling back to names if empty)
- Last updated timestamps
- Programming languages used
- Commit counts

Chapter 2 Implementation

2.1 GitHubAPISpider Class

The main spider class that handles data extraction.

2.1.1 Fields

- `GITHUB_USER`: Target GitHub username
- `ACCESS_TOKEN`: Optional API token
- 'OUTPUT_FILE': Optional outputted name file

2.1.2 Methods

- `start_requests()`: Initiates API calls
- `parse_repos()`: Processes repository listings
- `parse_languages()`: Extracts language data

2.1.3 Functions

There are no other functions within this class that has not already been explained in the Methods category.

2.2 'run_spider()' Function

This function acts as a CrawlerProcess() starter. If during the python file execution the user included username and acces token (github API) as arguments, they're both will be declared in this function to be put in the main spider class.

```
def run_spider(username=None, token=None):
    """Run the GitHub spider programmatically"""
    process = CrawlerProcess(get_project_settings())

    # Allow runtime configuration
    spider_kwargs = {}
    if username:
        spider_kwargs['GITHUB_USER'] = username
    if token:
        spider_kwargs['ACCESS_TOKEN'] = token

    process.crawl(GitHubAPISpider, **spider_kwargs)
    process.start()
```

2.3 ‘main()’

This is just a regular main function. This main function will be called whenever the python program is being executed. The program itself will accepts two arguments as parameter:

1. Username (-u) : Github username to scrape.
2. Token (-t) : Github API access token.

```
def main():
    """Main entry point for command line execution"""
    import argparse

    parser = argparse.ArgumentParser(description='Scrape GitHub repositories')
    parser.add_argument('-u', '--username', help='GitHub username to scrape')
    parser.add_argument('-t', '--token', help='GitHub access token')
    args = parser.parse_args()

    run_spider(username=args.username, token=args.token)

if __name__ == "__main__":
    main()
```

Chapter 3 Results

3.1 Result 1 (using command line)

Using the command line, in this meaning starting the process from the terminal instead of executing the python file. The command used is as below.

```
\Midterm-ACP> scrapy crawl github_api_spider
```

This command line will start the crawling process of the github_api_spider without using an access token and specifying github account to be scrape. The results are as follows.


```
<items>
  <item>
    <url>https://github.com/abi-gymnastiar/virtual_pet_tamagacha</url>
    <about>a virtual pet application made in java. Inspired by tamagotchi.</about>
    <last_updated>2023-04-01T22:14:31Z</last_updated>
    <languages>Java</languages>
    <commits>17</commits>
  </item>
  <item>
    <url>https://github.com/abi-gymnastiar/pethub2_FP</url>
    <about>Web Programming FP</about>
    <last_updated>2024-06-19T02:35:11Z</last_updated>
    <languages>PHP, Blade, CSS, JavaScript</languages>
    <commits>20</commits>
  </item>
  <item>
    <url>https://github.com/abi-gymnastiar/Maze_Game</url>
    <about>A Maze game created for the LBE GiGA game-dev assignment</about>
    <last_updated>2022-10-15T00:48:01Z</last_updated>
    <languages>ShaderLab, HLSL, C#</languages>
    <commits>7</commits>
  </item>
  <item>
    <url>https://github.com/abi-gymnastiar/Quiz-2-PAA</url>
    <about>Quiz-2-PAA</about>
    <last_updated>2022-11-23T12:11:46Z</last_updated>
    <languages>Java</languages>
    <commits>17</commits>
  </item>
  <item>
    <url>https://github.com/abi-gymnastiar/project-24424</url>
    <about>2D rogue-like platformer with procedural generation for special project course</about>
    <last_updated>2025-01-16T06:18:54Z</last_updated>
    <languages>C#, ShaderLab, HLSL, HTML</languages>
    <commits>37</commits>
  </item>
  <item>
    <url>https://github.com/abi-gymnastiar/portfolio-abi</url>
    <about>portfolio-abi</about>
    <last_updated>2025-01-30T18:02:43Z</last_updated>
    <languages>TypeScript, JavaScript, CSS, HTML</languages>
    <commits>6</commits>
  </item>
  <item>
    <url>https://github.com/abi-gymnastiar/OOPAbstractAssignment</url>
    <about>Abiansyah Adzani Gymnastiar (5025211077)</about>
    <last_updated>2022-11-02T05:46:11Z</last_updated>
    <languages>Java</languages>
    <commits>1</commits>
  </item>
  <item>
    <url>https://github.com/abi-gymnastiar/OOP-FinalProject</url>
    <about>OOP-FinalProject</about>
    <last_updated>2022-11-30T05:38:46Z</last_updated>
    <languages>None</languages>
    <commits>4</commits>
  </item>
```

3.2 Result 2 (Executing the Python File)

By executing the python file, which in this case is the ‘scrappy-crawler.py’ file, the program will start executing the main() function. The main() function itself will execute the run_spider() functin, which will start the crawling process. By executing the python file without specifying the arguments (-t for access token and –u for github userename), it will try to scrape the github repositories of ‘abi-gymnastiar’. Below are the results (XML file) of the scraping process.


```
<items>
  <item>
    <url>https://github.com/abi-gymnastiar/virtual_pet_tamagacha</url>
    <about>a virtual pet application made in java. Inspired by tamagotchi.</about>
    <last_updated>2023-04-01T22:14:31Z</last_updated>
    <languages>Java</languages>
    <commits>17</commits>
  </item>
  <item>
    <url>https://github.com/abi-gymnastiar/pethub2_FP</url>
    <about>Web Programming FP</about>
    <last_updated>2024-06-19T02:35:11Z</last_updated>
    <languages>PHP, Blade, CSS, JavaScript</languages>
    <commits>20</commits>
  </item>
  <item>
    <url>https://github.com/abi-gymnastiar/Maze_Game</url>
    <about>A Maze game created for the LBE GiGA game-dev assignment</about>
    <last_updated>2022-10-15T00:48:01Z</last_updated>
    <languages>ShaderLab, HLSL, C#</languages>
    <commits>7</commits>
  </item>
  <item>
    <url>https://github.com/abi-gymnastiar/Quiz-2-PAA</url>
    <about>Quiz-2-PAA</about>
    <last_updated>2022-11-23T12:11:46Z</last_updated>
    <languages>Java</languages>
    <commits>17</commits>
  </item>
  <item>
    <url>https://github.com/abi-gymnastiar/project-24424</url>
    <about>2D rogue-like platformer with procedural generation for special project course</about>
    <last_updated>2025-01-16T06:18:54Z</last_updated>
    <languages>C#, ShaderLab, HLSL, HTML</languages>
    <commits>37</commits>
  </item>
  <item>
    <url>https://github.com/abi-gymnastiar/portfolio-abi</url>
    <about>portfolio-abi</about>
    <last_updated>2025-01-30T18:02:43Z</last_updated>
    <languages>TypeScript, JavaScript, CSS, HTML</languages>
    <commits>6</commits>
  </item>
  <item>
    <url>https://github.com/abi-gymnastiar/OOPAbstractAssignment</url>
    <about>Abiansyah Adzani Gymnastiar (5025211077)</about>
    <last_updated>2022-11-02T05:46:11Z</last_updated>
    <languages>Java</languages>
    <commits>1</commits>
  </item>
  <item>
    <url>https://github.com/abi-gymnastiar/OOP-FinalProject</url>
    <about>OOP-FinalProject</about>
    <last_updated>2022-11-30T05:38:46Z</last_updated>
    <languages>None</languages>
    <commits>4</commits>
  </item>
```

Chapter 4 Conclusions

In conclusion, the scraping process from this program can successfully scrapped public repositories of a GitHub user. The Scrapy framework itself proved effective for GitHub API interaction. Whether by using command line in the terminal, or by executing the python file, both seem to result in generating the same output XML file.

There are some limitations however, as without using an access token, a process is limited to 60 requests/hour for unauthenticated requests. One other future improvement that can be made includes a graphical output visualization.