



亞洲大學
ASIA UNIVERSITY

Midterm Project Report

Advanced Computer Programming

Student Name : M. Fadhlan A. Harashta
Student ID : 112021222
Teacher : DINH-TRUNG VU

2024-04

Chapter 1 Introduction

1.1 Github

- 1) **Personal Github Account:** fadhlanharashta
- 2) **Group Project Repository:** <https://github.com/fadhlanharashta/ACP---Group-3>

1.2 Overview

In my mid term project, I develop a web scraper using Scrapy and Python web crawling framework, I use several useful libraries which includes:

- CrawlSpider
- Regular Expression
- CSS and XPath Selectors
- Conditional Logic and String Manipulation

In this project I use a few libraries, Scrapy which is the main scrapy module, LinkExtractor which helps extract links that match a pattern, and CrawlSpider to let the code to follow links based on rules.

While some part of the code works as intended, such as the code ability to capture the repositories link, title, and number of commit and language, it is somehow unable to get the last updated.

The captured data then put on an XML file and also to make it easier to check, I also make a JSON file.

Chapter 2 Implementation

2.1 Class 1: GithubSpider

This class inherits from CrawlSpider, a specialized spider in Scrapy used for crawling websites using Rules. It is designed to scrape Github repository data from a specific user.

2.1.1 Fields

Field	Description
name	Unique name to identify spider: "github_spider"
allowed_domains	A list that limits the spider crawling to github: "github.com"
Start_urls	Starting URL for the spider's to begin crawling "https://github.com/fadhlanharashta?tab=repositories"
rules	Some rules that tell the spider how to follow links and which callback to use. In this case, it follows repository links and calls "parse_repo"

2.1.2 Methods

Parse Repo

The main callback that extracts data from each repository page. It handles both empty and not empty repositories and collects all the needed data such as link, name, number of commit, language, and last update.

```

42 def parse_repo(self, response):
43     url = response.url
44     repo_name = url.rstrip('/').split('/')[-1]
45
46     # About section
47     about = response.css('p.f4.my-3::text, div.BorderGrid p.f4::text').get()
48     about = about.strip() if about else None
49
50     # Check for empty repo
51     is_empty = response.css('div.Box.mt-3 h3::text').re_first(r'This repository is (.*)')
52
53     if is_empty:
54         yield {
55             "url": url,
56             "about": about if about else repo_name,
57             "last_updated": None,
58             "languages": None,
59             "number_of_commits": None,
60         }
61     else:
62         last_updated = response.css('div[data-testid="latest-commit-details"] relative-time::attr(datetime)').get()
63         if not last_updated:
64             last_updated = response.xpath('//relative-time/@datetime').get()
65
66         # Language
67         languages = response.css('ul.list-style-none .d-inline .color-fg-default::text').getall()
68         if not languages:
69             languages = response.css('.language-color + span::text').getall()
70         languages = [lang.strip() for lang in languages if lang.strip()]
71         languages_str = ", ".join(languages) if languages else None
72         # Commit
73         commit_text = response.css('div[data-component="text"] span.fgcolor-default::text').re_first(r'(\d+)\s+commits?')
74         number_of_commits = int(commit_text) if commit_text else None
75
76
77     yield {
78         "url": url,
79         "about": about if about else repo_name,
80         "last_updated": last_updated,
81         "languages": languages_str,
82         "number_of_commits": number_of_commits,
83     }
84
85

```

2.1.3 Functions

- **Extract Repositories**

Extract current repository and then parse the repositories name from the URL string. The spider only crawls github.com and starts from repositories tab. It has a rule: the program will extract links for githubname/reponame but no more slashes. Since it only follow links to individual repositories from the profile's repositories page, it only take from anchor tags (<a>) that match the given CSS selector. And then each link call the parse repo method. Lastly it wont follow links from the individual repo page.

```

7 class GithubSpider(CrawlSpider):
8     name = "github_spider"
9     allowed_domains = ["github.com"]
10    start_urls = ["https://github.com/fadhlanharashta?tab=repositories"]
11
12    #rules
13    rules = (
14        Rule(
15            LinkExtractor(
16                allow=r'/fadhlanharashta/[^/]+$ ',
17                restrict_css='a[itemprop="name codeRepository"]'
18            ),
19            callback='parse_repo',
20            follow=False
21        ),
22    )
23
24
25 def parse_repo(self, response):
26     url = response.url
27     repo_name = url.rstrip('/').split('/')[-1]
28

```

- **About**

Uses CSS selector to get the repository description if available. It extract the repository description.

```
29         # About section
30         about = response.css('p.f4.my-3::text, div.BorderGrid p.f4::text').get()
31         about = about.strip() if about else None
```

- **Empty Repo check**

Uses regex to check if the repository is empty or not. If repo is empty yield as follows:

```
33         # Check for empty repo
34         is_empty = response.css('div.Box.mt-3 h3::text').re_first(r'This repository is (.*)')
35
36         if is_empty:
37             yield {
38                 "url": url,
39                 "about": about if about else repo_name,
40                 "last_updated": None,
41                 "languages": None,
42                 "number_of_commits": None,
43             }
44         else:
```

- **Last Update**

Scrapes the <relative-time> tag to get the latest update timestamp.

```
45         last_updated = response.css('div[data-testid="latest-commit-details"] relative-time::attr(datetime)').get()
46         if not last_updated:
47             last_updated = response.xpath('//relative-time/@datetime').get()
```

- **Language**

Extract programing languages used in the repository using css selector.

```
49         #Language
50         languages = response.css('ul.list-style-none .d-inline .color-fg-default::text').getall()
51         if not languages:
52             languages = response.css('.language-color + span::text').getall()
53         languages = [lang.strip() for lang in languages if lang.strip()]
54         languages_str = ", ".join(languages) if languages else None
```

- **Number of Commit**

Uses XPath and regex to find and parse the number of commits.

```
92         # Commit
93         commit_text = response.css('a[href*="commits"] span::text').get() \
94             or response.css('strong[data-test-id="commits"]::text').get()
```

- **Yield**

Each yield returns a dictionary for each repository. The data of which we want to crawl

```
61         yield {
62             "url": url,
63             "about": about if about else repo_name,
64             "last_updated": last_updated,
65             "languages": languages_str,
66             "number_of_commits": number_of_commits,
67         }
```

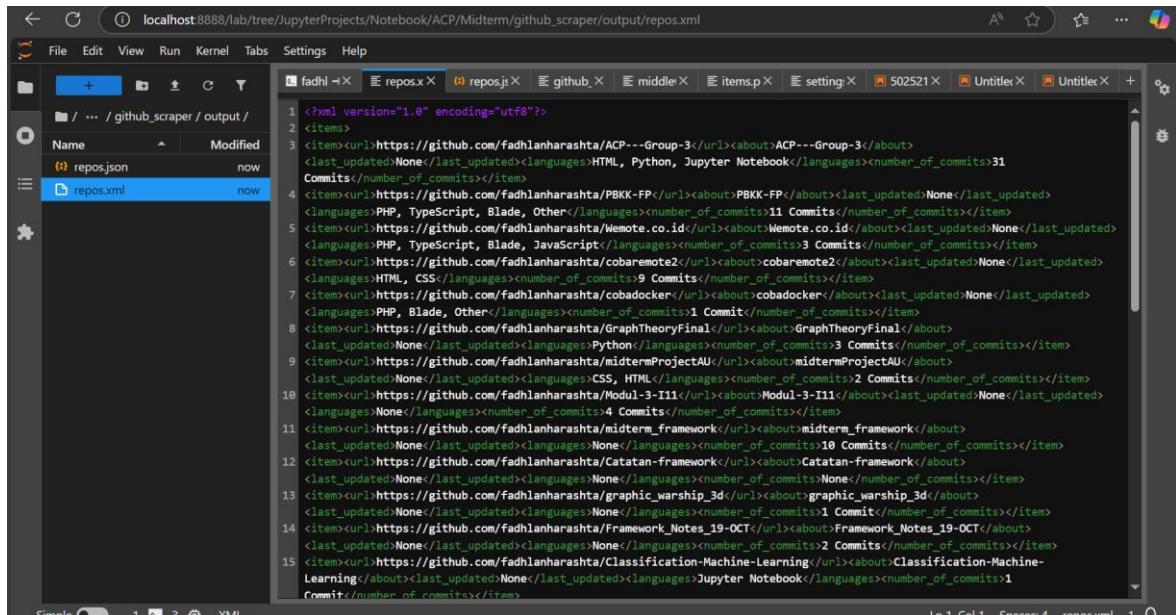
2.2 Class 2: Setting

```
10 BOT_NAME = "github_scraper"
11
12 SPIDER_MODULES = ["github_scraper.spiders"]
13 NEWSPIDER_MODULE = "github_scraper.spiders"
14
15 FEEDS = {
16     "output/repos.xml": {
17         "format": "xml",
18         "encoding": "utf8",
19         "overwrite": True,
20     }
21 }
22 # Crawl responsibly by identifying yourself (and your website) on the user-agent
23 #USER_AGENT = "github_scraper (+http://www.yourdomain.com)"
24
25 # Obey robots.txt rules
26 USER_AGENT = "Mozilla/5.0"
27 ROBOTSTXT_OBEY = False
```

- **BOT NAME**
Identifies the scrapy bot, used internally.
- **SPIDER_MODULES**
Module path for spider definition.
- **NEWSPIDER_MODULE**
Default path for spider created via command line
- **FEEDS**
Specifies export format and file path XML file exported to output/repos.xml
- **USER_AGENT**
Overrides default user-agent string to mimic a real browser.
- **ROBOTSTXT_OBEY**
Sets to false to ignore robot.txt rules and allow full crawling
- **TWISTED_REACTOR**
Specifies the event loop reactor for asynchronous processing.
- **FEED_EXPORT_ENCODING**
Sets UTF-8 Encoding for exported file.

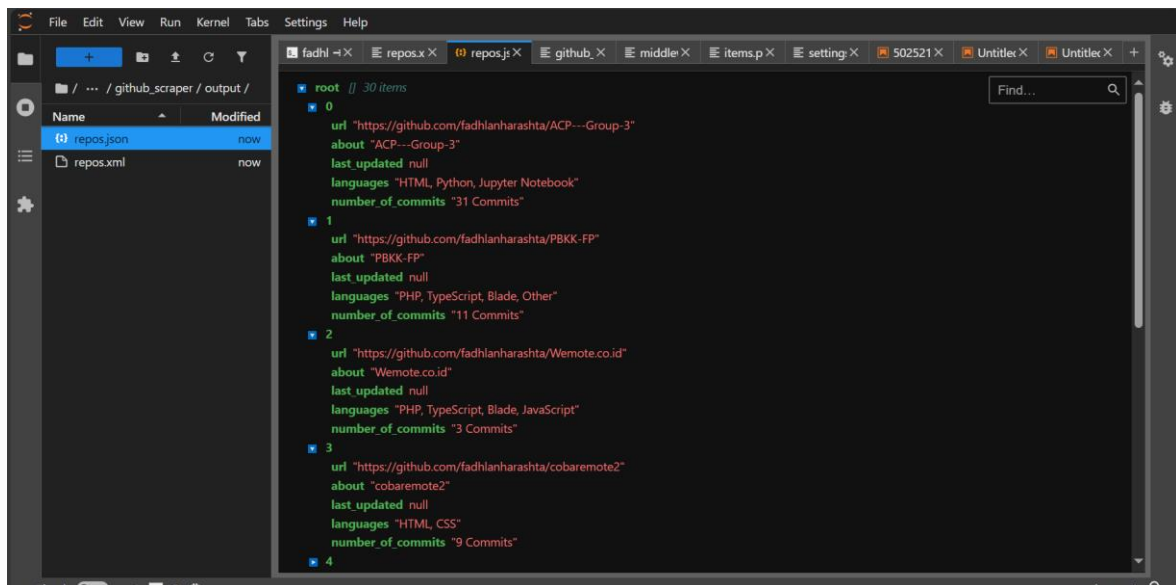
Chapter 3 Results

3.1 Result 1: XML File



```
<?xml version="1.0" encoding="utf8"?>
<items>
  <item><url>https://github.com/fadhlanharashta/ACP---Group-3</url><about>ACP---Group-3</about>
  <last_updated>None</last_updated><languages>HTML, Python, Jupyter Notebook</languages><number_of_commits>31
  Commits</number_of_commits></item>
  <item><url>https://github.com/fadhlanharashta/PBKK-FP</url><about>PBKK-FP</about><last_updated>None</last_updated>
  <languages>PHP, TypeScript, Blade, Other</languages><number_of_commits>11 Commits</number_of_commits></item>
  <item><url>https://github.com/fadhlanharashta/Wemote.co.id</url><about>Wemote.co.id</about><last_updated>None</last_updated>
  <languages>PHP, TypeScript, Blade, JavaScript</languages><number_of_commits>3 Commits</number_of_commits></item>
  <item><url>https://github.com/fadhlanharashta/cobaremove2</url><about>cobaremove2</about><last_updated>None</last_updated>
  <languages>HTML, CSS</languages><number_of_commits>9 Commits</number_of_commits></item>
  <item><url>https://github.com/fadhlanharashta/cobadocker</url><about>cobadocker</about><last_updated>None</last_updated>
  <languages>PHP, Blade, Other</languages><number_of_commits>1 Commit</number_of_commits></item>
  <item><url>https://github.com/fadhlanharashta/GraphTheoryFinal</url><about>GraphTheoryFinal</about>
  <last_updated>None</last_updated><languages>Python</languages><number_of_commits>3 Commits</number_of_commits></item>
  <item><url>https://github.com/fadhlanharashta/midtermProjectAU</url><about>midtermProjectAU</about>
  <last_updated>None</last_updated><languages>CSS, HTML</languages><number_of_commits>2 Commits</number_of_commits></item>
  <item><url>https://github.com/fadhlanharashta/Modul-3-111</url><about>Modul-3-111</about><last_updated>None</last_updated>
  <languages>None</languages><number_of_commits>4 Commits</number_of_commits></item>
  <item><url>https://github.com/fadhlanharashta/midterm_framework</url><about>midterm_framework</about>
  <last_updated>None</last_updated><languages>None</languages><number_of_commits>10 Commits</number_of_commits></item>
  <item><url>https://github.com/fadhlanharashta/Catatan-framework</url><about>Catatan-framework</about>
  <last_updated>None</last_updated><languages>None</languages><number_of_commits>None</number_of_commits></item>
  <item><url>https://github.com/fadhlanharashta/graphic_warship_3d</url><about>graphic_warship_3d</about>
  <last_updated>None</last_updated><languages>None</languages><number_of_commits>1 Commit</number_of_commits></item>
  <item><url>https://github.com/fadhlanharashta/Framework_Notes_19-OCT</url><about>Framework_Notes_19-OCT</about>
  <last_updated>None</last_updated><languages>None</languages><number_of_commits>2 Commits</number_of_commits></item>
  <item><url>https://github.com/fadhlanharashta/Classification-Machine-Learning</url><about>Classification-Machine-
  Learning</about><last_updated>None</last_updated><languages>Jupyter Notebook</languages><number_of_commits>1
  Commit</number_of_commits></item>
</items>
```

3.2 Result 2: Json File



```
{
  "url": "https://github.com/fadhlanharashta/ACP---Group-3",
  "about": "ACP---Group-3",
  "last_updated": null,
  "languages": "HTML, Python, Jupyter Notebook",
  "number_of_commits": "31 Commits"
},
{
  "url": "https://github.com/fadhlanharashta/PBKK-FP",
  "about": "PBKK-FP",
  "last_updated": null,
  "languages": "PHP, TypeScript, Blade, Other",
  "number_of_commits": "11 Commits"
},
{
  "url": "https://github.com/fadhlanharashta/Wemote.co.id",
  "about": "Wemote.co.id",
  "last_updated": null,
  "languages": "PHP, TypeScript, Blade, JavaScript",
  "number_of_commits": "3 Commits"
},
{
  "url": "https://github.com/fadhlanharashta/cobaremove2",
  "about": "cobaremove2",
  "last_updated": null,
  "languages": "HTML, CSS",
  "number_of_commits": "9 Commits"
},
{
  "url": "https://github.com/fadhlanharashta/cobadocker",
  "about": "cobadocker",
  "last_updated": null,
  "languages": "PHP, Blade, Other",
  "number_of_commits": "1 Commit"
},
{
  "url": "https://github.com/fadhlanharashta/GraphTheoryFinal",
  "about": "GraphTheoryFinal",
  "last_updated": null,
  "languages": "Python",
  "number_of_commits": "3 Commits"
},
{
  "url": "https://github.com/fadhlanharashta/midtermProjectAU",
  "about": "midtermProjectAU",
  "last_updated": null,
  "languages": "CSS, HTML",
  "number_of_commits": "2 Commits"
},
{
  "url": "https://github.com/fadhlanharashta/Modul-3-111",
  "about": "Modul-3-111",
  "last_updated": null,
  "languages": null,
  "number_of_commits": "4 Commits"
},
{
  "url": "https://github.com/fadhlanharashta/midterm_framework",
  "about": "midterm_framework",
  "last_updated": null,
  "languages": null,
  "number_of_commits": "10 Commits"
},
{
  "url": "https://github.com/fadhlanharashta/Catatan-framework",
  "about": "Catatan-framework",
  "last_updated": null,
  "languages": null,
  "number_of_commits": null
},
{
  "url": "https://github.com/fadhlanharashta/graphic_warship_3d",
  "about": "graphic_warship_3d",
  "last_updated": null,
  "languages": null,
  "number_of_commits": "1 Commit"
},
{
  "url": "https://github.com/fadhlanharashta/Framework_Notes_19-OCT",
  "about": "Framework_Notes_19-OCT",
  "last_updated": null,
  "languages": null,
  "number_of_commits": "2 Commits"
},
{
  "url": "https://github.com/fadhlanharashta/Classification-Machine-Learning",
  "about": "Classification-Machine-Learning",
  "last_updated": null,
  "languages": "Jupyter Notebook",
  "number_of_commits": "1 Commit"
}
```

Chapter 4 Conclusions

While I successfully crawl some data inside my repositories, there are some data that I still unable to get. Data like last updated still unable to be crawled. The problem is that my code is unable to recognize the last updated date from the HTML. While I have try to inspect the HTML manually to find the number of commit and the last update detail, I still unable to get the data and put it into the XML file. As a result, the data im able to put on the XML file which is not none is links, about, number of commit, and languages, while the last update and number of commit remain none.