



(/)

0x03. Log Parsing

Algorithm My Playlist Planning (me)



Weight: 1

Projects(/projects/current)



Project over - took place from Jul 15, 2024 6:00 AM to Jul 19, 2024 6:00 AM



An auto QA review will automatically take corrections(/to_review)



Evaluation quizzes(/dashboards/my_current_evaluation_quizzes)

In a nutshell...

- **Auto QA review:** 5.5/11 mandatory

- **Altogether:** 50.0%



Curriculums(/dashboards/my_curriculums)

◦ Mandatory: 50.0%

◦ Optional: no optional tasks



Concepts(/concepts)



For the "0x03. Log Parsing" project, you will need to apply your knowledge of Python programming, focusing on parsing and processing data streams in real-time. This project involves reading from standard input (stdin), handling data in a specific format, and performing calculations based on the input data. Here are some concepts and resources that you might find useful:



Servers(/servers)

>_ Concepts Needed:

Sandboxes(/user_containers/current)

1. File I/O in Python:



- Understand how to read from `sys.stdin` line by line.
- Python Input and Output (/rltoken/f7U2MDsBT_rd9AfUUaqVnQ)

2. Signal Handling in Python:



- Video on demand(/dashboards/videos)
- Handling keyboard interruption (CTRL + C) using signal handling in Python.
- Python Signal Handling (/rltoken/1nDqPJe80rSD-NMulzJBw)

3. Data Processing:



- Parsing strings to extract specific data points.
- Aggregating data to compute summaries.

4. Regular Expressions:



- Discord(https://discord.com/app)
- Using regular expressions to validate the format of each line.
- Python Regular Expressions (/rltoken/ZsD-YLisfaHFeMT_sZxX1Q)

5. Dictionaries in Python:

- Using dictionaries to count occurrences of status codes and accumulate file sizes.
- Python Dictionaries (/rltoken/JM-RpavKkb8yanxWEnNYJw)

6. Exception Handling

My Profile(/users/my_profile)



- Handling possible exceptions that may arise during file reading and data processing.
- Python Exceptions (/rltoken/OA2PlryYA2gyCCKIsdgUw)



By studying these concepts and utilizing the resources provided, you will be well-prepared to tackle the log parsing project, effectively handling data streams, parsing log entries, and computing metrics based on the processed data.



Home(/)

Additional Resources



My Planning(/planning/me)

- Mock Technical Interview (/rltoken/VIOaXKkbecRYdnTLaLU1lg)



Projects(/projects/current)

Requirements

General

QA Reviews I can make(/corrections/to_review)

- Allowed editors: `vi`, `vim`, `emacs`
- All your files will be tested on Ubuntu 20.04 LTS using `python3` (version 3.4.3)
- All your files should end with a new line
- The first line of all your files should be exactly `#!/usr/bin/python3`
- A `README.md` file, at the root of the folder of the project, is mandatory
- Your code should be able to run on 1.7.x)
- All your files must be executable
- The length of your files will be tested using `wc`



Quizzes(/dashboards/my_style/vers)



Concepts(/concepts)



Conference rooms(/dashboards/video_rooms)

Tasks



Servers(/servers)

0. Log parsing

mandatory



Sandboxes(/user_containers/current)

Score: 50.0% (Checks completed: 100.0%)



Tools(/dashboards/my_tools)

Write a script that reads stdin line by line and computes metrics:

- Input format: `<IP Address> - [<date>] "GET /projects/260 HTTP/1.1" <status code>`
- Videos on the dashboard (if the dashboard is on, the line must be skipped)
- After every 10 lines and/or a keyboard interruption (`CTRL + C`), print these statistics from the beginning.



Video on the dashboard (if the dashboard is on, the line must be skipped)

- After every 10 lines and/or a keyboard interruption (`CTRL + C`), print these statistics from the beginning.



Peers (/users/peer)

- Total file size: File size: `<total size>`
- `<peer size>` is the sum of all previous `<file size>` (see input format above)

- Number of lines by status code:



Discord (https://discord.com/app).

- possible status code: 200 , 301 , 400 , 401 , 403 , 404 , 405 and 500
- If a status code doesn't appear or is not an integer, don't print anything for this status code
- format: `<status code>: <number>`
- status codes should be printed in ascending order



Warning: In this sample, you will have random value - it's normal to not have the same output as this one.

My Profile(/users/my_profile)

```

alex@ubuntu:~/0x03-log_parsing$ cat 0-generator.py
#!/usr/bin/python3
import random
import sys
from time import sleep
import datetime

for i in range(10000):
    sleep(1)
    sys.stdout.write("{:d}.{:d}.{:d}.{:d} - [{}]\n"GET /projects/260 HTTP/1.1\"
    {} {}".format(
        Projects/projects/current, random.randint(1, 255), random.randint(1, 255), r
        andom.randint(1, 255),
        datetime.datetime.now(),
        QA_Reviews/qa_reviews/corrections[200, 300, 400, 401, 403, 404, 405, 500]],
        random.randint(1, 1024)
    ))
    sys.stdout.flush()
Evaluation quizzes(/dashboards/my_current_evaluation_quizzes)

```


```



alex@ubuntu:~/0x03-log_parsing$ ./0-generator.py | ./0-stats.py
File size: 5213
200: 2
401: 1 Curriculumums(/dashboards/my_curriculumums)
403: 2
404: 1
405: 1 Concepts(/concepts)
500: 3
File size: 11320
200: 3 Conference rooms(/dashboards/video_rooms)
301: 2
400: 1
401: 2 Servers(/servers)
403: 3
404: 4
405: 2 Sandboxes(/user_containers/current)
500: 3
File size: 16305
200: 3 Tools(/dashboards/my_tools)
301: 3
400: 4
401: 2 Video on demand(/dashboards/videos)
403: 5
404: 5
405: 4
500: 4
Peers(/users/peers)
File size: 17146
200: 4
301: 3
400: 4 Discord(https://discord.com/app)
401: 2
403: 6
404: 6
405: 4
500: 4
My Profile(/users/my_profile):
Traceback (most recent call last):

```

```
File "./0-stats.py", line 15, in <module>
Traceback (most recent call last):
File "./0-generator.py", line 8, in <module>
() for line in sys.stdin:
```

```
KeyboardInterrupt
sleep(random.random())
KeyboardInterrupt
alexa@ubuntu:~/0x03-log_parsing$
```


 My Planning(/planning/me)

- Repo:**
-  Projects(/projects/current)
 - GitHub repository: alx-interview
 - Directory: 0x03-log_parsing
 - ✓  • File QA-Reviews.py can make(/corrections/to_review)




 Curriculums(/dashboards/my_curriculums)

 Concepts(/concepts)


 Conference rooms(/dashboards/video_rooms)

 Servers(/servers)

 Sandboxes(/user_containers/current)

 Tools(/dashboards/my_tools)

 Video on demand(/dashboards/videos)

 Peers(/users/peers)

 Discord(<https://discord.com/app>)



My Profile(/users/my_profile)

Copyright © 2024 ALX, All rights reserved.