# Обработка лог файла

## Что нужно сделать?

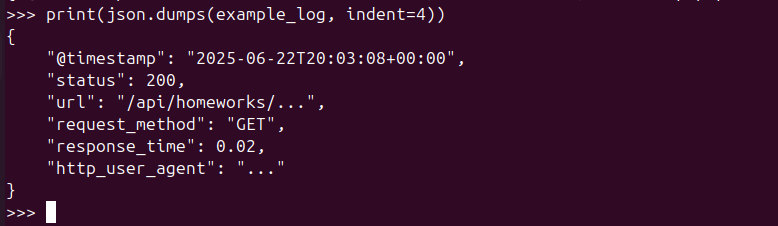
**Нужно написать скрипт для обработки лог-файла.**

Скрипт читает файл и формирует отчет со списком эндпоинтов, количеством запросов по каждому эндпоинту и средним временем ответа. Пути к файлам и название отчета передаются как параметры, например python [main.py](http://main.py) --file file.log --report average.

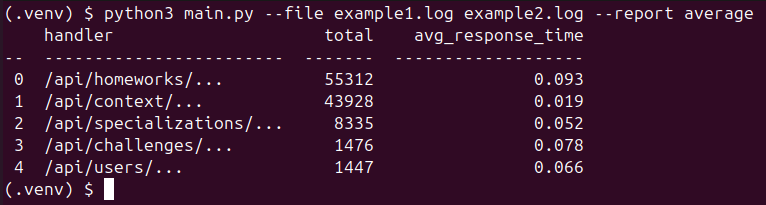
Админ о нас позаботился, поэтому логи пишутся в JSON, они легко читаются и парсятся, большая часть информации которая не пригодится при выполнении задания так же убрана.

Чтобы сфокусироваться на функционале формирования отчёта и не отвлекаться на рутинные задачи (обработку аргументов скрипта и форматирование вывода), можно использовать стандартную библиотеку argparse, а для красивого отображения в консоли — библиотеку tabulate.

##### Примеры лог-файлов можно посмотреть [здесь](https://drive.google.com/drive/folders/1b1GmaUndl52nidgOEEbDAQotpFlJBAsq?usp=sharing). Распарсенный вариант выглядит так:



##### Пример запуска скрипта:



## Какие функциональные требования?

* можно передать путь к файлу, файлов может быть несколько
* можно указать название отчета, нужен отчет average
* в консоль выводится отчёт в виде таблице

## Какие не функциональные требования?

* для всего кроме тестов и красивого вывода в консоль, можно использовать только стандартную библиотеку, например:
  + для обработки параметров скрипта нельзя использовать click, но можно использовать argparse из стандартной библиотеки
* код покрыт тестами написанных на pytest
* для тестов можно использовать любые дополнительные библиотеки
* код соответствует:
  + общепринятым стандартам написания проектов на python
  + общепринятому стилю

## Как сдавать задание?

* присылайте ссылку на git репозиторий, ссылки на google drive или yandex не подходят
* присылайте примеры запуска скрипта, например:
  + можно сделать скриншот запуска скрипта и добавить его репозиторий, для примера работы можно использовать [эти](https://drive.google.com/drive/folders/1b1GmaUndl52nidgOEEbDAQotpFlJBAsq?usp=sharing) файлы. За приложенные примеры запуска ревьюер скажет вам спасибо и добавит баллы.
* перед отправкой ссылки на репозиторий проверьте, пожалуйста, что репозиторий публичный и его можно посмотреть

## Как получить дополнительные баллы за тестовое?

Это дополнительные требования, их не обязательно реализовывать, но они помогут проявить себя. Если эти требования будут реализованы, хотя бы частично - это даст дополнительные баллы:

* в архитектуру заложена возможность быстрого добавления новых отчётов. Например, если захотим проанализировать User-Agent пользователей и понять, какими браузерами они пользуются, то это можно будет сделать, не переписывая половину скрипта
* при формировании отчета есть возможность указать дату, чтобы сформировать отчет только по записям с указанной датой, например --date 2025-22-06.

## FAQ

* Можно ли использовать нейросети?
  + Рекомендуем не использовать. Сталкиваемся со случаями, когда кандидаты увлекаются нейросетями, чтобы сделать тестовое, а потом не проходят техническое интервью, потому что не понимают, почему нейросеть написала тот или иной код.
* Будет ли приниматься задание без тестов?
  + Нет, приниматься не будет. Наличие тестов входит в основные требования.
* Код покрыт тестами - это какой процент покрытия?
  + Можно ориентироваться на 80% покрытия по [pytest-cov](https://pypi.org/project/pytest-cov/), можно больше, можно меньше, но главное чтобы был протестирован критически важный функционал.
* Можно ли использовать какие-то дополнительные библиотеки к pytest?
  + Да, всё что помогает вам тестировать код можно использовать.
* Можно ли пользоваться линтерами или форматтерами кода?
  + Да, можно использовать любой линтер или форматтер - это хорошая практика.
* Можно ли менять API, название и формат параметров скрипта?
  + Нет, лучше не менять, используем file, report и date.
* Нужно ли учитывать случаи, когда пользователь при запуске скрипта ввёл что-то не то?
  + Да, стоит учитывать.
* Нужно ли писать комментарии в коде?
  + Если вы считаете что они нужны, то пишите.
* Нужно ли писать readme.md?
  + Писать не обязательно, но в него можно положить примеры запуска скрипта и написать то, что на ваш взгляд важно знать ревьюеру. Если будете писать readme, то пусть он будет небольшим, но по делу, чем большим и сгенерированным нейросетью.