

Общее описание

Реализовать три сервиса:

- Клиентский
- Web
- Фоновый

Все сервисы запаковать в Docker, docker-compose. Код можно выложить на любой открытый репозиторий [GitHub, BibBucket etc].

Асинхронный Web API сервис

Сервис должен принимать на вход строку вида `'{IP адрес} {HTTP method} {URI} {HTTP status code}'`, парсить ее и сохранять в базу PostgreSQL, пометчая каждую строку уникальным идентификатором и временем сохранения. А также, возвращать результат из базы по запросу.

Stack: FastAPI, Pydantic, PostgreSQL

Спецификация endpoint'ов:

POST /api/data:

Request:

```
{
  "log": str
}
```

Response:

status code	Описание
201	Лог сохранен
418	Что-то пошло не так

GET /api/data:

Request параметры придумать удобные под сервис, имитирующий фоновую обработку.

Response:

```
[
  {
    "id": uuid4,
```

```
    "created": datetime,
    "log": {
        "ip": IPv4,
        "method": str,
        "uri": URI,
        "status_code": int
    },
}
```

Сервис, имитирующий клиент:

Сервис должен генерировать текст вида: '{IP адрес} {HTTP method} {URI} {HTTP status code}' и отправлять его Web API сервису POST запросами.

Сервис должен работать в N потоков|процессов с рандомной задержкой до M мс, между запросами. N и M - задаются переменными окружения, с осмысленными названиями.

Сгенерированный текст логировать в файл.

Сервис, имитирующий фоновую обработку:

Сервис должен периодически получать GET запросами данные с Web API сервиса и сохранять их в примонтированный файл.

Реализация сервиса должна учитывать, что сервис может быть запущен в несколько instance'ов с общим разделяемым ресурсом - монтированным файлом.