

Задание

Для выполнения задания вам понадобится [mercury](#).

Для выполнения задания нужно использовать хранилище S3 minio, которое разворачивается в mercury.

Дополните файл **clients/fapi/s3.py** и реализуйте функции:

1. upload_file — загрузить файл в хранилище S3 по указанному пути

Функция принимает:

bucket, path — путь до объекта в S3

buffer — массив байт

Функция ничего не должна возвращать.

Пример вызова:

```
client = S3Client(endpoint_url='', aws_access_key_id='', aws_secret_access_key='')
await client.upload_file('some_bucket', 'metaclass.txt', b'test')
```

2. fetch_and_upload — загрузить файл по ссылке в хранилище S3. Нужно полностью скачать файл, положить его в буфер в память и затем отправить в хранилище S3.

Функция принимает:

bucket, path — путь до объекта в S3

url — ссылка на файл, который нужно загрузить

Функция ничего не должна возвращать.

```
client = S3Client(endpoint_url='', aws_access_key_id='', aws_secret_access_key='')
await client.fetch_and_upload('some_bucket', 'dog.gif',
'https://random.dog/d239ed62-ea35-4e89-8931-5b0c732bd2ae.gif')
```

3. stream_upload — загрузить файл по ссылке в хранилище S3.

Функционально stream_upload повторяет fetch_and_upload. Отличие заключается в том, что stream_upload должен использовать **MultipartUploader** и загружать файл *частями* в хранилище S3, а не целиком, как fetch_and_upload.

Функция принимает:

bucket, path — путь до объекта в S3

url — ссылка на файл, который нужно загрузить

Функция ничего не должна возвращать.

```
client = S3Client(endpoint_url='', aws_access_key_id='', aws_secret_access_key='')
await client.stream_upload('some_bucket', 'dog.gif',
'https://random.dog/d239ed62-ea35-4e89-8931-5b0c732bd2ae.gif')
```

Подсказки

- для получения потока данных при скачивании стоит использовать корутину **resp.content.iter_chunked**
- минимальный размер части при загрузке в S3 должен быть больше 5 Мб
- iter_chunked может возвращать меньшее количество байт, чем указано в аргументе

Для запуска функций используйте файл **run_clients.py** и функцию **async def s3**.

Для проверки задания в терминале запустите:

```
pytest tests/clients/fapi/test_s3.py -vv -s
```