1. Для запуска сервиса с помощью Docker-compose необходимо выполнить следующие команды:

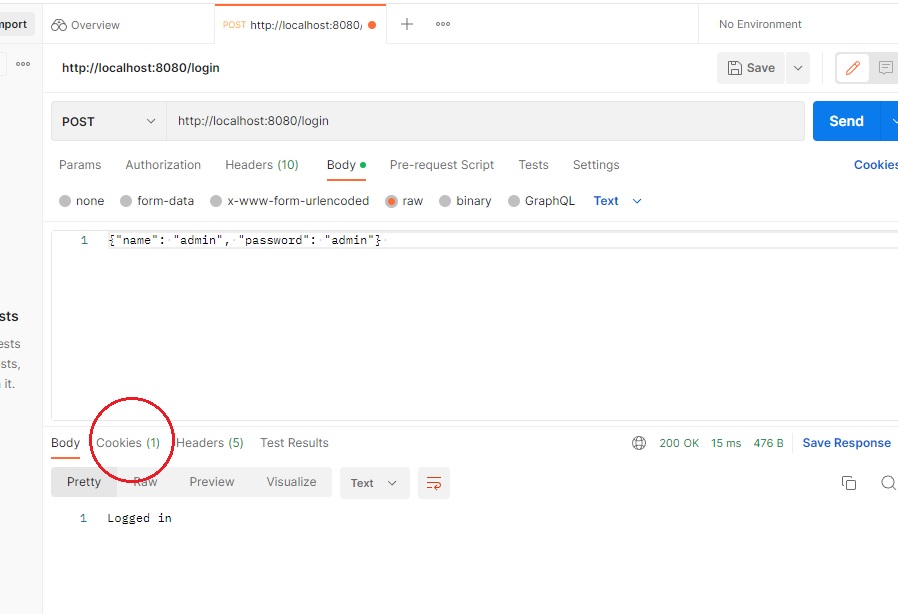
docker-compose -f docker-dev.yml build

docker-compose -f docker-dev.yml up

сервис запустится по адресу <http://localhost:8080/users> Для работы с сервисом удобнее всего использовать Postman.

1. Для создания таблиц я последовательно запустил файлы models.py и prepopulate\_bd.py, затем был создан дамп полученной базы Postgresql, затем я смонтировал дамп в /docker-entrypoint-initdb.d/ файла docker-dev.yml, что позволяет инициировать создание базы данных.

В базе данных есть 3 юзера, описанных в prepopulate\_bd.py, первый юзер с role\_id == 1 это администратор, администратор имеет права, описанные в задании.

1. Тригер на Users описан в файле app/models.py
2. Обработчики ендпоинтов описаны в файле app/views.py
3. Все методы доступны только после авторизации, для авторизации необходимо отправить запрос POST на <http://127.0.0.1:8080/login> с телом {"name": "admin", password": "admin"} для админа или {"name": "user", password": "user"} для юзера. В ответ придёт заголовок set\_cookies и установиться сессия, данные в куки будут хранится в зашифрованном виде.

Для выхода необходимо отправить POST на <http://localhost:8080/logout> с любым телом запроса.

Сессия закроется, кука будет удалена из браузера.

Можно протестировать все точки и методы, описанные в app/views.py под админом и под юзером.

1. Для регистрации нового юзера необходимо отправить POST на <http://localhost:8080/users> с телом, например {"name": "new\_users","last\_name": "Swanson","password": "123456","birthdate": "10102012", "role\_id": 2}
2. В файле app/permission.py описаны права доступа, необходимые по заданию.
3. В сессионную куку можно добавить любую информацию, так же, как и сервер может прочитать любую информацию из неё. В обработчике async def login(request) (185 строка) я добавил код для записи в куку allowed\_methods, сами методы описаны permission.py. Эти методы в логике сервера не используются, я их ввёл как пример. Для проверки прав доступа сервер считывает только информацию о id юзера и его роли.
4. В файле tests.py описана тривиальная функция тестирования, к сожалению, не успеваю её доработать, тк тестирование асинхронного сервера имеет некоторые отличия от привычного мне тестирования джанго сервера, после отправки задания планирую разобраться в нём лучше.
5. В сервере подключены логи.
6. По адресу <http://localhost:8080/api/v1/docs> можно получить документацию swagger, к сожалению не успел подключить авторизацию в интерфейсе, поэтому доступны только методы, не требующие авторизации.

Благодарю за внимание, с уважением Опарин Д.В.