**ОБЩАЯ ПОСТАНОВКА**

Задание ориентировано на решение задач разработки сервисов в составе абстрактной площадки онлайн торговли с функциями агрегатора магазинов (товарных предложений) в широком диапазоне пользовательских запросов. Примеры: Ozon, Wildberries, Beru.ru, AliExpress.

**ЗАДАЧА №1**

Спроектировать базу данных со следующим минимальным набором таблиц:

* Пользователь - User(id, name, surname, fathers\_name, email);
* Товар(Книга) - Book(id, name, author, isbn);
* Магазин - Shop(id, name, address, post\_code);
* Заказ - Order(id, reg\_date(дата оформления заказа), user\_id(внешгий ключ для User.id));
* Позиция заказа - OrderItem(id, order\_id(внешний ключ для Order.id), book\_id(внешний ключ для Book.id), book\_quantity(количество экземпляров книги), shop\_id(внешний ключ для Shop.id)).
* *Создана доп. Таблица – Stock (id, shop\_id (внешний ключ для Shop.id), book\_id (внешний ключ для Book.id), available qt) для возможности просмотра ассортимента определенного магазина*

Таблицы БД необходимо заполнить произвольными данными (количество записей не имеет значения). Выбор СУБД остается за Соискателем, в рамках тестового задания можно ограничиться sqlite.

*БД инициализируется файлом utils/init\_db.py. В файле два метода:*

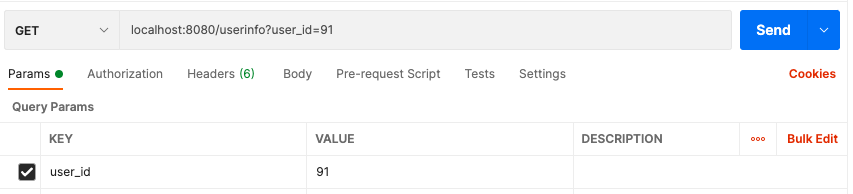
1. *create\_database – создает саму БД в папке database и соответствующие таблицы*
2. *fillin\_database -  заполняет произвольными данными таблицы из csv файлов в папке static*

**ЗАДАЧА №2**

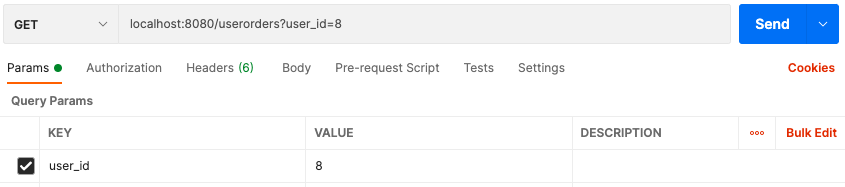
*API запускается из файла main.py (предполагается, что к этому моменту БД создана и проинициализирована)*

Реализовать API со следующим набором функций:

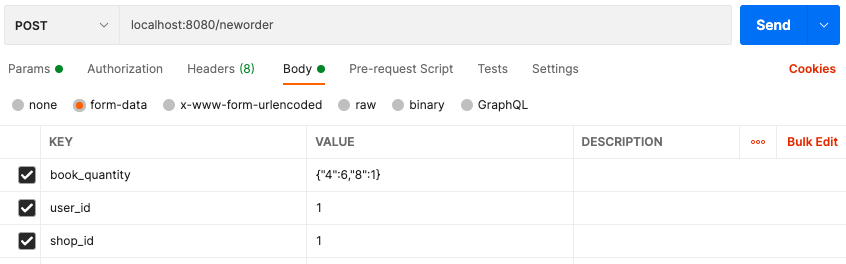
* Получение данных пользователя (имя, адрес эл. Почты и т.п.); *Осуществляется GET запросом вида localhost:8080/userinfo?user\_id=97*



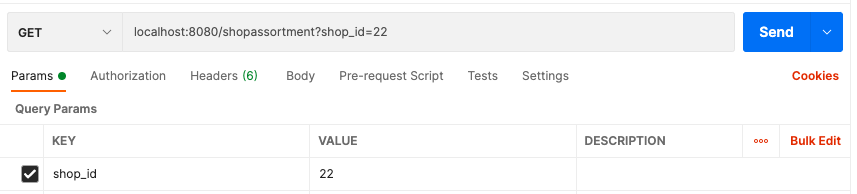
* Просмотр истории заказов пользователя; *Осуществляется GET запросом вида localhost:8080/userorders?user\_id=2*



* Добавление нового заказа (N книг каждая из которых в M количестве); *осуществляется POST запросом вида localhost:8080/neworder. Ниже список аргументов «ключ-значение»*
  + *Ключ book\_quantity, значение вида {"4":6,"8":1} (где в качестве ключа передается book\_id, в качестве значения кол-во)*
  + *Ключ user\_id, значение число*
  + *Ключ shop\_id, значение число*

**

* Просмотр ассортимента определенного магазина; *Осуществляется GET запросом вида localhost:8080/shopassortment?shop\_id=22*



Методы для каждого из запросов («GET», «POST», «PUT» и др.) Соискатель может назначить самостоятельно. Нет необходимости в реализации графического или веб интерфейса— все запросы будут тестироваться с помощью Postman. Фреймворк для написания API — aiohttp. Библиотека для общения с базой данных — sqlalchemy.

Дополнительно приветствуется:

* Добавление логгирования;
* Написание скрипта для создания и заполнения тестовой базы данных дабы облегчить тестирование полученной программы;
* Написание setup.py или requirements.txt с указанием зависимостей;
* Добавление краткого туториала с описанием запуска программы и требований, необходимых для ее работы;
* Краткое документирование кода.

Вся работа должна выполняться на python 3.5 или более поздних версий. Результат необходимо разместить в публичном репозитории.