

Тестовое задание Backend developer

Описание

Необходимо разработать серверную часть приложения для редактирования Cover Build-Up (кривые накопления охвата для разных частот) с использованием фреймворка Django и Django REST Framework (DRF).

API должно состоять из 3 endpoint'ов:

- Авторизация
- Ввод таблицы CBU
- Вывод сохраненных значений

Авторизация

Авторизация используется для защиты всех остальных endpoint'ов от несанкционированного доступа к данным с помощью логина / пароля, выдаваемых пользователям системы.

Авторизация должна осуществляться с помощью JWT. Можно использовать любую готовую библиотеку (например, `django-rest-framework-simplejwt`).

В рамках данного задания ситуацию, когда истекает срок действия токена (т.е. обновлять уже полученный токен), рассматривать не нужно.

Пример POST запроса на авторизацию по адресу

`http://127.0.0.1:8000/user/token/` :

```
{
  "username": "login",
  "password": "pass"
}
```

Пример ответа на запрос (может отличаться в зависимости от выбранной библиотеки):

```
{
  "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9...."
```

```
}
```

Ввод таблицы

СВУ представляет собой таблицу с 11 столбцами и неограниченным количеством строк. Пример таблицы с данными в файле `test_data.xlsx` в приложении к заданию. Столбец `Unit` - рейтинг, столбцы `Reach` - охват (в процентах).

Данный endpoint должен:

- Принять на вход таблицу в определенном формате
- Провалидировать значения
- Перезаписать в БД целиком таблицу для текущего пользователя

Каждый пользователь видит и может редактировать только свои данные.

Пример POST запроса по адресу `http://127.0.0.1:8000/curve/`:

```
[
  {
    "unit": 10000,
    "reach": [9.39, 2.88, 1.71, 1.48, 1.05, 0.83, 0.66, 0.53, 0.44, 0.4],
  },
  {
    "unit": 20000,
    "reach": [17.13, 6.68, 4.37, 2.97, 2.13, 1.68, 1.34, 1.08, 0.9, 0.8]
  }
]
```

Валидация данных

Правила валидации:

- В массиве `reach` всегда 10 элементов
- Вводить можно только числа (дробные)
- Значения в столбце `Unit`: $0 \leq \text{value}$
- Значения в столбцах `Reach`: $0 \leq \text{value} \leq 100$
- Значения охватов в строке (столбцы `Reach`) должны убывать

Если хотя бы одно правило нарушено - вернуть статус 400 с указанием причины.

Вывод сохраненных значений

Необходимо вывести ранее сохраненную таблицу пользователю в определенном формате.

Пример ответа на GET запрос по адресу `http://127.0.0.1:8000/curve/` :

```
[
  {
    "unit": 10000,
    "reach": [9.39, 2.88, 1.71, 1.48, 1.05, 0.83, 0.66, 0.53, 0.44, 0.4]
  },
  {
    "unit": 20000,
    "reach": [17.13, 6.68, 4.37, 2.97, 2.13, 1.68, 1.34, 1.08, 0.9, 0.8]
  }
]
```

Если данных у пользователя еще нет - в ответ должен прийти пустой массив.

Стэк технологий

- Django + Django Rest Framework (DRF) – обязательно
- Библиотека для JWT - любая на выбор (`django-rest-framework-simplejwt` , ...)
- БД – любая на выбор (SQLite, PostgreSQL, ...)

Срок выполнения

1 неделя

Предоставление результата

Необходимо предоставить:

- Исходный код на GitHub/GitLab (предпочтительно) или в архиве
- Инструкцию по сборке и запуску проекта