**Документация MTUCI please**

Backend составляющая данного проекта выполнена на языке программирования Python, фреймворк Django-rest-framework. В основе проекта строится 2 приложения – first\_level и user\_api. Также задействована база данных, в которой реализованы такие таблицы, как AppUser и Person. В проекте задействован механизм регистрации и аутентификации пользователя согласно CORS политике. Далее приводится подробное описание каждого аспекта backend части.

# **Приложение user\_api**

Приложение user\_api служит для реализации регистрации новых пользователей, аутентификации имеющихся пользователей, выхода из учетной записи и присваивания заработанных очков пользователям.

## **Контроллеры**

1. UserRegister

Url-адрес доступа: <http://127.0.0.1:8000/api/register>

Виды запросов:

* POST

Принимает email, username и password. В случае некорректных данных или использования уже существующего пользователя выдает статус 400\_BAD\_REQUEST. При корректных данных создается и сразу авторизуется новый пользователь, статус – 201\_CREATED.

1. UserLogin

Url-адрес доступа: <http://127.0.0.1:8000/api/login>

Виды запросов:

* POST

Принимает username и password. В случае неверного логина или пароля выдает статус 400\_BAD\_REQUEST. При вводе корректных данных происходит авторизация пользователя, статус – 200\_OK.

1. UserLogout

Url-адрес доступа: <http://127.0.0.1:8000/api/logout>

Виды запросов:

* POST

Принимает данные текущего пользователя и происходит выход из учетной записа, статус – 200\_OK.

1. UserView

Url-адрес доступа: <http://127.0.0.1:8000/api/user>

Виды запросов:

* GET

Если пользователь авторизован, выдает данные текущего пользователя, его имя, текущий уровень доступа и набранные очки. Статус – 200\_OK.

1. SetValue

Url-адрес доступа: <http://127.0.0.1:8000/api/set_value>

Виды запросов:

* POST

Принимает данные текущего пользователя, пройденный уровень и заработанные на нем очки. Обновляет рекорд и доступ к уровням данного пользователя. Статус – 200\_OK.

## **Модели**

Модель AppUser с полями:

1. User\_id
2. Last\_login
3. Username
4. Email
5. Password
6. Is\_staff
7. Is\_superuser
8. Level
9. First\_level\_score
10. Second\_level\_score (поле для будущей разработки)

Также реализован менеджер для данной модели – AppUserManager.

## **URL-адреса**

1. <http://127.0.0.1:8000/api/register>
2. <http://127.0.0.1:8000/api/login>
3. <http://127.0.0.1:8000/api/logout>
4. <http://127.0.0.1:8000/api/user>
5. <http://127.0.0.1:8000/api/set_value>

Устройство данных url-адресов указан в пункте «Контроллеры».

# **Приложение first\_level**

Приложение first\_level служит для реализации игрового процесса с генерацией различных студентов со своими ошибками в данных.

## **Контроллеры**

1. PersonsRandomize

Url-адрес доступа: <http://127.0.0.1:8000/first_level/persons>

Виды запросов:

* GET

Создает случайного пользователя и отправляет его данные и код ошибки, которая находится в его данных. Статус – 200\_OK.

## **Модели**

Модель Person с полями:

1. Photo
2. Surname
3. Name
4. Middle\_name
5. Gender

## **URL-адреса**

1. <http://127.0.0.1:8000/first_level/persons>