

Практика Цифровая Кафедра 2023

14 сен 2023, 18:08:30

старт: 14 апр 2023, 15:00:00

начало: 14 апр 2023, 15:00:00

10. Flask Rest API

Ограничение времени	2 секунды
Ограничение памяти	64.0 Мб
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Напишите rest-api веб сервис осуществив стандартные CRUD операции над пользователями (users)

Используя микрофреймворк Flask.

Для работы с json данными необходимо осуществить импорт библиотек (from flask import Flask, request, jsonify)

Для того чтобы симитировать работу с веб клиентом используйте postman

<https://www.postman.com/>

(скачать, установить, зарегистрироваться не обязательно)

указываем заголовки с Accept и ContentType, т.к работаем с json данными

<https://i.gyazo.com/0c1ea86075d9d0b58cccc2e718e70d16.png>

Для того чтобы симитировать в тестовом задании работу с базой данных, вам необходимо использовать глобальную переменную, например USERS можно ее определить как Dictionary или как List

Данные на сервер нужно отправлять в формате json.

При создание и редактировании используем следующую структуру полей пользователей

```
name: string
email: string
password: string
```

В своем программном коде вы должны добавить след. методы (едпоинты):

POST - создание пользователя

/users

вывод json response вида:

```
{"status": "ok", "message": "user {сюда подставить name} successfully created"}
```

GET - выводит список всех созданных пользователей

/users

вывод json response вида: [ссылка](#)

GET - получить информацию о пользователе по его почте (не выводить пароль)

/users/<user_email>

вывод json response вида:

```
{"email": "example1@gm.com", "name": "name1"}
```

если передаем не существующей емейл, т.е не нашли пользователя, то выводим ошибку

```
{"message": "user not found", "status": "error"}
```

PUT - обновление пользователя по почте

/users/<user_email>

вывод json response вида:

```
{"message": "name changed from {тут старое name} to {тут новое name}", "status": "ok"}
```

если передаем не сущ. емейл, т.е не нашли пользователя, то выводим ошибку

```
{"message": "user not found", "status": "error"}
```

DELETE - удаляем пользователя по почте

`/users/<user_email>`

вывод json response вида:

```
{"message": "user with email {тут почта удаленного пользователя} has been deleted", "status": "ok"}
```

GET - вернуть количество созданных пользователей

`/users/count`

метод, который будет возвращать количество пользователей, уже не в формате json, а просто текстом

Примечания

Проверяющая система использует такую команду для запуска веб-приложения:

```
FLASK_APP=solution_rest.py flask run
```

Для отправки решения, выберите в качестве Языка **Python 3.7 + network + requests**

Язык

Python 3.9 (PyPy 7.3.11)

Набрать здесь

Отправить файл

1

Отправить

Предыдущая