

Домашнє завдання (18)

▼ ДЗ №1

▼ Назва

Mars Rover "Curiosity" photos

▼ Опис

Є відкритий API NASA який дозволяє за певними параметрами отримати дані у вигляді JSON про фото зроблені ровером "Curiosity" на Марсі. Серед цих даних є посилання на фото які потрібно розпарсити і потім за допомогою додаткових запитів скачати і зберегти ці фото як локальні файли `mars_photo1.jpg` , `mars_photo2.jpg` . Завдання потрібно зробити використовуючи модуль `requests`

▼ Складність

Середня

▼ Початкові дані

```
import requests

url = 'https://api.nasa.gov/mars-photos/api/v1/rovers/curiosity/photos'
params = {'sol': 1000, 'camera': 'fhaz', 'api_key': 'DEMO_KEY'}
```

▼ ДЗ №2

▼ Назва

POST/GET/DELETE

▼ Опис

У venv Python встановіть Flask за допомогою команди `pip install flask`

Створіть у окремій директорії файл `app.py` та скопіюйте у нього код файлу `app.py` який приведено нижче в початкових даних.

Запустіть http сервер за допомогою команди `python app.py`

Сервер стартує за базовою адресою `http://127.0.0.1:8080`

Враховуючи документацію яку наведено нижче вам потрібно написати код який використовуючи модуль `request` зробить через POST upload якогось зображення на сервер, за допомогою GET отримає посилання на цей файл и потім за допомогою DELETE зробить видалення файлу з сервера

▼ Документація для `app.py`

Серверна частина надає можливість завантажувати, отримувати та видаляти зображення.

Завантаження зображення

Метод: POST

Шлях: `/upload`

Опис: Завантажує зображення на сервер.

Параметри запиту:

- `image`: файл зображення (тип MIME: image/*)

Відповідь:

- Код стану 201 (Created) у разі успішного завантаження.
- Повертає URL завантаженого зображення у форматі JSON:

```
{
  "image_url": "http://127.0.0.1:8080/uploads/example.jpg"
}
```

Отримання URL завантаженого зображення

Метод: GET

Шлях: `/image/<filename>`

Опис: Повертає URL або саме зображення в залежності від заголовка Content-Type.

Параметри запиту:

- `filename` : ім'я файлу зображення (рядок)

Відповідь:

- Код стану 200 (OK)
- Повертає URL завантаженого зображення у форматі JSON, якщо Content-Type рівний `text` :

```
{
  "image_url": "http://127.0.0.1:8080/uploads/example.jpg"
}
```

- Повертає саме зображення, якщо Content-Type рівний `image` .

Видалення зображення

Метод: DELETE

Шлях: `/delete/<filename>`

Опис: Видаляє завантажене зображення з серверу.

Параметри запиту:

- `filename` : ім'я файлу зображення (рядок)

Відповідь:

- Код стану 200 (OK) у разі успішного видалення.
- Повертає повідомлення про успішне видалення у форматі JSON:

```
{
  "message": "Image example.jpg видалено"
}
```

▼ Складність

Середня

▼ Початкові дані

Файл `app.py`:

```
from flask import Flask, request, jsonify, send_from_directory
import os

app = Flask(__name__)

upload_directory = './uploads'
if not os.path.exists(upload_directory):
    os.makedirs(upload_directory)

@app.route('/upload', methods=['POST'])
def upload_image():
    if 'image' not in request.files:
        return jsonify({'error': 'No image provided'}), 400

    image = request.files['image']
    if image.filename == '':
        return jsonify({'error': 'No selected file'}), 400

    filename = os.path.join(upload_directory, image.filename)
    image.save(filename)

    return jsonify({'image_url': request.host_url + 'uploads/' + image.filename}), 201

@app.route('/image/<filename>', methods=['GET'])
```

```

def get_image(filename):
    content_type = request.headers.get('Content-Type')
    filepath = os.path.join(upload_directory, filename)
    if os.path.exists(filepath):
        if content_type == 'text':
            return jsonify({'image_url': request.host_url + 'uploads/' + filename}), 200
        elif content_type == 'image':
            return send_from_directory(upload_directory, filename)
        else:
            return jsonify({'error': 'Unsupported Content-Type'}), 400
    else:
        return jsonify({'error': 'Image not found'}), 404

@app.route('/delete/<filename>', methods=['DELETE'])
def delete_image(filename):
    filepath = os.path.join(upload_directory, filename)
    if not os.path.exists(filepath):
        return jsonify({'error': 'Image not found'}), 404

    os.remove(filepath)
    return jsonify({'message': f'Image {filename} deleted'}), 200

if __name__ == '__main__':
    host = '127.0.0.1'
    port = 8080
    app.run(host=host, port=port, debug=True)

```