Національний технічний університет України «КПІ ім. Ігоря Сікорського»

Факультет Інформатики та Обчислювальної Техніки

Кафедра інформаційних систем та технологій

Лабораторна робота №5

з дисципліни «Проектування інформаційних систем»

на тему

«Використання і створення API»

Виконав:

студент групи ІС-21 Мельник С. В.

Викладач:

Ковальов М. О.

Київ – 2024

**Постановка задачі**

**Завдання:** Підключити зовнішній сервіс до свого рішення. Сервіс можна використовувати існуючий або створити власний. Приклади зовнішніх сервісів наведені в Додатку.

**Вправа 1.** Обрати зовнішній сервіс і описати сценарій його використання. Створити в GitHub репозиторій з описом сценарію використання API зовнішнього застосування.

**Вправа 2.** Реалізувати на будь-якій мові програмування виклик API зовнішнього застосування і візуалізацію відповіді. Додати в GitHub код і опис прикладу застосування API.

**Вправа 3.** Реалізувати на будь-якій мові програмування http-сервер, організувати звертання до нього і отримати відповідь у браузері. Звертання має містити Ваш логін у Moodle. Відповідь має містити Ваші особисті дані (прізвище, ім’я, курс, група).

**Виконання**

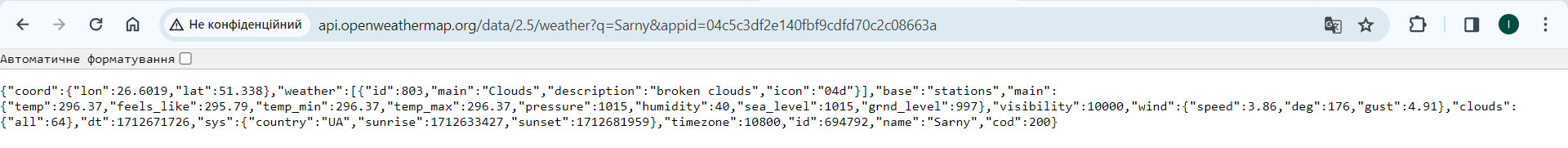
**Вправа 1**

Для доступу до зовнішнього API я обрала WEB-сервіс прогнозування погоди *OpenWeatherMap*. Документація по його API знаходиться [тут](https://idratherbewriting.com/learnapidoc/docapis_scenario_for_using_weather_api.html).

Спершу отримала вільний логін [тут](https://home.openweathermap.org/users/sign_in). Далі на сторінці свого логіну до сервісу <https://home.openweathermap.org/api_keys> подивилась значення API key:

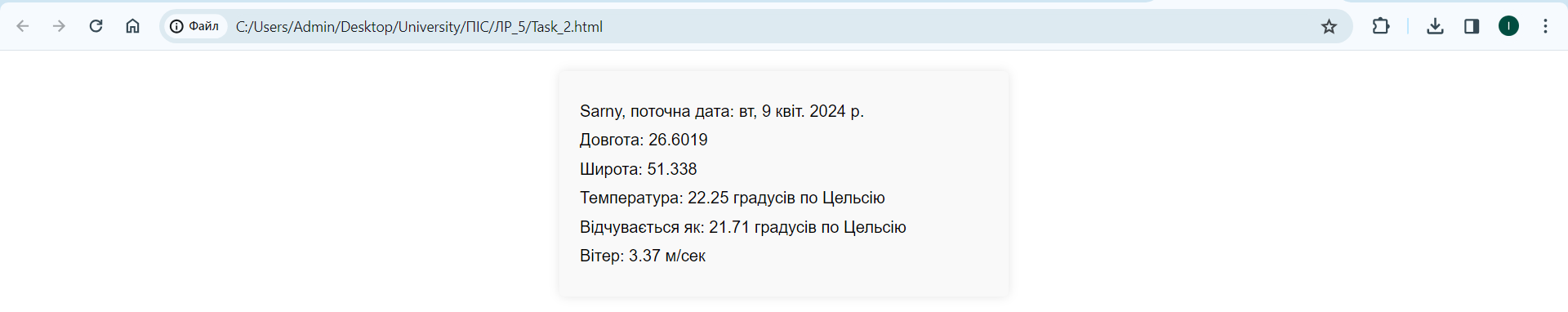
04c5c3df2e140fbf9cdfd70c2c08663a

Виконала один з прикладів виклику цього API, а саме запит погоди по Id міста, де заданий параметр *appid*, який дорівнює API Key. У підсумку запит <http://api.openweathermap.org/data/2.5/weather?q=Sarny&appid=04c5c3df2e140fbf9cdfd70c2c08663a> згенерував наступну відповідь



**Вправа 2**

Для реалізації виклику API зовнішнього застосування я обрала мову програмування JavaScript. Таким чином, можна відіслати запит до API і подивитись відповідь у браузері:



В HTML-документі з вбудованим кодом JavaScript виклик виглядає так:

<!DOCTYPE html>

<html lang="uk"> <!-- Вказуємо мову документа -->

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Прогноз погоди</title> <!-- Заголовок сторінки -->

    <style>

        body {

            font-family: Arial, sans-serif; /\* Шрифт для тексту \*/

            margin: 0; /\* Видаляємо зовнішні відступи \*/

            padding: 20px; /\* Додаємо внутрішні відступи \*/

        }

        #weather-info {

            background-color: #f9f9f9; /\* Колір фону блоку \*/

            padding: 20px; /\* Відступи в середині блоку \*/

            border-radius: 5px; /\* Закруглені кути блоку \*/

            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1); /\* Тінь блоку \*/

            max-width: 400px; /\* Максимальна ширина блоку \*/

            margin: 0 auto; /\* Вирівнюємо по центру \*/

        }

        p {

            margin: 10px 0; /\* Відступи між абзацами \*/

        }

    </style>

</head>

<body>

    <div id="weather-info"></div> <!-- Блок для відображення погодової інформації -->

    <script>

        const API\_KEY = "04c5c3df2e140fbf9cdfd70c2c08663a";

        const city = "Sarny";

        const apiUrl = `http://api.openweathermap.org/data/2.5/weather?q=${city}&appid=${API\_KEY}`;

        fetch(apiUrl)

            .then(response => response.json())

            .then(data => {

                const weatherInfo = document.getElementById('weather-info');

                const cityName = data.name;

                const date = new Date(data.dt \* 1000);

                const temperature = (data.main.temp - 273.15).toFixed(2); // Переводимо температуру з Кельвіна у Цельсій

                const feelsLike = (data.main.feels\_like - 273.15).toFixed(2); // Переводимо "відчувається як" з Кельвіна у Цельсій

                const windSpeed = data.wind.speed;

                // Опції для форматування дати

                const options = { weekday: 'short', month: 'short', day: 'numeric', year: 'numeric' };

                const formattedDate = date.toLocaleDateString('uk-UA', options);

                // Виводимо погодову інформацію у блок

                weatherInfo.innerHTML = `

                    <p>${cityName}, поточна дата: ${formattedDate}</p>

                    <p>Довгота: ${data.coord.lon}</p>

                    <p>Широта: ${data.coord.lat}</p>

                    <p>Температура: ${temperature} градусів по Цельсію</p>

                    <p>Відчувається як: ${feelsLike} градусів по Цельсію</p>

                    <p>Вітер: ${windSpeed} м/сек</p>

                `;

            })

            .catch(error => console.error('Помилка отримання погодових даних:', error));

    </script>

</body>

</html>

**Вправа 3**

Для реалізації цієї вправи я використала мову програмування JavaScript та фреймворк Node.js для створення простого HTTP-сервера. Створила файл з розширенням .js та назвала його server.js. Код сервера в цьому файлі:

const http = require('http');

const url = require('url');

const server = http.createServer((req, res) => {

  // Отримуємо логін з запиту

  const queryObject = url.parse(req.url, true).query;

  const login = queryObject.login;

  // Перевіряємо, чи логін співпадає з логіном у Moodle

  if (login === 'is-21fiot-22-050') {

    // Встановлюємо HTTP-статус 200 (ОК)

    res.writeHead(200, {'Content-Type': 'text/html; charset=utf-8'});

    // Відправляємо відповідь з особистими даними у форматі HTML

    res.end(`

      <html>

        <head>

          <title>Особисті дані</title>

        </head>

        <body>

          <h1>Особисті дані</h1>

          <p><strong>Прізвище:</strong> Мельник</p>

          <p><strong>Ім'я:</strong> Софія</p>

          <p><strong>Курс:</strong> 2 курс</p>

          <p><strong>Група:</strong> ІС-21</p>

        </body>

      </html>

    `);

  } else {

    // Встановлюємо HTTP-статус 404 (Not Found) у разі невірного логіну

    res.writeHead(404, {'Content-Type': 'text/plain; charset=utf-8'});

    res.end('Користувач не знайдений');

  }

});

// Сервер слухає порт 3000

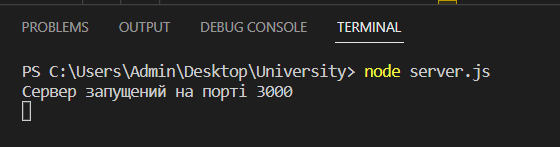
const port = 3000;

server.listen(port, () => {

  console.log(`Сервер запущений на порті ${port}`);

});

Після створення цього файлу і збереження його, запускаю сервер за допомогою Node.js, виконавши команду:



Тепер, перейшовши за посиланням <http://localhost:3000/?login=is-21fiot-22-050>, бачу відповідь зі своїми особистими даними:

