Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра автоматизованих систем обробки інформації та управління

Лабораторна робота №6

З дисципліни «Проектування інформаційних систем»

На тему: «Використання і створення API»

Виконала:

Студентка групи ІС-91

Куник Неля

## Постановка задачі:

**Завдання:** Підключити зовнішній сервіс до свого рішення. Сервіс можна використовувати

існуючий або створити власний. Приклади зовнішніх сервісів наведені в Додатку.

**Вправа 6.1.** Обрати зовнішній сервіс. і описати сценарій його використання. Створити в

GitHub репозіторій з описом сценарію використання API зовнішнього застосування.

**Вправа 6.2.** Реалізувати на будь-якій мові програмування виклик API зовнішнього

застосування і візуалізацію відповіді. Додати в GitHub код і опис прикладу застосування

API.

**Вправа 6.3.** 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:

**Посилання на репозиторій на GitHub**:   
https://github.com/neonilakunyk/is-91-012

**Вправа 6.1.**

WeatherAPI.com - це потужний повністю керований безкоштовний постачальник API-інтерфейсів погоди і геолокації, який надає API-інтерфейси широкого застосування, які варіюються від прогнозу погоди в реальному часі, історичної погоди, даних про якість повітря, IP-пошуку і астрономії до спорту, часових поясів і геолокації.

За замовчуванням у відповідь API повертає всі поля погоди, однак можна налаштувати у власному кабінеті поля, які потрібно, щоб сервіс повертав. Мені був не потрібний погодинний прогноз та прогноз на зараз, тож я вимкнула дані поля. Нижче наведено повний перелік, де галочка означає, що поле буде присутнє у відповіді.

### Current Weather

last\_updated\_epoch  
last\_updated  
temp\_c  
temp\_f  
is\_day  
text  
icon  
code  
wind\_mph  
wind\_kph  
wind\_degree  
wind\_dir  
pressure\_mb  
pressure\_in  
precip\_mm  
precip\_in  
humidity  
cloud  
feelslike\_c  
feelslike\_f  
vis\_km  
vis\_miles  
gust\_mph  
gust\_kph  
uv

### Forecast Weather

### forecastDay

date  
date\_epoch

### Day

maxtemp\_c  
maxtemp\_f  
mintemp\_c  
mintemp\_f  
avgtemp\_c  
avgtemp\_f  
maxwind\_mph  
maxwind\_kph  
totalprecip\_mm  
totalprecip\_in  
avgvis\_km  
avgvis\_miles  
avghumidity  
text  
icon  
code  
daily\_will\_it\_rain  
daily\_will\_it\_snow  
daily\_chance\_of\_rain  
daily\_chance\_of\_snow  
uv

### Astro

sunrise  
sunset  
moonrise  
moonset  
moon\_phase  
moon\_illumination

### Hour

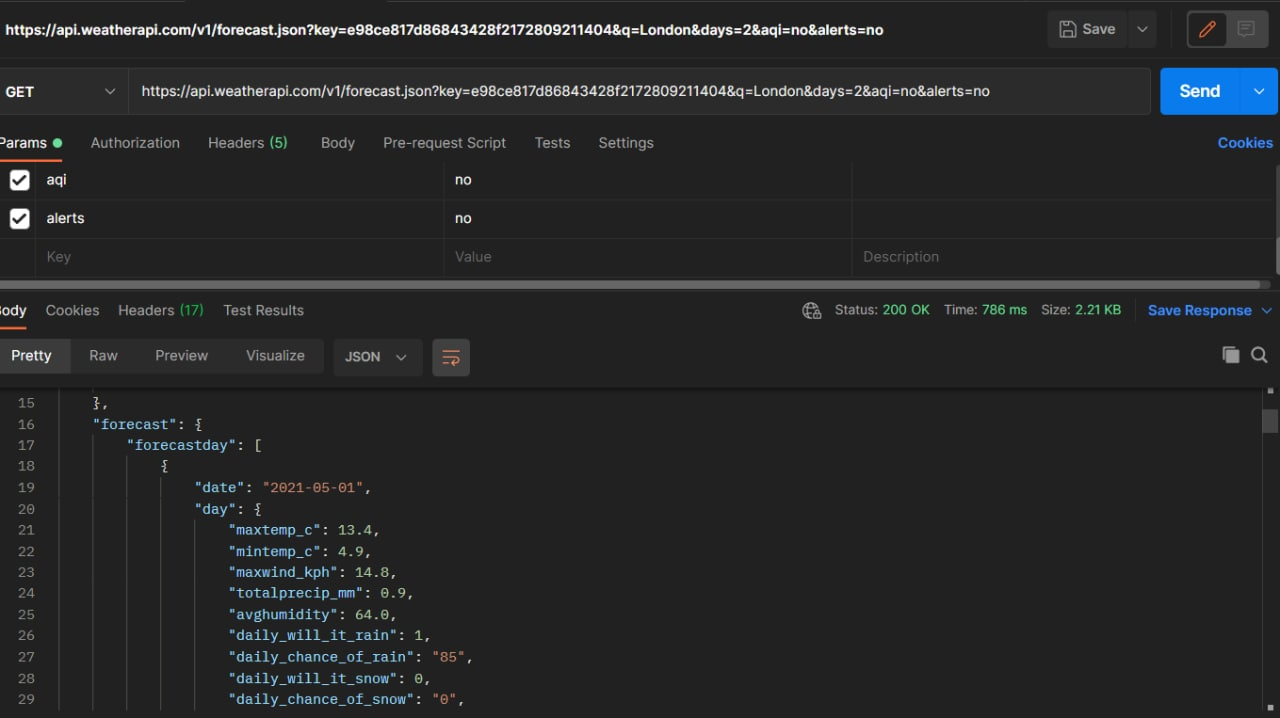
time\_epoch  
time  
temp\_c  
temp\_f  
is\_day  
text  
icon  
code  
wind\_mph  
wind\_kph  
wind\_degree  
wind\_dir  
pressure\_mb  
pressure\_in  
precip\_mm  
precip\_in  
humidity  
cloud  
feelslike\_c  
feelslike\_f  
windchill\_c  
windchill\_f  
heatindex\_c  
heatindex\_f  
dewpoint\_c  
dewpoint\_f  
will\_it\_rain  
will\_it\_snow  
vis\_km  
vis\_miles  
chance\_of\_rain  
chance\_of\_snow  
gust\_mph  
gust\_kph  
uv

У запиту є наступні параметри:

|  |  |
| --- | --- |
| **Параметр** | **Опис** |
| key | Ключ API |
| q | Параметр запиту, на основі якого дані надсилаються назад. Це може бути наступне: |
| Широта та довгота (десятковий градус), наприклад: q = 48,8567,2,3508 |
| назва міста, наприклад: q = Париж |
| США, наприклад: q = 10001 |
| Поштовий індекс Великобританії, наприклад: q = SW1 |
| Поштовий індекс Канади, наприклад: q = G2J |
| metar: <код metar> наприклад: q = metar: EGLL |
| iata: <3-значний код аеропорту> наприклад: q = iata: DXB |
| auto: ip IP-пошук, наприклад: q = auto: ip |
| IP-адреса (підтримується IPv4 та IPv6), наприклад: q = 100.0.0.1 |
| days | Кількість днів прогнозу. |
| Значення параметра днів становить від 1 до 10. наприклад: days = 5 |
| Якщо параметр днів не вказаний, повертається лише сьогоднішня погода. |
| dt (необхідний для API історії) | Для історії API 'dt' повинен бути 1 січня 2010 р. Або пізніше у форматі рррр-MM-dd (тобто dt = 01.01.2010) |
| Для прогнозу API 'dt' повинен бути між сьогодні і наступним 10 днем ​​у форматі рррр-MM-dd (тобто dt = 01.01.2010) |
| (необов’язковий) unixdt | unixdt має те саме обмеження, що і параметр 'dt'. Будь ласка, передайте "dt" або "unixdt", а не обидва в одному запиті. наприклад: unixdt = 1490227200 |
| (необов’язковий) end\_dt (доступний для API історії) | Для історії API 'end\_dt' має бути 1 січня 2010 р. Або пізніше у форматі рррр-ММ-дд (тобто dt = 01.01.2010) |
| "end\_dt" має бути більшим за параметр "dt", а різниця не повинна перевищувати 30 днів між двома датами. |
| **Працює лише для API на плані Pro та вище.** |
| (необов’язковий) unixend\_dt | unixend\_dt має те саме обмеження, що і параметр 'end\_dt'. Будь ласка, передайте 'end\_dt' або 'unixend\_dt', а не обидва в одному запиті. наприклад: unixend\_dt = 1490227200 |
| (Необов’язково) hour | Має бути через 24 години. Наприклад, 17:00 має бути hour = 17, 6 ранку як hour = 6 |
| (Необов’язкові) alerts | alerts=yes or alerts=no |
| (Необов’язково) aqi | aqi=yes or aqi=no |
| (Необов’язково) lang | "Код мови". Наприклад: lang = fr |

Далі я хотіла подивитися прогноз на найближчі 2 дні у Лондоні. Тому зробила запит за посиланням: https://api.weatherapi.com/v1/forecast.json?key=e98ce817d86843428f2172809211404&q=London&days=2&aqi=no&alerts=no

Дані повертаються в наступному вигляді:



*Малюнок 1. Перевірка роботи API за допомогою Postman*

Повний текст відповіді (три крапки вставлені замість пустих обьектів без значень, оскільки ці поля були виключені з відповіді раніше):

{

"location":{

"name":"London",

"region":"City of London, Greater London",

"country":"United Kingdom",

"lat":51.52,

"lon":-0.11,

"tz\_id":"Europe/London",

"localtime\_epoch":1619865785,

"localtime":"2021-05-01 11:43"

},

"current":{…},

"forecast":{

"forecastday":[

{

"date":"2021-05-01",

"day":{

"maxtemp\_c":13.4,

"mintemp\_c":4.9,

"maxwind\_kph":14.8,

"totalprecip\_mm":0.9,

"avghumidity":64.0,

"daily\_will\_it\_rain":1,

"daily\_chance\_of\_rain":"74",

"daily\_will\_it\_snow":0,

"daily\_chance\_of\_snow":"0",

"condition":{

"text":"Patchy rain possible"

}

},

"astro":{

"sunrise":"05:32 AM",

"sunset":"08:24 PM"

},

"hour":[…]

},

{

"date":"2021-05-02",

"day":{

"maxtemp\_c":14.8,

"mintemp\_c":5.3,

"maxwind\_kph":12.6,

"totalprecip\_mm":1.1,

"avghumidity":66.0,

"daily\_will\_it\_rain":1,

"daily\_chance\_of\_rain":"87",

"daily\_will\_it\_snow":0,

"daily\_chance\_of\_snow":"0",

"condition":{

"text":"Patchy rain possible"

}

},

"astro":{

"sunrise":"05:30 AM",

"sunset":"08:26 PM"

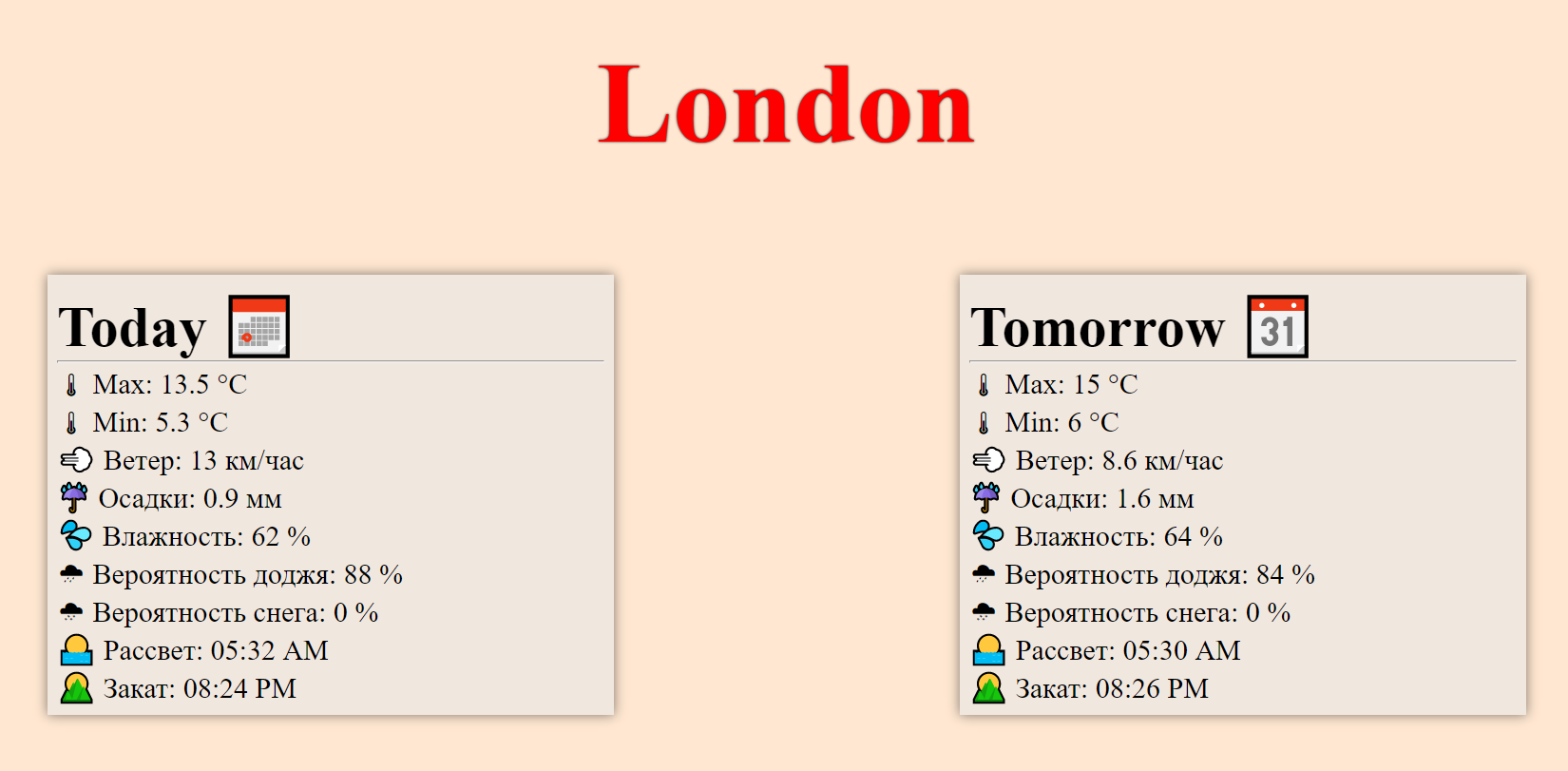
},

"hour":[…]

}

}

**Вправа 6.2.**



*Малюнок 2. Відображення даних отриманих з API*

Використовуючи наведений нижче код, можна відіслати запит до API і переглянути відповідь у браузері. Достятньо лише змінити змінну URL. Дана функціональність реалізується таким чином:

index.html

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <link rel="stylesheet" href="./style.css" />

    <title>Weather</title>

</head>

<body onload="LoadData()">

    <main>

        <div id="weather\_container">

            <div id="today\_container" class="day">

                <h1>Today 📅</h1>

                <hr>

                <span class="caption">🌡 Max: </span><span class="unit"> °C</span><br>

                <span class="caption">🌡 Min: </span><span class="unit"> °C</span><br>

                <span class="caption">💨 Ветер: </span><span class="unit"> км/час</span><br>

                <span class="caption">☔️ Осадки: </span><span class="unit"> мм</span><br>

                <span class="caption">💦 Влажность: </span><span class="unit"> %</span><br>

                <span class="caption">🌧 Вероятность доджя: </span><span class="unit"> %</span><br>

                <span class="caption">🌨 Вероятность снега: </span><span class="unit"> %</span><br>

                <span class="caption">🌅 Рассвет: </span><br>

                <span class="caption">🌄 Закат: </span>

            </div>

            <div id="tomorrow\_container" class="day">

                <h1>Tomorrow 📆</h1>

                <hr>

                <span class="caption">🌡 Max: </span><span class="unit"> °C</span><br>

                <span class="caption">🌡 Min: </span><span class="unit"> °C</span><br>

                <span class="caption">💨 Ветер: </span><span class="unit"> км/час</span><br>

                <span class="caption">☔️ Осадки: </span><span class="unit"> мм</span><br>

                <span class="caption">💦 Влажность: </span><span class="unit"> %</span><br>

                <span class="caption">🌧 Вероятность доджя: </span><span class="unit"> %</span><br>

                <span class="caption">🌨 Вероятность снега: </span><span class="unit"> %</span><br>

                <span class="caption">🌅 Рассвет: </span><br>

                <span class="caption">🌄 Закат: </span>

            </div>

        </div>

    </main>

    <script src="app.js"></script>

</body>

</html>

script.js

const LoadData = () => {

    let URL =

      "https://api.weatherapi.com/v1/forecast.json?key=e98ce817d86843428f2172809211404&q=London&days=2&aqi=no&alerts=no";

    let req = new XMLHttpreq();

    req.open("GET", URL);

    req.responseType = "json";

    req.send();

    req.onload = function () {

      ShowWeather(req.response);

    };

};

const ShowWeather = (obj) => {

    let city = document.createElement("h1");

    city.id = "city"

    city.textContent = obj.location.name;

    document.getElementById("weather\_container").before(city);

    for(let i=0; i<2; i++){

        const day = document.getElementsByClassName("day")[i]

        const d = obj.forecast.forecastday[i]

        let maxtemp = document.createElement("span");

        maxtemp.textContent = d.day.maxtemp\_c;

        day.getElementsByClassName("caption")[0].after(maxtemp);

        let mintemp = document.createElement("span");

        mintemp.textContent = d.day.mintemp\_c;

        day.getElementsByClassName("caption")[1].after(mintemp);

        let maxwind = document.createElement("span");

        maxwind.textContent = d.day.maxwind\_kph;

        day.getElementsByClassName("caption")[2].after(maxwind);

        let totalprecip = document.createElement("span");

        totalprecip.textContent = d.day.totalprecip\_mm;

        day.getElementsByClassName("caption")[3].after(totalprecip);

        let avghumidity = document.createElement("span");

        avghumidity.textContent = d.day.avghumidity;

        day.getElementsByClassName("caption")[4].after(avghumidity);

        let daily\_chance\_of\_rain = document.createElement("span");

        daily\_chance\_of\_rain.textContent = d.day.daily\_chance\_of\_rain;

        day.getElementsByClassName("caption")[5].after(daily\_chance\_of\_rain);

        let daily\_chance\_of\_snow = document.createElement("span");

        daily\_chance\_of\_snow.textContent = d.day.daily\_chance\_of\_snow;

        day.getElementsByClassName("caption")[6].after(daily\_chance\_of\_snow);

        let sunrise = document.createElement("span");

        sunrise.textContent = d.astro.sunrise;

        day.getElementsByClassName("caption")[7].after(sunrise);

        let sunset = document.createElement("span");

        sunset.textContent = d.astro.sunset;

        day.getElementsByClassName("caption")[8].after(sunset);

    }

}

**Вправа 6.3.**

Створимо сервер на локальному комп’ютері, отримавши доступ до нього через порт 3000. Сервер створюватимемо за допомогою пакету http та властивості url об'єкта Request. Нижче наведено код:

index.js

const http = require('http')

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

    res.setHeader("Content-Type","text/html; charset=utf-8;");

    if(req.url === "/" || req.url === "/home"){

        res.write('<h1>Main Page</h1>');

    }

    else if(req.url === "/about"){

        res.write('<h2>About</h2>');

    }

    else if(req.url === "/contact"){

        res.write('<h2>Contact</h2>');

    }

    else if(req.url === "/is-91-012"){

        res.write('<h2>Куник Неля IC-91 2 курс</h2>');

    }

    else{

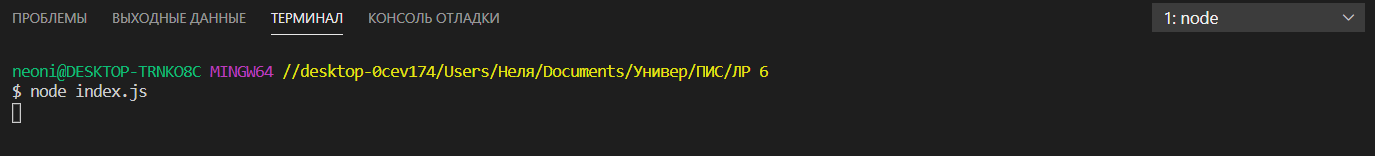
        res.write('<h2>This page is not found</h2>');

    }

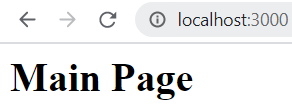
    res.end();

}).listen(3000);

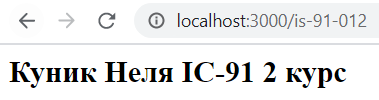
Файл виконаємо за допомогою команди node у вбудованому термінали програми vs code.



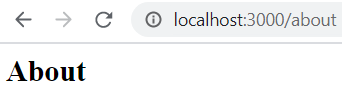
У браузері по адресі http://localhost:3000/ бачимо відповідь сервера.



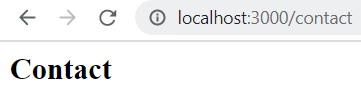
Організуємо звертання, яке містить мій логін у Moodle.



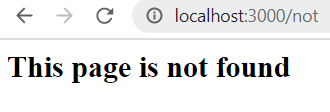
Організуємо звертання, яке містить about.



Організуємо звертання, яке містить contact.



Організуємо звертання, яке не обробляється.



Всі відповіді сервера відповідають нашим очікуванням.