

УДК 004.415

*Цирульник С. М., канд. техн. наук,
доцент кафедри інформаційних та прикладних технологій;
Цирульник М. С., здобувач вищої освіти,
Донецький національний університет імені Василя Стуса*

РОЗРОБКА ДОДАТКА «ПОГОДА» ЗАСОБАМИ ФРЕЙМВОРКУ XAMARIN

Ключові слова: REST API, JSON, HTTP-запит, фреймворк Xamarin.

Вступ. Більшість сучасних додатків працюють за клієнт-серверною архітектурою. Програма додатку складається з двох частин – клієнтської та серверної. Клієнтська та серверна частини взаємодіють між собою через Application Programming Interface (API). API є набором готових класів, методів, структур і констант, що надаються бібліотекою, сервісом для використання в зовнішніх програмах, не звертаючи уваги на різні мови програмування, якими зроблено клієнтську та серверну частину. Розробка API складається з двох частин – написання коду та складання правил спілкування з додатком. Складання правил включає: виконання операції, отримання даних на вхід, передачу даних на вихід.

Актуальність. У мобільних або десктоп-додатках користувач звертається до деякого набору візуальних елементів (кнопки, списки, чекбокси) графічної оболонки GUI, які спрощують спілкування з функціями API. Найпопулярніший стандарт API – REST API. Rest надає дані у вигляді ресурсів, викликати які можна за унікальним URL-посиланням. REST API використовують HTTP-запити для роботи з ресурсами: GET (для отримання інформації), POST (для надсилання даних), PUT/PATCH (для оновлення або зміни даних), DELETE (для видалення даних). REST API підтримує обмін інформацією у таких форматах: звичайний текст, YAML, HTML, XML, JSON [1–3].

Розглянемо практичну реалізацію мобільного додатка «Погода» засобами фреймворку Xamarin в Visual Studio. Щоб дізнатися прогноз погоди, програма (Android IOS Windows) звертається до сервера та запитує фрагмент даних з інформацією про погоду. Дизайн додатка проектується в файлі MainPage.xaml мовою розмітки XAML (рис. 1).

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
  xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
  x:Class="Pogoda.MainPage">
  <StackLayout BackgroundColor="#ededed">
    <StackLayout BackgroundColor="#194880">
      <Label Text="Погода" HorizontalOptions="Center" Margin="0,50,0,15"
        FontSize="30" TextColor="#091c33" FontAttributes="Bold" />
      <Image Margin="0,0,0,45"
        Source="https://cdn-icons-png.flaticon.com/256/9139/9139799.png"/>
    </StackLayout>
  </StackLayout>
</RelativeLayout>
```

```

<Frame RelativeLayout.YConstraint="-30" BackgroundColor="#fcfcfc"
  Margin="80, 0" Padding="15" CornerRadius="15">
  <StackLayout>
    <Entry x:Name="userInput" WidthRequest="200"
      Placeholder="Введіть місто"/>
    <Button x:Name="getWeather" Clicked="getWeather_Clicked"
      Text="Отримати" BackgroundColor="#e6b035"
      TextColor="#8f6b18" FontAttributes="Bold"/>
  </StackLayout>
</Frame>
</RelativeLayout>
<Label x:Name="resultLabel" Text="" FontSize="25" HorizontalOptions="Center"
  TextColor="#333" Margin="0,0,0,50"/>
</StackLayout>
</ContentPage>

```

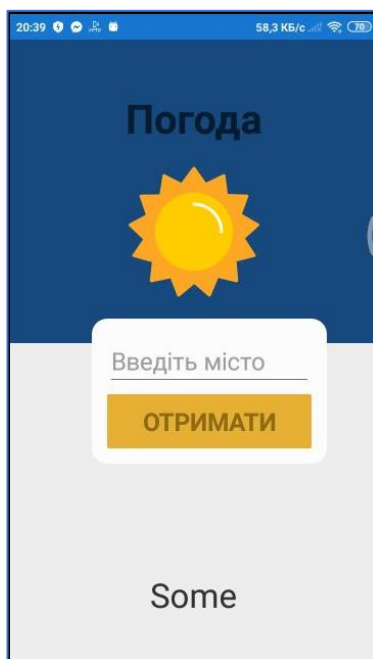


Рисунок 1 – Дизайн мобільного додатка «Погода»

Для отримання інформації про погоду потрібно сформувати HTTP-запит до сервера погоди та обробити отримані дані. Як сервер погоди використаємо ресурс OpenWeather [4]. Необхідно зареєструватись, скопіювати персональний API key та додати його як константу у додаток. На сайті знаходимо документацію про Build-in API request by city name (рис. 2).

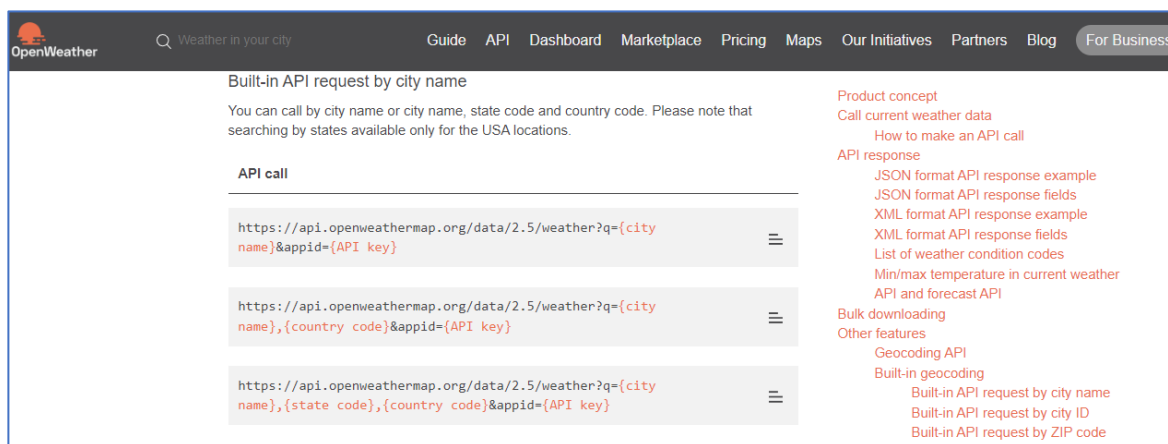


Рисунок 2 – Документація про Build-in API request by city name

Вводимо такий рядок у браузері:

```
https://api.openweathermap.org/data/2.5/weather?q=Kyiv&appid={API}
```

та отримаємо відповідь у JSON форматі.

```
{ "coord": { "lon": 30.5167, "lat": 50.4333 }, "weather": [ { "id": 803, "main": "Clouds", "description": "broken clouds", "icon": "04d" } ], "base": "stations", "main": { "temp": 27.77, "feels_like": 32.26, "temp_min": 27.62, "temp_max": 29.25, "pressure": 1012, "humidity": 85 }, "visibility": 1000, "wind": { "speed": 2.59, "deg": 34, "gust": 4.56 }, "clouds": { "all": 77 }, "dt": 169262675, "sys": { "type": 2, "id": 2003742, "country": "UA", "sunrise": 1692586495, "sunset": 1692637671 }, "timezone": 10800, "id": 703448, "name": "Kyiv", "cod": 200 }
```

Створимо обробник події за натисненням кнопки «Отримати» (рис. 1), у якому зчитуємо місто, що ввів користувач у елемент Entry з назвою userInput до змінної city. Формуємо API-запит. Для цього підключаємо бібліотеку System.Net.Http, створюємо об'єкт client-класу HttpClient та формуємо запит Get:

```
string city = userInput.Text.Trim();
HttpClient client = new HttpClient();
string url = $"https://api.openweathermap.org/data/2.5/weather?q="+
    $"{city}&appid={API}&units=metric";
string response = await client.GetStringAsync(url);
```

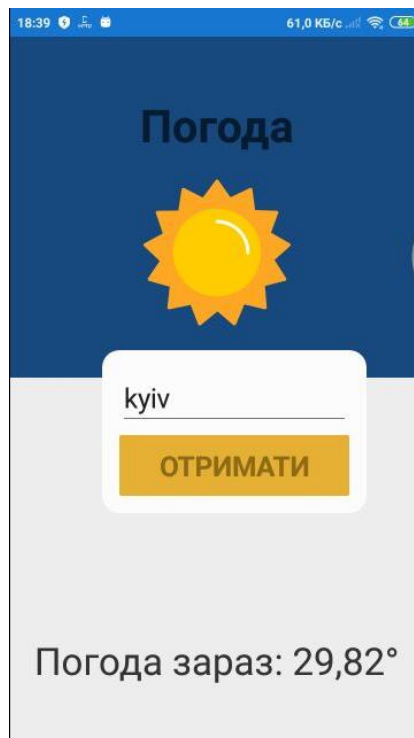


Рисунок 3 – Тестування роботи мобільного додатка «Погода»

Наступним етапом реалізації є обробка даних, що отримані в JSON-форматі. Для роботи з JSON є зручна бібліотека Newtonsoft.Json, яку необхідно встановити за допомогою менеджера пакетів NuGet у Visual Studio. У класі JObject є метод Parse, який дає змогу перетворити JSON-дані з рядка у відповідний об'єкт. Структура даних складається з двох елементів, кожен із яких

є парою ключ-значення. Необхідно отримати ключ `main` та отримати значення за ключем `temp`. Отримані дані виводимо у текстову змінну `resultLabel` (рис. 3).

Повний код обробника події за натисненням кнопки «Отримати»:

```
private async void getWeather_Clicked(object sender, EventArgs e)
{
    string city = userInput.Text.Trim();
    HttpClient client = new HttpClient();
    string url = $"https://api.openweathermap.org/data/2.5/+
    $"weather?q={city}&appid={API}&units=metric";
    string response = await client.GetStringAsync(url);
    var json = JObject.Parse(response);
    string temp = json["main"]["temp"].ToString();
    resultLabel.Text = "Погода зараз: " + temp + "\x00B0";
}
}
```

Висновки

Результатом роботи є дослідження формування API-запиту GET до сервера погоди OpenWeather об'єктом `client`-класу `HttpClient` та отримання відповіді у JSON-форматі. Для отримання інформації про погоду використовується бібліотека `Newtonsoft`. Метод `Parse`-класу `JObject` дає змогу за ключем отримати значення, яке відображається у текстовій змінній. Для практичної реалізації запропонованого підходу розроблений мобільний додаток «Погода» засобами фреймворку `Xamarin` в середовищі `Visual Studio 2019` мовою програмування `C#` та мовою розмітки `XAML`.

Список використаних джерел

1. Коротко про API та його тестування: вебсайт. URL: <https://qagroup.com.ua/publications/korotko-pro-ari-ta-jogo-testuvannia/> (дата звернення: 29.10.2023).
2. Що таке API, навіщо він необхідний і яку користь несе: вебсайт. URL: <https://highload.today/uk/shho-take-api-navishho-vin-neobhidnij-i-yaku-korist-nese> (дата звернення: 29.10.2023).
3. Що таке API і де їх шукати: вебсайт. URL: <https://info.nic.ua/uk/blog-uk/api-2> (дата звернення: 29.10.2023).
4. OpenWeather: вебсайт. URL: <https://openweathermap.org> (дата звернення: 29.10.2023).
5. Музика І. О., Кузнецов Д. І. Методичні вказівки до виконання лабораторних робіт з дисципліни «Об'єктно-орієнтоване програмування» для студентів спеціальності 123 «Комп'ютерна інженерія» усіх форм навчання. Частина 2. Кривий Ріг: Видавничий центр Криворізького національного університету, 2021. 111 с.
6. Цирульник С. М. Програмування мобільних додатків. Методичні вказівки для практичних занять для здобувачів першого (бакалаврського) рівня вищої освіти за галуззю знань 12 Інформаційні технології спеціальності 122 Комп'ютерні науки освітньо-професійної програми «Комп'ютерні науки». Вінниця, 2022. 127 с.
7. Шевчук Р. П., Сусла М. В. Методичні вказівки до виконання практичних робіт з дисципліни «Програмування для мобільних платформ», для студентів за спеціальністю 121 Інженерія програмного забезпечення. Тернопіль, 2017. 110 с.
8. What is Xamarin.Forms? URL: <https://docs.microsoft.com/en-us/xamarin/get-started/what-is-xamarin-forms> (дата звернення: 29.10.2023).