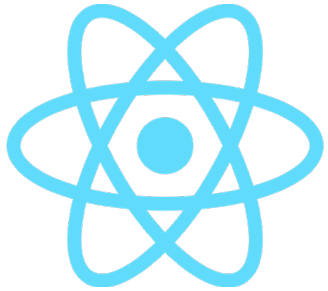




Microsoft Partner
Silver Learning



React Базовий

Робота з API, створення додатка для визначення погоди

React Базовий

Introduction

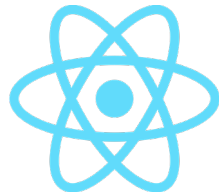


Кочев Денис

Web-розробник в Data Bridge

Магістр комп'ютерних наук Національного аерокосмічного університету ім. М.Є Жуковського

 [denkochev](https://www.linkedin.com/in/denkochev)



React Базовий

Тема уроку

Робота з API, створення додатка для визначення погоди

React Базовий

План уроку

1. Механізми мережевих запитів API
2. Методи API
3. Відповіді від API
4. Де розміщувати мережеві запити в React?
5. Додаткові бібліотеки для API запитів
6. Застосунок для визначення погоди

React Базовий

Механізми мережевих запитів API

Сучасні технології розробки вебзастосунків дозволяють розподілити програмний продукт на Frontend та Backend частини. Бібліотека React - це інструмент розробки Frontend частини вебдодатка тієї частини, з якою користувач взаємодіє безпосередньо. Дана частина вебдодатка є суто клієнтською, тобто вона не має власних ресурсів для збереження файлів на сервер, роботою з БД і таке інше. Для цих цілей розробляється незалежний Backend сервіс, з яким потім за допомогою API взаємодіє клієнтська частина вебдодатку. Механізми вебінтерфейсу API надають інструменти для взаємодії між сервісами. Це можуть бути як і непов'язані між собою сервіси, так і Frontend та Backend частина одного вебзастосунку - такий підхід називається REST API.



React Базовий

Методи API

Вебінтерфейси API мають чотири основних методів для обробки запитів:

- GET – отримання (щоденне оновлення погоди на смартфоні);
- POST – створення (реєстрація на форумі);
- PUT – оновлення (редагування профілю в соціальній мережі);
- DELETE - видалення (видалення акаунту).



React Базовий

Відповіді API

Після обробки мережевого запиту вебінтерфейсом API повинна прийти відповідь. Кожна відповідь має свій статус, який може різнитись в залежності від успішності обробки запиту.

Існує п'ять типів кодів статусу відповіді:

- 1XX - запит в процесі передачі;
- 2XX - запит успішно обробився, метод відпрацював коректно;
- 3XX - перенаправлення (перенаправити користувача з однієї сторінки на іншу);
- 4XX - помилка на клієнтській стороні (неіснуючі параметри в запиті і таке інше);
- 5XX - помилка на стороні серверу (сервер або вимкнений або недоступний по якимсь причинам).

▼ General
Request URL: https://api.openweathermap.org/data/03dc757b8eec2628c1
Request Method: GET
Status Code: 🟢 200 OK
Remote Address: 37.139.20.5:443
Referrer Policy: strict-origin-when-cross-origin

Успішний запит

▼ General
Request URL: https://api.openweathermap.org/data/2.7b8eec2628c1
Request Method: GET
Status Code: 🛑 404 Not Found
Remote Address: 37.139.20.5:443
Referrer Policy: strict-origin-when-cross-origin

Запит на неіснуючий параметр

React Базовий

Де розміщувати мережеві запити в React

До появи хуків більшість компонентів в React-застосунках розроблялись в парадигмі об'єктно-орієнтовного програмування за допомогою класів. В таких компонентах запити до API, які повинні були виконатись відразу після виклику компоненту, розміщувались в методі `componentDidMount`. Даний метод дозволяв виконувати передану в нього функцію відразу після рендеру. Для емуляції даного методу в функціональних компонентах використовується хук `useEffect` з пустим масивом як другий аргумент.

```
useEffect(()=>{  
  fetch('https://jsonplaceholder.typicode.com/users')  
    .then(response => response.json())  
    .then(json => setUsers(json))  
}, []);
```

Приклад використання `useEffect` для встановлення відповіді від API в стейт `users`

React Базовий

Додаткові бібліотеки для API запитів

Нативний JavaScript метод для обробки HTTP-запитів `fetch` неідеальний. Для коректної обробки помилок та стабільного отримання відповідей від API при `fetch` запитах потрібно писати десятки рядків власного коду та додаткових функцій.

Зазвичай в світі розробки програмного забезпечення майже будь-яка проблема, з якою ми можемо зіткнутись, вже була вирішена до нас. Робота з мережевими запитами не є виключенням. На сьогодні існують десятки бібліотек для полегшення роботи з API, самі популярні з них це:

- Axios;
- Superagent;
- Request.

```
npm i axios
```

Встановлення axios в React застосунок

```
import axios from 'axios';
```

Імпорт axios в компонент

React Базовий

Застосунок для визначення погоди

Для розробки додатку визначення погоди скористаємось безкоштовним API від openweathermap. Щоб отримати доступ до даного API, потрібно створити акаунт на порталі <https://openweathermap.org/> та отримати безкоштовний ключ.

Даний сервіс надає можливість розробити застосунок з реальними API запитам, які повертають інформацію щодо погоди у вказаному регіоні або населеному пункті.



Інформаційний відеосервіс для розробників програмного забезпечення



Перевірка знань

TestProvider.com



Перевірте, як ви засвоїли даний матеріал на [TestProvider.com](https://testprovider.com)

TestProvider – це online-сервіс перевірки знань з інформаційних технологій. За його допомогою Ви можете оцінити ваш рівень та виявити слабкі місця. Він буде корисним як у процесі вивчення технології, так і для загальної оцінки знань IT-спеціаліста.

Успішне проходження фінального тестування дозволить вам отримати відповідний сертифікат.

React Базовий

Дякую за увагу! До нових зустрічей!



Кочев Денис
Web-розробник в Data Bridge

