Raport z Laboratorium 4

Programowanie w Chmurze: REST API, JSON, Komunikacja z Serwerem

Mikołaj Kubś
272662

15 maja 2025

Streszczenie

Niniejszy raport podsumowuje realizację zadań w ramach Laboratorium 4, dotyczącego praktycznego wykorzystania REST API, formatu JSON oraz komunikacji klient-serwer. Opracowano dwie aplikacje: klienta API pogodowego oraz system klient-serwer do zarządzania listą albumów muzycznych.

1 Wprowadzenie

Laboratorium miało na celu zapoznanie studentów z fundamentalnymi koncepcjami związanymi z architekturą REST, formatem wymiany danych JSON oraz implementacją komunikacji sieciowej. Zadania obejmowały zarówno konsumpcję zewnętrznego API, jak i tworzenie własnego serwera REST API.

2 Zadanie 1: Aplikacja Pogodowa z OpenWeatherMap

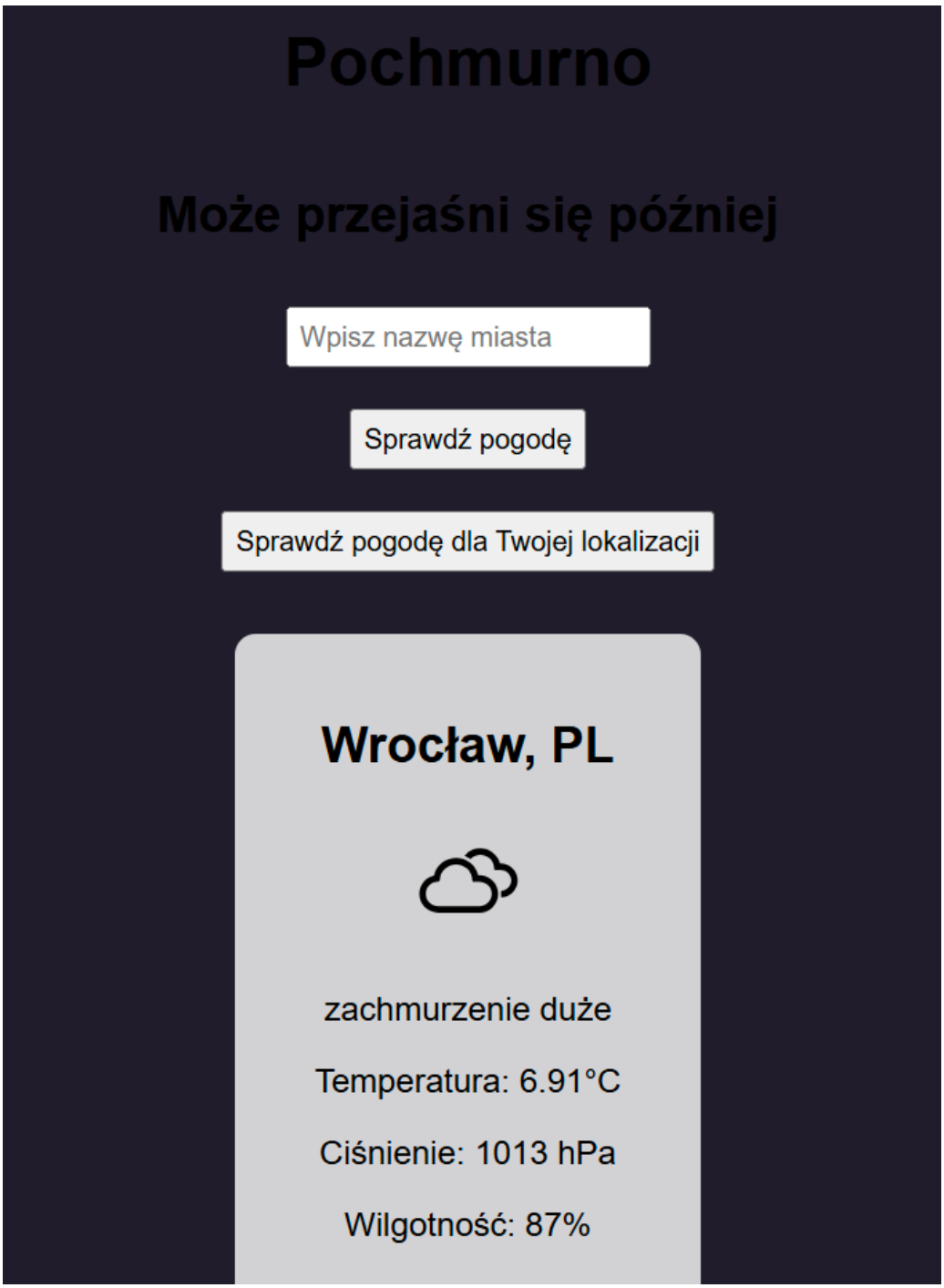
2.1 Cel zadania

Celem zadania było stworzenie aplikacji klienckiej w technologii React, która pobiera i wyświetla dane pogodowe z serwisu OpenWeatherMap dla wskazanej przez użytkownika lokalizacji lub na podstawie geolokalizacji.

2.2 Realizacja

Aplikacja została zaimplementowana zgodnie z wytycznymi. Kluczowe etapy realizacji obejmowały:

- Rejestrację w serwisie OpenWeatherMap i uzyskanie klucza API.
- Stworzenie komponentu React z interfejsem użytkownika umożliwiającym wprowadzenie nazwy miasta.
- Implementację funkcji asynchronicznej do wysyłania żądania GET do API OpenWeatherMap.
- Przetwarzanie odpowiedzi JSON i wyświetlanie danych pogodowych.
- Dodanie obsługi geolokalizacji oraz dynamiczne dostosowanie wyglądu aplikacji (np. tła, ikon) do aktualnych warunków pogodowych.
- Użycie pliku .env do przechowywania sekretów.



Rysunek 1: Interfejs użytkownika aplikacji pogodowej - pogoda dla lokalizacji.

Rysunek 2: Wyświetlanie danych pogodowych dla przykładowego miasta.

3 Zadanie 2: Własny Serwer REST API i Klient

3.1 Cel zadania

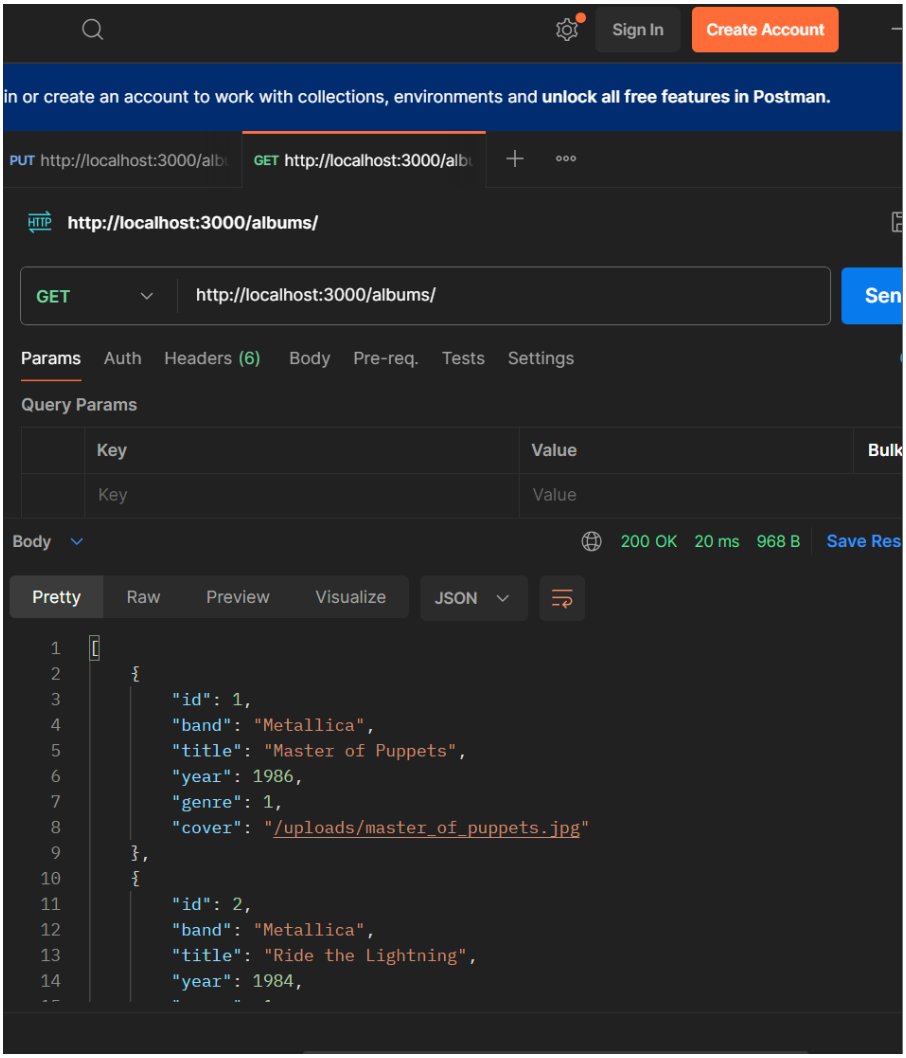
Zadanie polegało na zbudowaniu prostego serwera REST API w Node.js z użyciem frameworka Express, serwującego dane o albumach muzycznych, oraz aplikacji klienckiej w React do interakcji z tym API (operacje CRUD).

3.2 Realizacja - Serwer (Node.js/Express)

Serwer został zaimplementowany w Node.js przy użyciu Express. Główne funkcjonalności serwera:

- Definicja danych (lista albumów) przechowywanych w pamięci serwera.
- Endpointy REST API:

  - `GET /albums` - pobranie wszystkich albumów.
  - `GET /genres` - pobranie wszystkich gatunków.
  - `GET /albums/:band` - pobranie albumów danego zespołu.
  - `POST /albums` - dodanie nowego albumu.
  - `PUT /albums/:id` - aktualizacja danych albumu.
  - `DELETE /albums/:id` - usunięcie albumu.
- Proste testy podstawowej funkcjonalności backendu.
- Dodanie cover i genre - zdjęć okładki przechowywanych na serwerze i zapisywanych po dodaniu z formularza i możliwość wyboru gatunku jako dropdown.



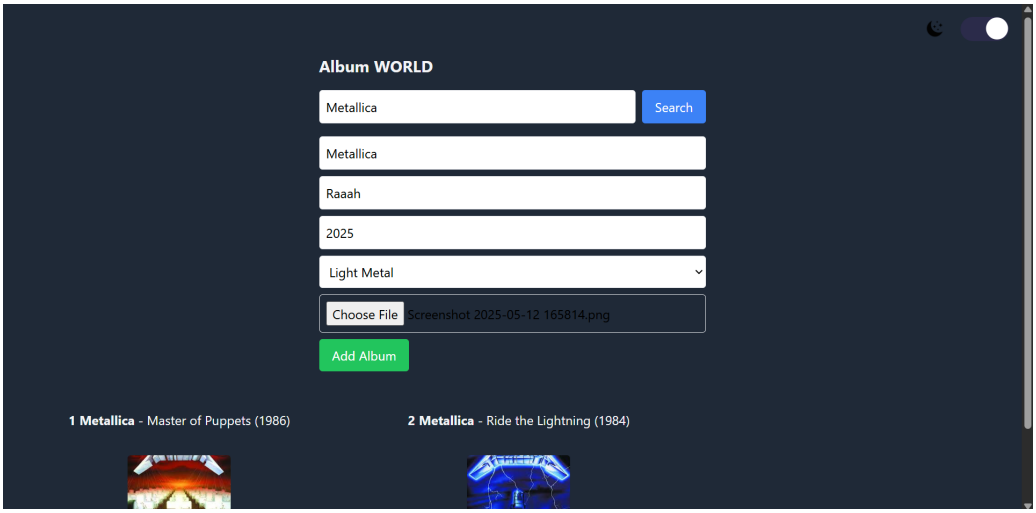
Rysunek 3: Testowanie endpointu serwera API w Postman.

3.3 Realizacja - Klient (React)

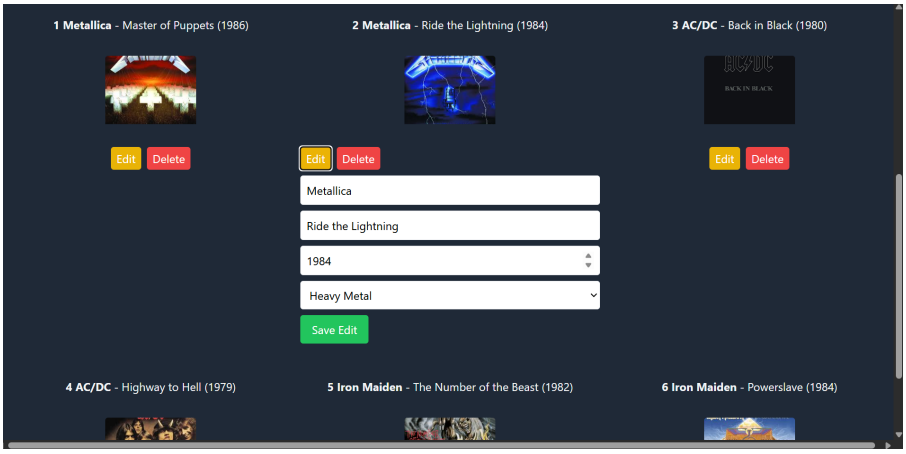
Aplikacja kliencka w React została stworzona do interakcji z serwerem API.

- Implementacja komponentu do wyświetlania listy albumów.
- Funkcje do pobierania, dodawania, edycji i usuwania albumów poprzez wysyłanie żądań HTTP do serwera.
- Interfejs użytkownika umożliwiający zarządzanie kolekcją albumów.

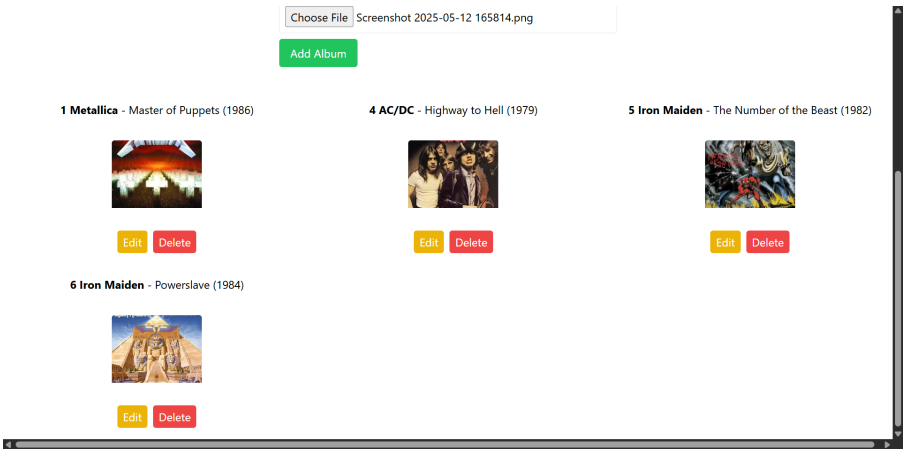
- Wykorzystanie `localStorage` do buforowania danych.
- Dodanie cover i genre - zdjęć okładki przechowywanych na serwerze i zapisywanych po dodaniu z formularza i możliwość wyboru gatunku jako dropdown.
- Jasny i ciemny tryb z custom Switch.



Rysunek 4: Formularz tworzenia albumu.



Rysunek 5: Formularz edycji albumu.



Rysunek 6: Widok aplikacji w trybie jasnym.

4 Podsumowanie

Laboratorium pozwoliło na praktyczne zastosowanie wiedzy dotyczącej REST API i formatu JSON. Zrealizowane zadania umożliwiły przećwiczenie procesu tworzenia aplikacji komunikujących się z serwerami za pomocą protokołu HTTP, zarówno w roli konsumenta istniejącego API, jak i twórcy własnego serwisu REST. Zdobyte umiejętności są kluczowe w nowoczesnym tworzeniu aplikacji webowych i mobilnych.

5 Link do repozytorium

Kod źródłowy aplikacji oraz instrukcje uruchomienia dostępne są w repozytorium GitHub:
`https://github.com/lmProgramming/uni-cloud-development`