

## WeatherController

### 1. Cel:

Kontroler `WeatherController` służy do pobierania aktualnych danych pogodowych dla określonego miasta. Wykorzystuje w tym celu zewnętrzne API `OpenWeatherMap`, które dostarcza informacje o pogodzie, takie jak temperatura, wilgotność czy opis warunków atmosferycznych.

### 2. Jak to działa?

- Użytkownik wysyła żądanie HTTP GET do endpointu `/api/weather/{city}`, gdzie `{city}` to nazwa miasta, dla którego chce uzyskać dane pogodowe (np. London).
- Kontroler odbiera nazwę miasta i wysyła zapytanie do API `OpenWeatherMap`, przekazując nazwę miasta oraz klucz API (`ApiKey`), który jest wymagany do autoryzacji.
- Jeśli miasto istnieje w bazie `OpenWeatherMap`, API zwraca odpowiedź z danymi pogodowymi w formacie JSON.
- Kontroler przetwarza tę odpowiedź, wyciąga najważniejsze informacje (np. temperaturę, wilgotność, opis pogody) i zwraca je użytkownikowi w formie uproszczonego obiektu JSON.
- Jeśli wystąpi błąd (np. miasto nie istnieje lub wystąpił problem z połączeniem), kontroler zwraca odpowiedni komunikat błędu.

### 3. Co zwraca API?

- a. W przypadku pomyślnego żądania, API zwraca obiekt JSON z następującymi danymi:
  - `City`: Nazwa miasta.
  - `Temperature`: Aktualna temperatura w stopniach Celsjusza.
  - `Humidity`: Wilgotność powietrza w procentach.
  - `WeatherDescription`: Krótki opis warunków pogodowych (np. "clear sky", "rainy").
- b. W przypadku błędów, API zwraca odpowiedni komunikat:
  - Jeśli nazwa miasta nie została podana, zwraca błąd `400 Bad Request` z komunikatem: "City name is required."
  - Jeśli miasto nie zostało znalezione w bazie `OpenWeatherMap`, zwraca błąd `404 Not Found` z komunikatem: "City 'XYZ' not found."
  - Jeśli wystąpił błąd po stronie serwera (np. problem z połączeniem do `OpenWeatherMap`), zwraca błąd `500 Internal Server Error` z komunikatem: "An error occurred while retrieving weather data."

#### 4. Co się dzieje w tle?

- a. Kontroler loguje wszystkie ważne zdarzenia, aby ułatwić monitorowanie i debugowanie. Przykłady logowanych zdarzeń:
  - Pomyślne pobranie danych pogodowych dla danego miasta.
  - Błędy żądań do OpenWeatherMap (np. brak połączenia, nieprawidłowa odpowiedź).
  - Nieoczekiwane błędy (np. wyjątki w kodzie).
- b. Logi pomagają zidentyfikować źródło problemów, jeśli coś pójdzie nie tak.

#### 5. Jakie są możliwe błędy?

- a. **400 Bad Request:** Występuje, gdy użytkownik nie poda nazwy miasta w żądaniu.
- b. **404 Not Found:** Występuje, gdy podane miasto nie istnieje w bazie OpenWeatherMap.
- c. **500 Internal Server Error:** Występuje, gdy wystąpi błąd po stronie serwera, np. problem z połączeniem do OpenWeatherMap lub nieoczekiwany wyjątek w kodzie.

### Podsumowanie

API WeatherController działa jako pośrednik między użytkownikiem a OpenWeatherMap. Jego głównym zadaniem jest pobieranie danych pogodowych dla określonego miasta i zwracanie ich w prostym formacie JSON. W przypadku błędów, API zwraca odpowiednie komunikaty, a także loguje zdarzenia, aby ułatwić diagnozowanie problemów. Działa w następujący sposób:

1. Odbiera nazwę miasta od użytkownika.
2. Wysyła zapytanie do OpenWeatherMap.
3. Przetwarza odpowiedź i zwraca dane pogodowe lub komunikat błędu.