

## REST API CLIENT

### SPIS TREŚCI

Spis treści .....	1
Cel zajęć.....	1
Rozpoczęcie .....	1
Uwaga .....	1
Wymagania.....	2
Badanie API .....	2
Implementacja .....	2
Commit projektu do GIT.....	7
Podsumowanie.....	7

### CEL ZAJĘĆ

Celem głównym zajęć jest zdobycie następujących umiejętności:

- pobieranie danych z zewnętrznych zasobów za pomocą REST API
- zdobywanie wiedzy na temat zewnętrznych API za pomocą dokumentacji typu Swagger
- wysyłanie asynchronicznych żądań z wykorzystaniem XMLHttpRequest i Fetch API

W praktycznym wymiarze uczestnicy stworzą dynamiczną stronę HTML pozwalającą na wyświetlanie bieżącej informacji pogodowej oraz prognoz dla zadanej przez użytkownika miejscowości.

### ROZPOCZĘCIE

Rozpoczęcie zajęć. Powtórzenie wykonywania połączeń synchronicznych i asynchronicznych z poziomu JS na stornie.

Wejściówka?

### UWAGA

Ten dokument aktywnie wykorzystuje niestandardowe właściwości. Podobnie jak w LAB A wejdź do Plik -> Informacje -> Właściwości -> Właściwości zaawansowane -> Niestandardowe i zaktualizuj pola. Następnie uruchom ten dokument ponownie lub Ctrl+A -> F9.

## WYMAGANIA

W ramach LAB D przygotowane powinny zostać:

- pojedyncza strona HTML ze skryptem ładowanym z zewnętrznego pliku JS
- pole tekstowe (input typu „text”) do wprowadzania adresu
- przycisk „Pogoda”, po kliknięciu którego wykonywane jest zapytanie asynchroniczne:
  - do API Current Weather: <https://openweathermap.org/current> za pomocą XMLHttpRequest
  - do API 5 day forecast: <https://openweathermap.org/forecast5> za pomocą Fetch API
- obsługa zwrotki z obu API – wypisanie pogody bieżącej oraz prognoz poniżej pola wyszukiwania.

Wygeneruj klucz do API. Ponieważ aktywacja może chwilę potrwać, na czas trwania laboratorium możesz wykorzystać „służbowy” klucz: `7ded80d91f2b280ec979100cc8bbba94`. **UWAGA!** Klucz zostanie dezaktywowany niedługo po zajęciach. Musisz wygenerować swój własny.

W przypadku blokady twórczej można posiłkować się filmem: <https://www.youtube.com/watch?v=WoKp2qDFxKk> jednakże spróbuj rozwiązać ten problem samodzielnie!

Prowadzący omówi powyższe wymagania. Upewnij się, czy wszystko rozumiesz.

Tu umieść swoje notatki:

...notatki...

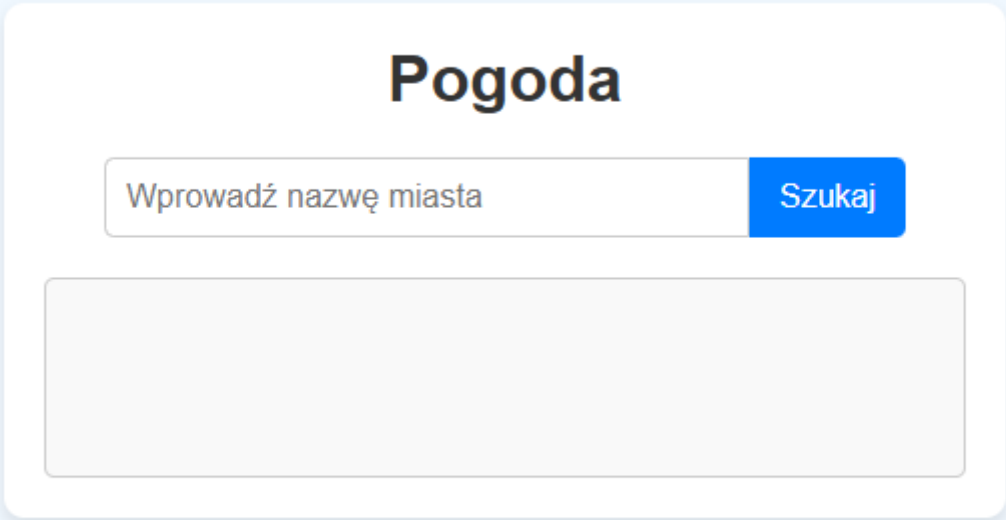
## BADANIE API

Poświęć kilka minut na wykonanie przykładowych zapytań do API z poziomu pasku adresu przeglądarki. Podaj wymagane parametry dla osiągnięcia różnych wyników. Zbadaj odpowiedzi API, aby uzyskać pełen obraz wymagań i możliwości API.

## IMPLEMENTACJA

Tradycyjnie implementację należy zacząć od zbudowania w HTML + CSS wszystkich wymaganych elementów / placeholderów na te elementy. Następnie krok po kroku należy implementować poszczególne zachowania.

Wstaw zrzut ekranu zawierającego stronę ze wszystkimi elementami, tj. pole tekstowe, przycisk, miejsce do wyświetlenia pogody i prognozy:



Punkty:	0	1
---------	---	---

Wstaw zrzut ekranu kodu odpowiedzialnego za wysyłanie żądania do current za pomocą XMLHttpRequest:

```
const xhr :XMLHttpRequest = new XMLHttpRequest();
xhr.open( method: "GET", url, async: true);

xhr.onload = function () :void {
  if (xhr.status === 200) {
    const response = JSON.parse(xhr.responseText);
    console.log("Dane o pogodzie:", response);
  } else {
    console.error("Błąd w żądaniu:", xhr.statusText);
  }
};

xhr.onerror = function () :void {
  console.error("Błąd podczas wysyłania żądania.");
}

xhr.send();
```

Wstaw zrzut ekranu pokazujący otrzymaną odpowiedź za pomocą `console.log()` w przeglądarce.

Odpowiedź z API: ▼ Object 1

- base: "stations"
- ▶ clouds: {all: 0}
- cod: 200
- ▶ coord: {lon: 14.553, lat: 53.4289}
- dt: 1737293338
- id: 3083829
- ▶ main: {temp: -0.92, feels\_like: -3.13, temp\_min: -0.92, temp\_max: -0.92, pressure: 1026, ...}
- name: "Szczecin"
- ▶ sys: {type: 1, id: 1712, country: 'PL', sunrise: 1737270311, sunset: 1737299968}
- timezone: 3600
- visibility: 10000
- ▶ weather: [{...}]
- ▶ wind: {speed: 1.71, deg: 118, gust: 3.83}
- ▶ [[Prototype]]: Object

Punkty:

0

1

Wstaw zrzut ekranu kodu odpowiedzialnego za wysyłanie żądania do forecast za pomocą Fetch:

```
fetch(url) Promise<Response>
  .then(response : Response => {
    if (!response.ok) {
      throw new Error(`HTTP error! status: ${response.status}`);
    }
    return response.json();
  }) Promise<any>
  .then(data => {
    console.log("Prognoza pogody (forecast):", data);
  }) Promise<void>
  .catch(error => {
    console.error("Wystąpił błąd podczas pobierania prognozy pogody (forecast):", error);
  });
}
```

Wstaw zrzut ekranu pokazujący otrzymaną odpowiedź za pomocą `console.log()` w przeglądarce.

```
Aktualna pogoda (current): ▶ {coord: {...}, weather: Array(1), base: 'stations', main: {...}, visibility: 10000, ...}
Prognoza pogody (forecast): ▶ {cod: '200', message: 0, cnt: 40, list: Array(40), city: {...}}
>
```

Punkty:

0

1

Wstaw zrzut ekranu przedstawiającego wizualizację prognoz pogody:

# Pogoda

Szczecin

Szukaj

## Miasto: Szczecin

Temperatura: 3.32°C

Opis: bezchmurnie

### Prognoza pogody na 5 dni:

**19.01.2025 16:00:00**

Temperatura: 3.32°C

Opis: bezchmurnie

**19.01.2025 19:00:00**

Temperatura: 2.97°C

Opis: bezchmurnie

**19.01.2025 22:00:00**

Temperatura: 2.33°C

Opis: bezchmurnie

**20.01.2025 01:00:00**

Temperatura: 1.64°C

Opis: zachmurzenie  
małe

**20.01.2025 04:00:00**

Temperatura: 1.33°C

Opis: pochmurnie

**20.01.2025 07:00:00**

Temperatura: 1.01°C

Opis: bezchmurnie

**20.01.2025 10:00:00**

Temperatura: 3.34°C

Opis: bezchmurnie

**20.01.2025 13:00:00**

Temperatura: 6.28°C

Opis: bezchmurnie

**20.01.2025 16:00:00**

Temperatura: 3.56°C

Opis: pochmurnie

**20.01.2025 19:00:00**

Temperatura: 2.12°C

Opis: zachmurzenie  
umiarkowane

**20.01.2025 22:00:00**

Temperatura: 0.77°C

Opis: zachmurzenie  
duże

**21.01.2025 01:00:00**

Temperatura: -0.51°C

Opis: zachmurzenie  
duże

**21.01.2025 04:00:00**

Temperatura: -0.92°C

Opis: zachmurzenie  
duże

**21.01.2025 07:00:00**

Temperatura: -0.5°C

Opis: zachmurzenie  
duże

Upewnij się, że widoczne są pasek wyszukiwania ze wskazaną miejscowością, a także zarówno pogoda bieżąca jak i prognozy pogody.

Punkty:	0	1
---------	---	---

## COMMIT PROJEKTU DO GIT

Zacommituj i pushnij swoje rozwiązanie do repozytorium GIT.

Upewnij się, czy wszystko dobrze się wysłało. Jeśli tak, to z poziomu przeglądarki utwórz branch o nazwie `lab-d` na podstawie głównej gałęzi kodu.

Podaj link do brancha `lab-d` w swoim repozytorium:  
. <https://github.com/mm51621/main/tree/main/AI/labD>

## PODSUMOWANIE

W kilku zdaniach podsumuj zdobyte podczas tego laboratorium umiejętności.  
Podczas tego laboratorium nauczyłem się, jak korzystać z API OpenWeatherMap do pobierania danych o pogodzie. Zrozumiałem, jak działa wysyłanie żądań HTTP za pomocą zarówno starszego obiektu XMLHttpRequest, jak i nowoczesnej metody fetch. Dowiedziałem się także, jak przetwarzać odpowiedzi z API w formacie JSON oraz wyświetlać je w przeglądarce, tworząc dynamiczne elementy HTML. Dzięki dodaniu parametru `lang=pl` mogłem dostosować język odpowiedzi API do polskiego.

...podsumowanie...

Zweryfikuj kompletność sprawozdania. Utwórz PDF i wyślij w terminie.