

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.....	5
Podsumowanie.....	6

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...

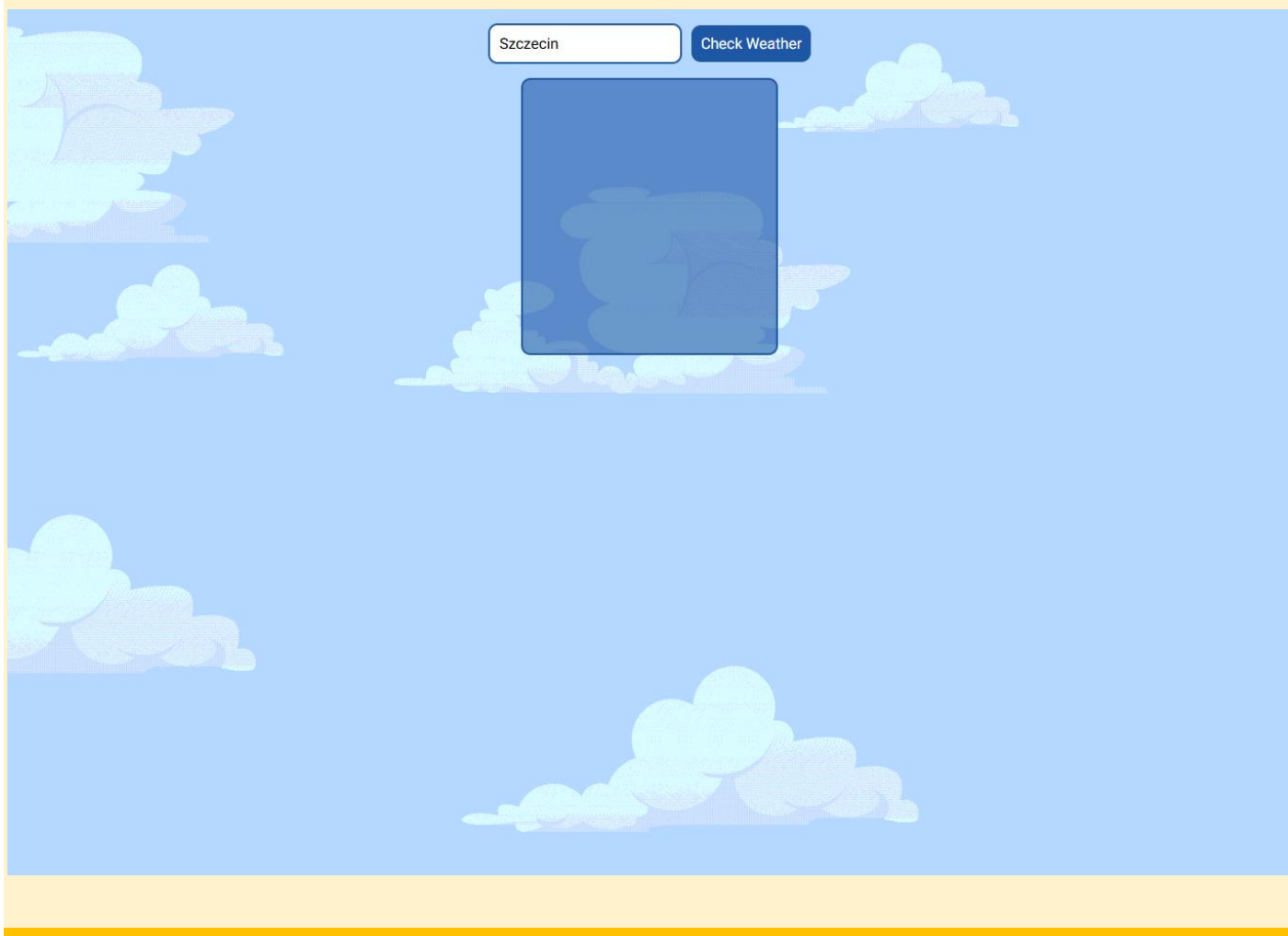
BADANIE API

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:

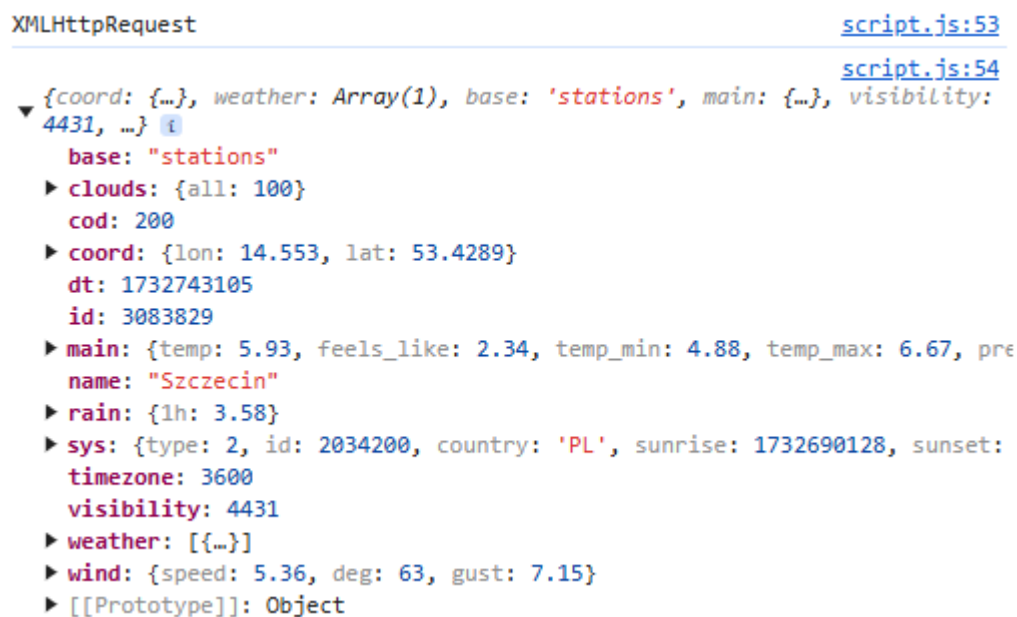
```
let url : string = `https://api.openweathermap.org/data/2.5/weather?q=${city}&units=metric&appid=7ded80d91f2b280ec979100cc8bbba94`

let req : XMLHttpRequest = new XMLHttpRequest();
req.open( method: "GET", url, async: true);

req.addEventListener( type: "load", listener: function(event : ProgressEvent<XMLHttpRequestEventTarget>) : void {

    let weather = JSON.parse(req.responseText);
    console.log("XMLHttpRequest");
    console.log(weather);
}
```

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



The screenshot shows the browser's developer console with the 'XMLHttpRequest' log expanded. It displays a JSON object representing weather data for Szczecin. The object includes coordinates, base station information, clouds, current weather conditions, and a 5-day forecast. The 'main' object shows a temperature of 5.93°C and a name of 'Szczecin'. The 'sys' object shows the country as 'PL' and the sunrise/sunset times.

```
{coord: {...}, weather: Array(1), base: 'stations', main: {...}, visibility: 4431, ...}
  base: "stations"
  clouds: {all: 100}
  cod: 200
  coord: {lon: 14.553, lat: 53.4289}
  dt: 1732743105
  id: 3083829
  main: {temp: 5.93, feels_like: 2.34, temp_min: 4.88, temp_max: 6.67, pressure: 1013, ...}
  name: "Szczecin"
  rain: {1h: 3.58}
  sys: {type: 2, id: 2034200, country: 'PL', sunrise: 1732690128, sunset: 1732743105, ...}
  timezone: 3600
  visibility: 4431
  weather: [{...}]
  wind: {speed: 5.36, deg: 63, gust: 7.15}
  [[Prototype]]: Object
```

Punkty:

0

1

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

```
let url : string = `https://api.openweathermap.org/data/2.5/forecast?q=${city}&units=metric&appid=7ded80d91f2b280ec979100cc8bbba94`;
let weather5 : HTMLElement = document.getElementById( elementId: "weather5Days");
weather5.innerHTML = "";

fetch(url) Promise<Response>
  .then(response : Response => response.json()) Promise<any>
  .then(data => {
    console.log("fetchAPI");
    console.log(data)
  })
```

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

The figure is a weather forecast interface for Los Angeles. At the top, there is a search bar containing 'Los Angeles' and a 'Check Weather' button. Below this is a large blue box for 'Today's weather' showing a clear sky icon, 'Clear Sky', and temperatures: Perceived temp: 20.06°C, Temperature: 20.3°C, Min temperature: 17.2°C, and Max temperature: 23.2°C. Below this are five smaller blue boxes for the following days: 2024-11-28 (Few Clouds), 2024-11-29 (Overcast Clouds), 2024-11-30 (Overcast Clouds), 2024-12-01 (Overcast Clouds), and 2024-12-02 (Broken Clouds). Each box displays the date, weather icon, weather description, and four temperature values: Perceived temp, Temperature, Min temperature, and Max temperature. The background features a large sun and clouds on the left and a large cloud on the right.

Date	Weather	Perceived temp	Temperature	Min temperature	Max temperature
2024-11-28	Few Clouds	19.3°C	19.59°C	17.98°C	19.59°C
2024-11-29	Overcast Clouds	20.34°C	21.74°C	21.74°C	21.74°C
2024-11-30	Overcast Clouds	18.34°C	20°C	20°C	20°C
2024-12-01	Overcast Clouds	18.92°C	20.41°C	20.41°C	20.41°C
2024-12-02	Broken Clouds	19.97°C	21.1°C	21.1°C	21.1°C

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

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/Xomil/Aplikacje-Internetowe-1/tree/main/lab-d>

PODSUMOWANIE

Nauczyłem się pobierać dane przez `fetchAPI` i `httpxmlrequest`

Zweryfikuj kompletność sprawozdania. Utwórz PDF i wyślij w terminie.