

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.....	4
Podsumowanie	5

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:

```
function fetchCurrentWeather(city) :void
{
    const xmlhttp :XMLHttpRequest = new XMLHttpRequest();
    const url :string = `https://api.openweathermap.org/data/2.5/weather?q=${city}&appid=${apiKey}&units=metric&lang=pl`;
```

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



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

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

```
function fetchForecast(city) :void
{
    const url :string = `https://api.openweathermap.org/data/2.5/forecast?q=${city}&appid=${apiKey}&units=metric&lang=pl`;

    fetch(url) .Promise<Response>
        .then(response :Response =>
        {
            if (!response.ok)
            {
                throw new Error('Błąd w pobieraniu prognozy');
            }
            return response.json();
        }) .Promise<any>
        .then(data => displayForecast(data)) .Promise<void>
        .catch(error =>
        {
            forecastDiv.innerHTML = 'Błąd w pobieraniu prognozy';
            console.error(error);
        });
}
```

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

```

> Object { code: "200", message: 0, cnt: 40, list: (40) [_,,], city: {_: -} }
city: Object { id: 3083829, name: "Szczecin", country: "PL", _ }
  > coord: Object { lat: 53.4289, lon: 14.553 }
    country: "PL"
    id: 3083829
    name: "Szczecin"
    population: 407811
    sunrise: 1730786928
    sunset: 1730882118
    timezone: 3600
  > <prototype>: Object { _ }
cnt: 40
code: "200"

▼ list: Array(40) [ [_,,], [_,,], [_,,], _ ]
  > 0: Object { dt: 1730818000, visibility: 10000, pop: 0, _ }
  > 1: Object { dt: 1730829600, visibility: 10000, pop: 0, _ }
  > 2: Object { dt: 1730840800, visibility: 10000, pop: 0, _ }
  > 3: Object { dt: 1730851200, visibility: 10000, pop: 0, _ }
  > 4: Object { dt: 1730862000, visibility: 10000, pop: 0, _ }
  > 5: Object { dt: 1730872800, visibility: 10000, pop: 0, _ }
  > 6: Object { dt: 1730883600, visibility: 10000, pop: 0, _ }
  > 7: Object { dt: 1730894400, visibility: 10000, pop: 0, _ }
  > 8: Object { dt: 1730905200, visibility: 10000, pop: 0, _ }
  > 9: Object { dt: 1730916000, visibility: 10000, pop: 0, _ }
  > 10: Object { dt: 1730926800, visibility: 10000, pop: 0, _ }
  > 11: Object { dt: 1730937600, visibility: 10000, pop: 0, _ }
  > 12: Object { dt: 1730948400, visibility: 10000, pop: 0, _ }
  > 13: Object { dt: 1730959200, visibility: 10000, pop: 0, _ }

```

[app.js:66:21](#)

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

Wstaw zrzut ekranu przedstawiającego wizualizację prognoz pogody:

Pogoda

Sprawdź

Aktualna pogoda w Szczecin

Temperatura: 6.08°C

Warunki: zachmurzenie duże

Prognoza na 5 dni

2024-11-05 15:00:00

Temperatura: 6.08°C

Warunki: zachmurzenie duże

2024-11-05 18:00:00

Temperatura: 6.33°C

Warunki: zachmurzenie duże

2024-11-05 21:00:00

Temperatura: 5.99°C

Warunki: 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/Smokesiren/achastas/tree/main/lab-d>

...link, np. <https://github.com/inazwisko/ai1-lab/tree/lab-d...>

PODSUMOWANIE

W kilku zdaniach podsumuj zdobyte podczas tego laboratorium umiejętności.

...podsumowanie...

Zweryfikuj kompletność sprawozdania. Utwórz PDF i wyślij w terminie.