

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.....	4

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:

Sprawdź pogodę

Pokaż pogodę

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

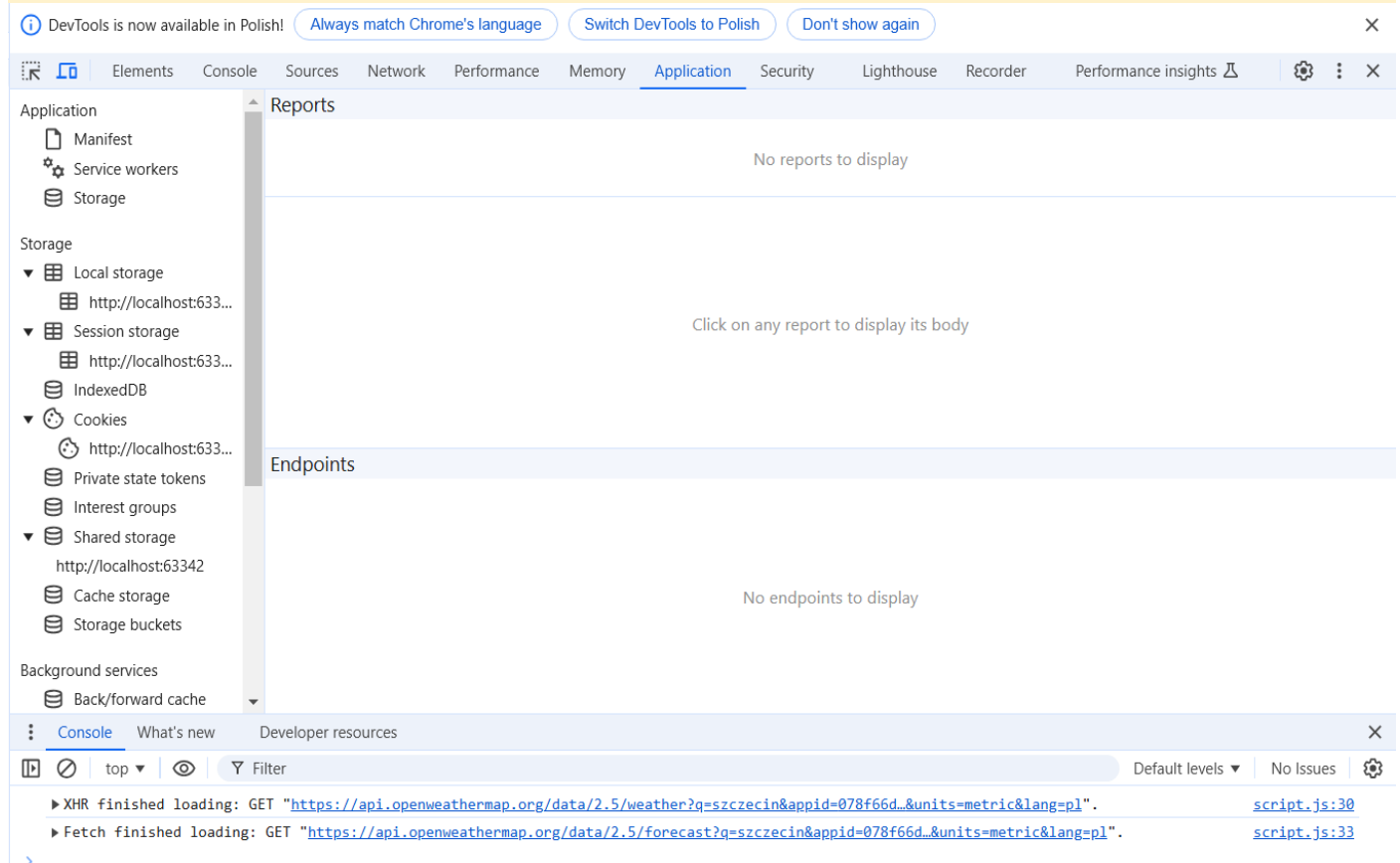
Wstaw zrzut ekranu kodu odpowiedzialnego za wysyłanie żądania do current za pomocą XMLHttpRequest:

```

13     const xhr : XMLHttpRequest = new XMLHttpRequest();
14     xhr.open( method: "GET", currentWeatherUrl, async: true);
15
16     xhr.onload = function () : void {
17         if (xhr.status === 200) {
18             const data = JSON.parse(xhr.responseText);
19             displayCurrentWeather(data);
20         } else {
21             alert("Nie znaleziono miasta lub wystąpił błąd.");
22         }
23     };
24
25     xhr.onerror = function () : void {
26         alert("Wystąpił błąd podczas komunikacji z API.");
27     };
28
29     xhr.send();

```

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

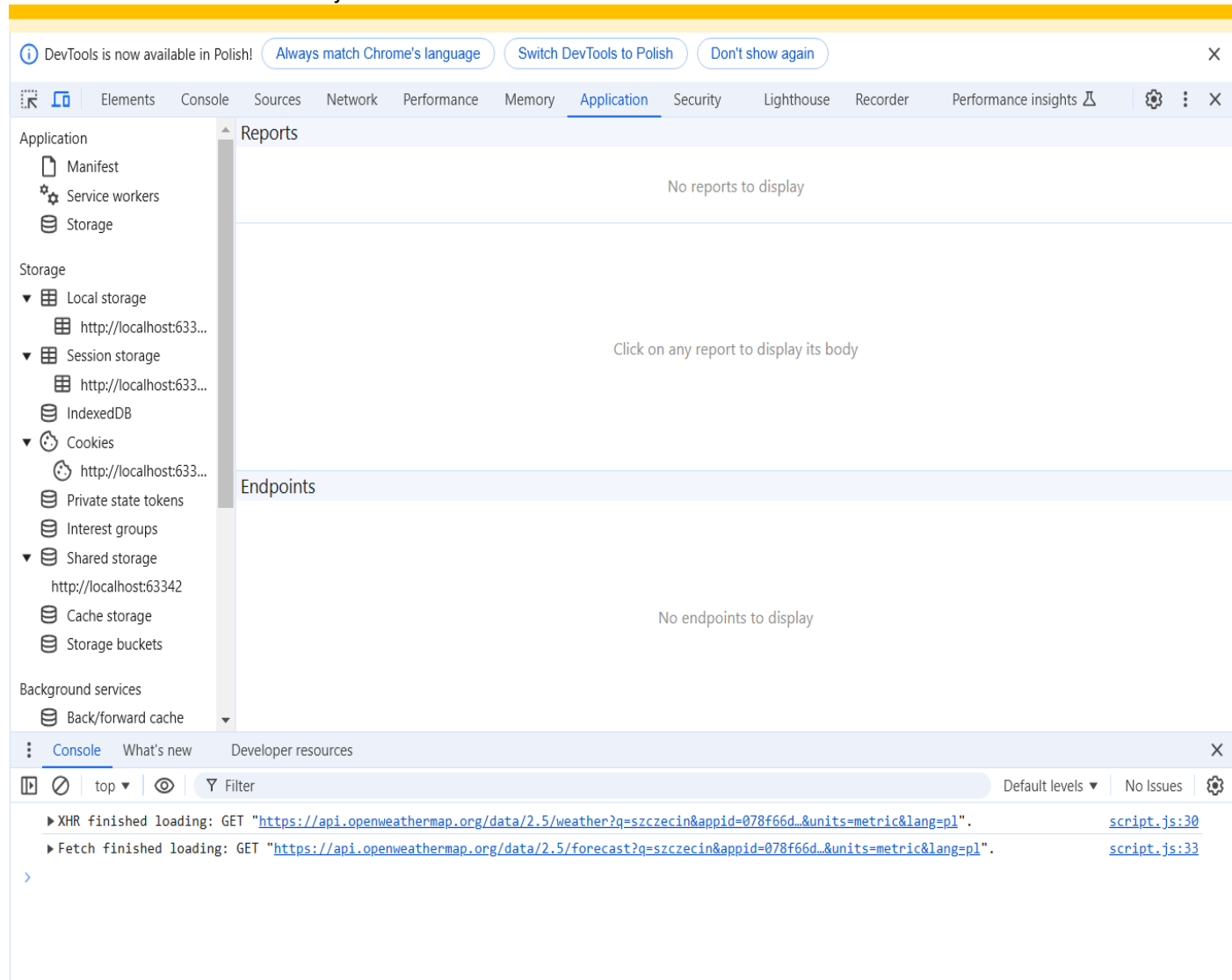


Punkty:	0	1
---------	---	---

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

```
33     fetch(forecastUrl) Promise<Response>
34     .then(response : Response => {
35         if (!response.ok) {
36             throw new Error("Nie udało się pobrać prognozy.");
37         }
38         return response.json();
39     }) Promise<any>
40     .then(data => {
41         displayForecast(data);
42     }) Promise<void>
43     .catch(error => {
44         alert(error.message);
45     });
46 });
```

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



Punkty:

0

1

Wstaw zrzut ekranu przedstawiającego wizualizację prognoz pogody:

Sprawdź pogodę

szczecin

Pokaż
pogodę

Szczecin

Temperatura: 2.57°C

Opis: zachmurzenie małe



- 2024-11-19 Temperatura: 0.62°C opady



śniegu

- 2024-11-20 Temperatura: 0.84°C opady



śniegu

- 2024-11-21 Temperatura: 1.89°C



zachmurzenie małe

- 2024-11-22 Temperatura: 1.83°C



zachmurzenie małe

- 2024-11-23 Temperatura: 2.91°C



zachmurzenie małe

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

Zacommittuj 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/Saggy28/AI1/tree/lab-d>

PODSUMOWANIE

W kilku zdaniach podsumuj zdobyte podczas tego laboratorium umiejętności.

Dowiedziałem się jak wykonywać żądanie za pomocą XMLHttpRequest i Fetch API.

Zweryfikuj kompletność sprawozdania. Utwórz PDF i wyślij w terminie.