

fetch

1. Definicja

Interfejs API Fetch zapewnia interfejs JavaScript do uzyskiwania dostępu i manipulowania częściami potoku HTTP, takimi jak żądania i odpowiedzi. Zapewnia również globalną metodę `fetch()`, która zapewnia łatwy, logiczny sposób asynchronicznego pobierania zasobów w sieci.

Ten rodzaj funkcjonalności został wcześniej osiągnięty za pomocą `XMLHttpRequest`. Fetch stanowi lepszą alternatywę, z której mogą łatwo korzystać inne technologie, takie jak `service workers`. Fetch zapewnia również jedno logiczne miejsce do definiowania innych pojęć związanych z HTTP, takich jak CORS i rozszerzenia HTTP.

2. Wykorzystanie

3. Przykłady z hostowaniem lokalnego pliku

Przypomnij sobie nasze zajęcia z Node'a - hostując naszą aplikację możemy za pomocą `fetcha` mieć dostęp do lokalnego pliku (np. poprzez adres `localhost:3000`)! Utwórz lub skopiuj aplikację z tamtych zajęć i dodaj poniższe kody.

3.1. - przykład 1.

```
fetch("sample.txt")
  .then((res) => res.text()) // Transform the data into text
  .then((data) => {
    console.log(data);
  })
  .catch((err) => console.log(err));
```

W powyższym przykładzie odczytujemy dane z pliku `txt`, stąd w linii 2. korzystamy z metody `text`.

3.2. - przykład 2.

```
fetch("airports_local.json")
  .then((resp) => resp.json()) // Transform the data into json
  .then(function (data) {
    console.log(data);
  });
```

4. Przykłady z hostowaniem zewnętrznego pliku

Wszystkie poniższe przykłady umieść w pliku JS. Podepnij go pod kod HTML (przed `</body>`). Odświeżając stronę zobaczysz efekty (w zależności od zadania to albo w konsoli, albo na stronie internetowej).

4.1. - przykład 1.

Wypisanie w konsoli danych z hostowanego JSONa

Dane z lotniskami w UK: <https://api.jsonbin.io/b/5ec2beca2bb52645e55330ba>

```
fetch("https://api.jsonbin.io/b/5ec2beca2bb52645e55330ba")
  .then((resp) => resp.json()) // Transform the data into json
  .then(function (data) {
    console.log(data);
  });
```

4.2. - przykład 2.

Wypisanie w konsoli nazw z hostowanego JSONa

```
fetch("https://api.jsonbin.io/b/5ec2beca2bb52645e55330ba")
  .then((resp) => resp.json()) // Transform the data into json
  .then(function (data) {
    data.forEach(function (element) {
      console.log(element.name);
    });
  });
```

Utirik Airport	index.js:5
Five Finger CG Heliport	index.js:5
False Island Seaplane Base	index.js:5
Blaine Municipal Airport	index.js:5
Aleknagik / New Airport	index.js:5
East 34th Street Heliport	index.js:5
Fort Bragg Airport	index.js:5
Atkamba Airport	index.js:5
Andorra la Vella Heliport	index.js:5
Andakombe Airport	index.js:5
Areopuerto Internacional Michael Gonzalez	index.js:5
Adareil Airport	index.js:5
Algeciras Heliport	index.js:5
Aseki Airport	index.js:5
Salerno Landing Zone Airport	index.js:5
Afore Airstrip	index.js:5
Afutara Aerodrome	index.js:5
Uru Harbour Airport	index.js:5
Barakoma Airport	index.js:5
Geva Airport	index.js:5
Angoram Airport	index.js:5
Auki Airport	index.js:5
Ballalae Airport	index.js:5
Fera/Maringe Airport	index.js:5
Honiara International Airport	index.js:5
Babanakira Airport	index.js:5
Ngorangora Airport	index.js:5
Santa Cruz/Graciosa Bay/Luova Airport	index.js:5
Munda Airport	index.js:5
Nusatupe Airport	index.js:5
Mono Airport	index.js:5
Parasi Airport	index.js:5

>

4.3. - przykład 3.

Wypisanie danych na stronie internetowej

Kod HTML:

```
<div id="output"></div>
```

Kod JS:

```
fetch("https://api.jsonbin.io/b/5ec2beca2bb52645e55330ba")
  .then((resp) => resp.json()) // Transform the data into json
  .then(function (data) {
    data.forEach(function (element) {
      console.log(element.name);
      document.getElementById(
        "output"
      ).innerHTML += `<div> ${element.name} </div>`;
    });
  });
```

4.4. - przykład 4.

Kod HTML:

```
<button id="btn">Kliknij mnie</button>
<div id="btnOutput"></div>
```

Kod JS:

```
document.getElementById("btn").addEventListener("click", getData);

function getData() {
  fetch("https://api.jsonbin.io/b/5ec2beca2bb52645e55330ba")
    .then((resp) => resp.json()) // Transform the data into json
    .then(function (data) {
      data.forEach(function (element) {
        console.log(element.name);
        document.getElementById(
          "btnOutput"
        ).innerHTML += `<div> ${element.name} </div>`;
      });
    });
}
```

5. - Obsługa błędów

```
fetch('sample.txt')
  .then((res) => res.text())
  .then((data) => {
    document.getElementById('output').innerHTML = data;
  })
  .catch((err) => console.log(err))
```

6. Hostowanie plików

W jaki sposób na zdalnym serwerze hostować plik JSON?

6.1. - sposób 1. - jsonbin.io

Wejdź na stronę: <https://jsonbin.io/>

Wkleić tutaj musisz jedną z opcji:

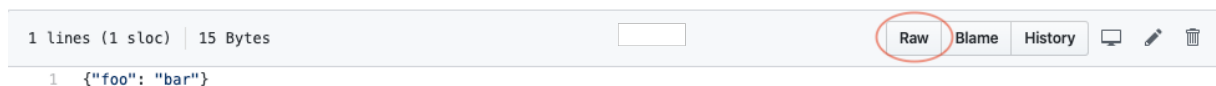
- obiekt
- tablicę obiektów

Mój przykład wyglądał następująco - zobacz plik `airports_local.json` .

Po zalogowaniu się githubem, gdy w JSONie nie wykryto żadnych błędów (o błędach zostaniesz poinformowany, możesz również użyć walidatora JSONa: <https://jsonbin.io/json-validator>) kliknij przycisk `create` i wygenerowany zostanie link (użyty jak w przykładzie powyżej).

6.2. - sposób 2. - github

Na swoim repozytorium wyślij plik JSON. Następnie kliknij w przycisk "Raw" aby uzyskać dostęp do pliku JSON. Skopiuj link z okna URL i wykorzystaj go w swojej aplikacji.



7. Inne

7.1. - iterowanie po tablicy obiektów

Dla przypomnienia: <https://stackoverflow.com/questions/19529403/javascript-loop-through-object-array>