API (Application Programming Interface)

to zestaw reguł, protokołów i narzędzi, które umożliwiają różnym aplikacjom, systemom lub usługom komunikowanie się i wymianę danych między sobą. Działa jako pośrednik, pozwalając jednemu programowi na bezpieczne wykorzystanie funkcji lub danych innego bez znajomości jego wewnętrznej budowy.

```
import { useEffect, useState } from "react";
import './App.css';

function App() {

  const [dogs, setDogs] = useState([]);      // przechowuje dane z API
  const [loading, setLoading] = useState(true); // informacja o ładowaniu
  const [error, setError] = useState(null);     // informacja o błędzie


  useEffect(() => {
    fetch("https://dog.ceo/api/breeds/image/random")
      .then((response) => {
        if (!response.ok) {
          throw new Error("Błąd pobierania danych");
        }
        return response.json();
      })
      .then((data) => {
        setDogs(data);
      })
      .catch((err) => {
        setError(err.message);
      })
```

```
    .finally(() => {
      setLoading(false);
    });
}, []);

const handleLosuj=()=>{
  fetch("https://dog.ceo/api/breeds/image/random")
    .then((response) => {
      if (!response.ok) {
        throw new Error("Błąd pobierania danych");
      }
      return response.json();
    })
    .then((data) => {
      setDogs(data);
    })
    .catch((err) => {
      setError(err.message);
    })
    .finally(() => {
      setLoading(false);
    });
}


if (loading) {
  return <p>Ładowanie...</p>;
}

if (error) {
  return <p>Błąd: {error}</p>;
```

```
  }


  return (
    <div className="App">

      <div className="obrazek">
        <img src={dogs.message}/>
      </div>

      <button onClick={handleLosuj}>Losuj zdjęcie</button>
    </div>
  );
}

export default App;
```

objaśnienia kodu

fetch("https://dog.ceo/api/breeds/image/random")

jest to adres API, z którego chcemy pobrać dane

można też napisać tak:

fetch('https://api.adviceslip.com/advice',{ cache: "no-store"})

wyłączy cache przeglądarki

response.ok

sprawdza, czy status jest od 200 do 299 (w HTTP te statusy oznaczają sukces)

throw new Error("Błąd pobierania danych");

oznacza przerwij działanie kodu i utwórz obiekt błędu

return response.json();

konwertuje dane na obiekt JS

```
.then((data) => {
    setDogs(data);
})
```

Zapisuje dane

```
.catch((err) => {
  setError(err.message);
})
```

Obsługa błędu, jeżeli zostanie wygenerowany błąd w throw to err.mesage to tekst „Błąd pobierania danych”

```
.finally(() => {
  setLoading(false);
});
```

Wykonuje się zawsze, niezależnie czy jest sukces czy błąd

Uwaga

Jeżeli z API dostajemy zagnieżdżone dane, np.


```
{
"slip": {
"id": 123,
"advice": "Always drink water"
}
}
```


I chcemy je odczytać, to trzeba zrobić zabezpieczenie, np.(porady to useState, w którym zapisane dane z API).

```
{porady?.slip?.advice}
```

Przykładowe adresy API

losowe psy

https://dog.ceo/api/breeds/image/random

Dane o krajach (flaga, populacja, stolica, region).

https://restcountries.com/v3.1/all

Filmy ze studia Ghibli.

https://ghibliapi.vercel.app/films

koty

https://api.thecatapi.com/v1/images/search

Produkty: nazwa, cena, obrazek, kategoria

https://fakestoreapi.com/products

Losowego użytkownika (imię, nazwisko, zdjęcie, kraj, email).

https://randomuser.me/api/

Wyniki wyszukiwania książek (tytuł, autor, rok).

https://openlibrary.org/search.json?q=harry+potter

Losowa porada

https://api.adviceslip.com/advice

Aktualną pogodę dla podanych współrzędnych.

https://api.open-meteo.com/v1/forecast?latitude=52.52&longitude=13.41&current_weather=true

Listę użytkowników GitHuba.

https://api.github.com/users