Note :

<https://swapi.dev/>

for dummy data

Sending http requests

* We cannot connect React directly with database server

Because database credentials would be exposed int the browser, performance issue,…

* We have to use backend app like Nodejs, django, …
* React js code can be seen by anyone.
* <https://github.com/axios/axios>
* fetch( ) web API:

use to accessing and manipulating part of http pipeline, such as request and response.

We can  fetch resources asynchronously across the network.

(sending get request)

fetch('http://example.com/movies.json')

.then(response => response.json())

.then(data => console.log(data));

-Here we are fetching json file across network and printing data to console.

-This is the simplest use of fetch function which takes only one argument “the path of resourse we want to fetch. And then return promise containing the response.

-To fetch JSON code from the response we use json( ) method.

* Promise :

The Promise object represents the eventual completion (or failure) of an asynchronous operation and its resulting value.

We can use then( ) on promises.

* Using async function and await :

async function fetchMoviesHandler() {

    const respone = await fetch("https://swapi.dev/api/films/");

    const data = await respone.json();

    const transformedMovies = data.results.map((movieData) => {

      return {

        id: movieData.episode\_id,

        title: movieData.title,

        openingText: movieData.opening\_crawl,

        releaseDate: movieData.release\_date,

      };

    });

    setMovies(transformedMovies);

  }

Get request complete process

import React, { useState, useEffect, useCallback } from 'react';

import MoviesList from './components/MoviesList';

import './App.css';

function App() {

  const [movies, setMovies] = useState([]);

  const [isLoading, setIsLoading] = useState(false);

  const [error, setError] = useState(null);

  const fetchMoviesHandler = useCallback(async () => {

    // set loading true as data fetching started

    setIsLoading(true);

    // set error to null

    setError(null);

    try {

      // get response from link store to response

      const response = await fetch('https://swapi.dev/api/films/');

      if (!response.ok) {

        throw new Error('Something went wrong!');

      }

      const data = await response.json();

      const transformedMovies = data.results.map((movieData) => {

        return {

          id: movieData.episode\_id,

          title: movieData.title,

          openingText: movieData.opening\_crawl,

          releaseDate: movieData.release\_date,

        };

      });

      setMovies(transformedMovies);

    } catch (error) {

      setError(error.message);

    }

    setIsLoading(false);

  }, []);

  useEffect(() => {

    fetchMoviesHandler();

  }, [fetchMoviesHandler]);

  let content = <p>Found no movies.</p>;

  if (movies.length > 0) {

    content = <MoviesList movies={movies} />;

  }

  if (error) {

    content = <p>{error}</p>;

  }

  if (isLoading) {

    content = <p>Loading...</p>;

  }

  return (

    <React.Fragment>

      <section>

        <button onClick={fetchMoviesHandler}>Fetch Movies</button>

      </section>

      <section>{content}</section>

    </React.Fragment>

  );

}

export default App;

* Post request to add data to server:

 async function addMovieHandler(movie) {

    // console.log(movie);

    const response = await fetch('https://react-http-3b3f1-default-rtdb.firebaseio.com/movies.json', {

      method : 'POST',        //by default method is GET

      body : JSON.stringify(movie),

      headers : {

        'Content-Type' : 'application/json'

      }

    });

    // firebase also send data

    const data =await response.json();

    console.log(data);

  }