```
// ================ Connecting a Backend & Database =================
// Sending Http Requests
/*
    API = application program Interface

    1) How do React Apps Interact with Database ?
    2) Sending Http Requests & Using Responses
    3) Handling Errors & Loading State
*/

/*
    Browser-side Apps Don't Directly Talk to Databases

    React App ----------------X ------------ Database (SQL, NoSQL...)

    But by the help of [Backedn App, (Node js App, PHP, Asp.net)] connect the sever and get the data

*/

/*  *************** fetch('url API or server', {pass object Method: "post", header, body chance on this object})
    we can use fetch() method to resuest and send data to in server

        fetch('url').then(response => response.json();).then(data => useData);
*/

// *************************** fetch() data using API *******************

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


const fetchMoviesHandler = () => {

    fetch('https://swapi.dev/api/films/').then(response => {
        return response.json(); // data simple formate json or js object by json() function
    }).then(data => {
        // data tranform acording to ours need change title, id, others things..
        const transformedMovies = data.results.map(movieData => {
            return {
                id: movieData.episode_id,
                title: movieData.title,
                openingText: movieData.opeing_crawl,

            };
        });
        setMovies(transformedMovies);
    });
}

// ***************************** END ***************************

// ======================== async and await ==================

// sometimes we get data After few milliseconds letter that time programm must not block that why we use async and await

// ************************ now code look like this it is better way....
const [isLoading, setIsLoading] = useState(false);

function App() {
    const [movies, setMovies] = useState([]);
    async function fetchMoviesHandler() {

        // when data is coming from server we can show that loading data .................
        // by showing one components as yours wish

        setIsLoading(true);

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

        setIsLoading(false);
        // After loading date make it is false and remove the loding components

    }

    return (
        <React.Fragment>
            {!isLoading && movies.length > 0 && <MovieList movies={moivies} />}
            {isLoading && <p>loading ....</p>}
        </React.Fragment>
    )
}




// ============================== END ===================


// ======================= full code fetch data by API url==================

// **************************** APP.js START ***********************

import React, { useState } from 'react';

import MoviesList from './components/MoviesList';
import './App.css';

function App() {

    const [movies, setMovies] = useState([]);
    /* const dummyMovies = [
       {
         id: 1,
         title: 'Some Dummy Movie',
         openingText: 'This is the opening text of the movie',
         releaseDate: '2021-05-18',
       },
       {
         id: 2,
         title: 'Some Dummy Movie 2',
         openingText: 'This is the second opening text of the movie',
         releaseDate: '2021-05-19',
       },
     ];
    */
    // we can use fetch() method to resuest and send data to in server
    // and fetch return a promise....js term

    const fetchMoviesHandler = () => {
        fetch('https://swapi.dev/api/films/').then(response => {
            return response.json(); // data simple formate json or js object by json() function
        }).then(data => {
            // data tranform acording to ours need
            const transformedMovies = data.results.map(movieData => {
                return {
                    id: movieData.episode_id,
                    title: movieData.title,
                    openingText: movieData.opeing_crawl,

                };
            });
            setMovies(transformedMovies);
        });
    }

    return (
        <React.Fragment>
            <section>
                <button onClick={fetchMoviesHandler}>Fetch Movies</button>
            </section>
            <section>
                <MoviesList movies={movies} />
```

```
            </section>
        </React.Fragment>
    );
}

export default App;


// ************************** END ********************************

// ************************** MoviesList.js START ***********************

import React from 'react';

import Movie from './Movie';
import classes from './MoviesList.module.css';

const MovieList = (props) => {
    return (
        <ul className={classes['movies-list']}>
            {props.movies.map((movie) => (
                <Movie
                    key={movie.id}
                    title={movie.title}
                    releaseDate={movie.release}
                    openingText={movie.openingText}
                />
            ))}
        </ul>
    );
};

export default MovieList;

// ***************************   END ************************************

// ************************** Movie.js ********************************

import React from 'react';

import classes from './Movie.module.css';

const Movie = (props) => {
    return (
        <li className={classes.movie}>
            <h2>{props.title}</h2>
            <h3>{props.releaseDate}</h3>
            <p>{props.openingText}</p>
        </li>
    );
};

export default Movie;

// ******************************** END ********************************




// ====================== Handling Http Errors ================================

function App() {
    const [movies, setMovies] = useState([]);
    const [isLoading, setIsLoading] = useState(false);
    const [error, setError] = useState(null);

    async function fetchMoviesHandler() {

        setIsLoading(true);
        // after running the if it has some error that will remove by calling that...below setError(null);
        setError(null);

        // when we use async and await the to handle the error by using => try {} catch{} way

        try {
            // try some code and find error or generating error and use to handel that...below
```

```
228
229             const response = await fetch('https://swapi.dev/api/films');
230
231             if(!response.ok) {
232                 throw new Error('somthing wrong there');
233             }
234
235             const data = response.json();
236
237             // we are actually generating error .. if the response not ok then we have to handel this by won your won
238
239             // some API will provide some json after the error ocured then .. we have to manage the After the data or berore the data get ....
240
241             /*
242             if(!response.ok) {
243                 throw new Error('somthing wrong there');
244             }
245             */
246
247             const transformedMovies = data.results.map(movieData => {
248                 return {
249                     id: movieData.episode_id,
250                     title: movieData.title,
251                     release: movieData.releaseDate,
252                 };
253             });
254             setMovies(transformedMovies);
255
256             setIsLoading(false);
257         }
258         catch (error){
259             setError(error.message);
260         }
261
262
263     }
264
265     return (
266
267         <React.Fragment>
268             {!isLoading && movies.length > 0 && <MoviesList movies={movies} />}
269             { isLoading && <p>loading....</p> }
270             { !isLoading && error && <p>{error}</p>}
271         </React.Fragment>
272     );
273 }
274
275
276 // ====================== END ========================
277
278 // ***************** using variables components using *****************
279
280     function App(){
281
282         let content = <p>Found no data</p>;
283
284         if(movies.length > 0){
285             content = <MoviesList movies={movies}/>;
286         }
287
288         if(error){
289             content = <p>{error}</p>;
290         }
291
292         if(isLoading){
293             content = <p>Loading...</p>;
294         }
295
296
297         return (
298             <React.Fragment>
299                 {content}
300             </React.Fragment>
301         );
302     }
303
```

```
//************************    END    *********************************

// ====================== useEffect() and callBack() ========================

    // in some Applicaton when user Enter that automitically loading data form database
    // that time we can use useEffect() hook to get data imidiately from database

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

    function App() {
        const [movies, setMovies] = useState([]);
        const [isLoading, setIsLoading] = useState(false);
        const [error, setError] = useState(null);

        // ************************** here callBack ******************

        // callBackk theke the hole function becase we want to make inside only one reference of that fetchMoviesHandlere function...

       const fetchMoviesHandler =  callBack(async ()=> {

            setIsLoading(true);
            setError(null);

            try {

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

                if(!response.ok) {
                    throw new Error('somthing wrong there');
                }

                const data = response.json();

                const transformedMovies = data.results.map(movieData => {
                    return {
                        id: movieData.episode_id,
                        title: movieData.title,
                        release: movieData.releaseDate,
                    };
                });
                setMovies(transformedMovies);

                setIsLoading(false);
            }
            catch (error){
                setError(error.message);
            }
        });


       // if we use A hole function as a dependence then it will create infinity looop cos function is a object it sometime some value will bechnage ..
       // components will call again and gain....that why we give the function inside the calback() to create only one reference of that function ....
        useEffect(()=>{
            fetchMoviesHandler();
        }, [fetchMoviesHandler]);


        return (

            <React.Fragment>
                {!isLoading && movies.length > 0 && <MoviesList movies={movies} />}
                { isLoading && <p>loading....</p> }
                { !isLoading && error && <p>{error}</p>}
            </React.Fragment>
        );
    }

// ========================= END ================================


// ====================== get and post on firebase Realtime server ================

/*
    1) create database
    2) name the datebase what you want
```

```
    3) select realtime database
    4) copy the link (rest API)
    5) fetch('https://react-http-2b87c-default-rtdb.firebaseio.com/AddFoder.json');
*/



// ********************** Post inside the database **********************

async function addMovieHandler() {
  const response = await fetch('https://react-http-2b87c-default-rtdb.firebaseio.com/AddFoder.json', {

    // by default method: 'GET',

    method: 'POST',
    body: JSNO.stringify(movie),
    header: {
        'content-Type':'application/json',
    }
    });

    const data = await response.json();
    console.log(data);
}


// ******************* get data form server

// when firebase feadback the data that will a object....now we have to moifiie like this way ...so that we can show the data....
// on our website.... how???

const fetchMoviesHandler =  callBack(async ()=> {

    setIsLoading(true);
    setError(null);

    try {

        const response = await fetch('https://react-http-2b87c-default-rtdb.firebaseio.com/AddFoder.json');

        if(!response.ok) {
            throw new Error('somthing wrong there');
        }

        const data = response.json();

        const loadedMovies = [];

        for(const key in data){
            loadedMovies.push({
                id: key,
                title: data[key].title,
                openingText: data[key].openingText,
                releaseDate: data[key].releaseDate,
            });
        }


         setMovies(loadedMovies);

        setIsLoading(false);
    }
    catch (error){
        setError(error.message);
    }
});



useEffect(()=>{
    fetchMoviesHandler();
}, [fetchMoviesHandler]);
```