First take the API (Application Programming Interface) key and test it in Thunder Client or Postman

How to fetch data from API ?

In your main Component file fetch the API

const getRequest = async () => {

const url = “API”

// #1 When the fetch request occurs means when the data is pulled from the API through fetch() we want to store inside our response variable, now from here all our API data will be inside response variable

// #2 Now why are using this async and await here because if you will just fetch the url then you will be getting Promises which might be in pending state, To move that Promise from pending state to normal state we are using await

const response = await fetch(url);

//Now http response data which was fetched from the API we want to be get it converted into JSON, so therefore we are using .json() and storing the JSON converted data into responseJson variable

const responseJson = await response.json();

console.log(responseJson); //You will be able to see all our json converted data into our browser console

setMovies(responseJson.NameoftheArrayWhichYouWanttoFetchFromApi) //Storing the data to display through UseState Hooks on Front End

}

Now after this call your API request function for this we will be using useEffect() hook in react Js

//This will fire our API whenever our page gets load

useEffect(()=>{

getRequest();

},[] );

Now after fetching the data how to loop all these data in the JSX component ?

Now after fetching the API we have to include this data into our useState Hooks so that we can pass this data to another component using Props

const [movies, Setmovies] = useState([]); //Don’t forget to call this Setmovies from where your Updated data is coming

//movies is the data which will be displayed

//inside our useState hook there will be shown the initial data which is currently empty

//Setmovies is the updated data coming from API which is moved into movies which will be that the data to be displayed

Now create a component where you will be going to loop this fetched data

// Consider the name of your component is MovieList

//Pass the Props into your component Props helps us to transfer data into components from hooks

const MovieList = (props) => {

<>

{props.movies.map((movie,index) => (

<div>

<h1>{movie.name}<h1/>

<div/>

))}

</>

}

Now add this components inside your file where you have fetched the API

{MovieList movies={movies}/}

How to Create Search Bar which is searching live data from the API ?

Create a useState Hook for Search functionalities in our App.js File

const App=()=>{

const [searchValue, setSearchValue] = useState()

}

Now create a Search Component and import it in our App.js file

<SearchBox/>

Now we have to Store the search entered by user searchValue Parameter, anytime the seachValue parameter changes we want to call getRequest function , getRequest function will help us to fetch the data from the API as we know

Now, for doing storing the search value in searchValue Parameter we need to do below task

STEP 1

Const SearchBox = (props) =>{

return {

<input value={props.value}

onChange = {(event)=>props.setSearchValue(event.target.value)}

}

}

STEP 2

<SearchBox searchValue={searchValue} setSearchValue={setSearchValue}

🔴 Create a function SearchXYZ and creare a useState hook inside it (remember that you have created separate J S file for Search Bar Component)

Const SearchXYZ = () => {

const[search,setSearch]=useState(“”); //currently keep it empty

}

1] Add value={search}

onChange={(e)=>setSearch(e.target.value.toLowerCase())}

inside the search input tag

2] Add onClick={handleSearch} on our button

3] Create a function handleSearch to handle Search functionalities logic inside our SearchXYZ function and below our useState

Const SearchXYZ = () => {

const[search,setSearch]=useState(“”); //currently keep it empty

const handleSearch = async ()=>{

if(search){

const XYZdata = await fetechdata(); //this fetchdata is another function which is fetching data from the API and storing it inside const XYZdata variable.

}

}

4] For creating the fetechdata function create an another folder 📂 named Utils in src and create a file fetchdata.js. This is where we are going to make the connection with rapid api

export const fetchdata = async (url, options) =>{

// #1 When the fetch request occurs means when the data is pulled from the API through fetch() we want to store inside our response variable, now from here all our API data will be inside response variable

// #2 Now why are using this async and await here because if you will just fetch the url then you will be getting Promises which might be in pending state, To move that Promise from pending state to normal state we are using await

const response = await fetch(url,options);

//Now http response data was fetched from the API we want to be get it converted into JSON, so therefore we are using .json() and storing the JSON converted data into responseJson variable

const responseJson = await response.json();

return responseJson;

}

5] Now after creating the fetchdata functions copy options from Rapid API and paste it above fetchdata function

export const XYZOptions = {

method:’’

headers:{

X-RapidAPI-Host : ‘…’,

X-RapidAPI-Key: ‘…’

}

}

Now copy the X-RapidAPI-Key🗝 from XYZOptions and store secretly🔐 in .env file (create this file in global folder📂)

REACT\_APP\_RAPID\_API\_KEY = …key…

After doing this stop the terminal and re-run the terminal and now our key will be populated in our project now the REACT\_APP\_RAPID\_API\_KEY is our secret word for our api key which we will be using in place of our actual rapid api key in our XYZOptions function

export const XYZOptions = {

method:’’

headers:{

X-RapidAPI-Host : ‘…’,

X-RapidAPI-Key: process.env.REACT\_APP\_RAPID\_API\_KEY

}

}

6] Now move back to Search Component file and import this fetchdata function and XYZOptions function

Import {XYZOptions, fetchdata} from …

7]

Const SearchXYZ = () => {

const[search,setSearch]=useState(“”);

const[ XYZ,setXYZ] = useState(“”); *⚽*

const handleSearch = async ()=>{

if(search){

const XYZdata = await fetechdata( ‘url’, XYZOptions ); //modify url according to your search need

console.log(XYZdata);

//Now go to our site search bar and search something if our data get displayed in our console then 🎊 😎

//APPLYING FILTER [ create a useState hook for applying filter ⚽ ]

Consider the example of the excerciseAPI

const searchedXYZ = XYZData.filter(

(XYZ) => XYZ.name.toLowerCase().includes(search)

|| XYZ.target.toLowerCase().includes(search)

|| XYZ.equipment.toLowerCase().includes(search)

|| XYZ.bodyPart.toLowerCase().includes(search)

*Target, name, equipment, bodypart, equipment are the keys of api json objects*

);

setSearch(‘’); //initially set our Search as empty string

setXYZ(searchedXYZ); //set our XYZ data to the searched data

}

}

useEffect hook is use to display data whenever the page loads

useEffect(()=> {

}

),[]) //if this array is kept empty, it is also used to display the data whenever the page id updates ,

Whenever we fetch data using function we pass that function into the updating parameter ex. Setdata of useState hook after that remember when to call that function , for example after button click, on page refresh