GET API:

Get data from server: returns all data in the form of text

async function getData(){

    const response = await fetch("https://jsonplaceholder.typicode.com/todos/1");//fetch data, await use to get resolve response

    //above we got the data, and to convert to json, we need await response.json();

    console.log("response = ", response);

    const data = await response.json();//converting to json

    console.log("data", data)

    //so by this, this api contains two promises, one for fetching the data and other to convert that data

    //into json

}

getData();

POST API:

Used to post data on the server, all data which you are sending is hidden in the body

This is the main data which we want to sent which is an object or in json format

        body: {

            title:"My info",

            body: 'Data',

            userId: '104'

        }

We cannot sent in json format directly, so we need to convert it into string with the help of JSONG.stringify

        body: JSON.stringify({

            title:"My info",

            body: 'Data',

            userId: 104,

        }),

Now to pass content type, we use headers, and pass json type in it

        headers: {

            "Content-type" : "application/json; charset=UTF-8",

        },

"application/json-> this is telling that that sent is of json type

This is the complete code

async function postData(){

    const response = await fetch('https://jsonplaceholder.typicode.com/posts',{

        method: 'POST',//post api type, we can also pass get here.

        body: JSON.stringify({

            title:"My info",

            body: 'Data',

            userId: 104,

        }), //This is the main data which is in json data which is converted into string with the help of JSON.stringify

        headers: {

            "Content-type" : "application/json; charset=UTF-8",

        },//header is used to pass content type which is json as mentioned with application/json

    });

    console.log("response = ", response);

    const data = await response.json();//converting to json

    console.log("data", data)

}

//getData();

postData();

POSTMAN:

Postman was initially extension, but now a separate framework

It is used to verify API and its data. The best thing is that instead of writing whole application or whole fetch code, we can directly fetch data through postman. We just need to select type of API like GET, POST etc and paste in the URL. So by doing that we can verify that the API from which we need to fetch data or sent data is working fine or not. And then we can do rest with code if needed.

This is the method of fetching data in JS. Now lets learn how to fetch data in react. So from now, lets continue in folder react-fetch