Day 6 : 19 Nov 2024

Access end point or rest api using react js

Using backend technologies like Django with python, spring boot, asp.net, express js with node js we can create rest full web service.

Web Service : Giving the service for web application when both the application running using different technologies or languages.

Amazon --🡪

Django payment --🡪 Gpay java

XML Phone Pay python with Django

JSON Paypal asp.net

Paytm php

Net banking express js

Credit card

SOAP web Service : Simple object access protocol. SOA (Service Oriented Architecture). In SOAP web service we need to consume and produce data only in the form of xml. eXtensible markup of language.

Rest full web service: IN Rest full web service we can consume and produce the data any other format base upon application requirement like XML or JSON(JavaScript Object Notation), html, plain text or any media type.

As front end developer we need to learn how to consume rest api or rest full web service develop using any language.

Next JS : Using Next JS we can create server side component.

According Rest api we need to use Http protocol methods.

1. Get method : Get the resources like Product, customer, order, employee etc.
2. Post method : this method is use to create the resource. Insert the record in database.
3. Put method : update existing recourse : update employee salary using id.
4. Delete method : this method use to delete the resource delete employee information using id.

In React we can consume rest api

1. Using fetch() function: fetch is pure JavaScript function which help to call rest api.
2. Using axios library : it is a third party library which internally use fetch context to access rest api.

Fetch and axios method return type of promise. Promise is use to handle asynchronous event of data.

ES5 style . if promise resolve then will call. If promise rejected catch will call.

ES6 style : we can use asyn and await keyword to handle asynchronous data.

create-react-app react-rest-api

useEffect(): this hook is use to do side effect whenever the component get loaded…

syntax :

useEffect(callback,[dependencies]) : if any dependencies get change then only execute else don’t execute.

When component get loaded it will call once initially and whenever any dependencies change then it will call.