**React – Json-Server And Firebase Real Time Database**

**• Question : What is Json-Server? How we use in React ?**

**✅ What is json-server**

**json-server** is a lightweight **mock REST API server** that uses a simple **JSON file as a database**.

It allows you to quickly create a **fake backend** to test and develop frontend applications (like React) **without setting up a real server**.

### 🔧 ****Why Use**** json-server****?****

* No backend setup needed
* Ideal for quick prototyping and front-end development
* Provides full **CRUD** support (GET, POST, PUT, DELETE)
* Easy to use and install

**Question : How do you fetch data from a Json-server API in React? Explain the role of fetch() or axios()in making API requests**

### ✅ ****How to Fetch Data from a**** json-server ****API in React****

To fetch data from a json-server API in React, you typically use **either**:

* The native **fetch()** method (built into browsers), or
* An external library like **axios** (more powerful and easier to use)

### 🔁 ****Basic Steps:****

1. Start your json-server:

bash

CopyEdit

json-server --watch db.json --port 3001

1. Use fetch() or axios() in a React component to get data from http://localhost:3001.

## 📌 Using fetch() Example:

jsx

CopyEdit

import { useEffect, useState } from 'react';

function UserList() {

const [users, setUsers] = useState([]);

useEffect(() => {

fetch('http://localhost:3001/users')

.then(response => response.json()) // convert response to JSON

.then(data => setUsers(data)) // update state

.catch(error => console.error('Error:', error));

}, []);

return (

<ul>

{users.map(user => <li key={user.id}>{user.name}</li>)}

</ul>

);

}

## 📌 Using axios() Example:

First, install axios:

bash

CopyEdit

npm install axios

Then in your component:

jsx

CopyEdit

import { useEffect, useState } from 'react';

import axios from 'axios';

function UserList() {

const [users, setUsers] = useState([]);

useEffect(() => {

axios.get('http://localhost:3001/users')

.then(response => setUsers(response.data))

.catch(error => console.error('Error:', error));

}, []);

return (

<ul>

{users.map(user => <li key={user.id}>{user.name}</li>)}

</ul>

);

}

 Use fetch() or axios() to **get data from json-server** in React.

 Place API calls inside useEffect() to fetch data **when the component mounts**.

 Store the response using useState() to render it in the UI.

**Question : What is Firebase? What features does Firebase offer?**

### ✅ ****What is Firebase?****

**Firebase** is a **platform developed by Google** that offers a collection of **cloud-based tools and services** to help developers build, manage, and scale **web and mobile applications** quickly and efficiently—**without managing server infrastructure**.

It acts as a **Backend-as-a-Service (BaaS)**.

### 🚀 ****Why Use Firebase?****

* ✅ No need to build your own backend
* ✅ Easy integration with React, Android, and iOS
* ✅ Real-time updates for chat apps, dashboards, etc.
* ✅ Scalable and secure
* ✅ Great for MVPs and production-ready apps

**Question : Discuss the importance of handling errors and loading states when working withAPIs in React**

### ✅ ****Why Handle Loading States?****

* When you fetch data, it takes time for the request to complete.
* Showing a **loading indicator** (like a spinner or message) lets users know something is happening.
* Prevents the UI from looking broken or empty.
* Improves user experience by giving feedback.

### ✅ ****Why Handle Error States?****

* API requests can **fail** due to network issues, server problems, or bad data.
* Showing an **error message** helps users understand what went wrong.
* Prevents the app from crashing or freezing.
* Allows you to gracefully recover or retry the request.

### ****How to Handle Loading and Error States in React****

Use useState to keep track of:

* loading (boolean): Is data loading?
* error (string or null): Is there an error message?
* data (API response): The fetched data