# React FetchUserApp - API Integration

**Explain how to consume REST APIs from React applications.**

Consuming REST APIs in React involves sending HTTP requests to an external server and using the received data to dynamically update the UI. This is commonly done using JavaScript methods like fetch(), or libraries such as axios. These requests are usually made from within React components — either functional or class-based — typically during component mounting using useEffect (for functional components) or componentDidMount (for class components).

When a component loads, it triggers an API call (e.g., to fetch user data, product lists, etc.). Using fetch, a GET request is made to the API endpoint, and the response is parsed using .json(). The result is stored in the component's state using useState or this.setState, and React re-renders the UI with the updated data. Error handling is also important — catching exceptions helps display fallback messages or retry options when something goes wrong.

To keep code clean and modular, it's good practice to abstract API calls into separate services or hooks. For example, you might create a userService.js file to handle all user-related API logic. This improves maintainability and testing. Additionally, developers often use tools like Axios for easier configuration of headers, timeouts, and interceptors. Libraries like SWR or React Query also simplify API state management with built-in caching, loading states, and refetching mechanisms.

# FetchUserApp - Full Working Code

## 1. App.js

import React from "react";  
import Getuser from "./Getuser";  
  
function App() {  
 return (  
 <div>  
 <Getuser />  
 </div>  
 );  
}  
  
export default App;

## 2. Getuser.js

import React, { Component } from "react";  
  
class Getuser extends Component {  
 constructor() {  
 super();  
 this.state = {  
 user: null,  
 };  
 }  
  
 async componentDidMount() {  
 try {  
 const response = await fetch("https://api.randomuser.me/");  
 const data = await response.json();  
 this.setState({ user: data.results[0] });  
 } catch (error) {  
 console.error("Failed to fetch user data:", error);  
 }  
 }  
  
 render() {  
 const { user } = this.state;  
  
 if (!user) {  
 return <h2>Loading...</h2>;  
 }  
  
 return (  
 <div style={{ textAlign: "center", marginTop: "50px" }}>  
 <h2>  
 {user.name.title} {user.name.first} {user.name.last}  
 </h2>  
 <img  
 src={user.picture.large}  
 alt="User"  
 style={{ borderRadius: "8px", boxShadow: "0 0 5px #aaa" }}  
 />  
 </div>  
 );  
 }  
}  
  
export default Getuser;

## Output

