

**Answer**

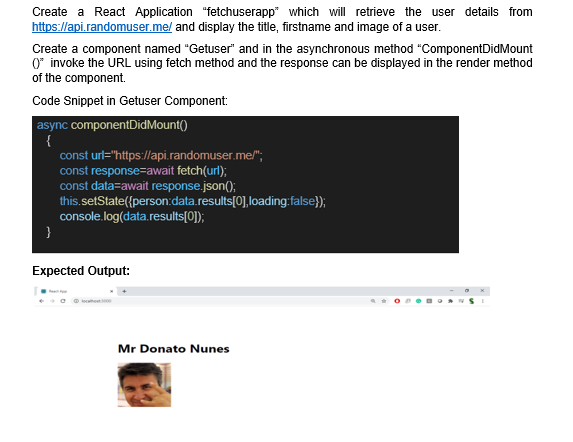
Consuming REST APIs from React applications is a fundamental task for building dynamic, data-driven user interfaces. The process involves making an HTTP request to an external data source and then using the retrieved data to update the component's state and render the UI. This is typically done within a component's lifecycle, and there are several ways to accomplish this:

1. Using the fetch API

The fetch API is a built-in browser function that provides a promise-based interface for making network requests. It is the most straightforward method for basic data fetching without external dependencies. The request and response cycle typically takes place within the useEffect Hook to manage the component's lifecycle.

2. Using a third-party library like Axios

Axios is a popular, promise-based HTTP client that provides a more robust and streamlined approach to making API requests. It offers features such as automatic JSON parsing, better error handling, and a cleaner syntax.



**Solution**

**Getuser.js**

import React, { Component } from 'react';

class Getuser extends Component {

constructor() {

super();

this.state = {

person: null,

loading: true,

};

}

async componentDidMount() {

const url = "https://api.randomuser.me/";

const response = await fetch(url);

const data = await response.json();

this.setState({ person: data.results[0], loading: false });

console.log(data.results[0]);

}

render() {

if (this.state.loading) {

return <div>Loading...</div>;

}

if (!this.state.person) {

return <div>No person found.</div>;

}

const { title, first } = this.state.person.name;

const { medium } = this.state.person.picture;

return (

<div>

<h1>{title} {first}</h1>

<img src={medium} alt="User" />

</div>

);

}

}

export default Getuser;

**App.js**

//import logo from './logo.svg';

import './App.css';

import Getuser from './Getuser';

function App() {

return (

<div className="App">

<Getuser />

</div>

);

}

export default App;

**App.css**

.App {

text-align: center;

margin-top: 50px;

}

h1 {

font-size: 36px;

font-weight: bold;

}

img {

width: 200px;

height: 200px;

border-radius: 10px;

margin-top: 20px;

}

**Output**



