

Hands-on 17

Objectives

- Explain how to consume REST APIs from React applications

In this hands-on lab, you will learn how to:

- Construct a React application that invokes the REST API and fetch data from the API

Prerequisites

The following is required to complete this hands-on lab:

- Node.js
- NPM
- Visual Studio Code

Notes

Estimated time to complete this lab: **60 minutes.**

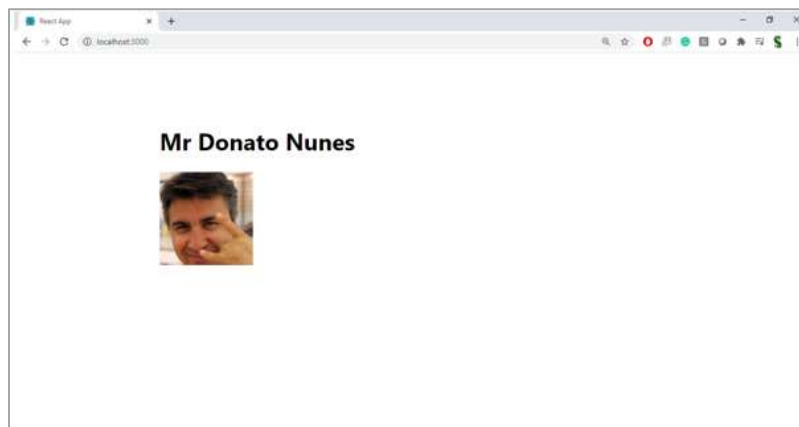
Create a React Application “fetchuserapp” which will retrieve the user details from <https://api.randomuser.me/> and display the title, firstname and image of a user.

Create a component named “Getuser” and in the asynchronous method “ComponentDidMount ()” invoke the URL using fetch method and the response can be displayed in the render method of the component.

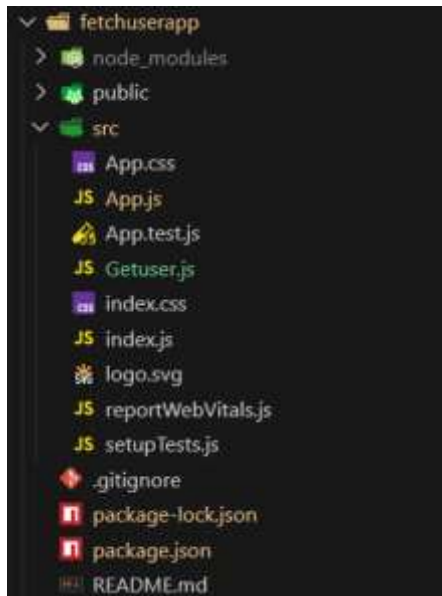
Code Snippet in Getuser Component:

```
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]);
}
```

Expected Output:



Implementation



App.js

```
import Getuser from './Getuser';
function App() {
  return (
    <div>
      <Getuser />
    </div>
  );
}
export default App;
```

Getuser.js

```
import React, { Component } from 'react';

class Getuser extends Component {
  constructor(props) {
    super(props);
    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() {
  const { person, loading } = this.state;

  if (loading) {
    return <h2>Loading user data...</h2>;
  }

  return (
    <div style={{ textAlign: 'center', marginTop: '20px' }}>
      <h1>User Info</h1>
      <h3>{person.name.title} {person.name.first}</h3>
      <img src={person.picture.large} alt="User" />
    </div>
  );
}
}

export default Getuser;
```

Output:

