1. Create a new React application using *create-react-app* tool and name it as “gitclientapp”.
2. Open the application using VS Code
3. Go to terminal in VS Code and install the support for “axios” react library to make calls to the GitHub API.

*Figure 2: Install axios library*

1. Create a new file with the name as **GitClient.js** in **src folder** of the application
2. Add the following code to create a class named **GitClient** which will make calls to the *api.github.com* to fetch the repositories as follows.

*Figure 3: GitClient module*

1. Modify the App component to use the declared module to fetch and display the repositories data as follows.

*Figure 4: App Component*

1. Build and Run the application using *npm start* command. The output should look similar to below.

A screenshot of a computer

AI-generated content may be incorrect.

*Figure 5: Application Output*

1. Create a new unit test file named as **GitClient.test.js** to unit test the newly created module.
2. Import **axios** and **GitClient** into the unit test file.
3. Describe the test name as “Git Client Tests”
4. Create a unit test using the **test()** and give the test name as “should return repository names for techiesyed”
5. Mock the axios object to return the dummy data
6. Invoke the **getRepositories()** method of **GitClient** and see it’s returning the mocked data instead of making an actual call to *api.github.com*
7. Run tests using *npm test* command.

**App.js**

import React, { useEffect, useState } from "react";

import GitClient from "./GitClient.js";

function App() {

const [repos, setRepos] = useState([]);

useEffect(() => {

const gitClient = new GitClient();

gitClient.getRepositories("techiesyed").then((data) => {

setRepos(data);

});

}, []);

return (

<div style={{ padding: "20px" }}>

<h1>Git repositories of User - TechieSyed</h1>

<ul>

{repos.map((repo, idx) => (

<li key={idx}>{repo}</li>

))}

</ul>

</div>

);

}

export default App;

**GitClient.js**

import axios from "axios";

class GitClient {

async getRepositories(username) {

const response = await axios.get(`https://api.github.com/users/${username}/repos`);

return response.data.map((repo) => repo.name);

}

}

export default GitClient;

**GitClient.test.js**

import axios from "axios";

import GitClient from "./GitClient.js";

jest.mock("axios");

describe("Git Client Tests", () => {

test("should return repository names for techiesyed", async () => {

const mockedResponse = {

data: [

{ name: "first-repo" },

{ name: "second-repo" },

{ name: "third-repo" }

]

};

axios.get.mockResolvedValue(mockedResponse);

const client = new GitClient();

const result = await client.getRepositories("techiesyed");

expect(result).toEqual(["first-repo", "second-repo", "third-repo"]);

});

});

**index.js**

import React from 'react';

import ReactDOM from 'react-dom/client';

import App from './App.js';

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(

<React.StrictMode>

<App />

</React.StrictMode>

);

**index.css**

body {

margin: 0;

padding: 0;

font-family: sans-serif;

background-color: #f7f7f7;

}

**index.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>GitClient App</title>

</head>

<body>

<div id="root"></div>

</body>

</html>

**package.json**

{

"name": "gitclientapp",

"version": "1.0.0",

"description": "React app to fetch GitHub repos with Jest tests",

"main": "index.js",

"type": "module",

"scripts": {

"start": "react-scripts start",

"test": "jest"

},

"dependencies": {

"axios": "^1.6.8",

"react": "^18.2.0",

"react-dom": "^18.2.0",

"react-scripts": "5.0.1"

},

"devDependencies": {

"@testing-library/react": "^14.0.0",

"jest": "^29.7.0"

}

}

**OUTPUT**

**A screenshot of a computer

AI-generated content may be incorrect.**