# Classroom Activity: Demonstrate how to use useEffect to perform a side effect, such as fetching data when the component mounts

**React component code for DataFetchingComponent.js:**

## Code:

|  |
| --- |
| import React, { useState, useEffect } from 'react';  const DataFetchingComponent = () => {  const [data, setData] = useState(null);   useEffect(() => {  // This function fetches data when the component mounts  fetch('https://jsonplaceholder.typicode.com/todos/1')  .then(response => response.json())  .then(data => setData(data));  }, []); // Empty array ensures this runs only once when the component mounts   return (  <div>  <h1>Fetched Data</h1>  {data && <p>{data.title}</p>}  </div>  ); };  export default DataFetchingComponent; |

**update App.js to include and use the DataFetchingComponent:**

## Code:

|  |
| --- |
| import React from 'react';  import DataFetchingComponent from './DataFetchingComponent';  function App() {  return (  <div className="App">  <header className="App-header">  <h1>My React App</h1>  <DataFetchingComponent />  </header>  </div>  );  }  export default App; |

**Output fetch API:**

**Output:**



## Execution Steps:

* Create a new React project if you don't have one already.
* Create a file named DataFetchingComponent.js in the src folder and write the code in the file.
* Import and use DataFetchingComponent in your main App.js file.
* Save the files.
* Run your React app using the terminal command: npm start.
* The fetched data will be displayed on the browser screen.