

React + Node.js Full-Stack Example (Using Axios and async/await)

This guide demonstrates a simple full-stack example using Node.js with Express on the backend and React.js on the frontend. The frontend fetches JSON data from the backend using Axios with `async/await` syntax.

■ Step 1: Backend (Node.js + Express)

Create a file named `server.js` and write the following code:

```
const express = require('express');
const cors = require('cors');
const app = express();
const PORT = 5000;

app.use(cors());

app.get('/json', (req, res) => {
  res.json({ name: 'Dielli', course: 'Node.js with Express' });
});

app.get('/html', (req, res) => {
  res.send('<h1>Express HTML Example</h1><p>Hello Students!</p>');
});

app.listen(PORT, () => console.log(`Server running on http://localhost:${PORT}`));
```

Run the server:

```
node server.js
```

Then open <http://localhost:5000/json> in your browser to confirm it works.

■ Step 2: Frontend (React + Axios + async/await)

Install Axios in your React project:

```
npm install axios
```

Inside `src/App.js`:

```
import React, { useEffect, useState } from 'react';
import axios from 'axios';

function App() {
  const [data, setData] = useState(null);
  const [loading, setLoading] = useState(true);
  const [error, setError] = useState(null);

  const fetchData = async () => {
    try {
      const response = await axios.get('http://localhost:5000/json');
      setData(response.data);
    } catch (err) {
```

```

        console.error('Error fetching data:', err);
        setError('Failed to load data');
    } finally {
        setLoading(false);
    }
};

useEffect(() => {
    fetchData();
}, []);

if (loading) return <h3>Loading data...</h3>;
if (error) return <h3 style={{ color: 'red' }}>{error}</h3>;

return (
    <div style={{ fontFamily: 'Poppins, sans-serif', padding: '2rem' }}>
        <h1>React + Node.js + Axios Example</h1>
        <div style={{
            marginTop: '1rem',
            background: '#f5f5f5',
            borderRadius: '10px',
            padding: '1rem',
            maxWidth: '400px',
        }}>
            <p><strong>Name:</strong> {data.name}</p>
            <p><strong>Course:</strong> {data.course}</p>
        </div>
    </div>
);
}

export default App;

```

Run your React app:

```
npm start
```

■ Your React app will fetch the JSON data from the Node.js server and display it dynamically using Axios and async/await.

■ Summary

- Backend: Node.js + Express serving JSON on /json
- Frontend: React using Axios with async/await for fetching
- CORS enabled for cross-origin access
- Clean error and loading state handling