

## Lesson 05 Demo 03

### Automating Boilerplate Code generation with GitHub Copilot

**Objective:** To demonstrate the automation of boilerplate code generation in a college management system using GitHub Copilot, specifically focusing on creating standard API endpoints

**Tools required:** Visual Studio Code, node js, json-server, and GitHub Copilot

**Prerequisites:** Lesson 05 Demos 01 and 02

Steps to be followed:

1. Input a specific endpoint creation prompt
2. Based upon prompt message whatever code generate follow all these steps and make the project ready.
3. Test the application.

**Note:** The generative AI tool used in this exercise can produce varied outputs even when presented with similar prompts. Thus, you may get different outputs for the same prompt.

## Step 1: Input a specific endpoint creation prompt

1.1 First create the folder with name as **react-demo-api-app**

1.2 Open this folder in vs code.

1.3 Open Visual Studio and provide the following prompt to GitHub Copilot:


**“The project is based on React JS using axios module to interact with emp.json file which is running using json-sever module and provide the end point. All the REST endpoints follow camelCasing for the URL paths. The emp.json file is as follows: departments with attribute [id, departmentName] and students with columns [id, StudentName, departmentId]. Create an API store and retrieve department and student details using react js. Please provide these provide all steps in details.”**

/help What can you do?

The project is based on React JS using axios module to interact with emp.json file which is running using json-sever module and provide the end point. All the REST endpoints follow camelCasing for the URL paths. The emp.json file is as follows: departments with attribute [id, departmentName] and students with columns [id, StudentName, departmentId]. Create an API store and retrieve department and student details using react js. Please provide these provide all steps in details.

@ 📎 🗣

The project is based on React JS using axios module to interact with emp.json file which is running using json-sever module and provide the end point. All the REST endpoints follow camelCasing for the URL paths. The emp.json file is as follows: departments with attribute [id, departmentName] and students with columns [id, StudentName, departmentId]. Create an API store and retrieve department and student details using react js. Please provide these provide all steps in details.

 GitHub Copilot

Sure, I'll guide you through the steps to create a React JS application that interacts with a **json-server** to store and retrieve department and student details using axios.

### Step 1: Set Up the Project

1. **Create a new React project:** Open your terminal and run:

```
npx create-react-app react-demo-api-app
cd react-demo-api-app
```

2. **Install axios and json-server:**

```
npm install axios json-server
```

### Step 2: Set Up the JSON Server

1. **Create **emp.json** file:** In the root of your project, create a file named **emp.json** with the following content:

```
{
```

**Step 2.** Based upon prompt message whatever code generate follow all these steps and make the project ready.

## 2.1 Set Up the Project

### 1. Initialize a new React project:

```
C:\Users\akash\Desktop\Project with Gen AI\Demos - Updated with MERN\Demos\Lesson_05 - with MERN Using Generative AI\React JS Project
using Generative AI\react-demo-api-app>npx create-react-app react-demo-api-app

You are running 'create-react-app' 5.0.1, which is behind the latest release (5.1.0).
We recommend always using the latest version of create-react-app if possible.

The latest instructions for creating a new app can be found here:
https://create-react-app.dev/docs/getting-started/

C:\Users\akash\Desktop\Project with Gen AI\Demos - Updated with MERN\Demos\Lesson_05 - with MERN Using Generative AI\React JS Project
using Generative AI\react-demo-api-app>
```

npx create-react-app react-demo-api-app

if you are getting the error

Then run the below command as

npx create-react-app@latest react-demo-api-app

```
C:\Users\akash\Desktop\Project with Gen AI\Demos - Updated with MERN\Demos\Lesson_05 - with MERN Using Generative AI\React JS Project
using Generative AI\react-demo-api-app>npx create-react-app@latest react-demo-api-app
Need to install the following packages:
  create-react-app@5.1.0
Ok to proceed? (y) y

Creating a new React app in C:\Users\akash\Desktop\Project with Gen AI\Demos - Updated with MERN\Demos\Lesson_05 - with MERN Using Ge
nerative AI\React JS Project using Generative AI\react-demo-api-app\react-demo-api-app.

Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...

[ ] \ idealTree:react-demo-api-app: sill idealTree buildDeps
```

2.2 after project create move inside this project using the command as

cd react-demo-api-app

```
C:\Users\akash\Desktop\Project with Gen AI\Demos - Updated with MERN\Demos\Lesson_05 - with MERN Using Generative AI\React JS Project
using Generative AI\react-demo-api-app>cd react-demo-api-app

C:\Users\akash\Desktop\Project with Gen AI\Demos - Updated with MERN\Demos\Lesson_05 - with MERN Using Generative AI\React JS Project
using Generative AI\react-demo-api-app\react-demo-api-app>
```

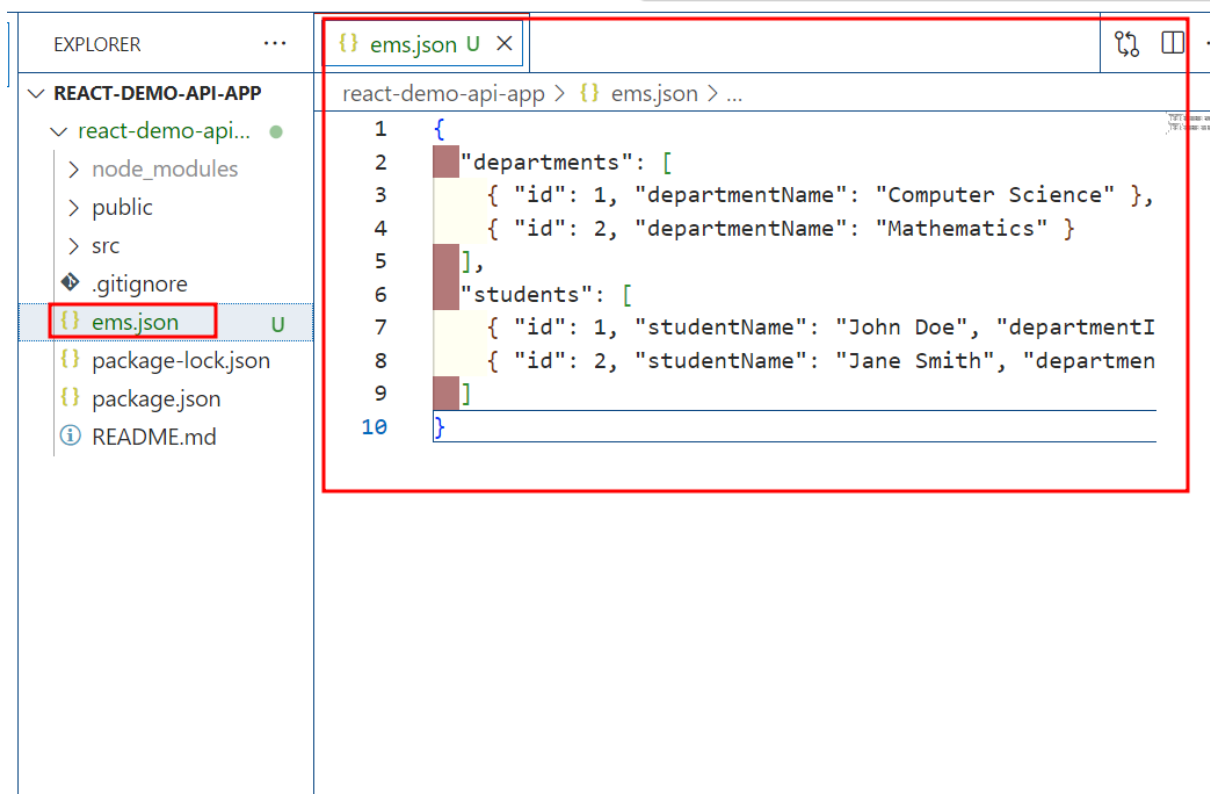
2.3 Now need to installed required dependencies.

**npm install axios json-serve**

2.4 Now we will create the **sms.json** file

```
{
  "departments": [
    { "id": 1, "departmentName": "Computer Science" },
    { "id": 2, "departmentName": "Mathematics" }
  ],
  "students": [
    { "id": 1, "studentName": "John Doe", "departmentId": 1 },
    { "id": 2, "studentName": "Jane Smith", "departmentId": 2 }
  ]
}
```

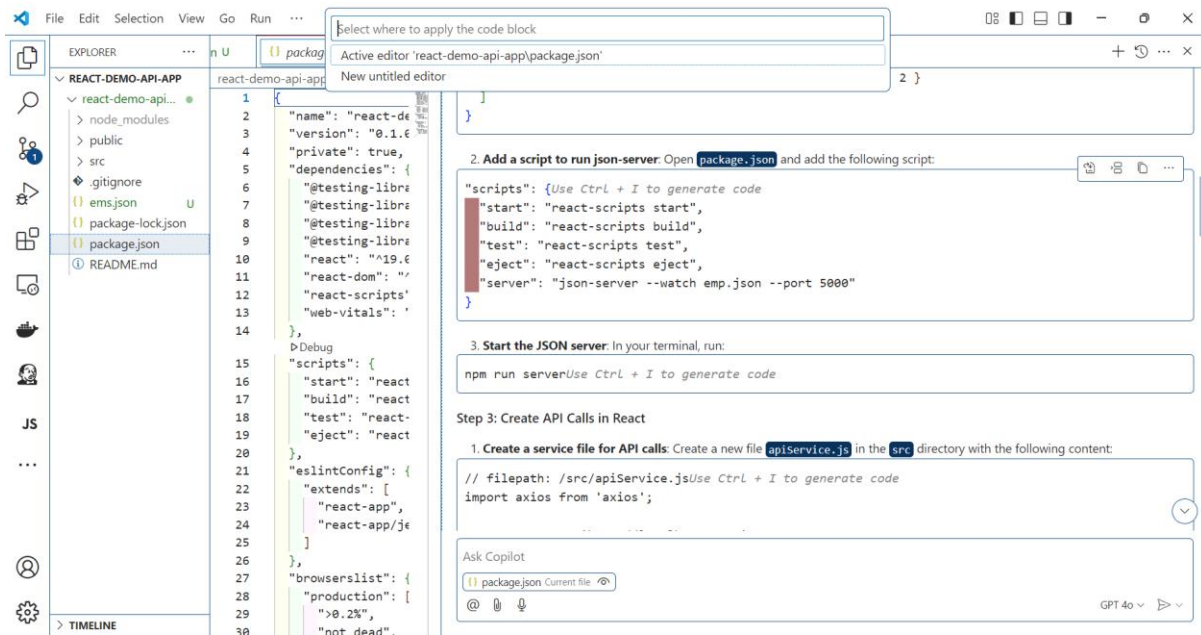
Apply this sample file in editor.



2.5 apply generate package.json sever code to existing package.json code

2. Add a script to run json-server: Open `package.json` and add the following script:

```
"scripts": {  
  "start": "react-scripts start",  
  "build": "react-scripts build",  
  "test": "react-scripts test",  
  "eject": "react-scripts eject",  
  "server": "json-server --watch emp.json --port 5000"  
}
```



2.6 Start the JSON server: In your terminal, run:

```
C:\Users\akash\Desktop\Project with Gen AI\Demos - Updated with MERIN\Demos\Lesson_05 - with MERIN using Generative AI\react-demo-api-app\react-demo-api-app>npm run server  
  
> react-demo-api-app@0.1.0 server  
> json-server --watch emp.json --port 5000  
  
--watch/-w can be omitted, JSON Server 1+ watches for file changes by default  
JSON Server started on PORT :5000  
Press CTRL-C to stop  
Watching emp.json...  
  
( ^ _ ^ )  
  
Index:  
http://localhost:5000/  
  
Static files:  
Serving ./public directory if it exists  
  
Endpoints:  
http://localhost:5000/departments  
http://localhost:5000/students
```

2.7 create API call in react js.

### **apiService.js**

```
// filepath: /src/apiService.js
import axios from 'axios';

const API_URL = 'http://localhost:5000';

export const getDepartments = async () => {
  const response = await axios.get(`${API_URL}/departments`);
  return response.data;
};

export const getStudents = async () => {
  const response = await axios.get(`${API_URL}/students`);
  return response.data;
};

export const addDepartment = async (department) => {
  const response = await axios.post(`${API_URL}/departments`, department);
  return response.data;
};

export const addStudent = async (student) => {
  const response = await axios.post(`${API_URL}/students`, student);
  return response.data;
};
```

Apply this code to editor and save with file name as **apiService.js** inside src folder.

## 2.8 Create Components to Display Data

**Create a Departments component:** Create a new file **Departments.js** in the src directory with the following content:

### **Department.js**

```
// filepath: /src/Departments.js
import React, { useEffect, useState } from 'react';
import { getDepartments } from './apiService';

const Departments = () => {
```

```

const [departments, setDepartments] = useState([]);

useEffect(() => {
  const fetchDepartments = async () => {
    const data = await getDepartments();
    setDepartments(data);
  };
  fetchDepartments();
}, []);

return (
  <div>
    <h2>Departments</h2>
    <ul>
      {departments.map(department => (
        <li key={department.id}>{department.departmentName}</li>
      ))}
    </ul>
  </div>
);
};

export default Departments;

```

**2.9 Create a Students component:** Create a new file Students.js in the src directory with the following content:

### Students.js

```

// filepath: /src/Students.js
import React, { useEffect, useState } from 'react';
import { getStudents } from './apiService';

const Students = () => {
  const [students, setStudents] = useState([]);

  useEffect(() => {
    const fetchStudents = async () => {
      const data = await getStudents();
      setStudents(data);
    };
    fetchStudents();
  }, []);

  return (

```

```

    <div>
      <h2>Students</h2>
      <ul>
        {students.map(student => (
          <li key={student.id}>{student.studentName}</li>
        ))}
      </ul>
    </div>
  );
};

export default Students;

```

## 2.10 Integrate Components in App

1. **Update App.js:** Open src/App.js and update it to include the new components:

### App.js

```

// filepath: /src/App.js
import React from 'react';
import Departments from './Departments';
import Students from './Students';

function App() {
  return (
    <div className="App">
      <header className="App-header">
        <h1>React Demo API App</h1>
        <Departments />
        <Students />
      </header>
    </div>
  );
}

export default App;

```

2.11 : If you need to add Add operation on Student and Department ask for the prompt message.

“please provide the code for add Student and deparment must be react js component”

2.12 **Create AddDepartment.js:** Create a new file AddDepartment.js in the src directory with the following content:



### AddDepartment.js

```
// filepath: /src/AddDepartment.js
import React, { useState } from 'react';
import { addDepartment } from './apiService';

const AddDepartment = () => {
  const [departmentName, setDepartmentName] = useState("");

  const handleSubmit = async (e) => {
    e.preventDefault();
    const newDepartment = { departmentName };
    await addDepartment(newDepartment);
    setDepartmentName("");
  };

  return (
    <div>
      <h2>Add Department</h2>
      <form onSubmit={handleSubmit}>
        <input
          type="text"
          value={departmentName}
          onChange={(e) => setDepartmentName(e.target.value)}
          placeholder="Department Name"
          required
        />
        <button type="submit">Add Department</button>
      </form>
    </div>
  );
};

export default AddDepartment;
```

### 2.13 Create AddStudent Component

**Create AddStudent.js:** Create a new file AddStudent.js in the src directory with the following content:

```
// filepath: /src/AddStudent.js
import React, { useState, useEffect } from 'react';
import { addStudent, getDepartments } from './apiService';

const AddStudent = () => {
  const [studentName, setStudentName] = useState("");
  const [departmentId, setDepartmentId] = useState("");
```

```

const [departments, setDepartments] = useState([]);

useEffect(() => {
  const fetchDepartments = async () => {
    const data = await getDepartments();
    setDepartments(data);
  };
  fetchDepartments();
}, []);

const handleSubmit = async (e) => {
  e.preventDefault();
  const newStudent = { studentName, departmentId: parseInt(departmentId) };
  await addStudent(newStudent);
  setStudentName("");
  setDepartmentId("");
};

return (
  <div>
    <h2>Add Student</h2>
    <form onSubmit={handleSubmit}>
      <input
        type="text"
        value={studentName}
        onChange={(e) => setStudentName(e.target.value)}
        placeholder="Student Name"
        required
      />
      <select
        value={departmentId}
        onChange={(e) => setDepartmentId(e.target.value)}
        required
      >
        <option value="">Select Department</option>
        {departments.map(department => (
          <option key={department.id} value={department.id}>
            {department.departmentName}
          </option>
        ))}
      </select>
      <button type="submit">Add Student</button>
    </form>
  </div>
);
};

```

```
export default AddStudent;
```

## 2.14 Integrate Components in App

**Update [App.js](#):** Open [App.js](#) and update it to include the new components:

```
// filepath: /src/App.js
import React from 'react';
import logo from './logo.svg';
import './App.css';
import Departments from './Departments';
import Students from './Students';
import AddDepartment from './AddDepartment';
import AddStudent from './AddStudent';

function App() {
  return (
    <div className="App">
      <header className="App-header">
        <h1>React Demo API App</h1>
        <AddDepartment />
        <AddStudent />
        <Departments />
        <Students />
      </header>
    </div>
  );
}

export default App;
```

## Step 3 : Run this application

npm start                      (this command you need to run on another terminal)

# React Demo API App

## Add Department

Department Name Add Department

## Add Student

Student Name Select Department Add Student

## Departments

- Computer Science
- Mathematics