**✅ Why Use Backend for APIs?**

Some APIs (like Swiggy) don’t allow direct calls from the browser due to:

* ❌ **CORS restrictions**
* 🔐 **API key protection**
* 🧱 **Rate limiting or auth requirements**

**✅ Full Setup: Using an API via a Backend Proxy**

**🧱 1. Create a Backend Folder (Node.js)**

📁 Folder Structure:

.1-Projects/

├── food-panda/ ← React frontend

└── swiggy-proxy/ ← Backend proxy (Node.js)

**📦 2. Initialize backend (one-time)**

In terminal:

cd path/to/.1-Projects/

mkdir swiggy-proxy

cd swiggy-proxy

npm init -y

npm install express cors node-fetch

**🧑‍💻 3. Create swiggyProxy.js**

Paste this code:

const express = require("express");

const fetch = require("node-fetch");

const cors = require("cors");

const app = express();

const PORT = 5000;

// ✅ Allow frontend (React) access

app.use(cors({ origin: "http://localhost:1234" })); // your React port

// ✅ Your proxy route

app.get("/api/swiggy", async (req, res) => {

try {

const swiggyUrl = "https://www.swiggy.com/mapi/restaurants/list/v5?..."; // full URL

const response = await fetch(swiggyUrl);

const data = await response.json();

res.json(data); // send to React

} catch (err) {

console.error("❌ Swiggy fetch error:", err);

res.status(500).json({ error: "Failed to fetch Swiggy data" });

}

});

app.listen(PORT, () => {

console.log(`🚀 Proxy running at http://localhost:${PORT}`);

});

**▶️ 4. Start Backend**

node swiggyProxy.js

You’ll see:

🚀 Proxy running at http://localhost:5000

**⚛️ 5. Call the Proxy from React (food-panda)**

Inside src/utils/useFetchRecipes.js:

import { useEffect } from "react";

const useFetchRestaurants = () => {

useEffect(() => {

const fetchData = async () => {

try {

const res = await fetch("http://localhost:5000/api/swiggy");

const json = await res.json();

console.log("🍽️ Swiggy data from proxy:", json);

} catch (err) {

console.error("❌ Failed to fetch from proxy:", err);

}

};

fetchData();

}, []);

};

export default useFetchRestaurants;

Then use this hook in Body.js:

useFetchRestaurants();

**✅ What Happens Behind the Scenes**

1. React → http://localhost:5000/api/swiggy
2. Your Node.js proxy fetches Swiggy data
3. Proxy sends the JSON back to React
4. No CORS issues ✅
5. No API keys exposed ✅

**❗ Must-Do Every Time**

Every time you start your project:

1. Start proxy:
2. cd swiggy-proxy
3. node swiggyProxy.js
4. Start frontend:
5. cd food-panda
6. npm run dev