

Gandhinagar Bites

Priyank Pitliya

Unit Testing

Technology used

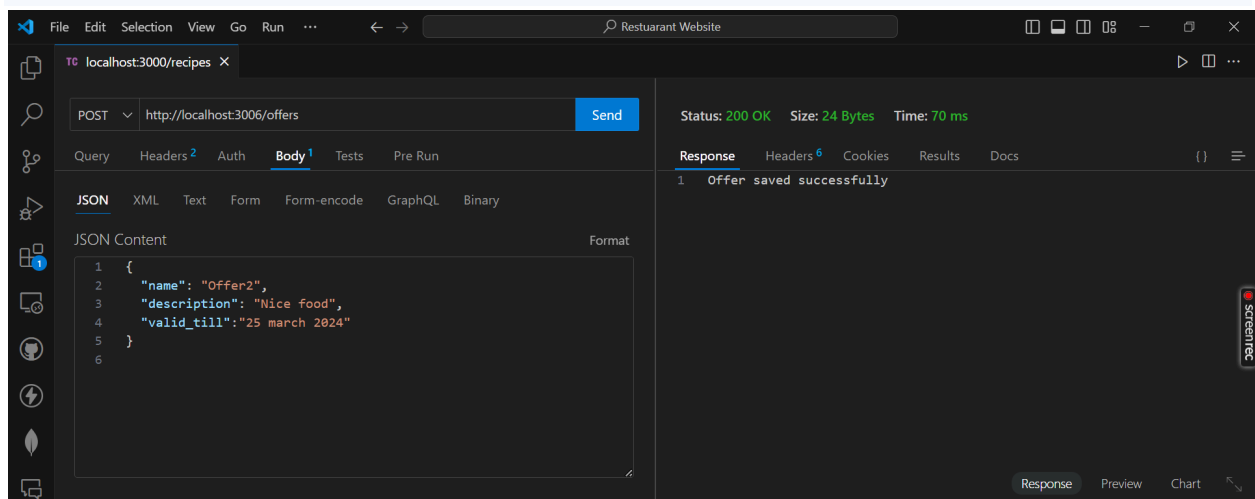
Thunderclient

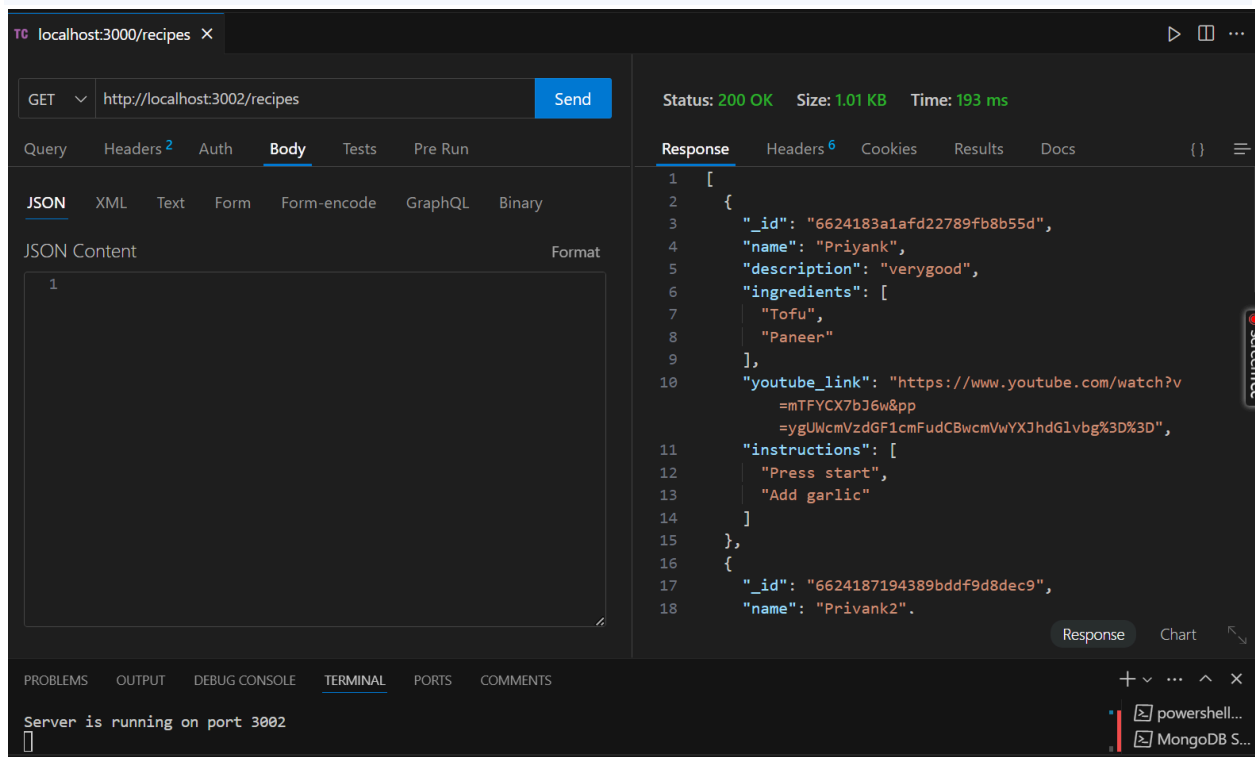
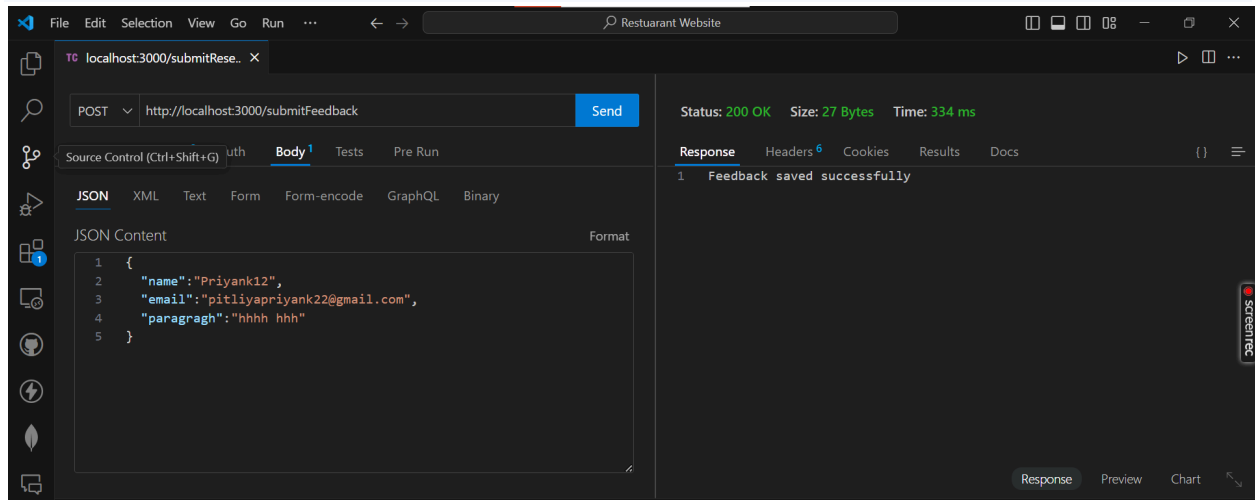
Thunder Client is a lightweight Rest API Client Extension for Visual Studio Code, hand-crafted with a focus on simplicity, clean design and local storage.

Main Features

- Lightweight and easy to use REST API Client.
- Supports Collections and Environment variables.
- Scriptless Testing: Easily test API responses with a GUI-based interface.
- Local Storage: The extension saves all data locally on the user's device.
- Git Sync: Save requests data in your Git repository for team collaboration.
- Advanced CLI: Run requests, multiple collections and cURL commands from the terminal.

Results:





localhost:3000/recipes X

POST http://localhost:3002/recipes Send

Query Headers 2 Auth Body 1 Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

JSON Content Format

```
1 {
2   "name": "Priyank",
3   "description": "verygood",
4   "ingredients": [
5     "Tofu",
6     "Paneer"
7   ],
8   "youtube_link": "https://www.youtube.com/watch?v
9     =mTFYCX7bJ6w&pp
10    =ygUWcmVzdGF1cmFudCBwcmVwYXJhdG1vbG%3D%3D",
11   "instructions": [
12     "Press start",
13     "Add garlic"
14   ]
15 }
```

Status: 200 OK Size: 25 Bytes Time: 169 ms

Response Headers 6 Cookies Results Docs {}

1 Recipe saved successfully

Response Preview Chart

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

Server is running on port 3002

Go Live

localhost:3000/recipes X

POST http://localhost:3006/offers Send

Query Headers 2 Auth Body 1 Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

JSON Content Format

```
1 {
2   "name": "offer3",
3   "description": "50%off",
4   "valid_till": "23 March 2024"}
5 }
```

Status: 200 OK Size: 24 Bytes Time: 215 ms

Response Headers 6 Cookies Results Docs {}

1 Offer saved successfully

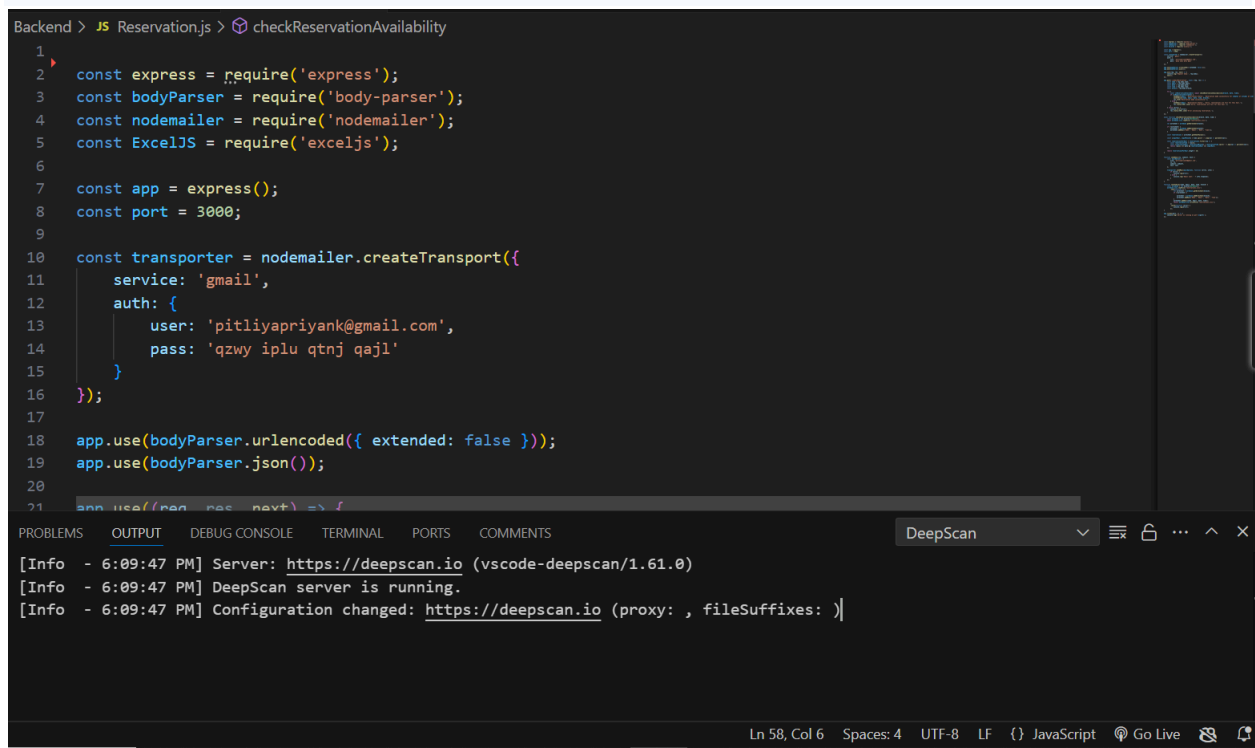
Response Preview Chart

Go Live

Static Analysis

Deep Scan

Deep Scan performs static analysis on the JavaScript code and can also be added in vs code, providing all sorts of errors and warnings. It also provides a summary of issues and their impacts on the code.



The screenshot shows a VS Code editor with a JavaScript file named `Reservation.js`. The code defines an Express.js application with a nodemailer transport for email. The terminal at the bottom shows the output of the DeepScan extension, indicating it is running and configured.

```
Backend > JS Reservation.js > checkReservationAvailability
1
2 const express = require('express');
3 const bodyParser = require('body-parser');
4 const nodemailer = require('nodemailer');
5 const ExcelJS = require('exceljs');
6
7 const app = express();
8 const port = 3000;
9
10 const transporter = nodemailer.createTransport({
11   service: 'gmail',
12   auth: {
13     user: 'pitliyapriyank@gmail.com',
14     pass: 'qzwy iplu qtnj qajl'
15   }
16 });
17
18 app.use(bodyParser.urlencoded({ extended: false }));
19 app.use(bodyParser.json());
20
21 app.use((req, res, next) => {
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS DeepScan

```
[Info - 6:09:47 PM] Server: https://deepscan.io (vscode-deepscan/1.61.0)
[Info - 6:09:47 PM] DeepScan server is running.
[Info - 6:09:47 PM] Configuration changed: https://deepscan.io (proxy: , fileSuffixes: )
```

Ln 58, Col 6 Spaces: 4 UTF-8 LF {} JavaScript Go Live