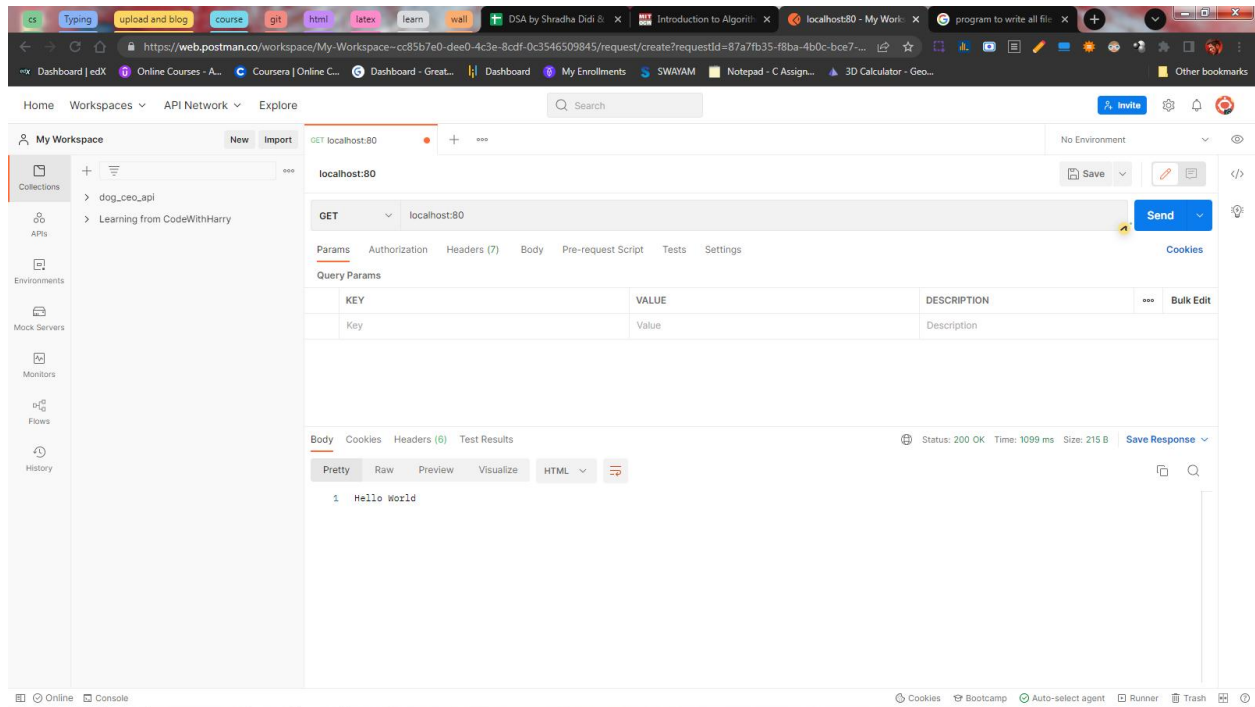


Testing our app on postman

Run the file (or type node filename in terminal)

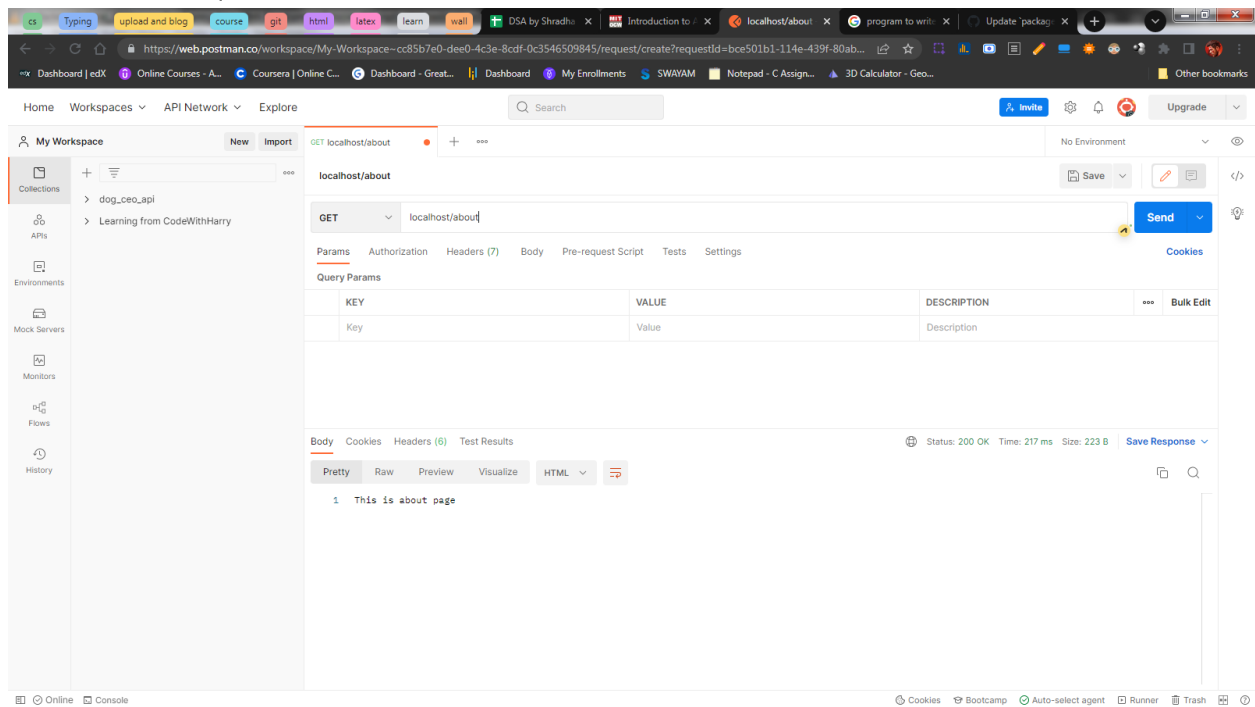
Type on postman → “localhost:80” or “localhost”.

This is the benefit of port 80 i.e. we do not need to write it.

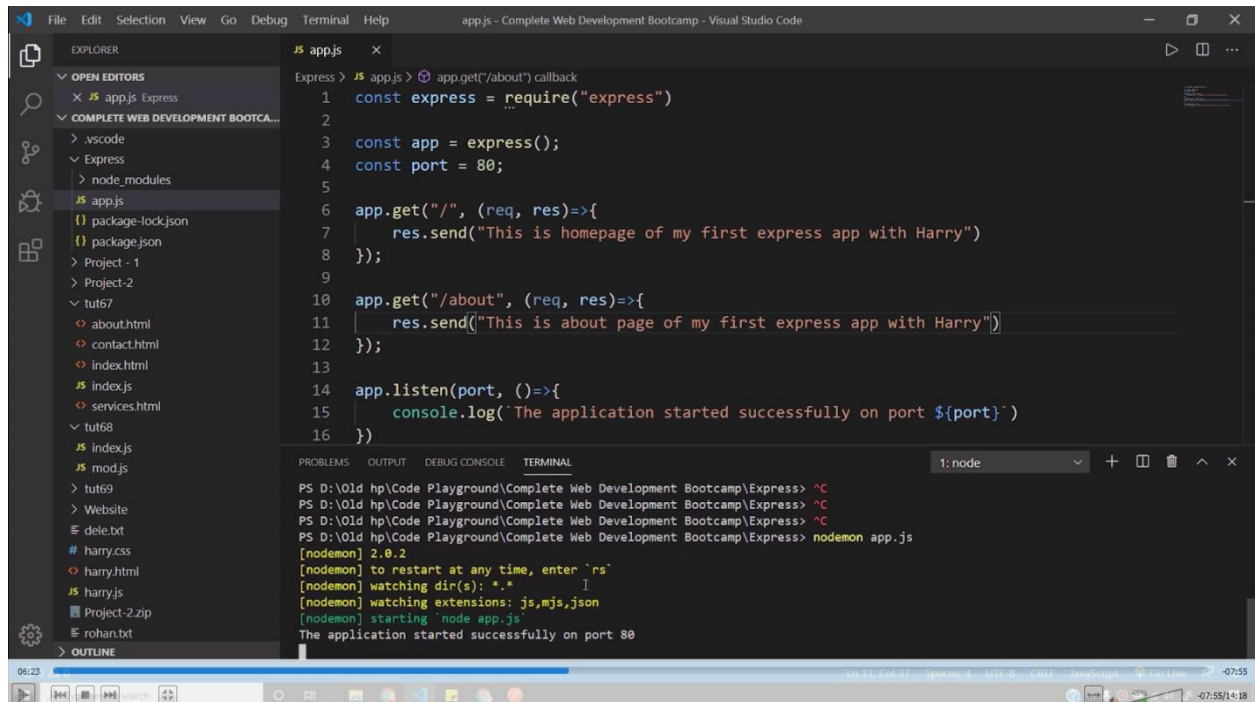


Above localhost command also work on chrome, but only get request will work on chrome. For post or other request we need postman.

/about section request



Install "nodemon" it will automatically restart the server when anything changes in the code.



Press "rs" to restart nodemon while server is running

A screenshot of a Windows desktop with a dark theme. The primary focus is the Visual Studio Code (VS Code) editor, which is open to a file named `app.js` in a project called `appjs - Complete Web Development Bootcamp`. The Explorer sidebar on the left shows the project structure, including `package-lock.json`, `package.json`, and several HTML files. The main editor area displays the following JavaScript code:

```
1 const express = require("express")
2
3 const app = express();
4 const port = 80;
5
6 app.get("/", (req, res)=>{
7     res.send("This is homepage of my first express app with Harry")
8 });
9
10 app.get("/about", (req, res)=>{
11     res.send("This is about page of my first express app with Harry")
12 });
13
14 app.listen(port, ()=>{
15     console.log(`The application started successfully on port ${port}`)
16 })
```

The TERMINAL panel at the bottom shows the command `nodemon app.js` being executed in a PowerShell window. The output indicates that the application is running successfully on port 80. The Windows taskbar at the bottom shows the time as 06:31 and the system tray includes icons for network, volume, and other background processes.

We can also status code

The image shows a development environment with Visual Studio Code and Postman. In Visual Studio Code, the file `app.js` is open, showing a Node.js Express application. The code includes comments and a status code of 200 for the `GET /about` endpoint. A lightbulb icon indicates a suggestion to use `res.status(200)` before `res.send()`.

```
1 //Notes:
2 /*
3 1) It is a web application framework of node js.
4 2) It will do url parse for us and help us routing
5 3) It will be scalability
6 */
7
8
9 const express = require("express");
10
11 const port = 80;
12 const app = express();
13
14 /* when user will send request to our server it will be
15 handled by this function */
16 app.get("/", (req, res) => {
17   res.send("This is homepage of my first express app");
18 });
19
20 app.get("/about", (req, res) => {
21   // res.send("This is about page via get");
22   // we can also send status code
23   res.status(200).send("This is about page via get");
24 });
25
26 app.post("/about", (req, res) => {
27   res.send("This is about page via posts via posts");
28 });
29
30
31 /* we have to tell on which port we want to listen */
32 app.listen(port, () => {
33   console.log(`Server is running on port ${port}`);
34 });
```

In Postman, an `GET` request to `localhost/` has been sent. The response is a `200 OK` status, indicating a successful HTTP request. The response body is `This is homepage of my first express app with Harry`.

200 OK
Standard response for successful HTTP requests. The actual response will depend on the request method used. In a GET request, the response will contain an entity corresponding to the requested resource. In a POST request the response will contain an entity describing or containing the result of the action.

Visual Studio Code interface showing a JavaScript file named `app.js` and the Postman application below it.

app.js

```
const port = 80;
const app = express();

/* when user will send request to our server it will be
handled by this function */
app.get("/", (req, res) => {
  res.send("This is homepage of my first express app");
});

app.get("/about", (req, res) => {
  // res.send("This is about page via get");

  // we can also send status code
  res.status(200).send("This is about page via get");
});

app.get("/this", (req, res) => {
  // ...res.status(400).send("This page is not found");
});

app.post("/about", (req, res) => {
  res.send("This is about page via posts via posts");
});

/* we have to tell on which port we want to listen */
app.listen(port, () => {
  console.log(`Server is running on port ${port}`);
});
```

Postman

Request: `GET localhost/this`

Status: 400 Bad Request

Message: The request cannot be fulfilled due to bad syntax.

Response Body (HTML):

```
1 This page is not found
```