

when the url hits app.get("/", ()=>{ console.log ("welcome to home page")})

The screenshot shows a developer's workspace with two main windows open:

- Thunder Client (REST API Assignment):** A tool for making HTTP requests. The left sidebar shows activity logs for various requests (GET localhost:8086, GET localhost:5081, etc.). The main interface has a "New Request" button, a URL input field containing "http://localhost:8086/", and a "Send" button. The response pane shows a 200 OK status with the message "Welcome to the Home page".
- VS Code Terminal:** The bottom half of the screen displays terminal output. It shows log entries from a Node.js application indicating successful access to the home page at various times and IP addresses (e.g., "Home page accessed" with Method: GET, URL: /, Status: 200, and specific timestamps like 2025-12-26T09:26:20.127Z).

when url hits app.get("/users",()=>{}) in response giving three users data

The screenshot shows a Node.js development environment with several windows open:

- Terminal:** Shows logs for a user access and a list access.
- Code Editor:** Displays a file named `index.js` containing the following code:

```
const express = require('express');
const app = express();
app.get('/users', () => {
  return [
    {
      "id": 1,
      "firstName": "Anshika",
      "lastName": "Agarwal",
      "hobby": "Teaching"
    },
    {
      "id": 2,
      "firstName": "Ritik",
      "lastName": "Agarwal",
      "hobby": "Travelling"
    },
    {
      "id": 3,
      "firstName": "Ankit",
      "lastName": "Agarwal",
      "hobby": "Reading"
    }
  ]
});
app.listen(8081, () => {
  console.log(`Server is running on port 8081`);
});
```
- Browser:** Shows a request to `http://localhost:8081/users` with a status of 200 OK, size of 211 bytes, and time of 11 ms. The response body is a JSON array of three user objects:

```
[{"id": 1, "firstName": "Anshika", "lastName": "Agarwal", "hobby": "Teaching"}, {"id": 2, "firstName": "Ritik", "lastName": "Agarwal", "hobby": "Travelling"}, {"id": 3, "firstName": "Ankit", "lastName": "Agarwal", "hobby": "Reading"}]
```

when url hits app.get("/users/:id", ()=>{})

The screenshot shows a developer's environment with a focus on a REST API request and its response.

Request:

- Method: GET
- URL: <http://localhost:8085/users/2>
- Headers: Content-Type: application/json
- Body: (Empty)

Response:

- Status: 200 OK
- Size: 70 Bytes
- Time: 10 ms

```
1 {
2   "id": 2,
3   "firstName": "Vittik",
4   "lastName": "Agarwal",
5   "hobby": "Travelling"
6 }
```

Terminal:

```
Status: 200
ip::ffff:127.0.0.1,
Time: 2025-12-26T09:28:50.357Z,
```

User list accessed
Method: GET,
URL: /users/2,
Status: 200
ip::ffff:127.0.0.1,
Time: 2025-12-26T09:30:46.816Z,

when url hits app.post('/user', ()=>{})

The screenshot shows the Thunder Client interface. In the top bar, there are tabs for 'File', 'Edit', 'Selection', 'View', 'Go', 'Run', and a search bar containing 'RESTAPIASSMENT'. Below the search bar, there are two tabs: 'New Request' and 'New Request X'. The main area shows a POST request to 'http://localhost:8086/user'. The 'Body' tab is selected, showing JSON content:

```
1 {  
2   "id": 7,  
3   "firstName": "john",  
4   "lastName": "doe",  
5   "hobby": "Travelling"  
6 }
```

The response status is '201 Created', size is '25 Bytes', and time is '38 ms'. The response body contains the message 'user added successfully'. The left sidebar shows activity logs with entries for POST localhost:8086/, GET localhost:8081/, PUT localhost:5081/, and GET localhost:8080/. The bottom terminal window shows command-line output for a node.js process.

```
URL: /users/2  
Status: 200  
ip:::ffff:127.0.0.1,  
Time: 2025-12-26T09:30:46.816Z,  
  
Method: POST,  
URL: /user,  
Status: 200  
ip:::ffff:127.0.0.1,  
Time: 2025-12-26T09:33:13.940Z,
```

when url hits app.put("/user/:id",()=>{})

The screenshot shows a Node.js development environment with the following details:

- Activity:** A sidebar on the left lists recent API interactions:
 - PUT localhost:8086/ 3 mins ago
 - GET localhost:8081/ 12 mins ago
 - PUT localhost:5081/ 2 hours ago
 - GET localhost:8080 1 day ago
 - POST localhost:8888 1 month ago
- Request Panel:** The main panel shows a PUT request to `http://localhost:8086/user/3`. The "Body" tab is selected, displaying the JSON payload:

```
1  {
2    "id": 3,
3    "firstName": "Ankit tiwari",
4    "lastName": "Agarwal",
5    "hobby": "Reading"
6 }
```
- Response Panel:** The response shows a `200 OK` status with a size of `35 Bytes` and a time of `12 ms`. The "Response" tab displays the JSON response:

```
1 "user details updated successfully"
```
- Terminal:** The terminal pane shows the command used to run the application:

```
URL: /user/3,
Status: 200
ip::ffff:127.0.0.1,
Time: 2025-12-26T09:38:22.022Z,
```

```
Method: PUT,
URL: /user/3,
Status: 200
ip::ffff:127.0.0.1,
Time: 2025-12-26T09:38:26.641Z,
```
- System Tray:** The bottom right corner shows system icons for battery, signal, and date/time.

when url hits app.delete("/user/:id",()=>{})

The screenshot shows a developer environment with a terminal, code editor, and a REST API testing tool.

REST API Testing Tool:

- Request:** DELETE http://localhost:8086/user/3
- Status:** 200 OK
- Response:** "user deleted successfully"

Code Editor (index.js):

```
app.delete('/user/:id', () => {})
```

Terminal:

```
> restfulapi@1.0.0 start
> node index.js
server is running at port: 8086
Method: DELETE,
URL: /user/3,
Status: 200
ip:::ffff:127.0.0.1,
Time: 2025-12-26T09:47:01.915Z,
```