

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

The screenshot displays the Visual Studio Code interface with the REST Client extension. The main editor shows a REST client request configuration for a GET method to `http://localhost:8080/`. The 'Send' button is visible. Below the configuration, the 'Query Parameters' section is empty. The right-hand pane shows the response details: Status 200 OK, Size: 24 Bytes, Time: 10 ms. The response body contains the text 'Welcome to the Home page'. The bottom terminal pane shows the output of the request, including the IP address `Ip::ffff:127.0.0.1`, the time `Time: 2025-12-26T09:26:28.127Z`, and the message 'Home page accessed'.

File Edit Selection View Go Run Terminal Help

RESTAPI/ASSIGNMENT

THUNDER CLIENT... JS index.js M localhost:5081/ TE New Request TE New Request X

New Request

Activity Collections Env

filter activity

GET localhost:8080/ just now

GET localhost:8081/ just now

GET localhost:5081/ 2 hours ago

GET localhost:6060 1 day ago

POST localhost:8088 1 month ago

GET http://localhost:8080/ Send

Query Headers² Auth Body Tests Pre Run

Query Parameters

parameter value

Status: 200 OK Size: 24 Bytes Time: 10 ms

Response Headers⁴ Cookies Results Docs

1 Welcome to the Home page

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Ip::ffff:127.0.0.1,
Time: 2025-12-26T09:26:28.127Z,

Home page accessed
Method: GET,
URL: /,
Status: 200
Ip::ffff:127.0.0.1,
Time: 2025-12-26T09:26:36.381Z,

Home page accessed

main

Type here to search

Top Stories

Go Live Quick Save

2:56 PM
12/26/2025

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

The screenshot shows the VS Code interface with a REST client extension. A GET request is sent to `http://localhost:8086/users`. The response is a JSON array containing three user objects. The status is 200 OK, size is 211 Bytes, and time is 11 ms.

Request:

```
GET http://localhost:8086/users
```

Response:

```
1 {
2   {
3     "id": 1,
4     "firstName": "Anshika",
5     "lastName": "Agarwal",
6     "hobby": "Teaching"
7   },
8   {
9     "id": 2,
10    "firstName": "Ritik",
11    "lastName": "Agarwal",
12    "hobby": "Travelling"
13  },
14  {
15    "id": 3,
16    "firstName": "Ankit",
17    "lastName": "Agarwal",
18    "hobby": "Reading"
19  }
20 }
```

Terminal Output:

```
ip::ffff:127.0.0.1,
Time: 2025-12-26T09:26:36.381Z,

Home page accessed
Method: GET,
URL: /users,
Status: 200
ip::ffff:127.0.0.1,
Time: 2025-12-26T09:28:50.357Z,

User list accessed
```

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

The screenshot displays the Thunder Client interface with a REST client request and response. The request is a GET method to the URL `http://localhost:8086/users/2`. The response is a 200 OK status with a size of 70 Bytes and a time of 10 ms. The response body is a JSON object containing user details.

Request:

```
GET http://localhost:8086/users/2
```

Response:

```
{
  "id": 2,
  "firstName": "Ritik",
  "lastName": "Agarwal",
  "hobby": "Travelling"
}
```

Terminal Output:

```
Status: 200
ip::ffff:127.0.0.1,
Time: 2025-12-26T00:28:50.357Z,

User list accessed
Method: GET,
URL: /users/2,
Status: 200
ip::ffff:127.0.0.1,
Time: 2025-12-26T00:30:46.816Z,
```

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

The screenshot displays the Thunder Client interface with a REST API named 'RESTAPIASSIGNMENT'. A POST request to 'http://localhost:8086/user' has been executed successfully. The status is '201 Created', the size is '25 Bytes', and the time taken is '38 ms'. The response body is a JSON object: `{ "id": 7, "firstName": "john", "lastName": "doe", "hobby": "Travelling" }`. The terminal at the bottom shows the request details, including the URL, status (200), IP address, and timestamp.

Thunder Client interface showing a REST API named 'RESTAPIASSIGNMENT'. The active request is a POST to 'http://localhost:8086/user'. The response status is '201 Created', size is '25 Bytes', and time is '38 ms'. The response body is a JSON object: `{ "id": 7, "firstName": "john", "lastName": "doe", "hobby": "Travelling" }`. The terminal shows the request details: URL: /users/2, Status: 200, ip: ::ffff:127.0.0.1, Time: 2025-12-26T09:30:45.816Z, 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 displays the Thunder Client interface with a REST API client. The main window shows a PUT request to `http://localhost:8086/user/3` that has been successfully executed. The response is a 200 OK status with a body of `"user details updated successfully"`. The JSON content of the request is visible in the left pane, showing a user object with id 3, first name "Ankit tiwari", last name "Agarwal", and hobby "Reading". The bottom pane shows the terminal output, which includes the URL, status, IP address, and time of the request.

Thunder Client interface showing a REST API client. The main window displays a PUT request to `http://localhost:8086/user/3` with a status of 200 OK. The response body is `"user details updated successfully"`. The JSON content of the request is visible in the left pane, showing a user object with `"id": 3`, `"firstName": "Ankit tiwari "`, `"lastName": "Agarwal"`, and `"hobby": "Reading"`. The bottom pane shows the terminal output, which includes the URL, status, IP address, and time of the request.

```
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,
```

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

The screenshot displays the Visual Studio Code interface with a REST client extension. The top panel shows a DELETE request to `http://localhost:8086/user/3`. The request body is a JSON object: `{ "id": 3, "firstName": "Ankit tiwari ", "lastName": "Agarwal", "hobby": "Reading" }`. The response is `200 OK` with the body `"user deleted successfully"`. The bottom panel shows the terminal output of the REST client, indicating the server is running at port 8086 and the request was successful.

Activity

- DEL localhost:8086/ 6 mins ago
- GET localhost:8081/ 21 mins ago
- PUT localhost:5081/ 3 hours ago
- GET localhost:8080 1 day ago
- POST localhost:8888 1 month ago

Request

DELETE `http://localhost:8086/user/3` **Send**

Body

JSON Content

```
1 {
2   "id": 3,
3   "firstName": "Ankit tiwari ",
4   "lastName": "Agarwal",
5   "hobby": "Reading"
6 }
```

Response

Status: 200 OK Size: 27 Bytes Time: 66 ms

1 "user deleted successfully"

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,
```