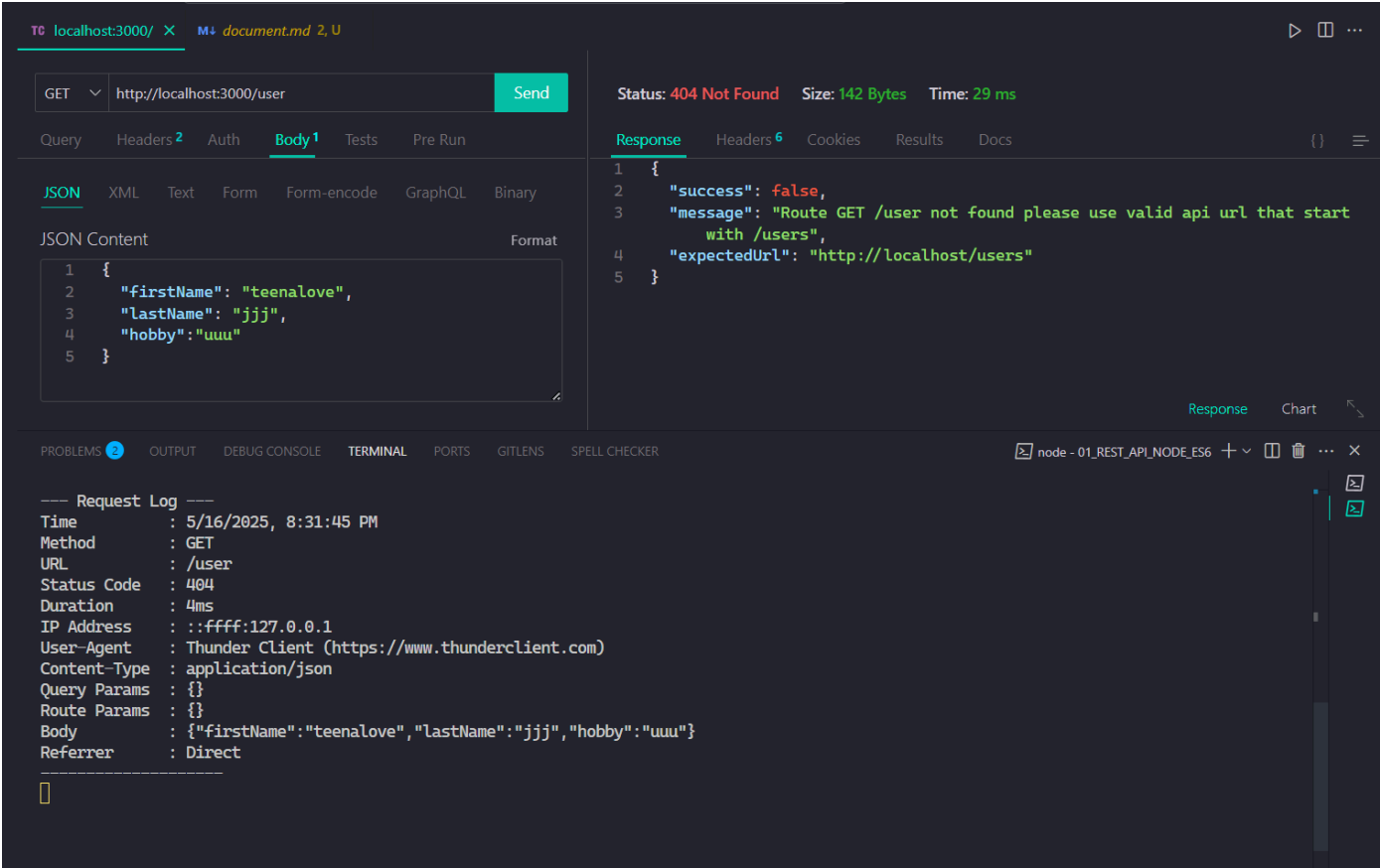


API Test Results Document

GET Request: Fetching All Users

(1) When API URL is `http://localhost:port/user`

- **Scenario:** API URL does **not** start with `/users` .
- **Expected:** Server returns an error due to invalid endpoint.
- **Actual:** Error thrown.
- **HTTP Status:** 404 Not Found



The screenshot displays the Thunder Client interface. The top bar shows the request method as GET and the URL as `http://localhost:3000/user`. The status bar indicates a 404 Not Found status, 142 bytes, and a response time of 29 ms. The response body is shown in JSON format, containing the following data:

```
{  "success": false,  "message": "Route GET /user not found please use valid api url that start with /users",  "expectedUrl": "http://localhost/users"}
```

The bottom panel shows the Request Log with the following details:

```
--- Request Log ---
Time       : 5/16/2025, 8:31:45 PM
Method     : GET
URL        : /user
Status Code : 404
Duration   : 4ms
IP Address  : ::ffff:127.0.0.1
User-Agent  : Thunder Client (https://www.thunderclient.com)
Content-Type : application/json
Query Params : {}
Route Params : {}
Body       : {"firstName": "teenalove", "lastName": "jjj", "hobby": "uuu"}
Referrer    : Direct
```

(2) When API URL is `http://localhost:port/users`

- **Scenario:** API URL is valid and starts with `/users` .
- **Expected:** Returns the list of all users.
- **Actual:** Successful response with user data.

- **HTTP Status: 200 OK**

localhost:3000/ X document.md 1, U

GET http://localhost:3000/users Send

Query Headers 2 Auth Body 1 Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

JSON Content Format

```
1 {
2   "firstName": "teenalove",
3   "lastName": "jjj",
4   "hobby": "uuu"
5 }
```

Status: 200 OK Size: 233 Bytes Time: 27 ms

Response Headers 6 Cookies Results Docs

```
1 {
2   "sucess": true,
3   "data": [
4     {
5       "id": "1",
6       "firstName": "ankit",
7       "lastName": "negi",
8       "hobby": "travelling"
9     },
10    {
11      "id": "2",
12      "firstName": "bingo",
13      "lastName": "live",
14      "hobby": "watching web-series"
15    }
16  ],
17  "message": "conguratulations all users fetched sucessfully"
18 }
```

Problems 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS SPELL CHECKER 1

Request Log

Time : 5/16/2025, 8:35:52 PM

Method : GET

URL : /users

Status Code : 200

Duration : 7ms

IP Address : ::ffff:127.0.0.1

User-Agent : Thunder Client (https://www.thunderclient.com)

Content-Type : application/json

Query Params : {}

Route Params : {}

Body : {"firstName":"teenalove","lastName":"jjj","hobby":"uuu"}

Referrer : Direct

node - 01_REST_API_NODE_ES6

GET Request: Fetching Single User Data

(1) When API URL is http://localhost:port/users/invalidId

- **Scenario:** The user ID in the URL is invalid.
- **Expected:** Server returns an error indicating invalid ID.
- **Actual:** Error thrown.
- **HTTP Status:** 400 Bad Request / 404 Not Found (based on implementation)

The screenshot shows the Thunder Client interface. The top bar indicates the target is `localhost:3000/` and the document is `document.md 1, U`. The request is a `GET` to `http://localhost:3000/users/5`. The response status is `404 Not Found` with a size of `71 Bytes` and a time of `5 ms`. The response body is a JSON object:

```
{
  "message": "Sorry User is not found with given id : 5",
  "success": false
}
```

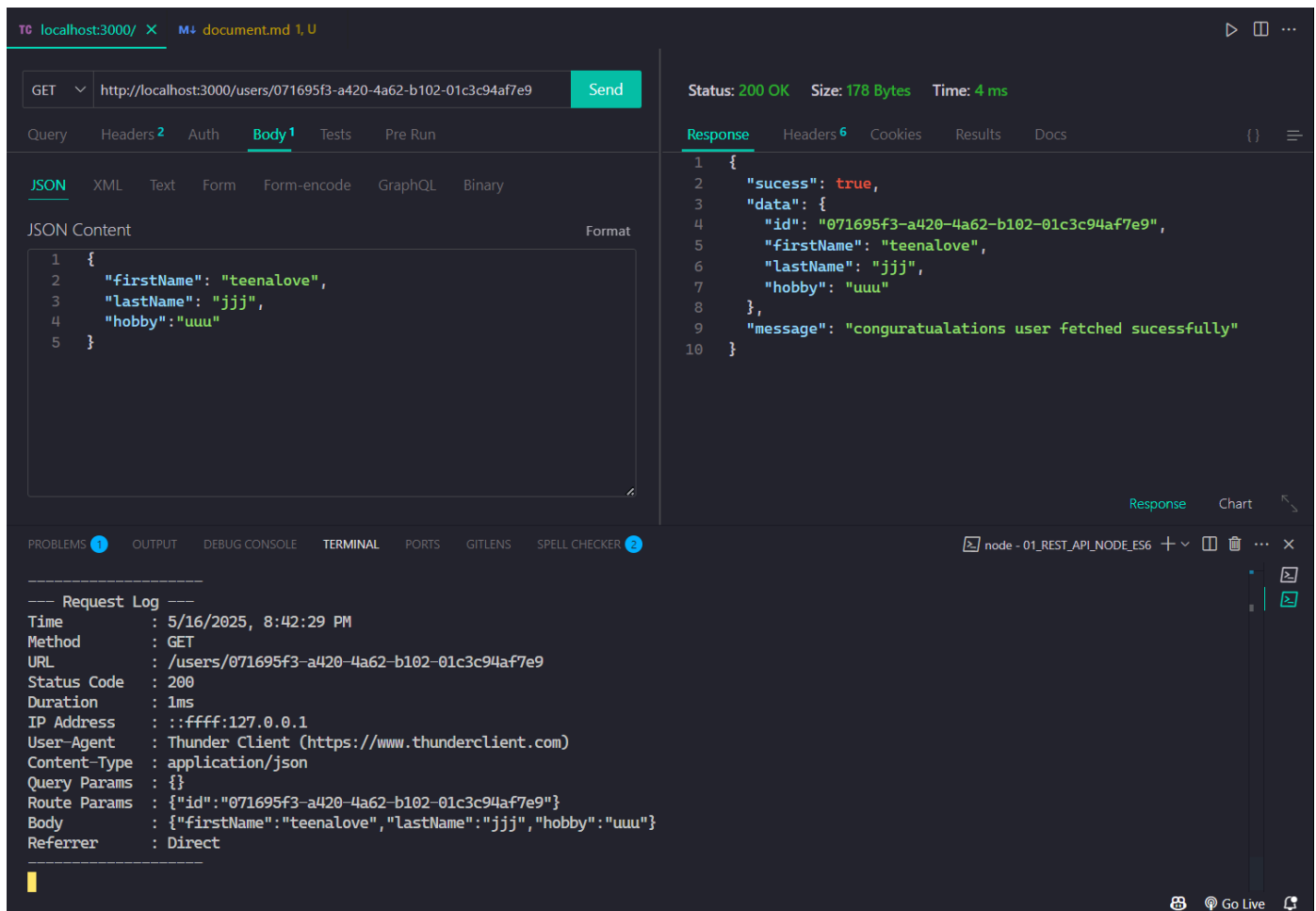
The bottom panel shows the terminal output:

```
no user found

error occurred in controller : Sorry User is not found with given id : 5
--- Request Log ---
Time       : 5/16/2025, 8:40:04 PM
Method     : GET
URL        : /users/5
Status Code : 404
Duration   : 2ms
IP Address  : ::ffff:127.0.0.1
User-Agent  : Thunder Client (https://www.thunderclient.com)
Content-Type : application/json
Query Params : {}
Route Params : {"id": "5"}
Body       : {"firstName": "teenalove", "lastName": "jjj", "hobby": "uuu"}
Referrer    : Direct
```

(2) When API URL is `http://localhost:port/users/validId`

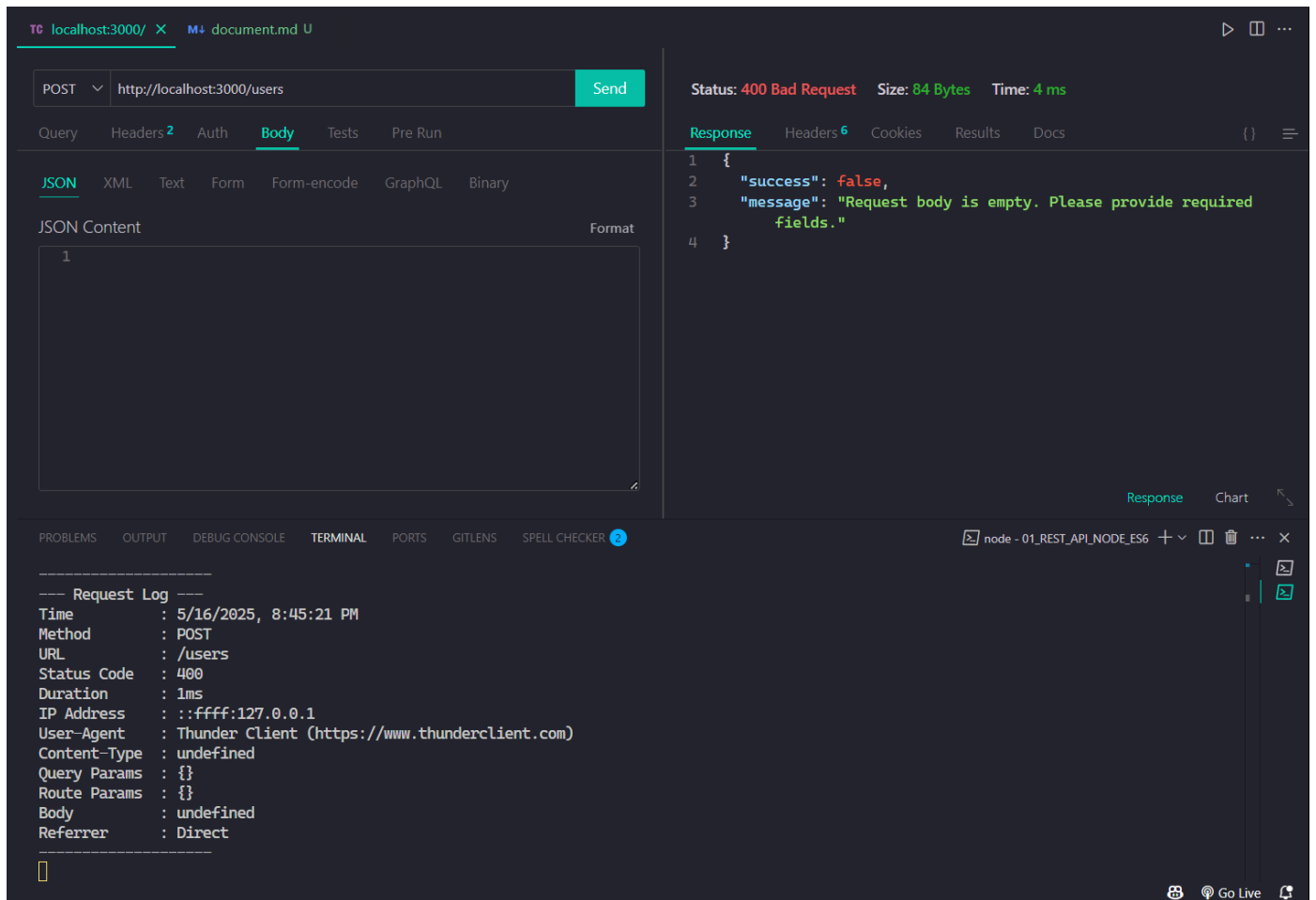
- **Scenario:** The user ID is valid.
- **Expected:** Returns the user data corresponding to the valid ID.
- **Actual:** Successful response with user data.
- **HTTP Status:** 200 OK



POST Request: Creating a User

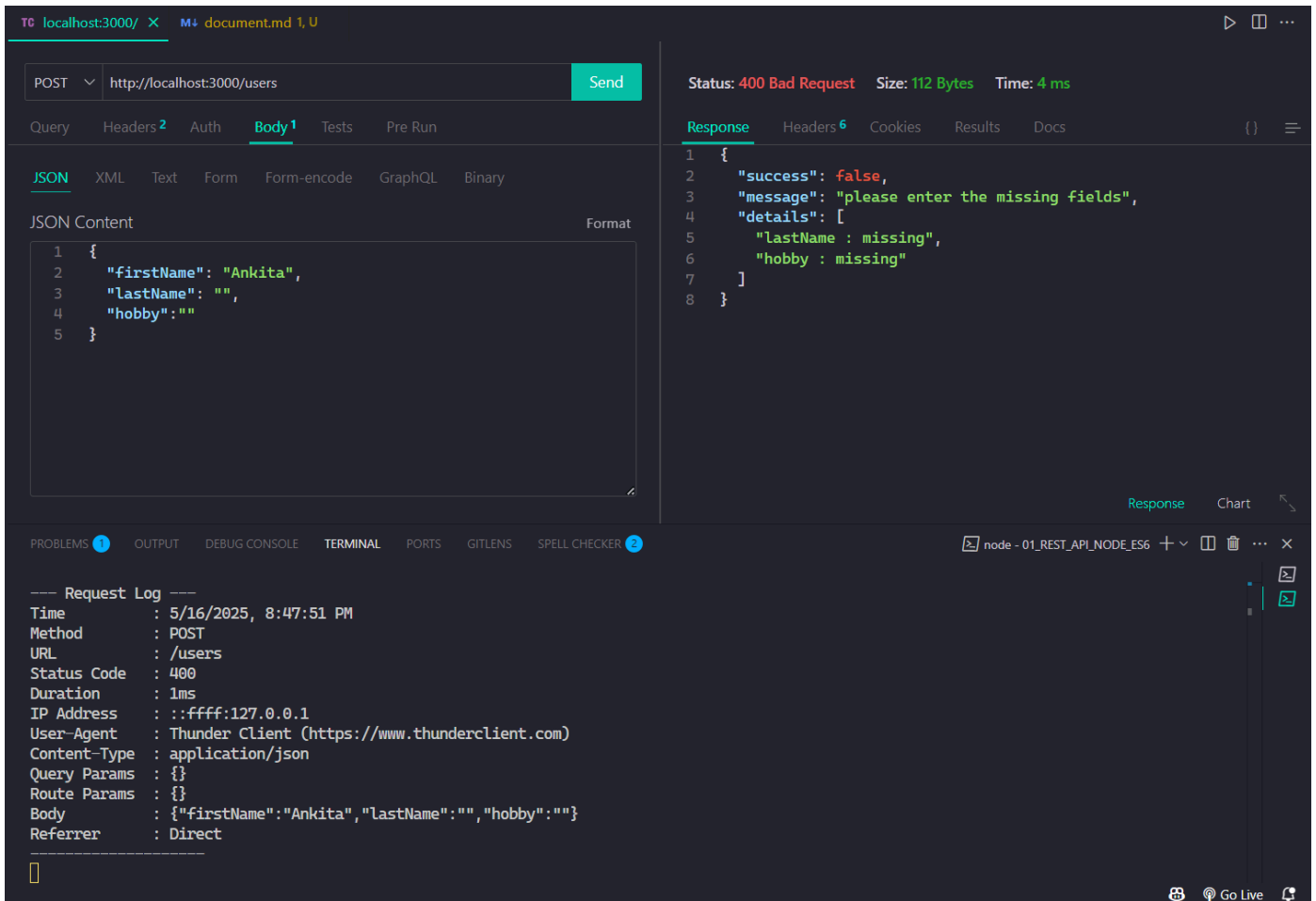
(1) When API URL is `http://localhost:port/users` — No request body provided

- **Scenario:** No request body sent.
- **Expected:** Server throws an error indicating missing request body.
- **Actual:** Error thrown.
- **HTTP Status:** 400 Bad Request



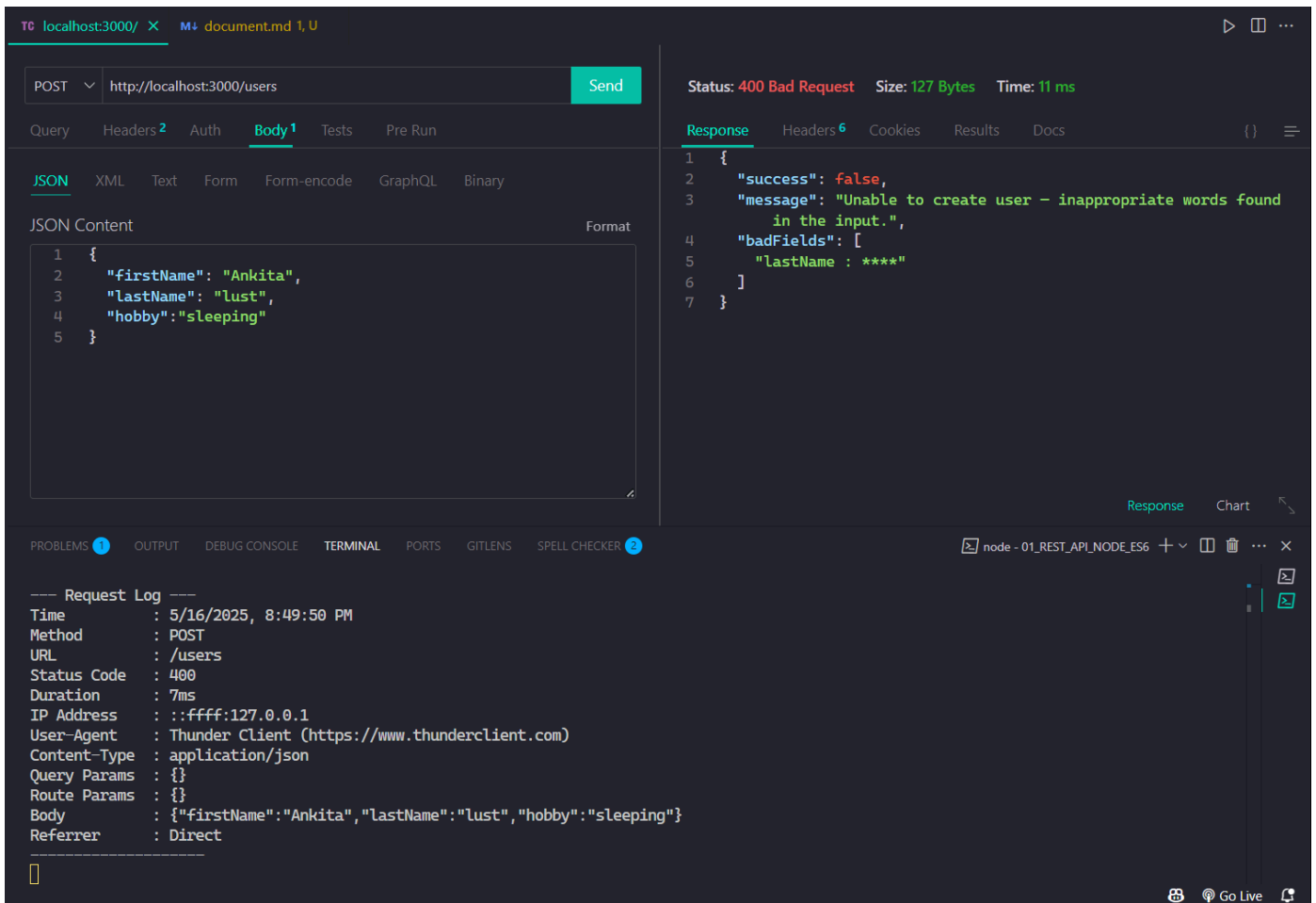
(2) When API URL is `http://localhost:port/users` — Some fields in the request body are empty

- **Scenario:** Request body is provided but some required fields are empty.
- **Expected:** Server returns an error indicating validation failure.
- **Actual:** Error thrown.
- **HTTP Status:** 400 Bad Request



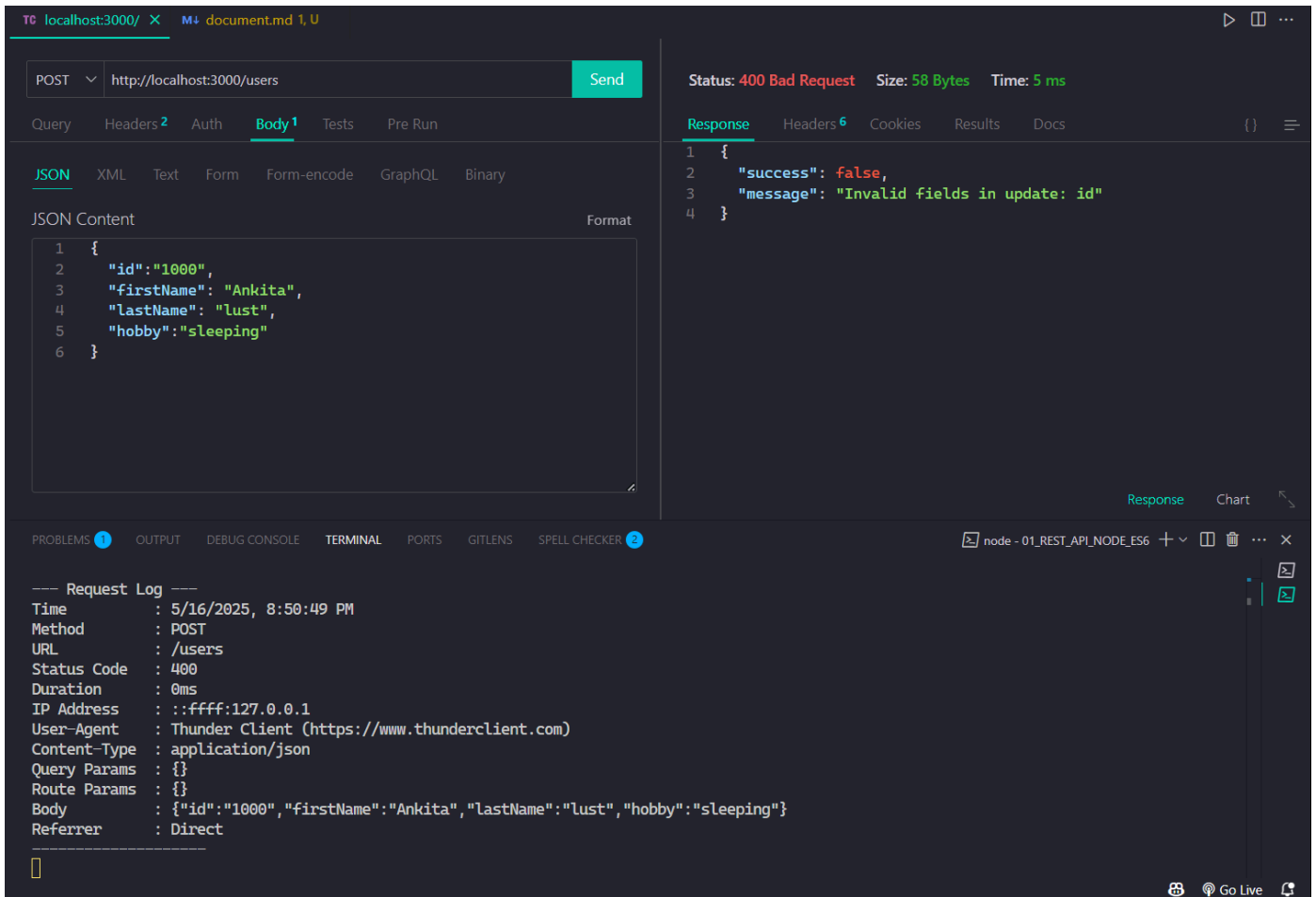
(3) When API URL is `http://localhost:port/users` — Some bad words are passed in request body

- **Scenario:** Request body contains inappropriate or bad words.
- **Expected:** Server rejects input with an error related to content policy.
- **Actual:** Error thrown.
- **HTTP Status:** 400 Bad Request



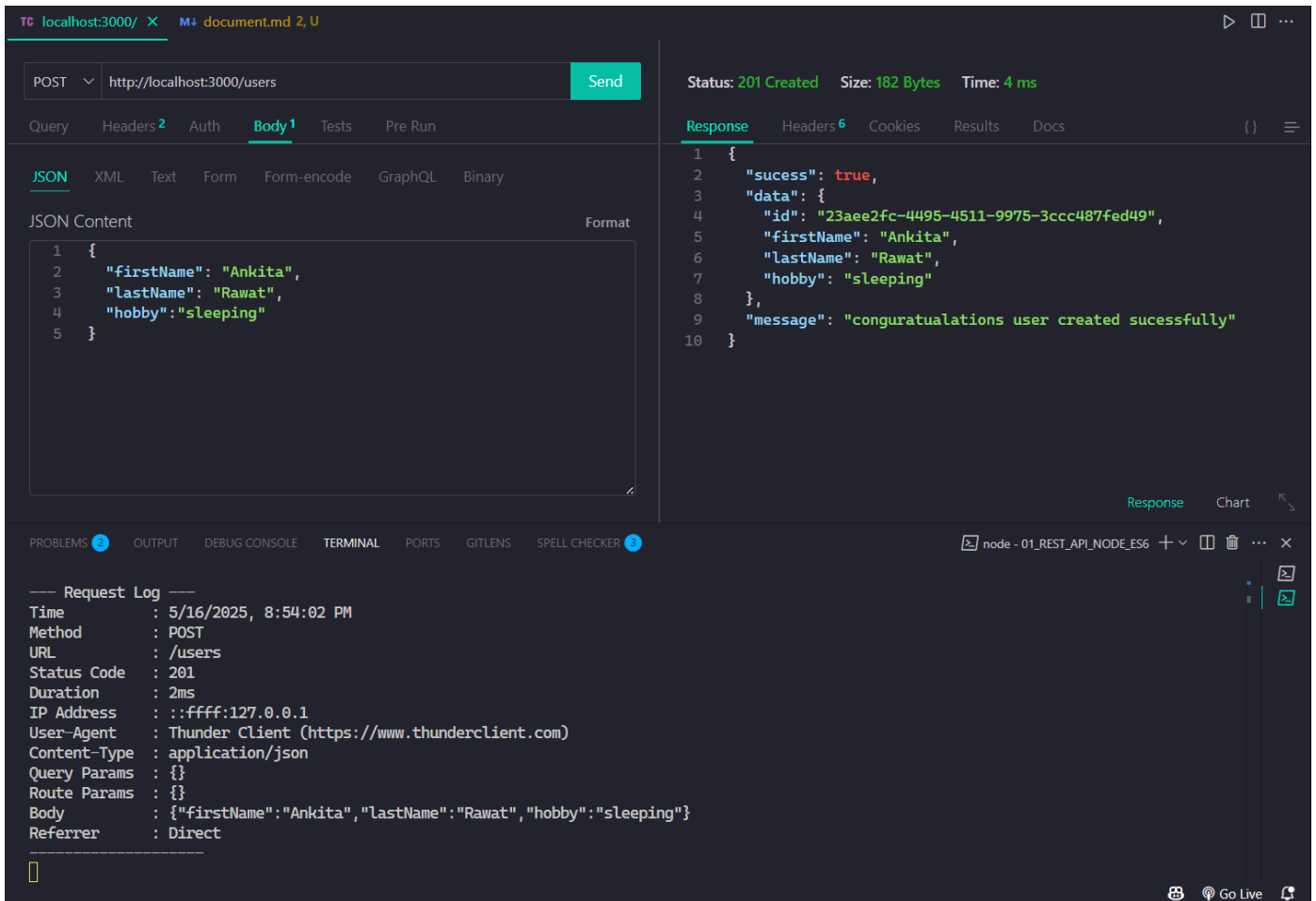
(4) When API URL is `http://localhost:port/users` — Invalid fields passed in request body

- **Scenario:** Request body contains fields that are not allowed, e.g., user tries to set the ID manually.
- **Expected:** Server ignores or throws error for invalid fields.
- **Actual:** Error thrown or invalid fields ignored.
- **HTTP Status:** 400 Bad Request



(5) When API URL is `http://localhost:port/users` — Correct fields passed

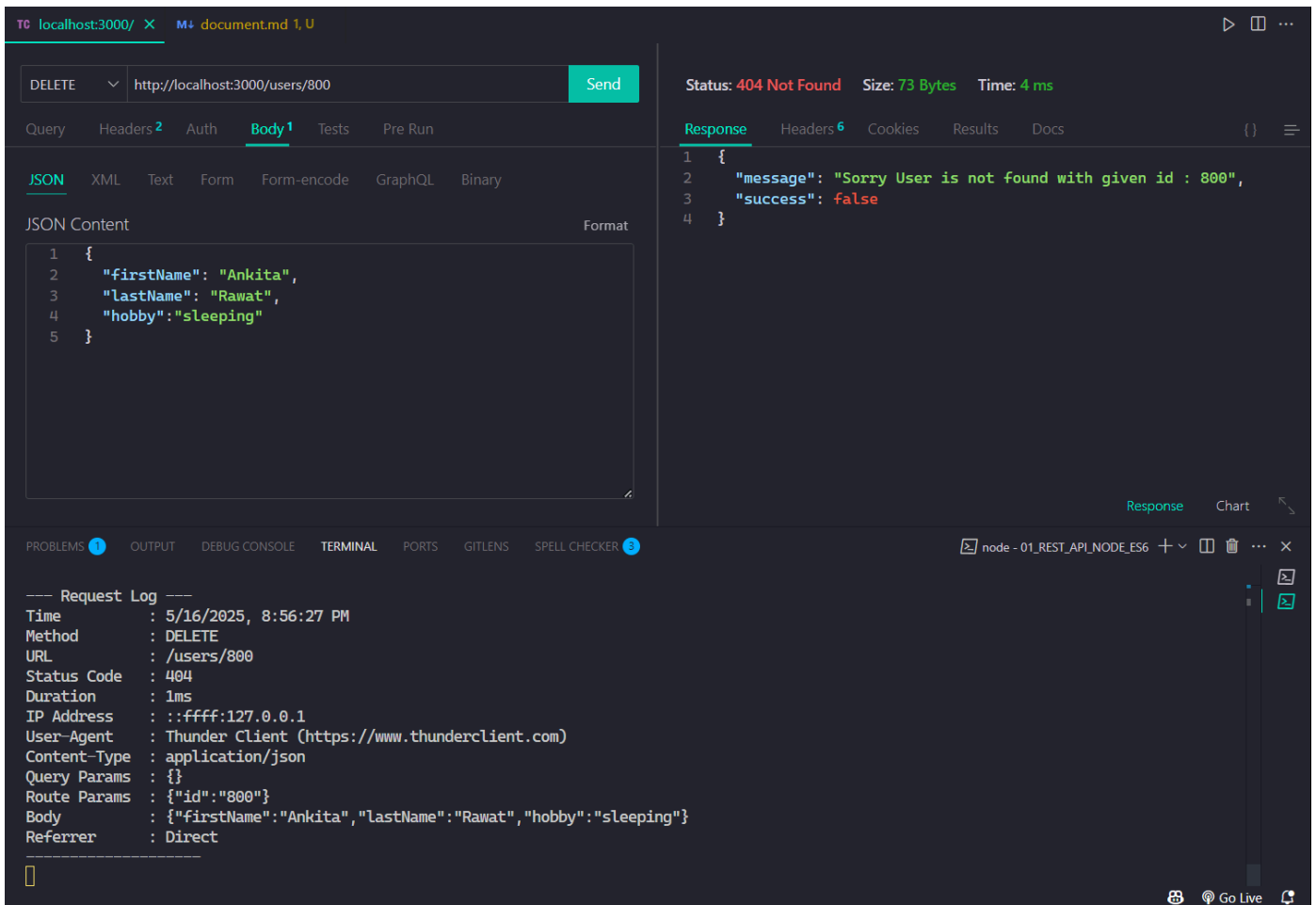
- **Scenario:** Request body contains all required fields correctly.
- **Expected:** User is created successfully.
- **Actual:** Success response.
- **HTTP Status:** 201 Created



DELETE Request: Deleting a User

(1) When API URL is `http://localhost:port/users/id` — Invalid ID passed

- **Scenario:** ID provided does not exist or is invalid.
- **Expected:** Server returns error for invalid ID.
- **Actual:** Error thrown.
- **HTTP Status:** 404 Not Found



(2) When API URL is `http://localhost:port/users/id` — Valid ID passed

- **Scenario:** Valid user ID provided.
- **Expected:** User is deleted successfully.
- **Actual:** Success response.
- **HTTP Status:** 200 OK

The screenshot shows the Thunder Client interface. At the top, the URL bar displays `http://localhost:3000/users/23aee2fc-4495-4511-9975-3ccc487fed49` with a `DELETE` method selected. The `Send` button is visible. Below the URL bar, the `Body` tab is active, showing a JSON request body:

```
1 {
2   "firstName": "Ankita",
3   "lastName": "Rawat",
4   "hobby": "sleeping"
5 }
```

The `Response` tab is also active, showing the server's response:

```
1 {
2   "sucess": true,
3   "data": {
4     "success": true,
5     "message": "User deleted successfully"
6   },
7   "message": "congratulations user deleted sucessfully"
8 }
```

At the bottom, the `Request Log` tab is open, displaying the following details:

```
----- Request Log -----
Time       : 5/16/2025, 8:58:33 PM
Method     : DELETE
URL        : /users/23aee2fc-4495-4511-9975-3ccc487fed49
Status Code : 200
Duration   : 1ms
IP Address  : ::ffff:127.0.0.1
User-Agent  : Thunder Client (https://www.thunderclient.com)
Content-Type : application/json
Query Params : {}
Route Params : {"id": "23aee2fc-4495-4511-9975-3ccc487fed49"}
Body       : {"firstName": "Ankita", "lastName": "Rawat", "hobby": "sleeping"}
Referrer    : Direct
```

(3) When API URL is `http://localhost:port/users/id` — Deleting a user already deleted

- **Scenario:** Attempting to delete a user that was already deleted.
- **Expected:** Server returns error indicating user not found.
- **Actual:** Error thrown.
- **HTTP Status:** 404 Not Found

TC localhost:3000/ X M+ document.md 2, U

DELETE http://localhost:3000/users/23aee2fc-4495-4511-9975-3ccc487fed49 Send

Status: 404 Not Found Size: 106 Bytes Time: 4 ms

Response Headers 6 Cookies Results Docs

```
1 {
2   "message": "Sorry User is not found with given id : 23aee2fc-4495-4511-9975-3ccc487fed49",
3   "success": false
4 }
```

JSON Content Format

```
1 {
2   "firstName": "Ankita",
3   "lastName": "Rawat",
4   "hobby": "sleeping"
5 }
```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS SPELL CHECKER 4

node - 01_REST_API_NODE_ES6

Request Log

Time : 5/16/2025, 8:58:33 PM

Method : DELETE

URL : /users/23aee2fc-4495-4511-9975-3ccc487fed49

Status Code : 200

Duration : 1ms

IP Address : ::ffff:127.0.0.1

User-Agent : Thunder Client (https://www.thunderclient.com)

Content-Type : application/json

Query Params : {}

Route Params : {"id": "23aee2fc-4495-4511-9975-3ccc487fed49"}

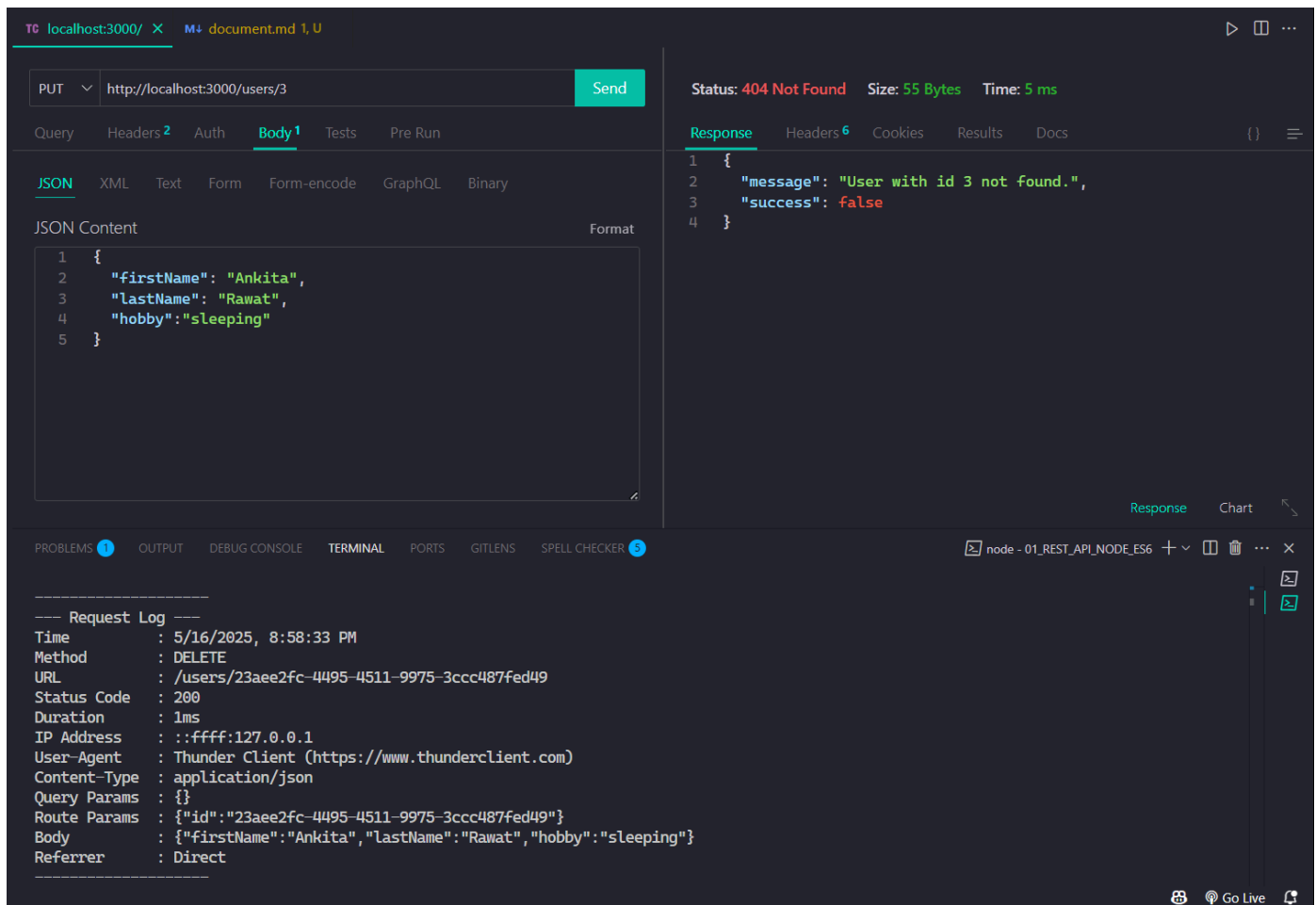
Body : {"firstName": "Ankita", "lastName": "Rawat", "hobby": "sleeping"}

Referrer : Direct

PUT Request: Updating Existing User

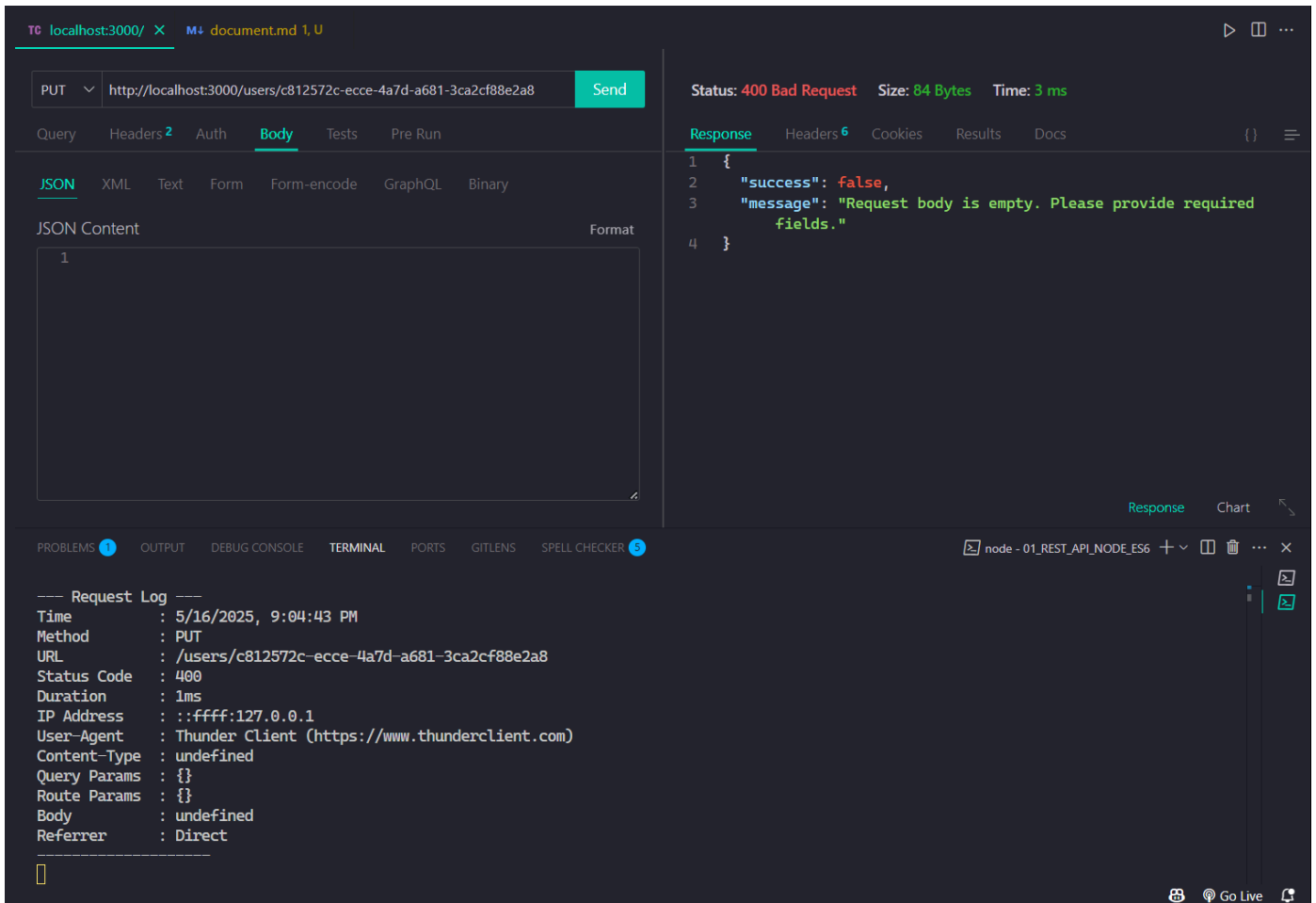
(1) When API URL is `http://localhost:port/users/id` — Invalid ID passed by user

- **Scenario:** User ID is invalid.
- **Expected:** Server returns error indicating invalid ID.
- **Actual:** Error thrown.
- **HTTP Status:** 400 Bad Request / 404 Not Found



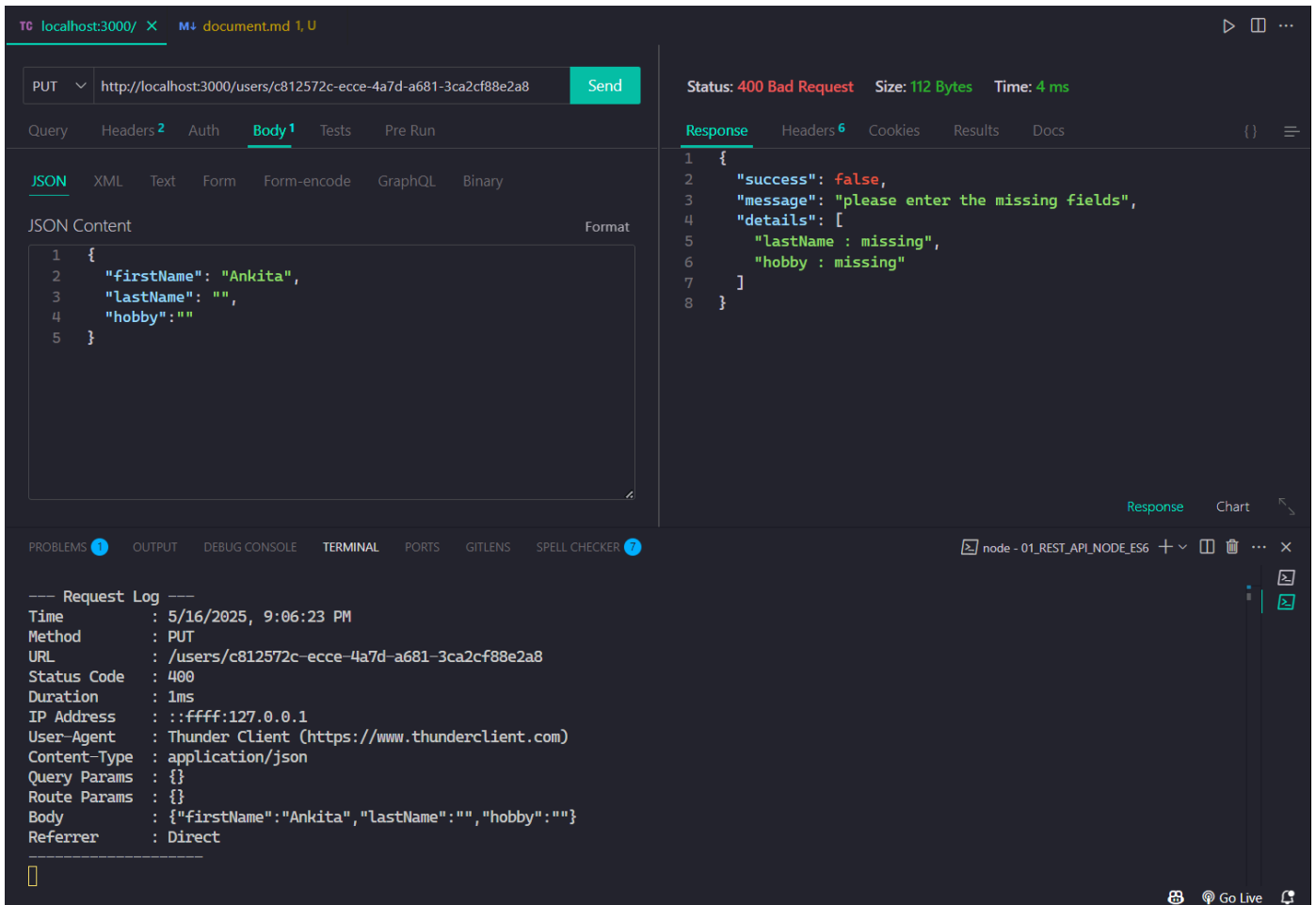
(2) When API URL is `http://localhost:port/users/id` — Valid ID but no request body passed

- **Scenario:** No request body sent with the update.
- **Expected:** Server returns error for missing update data.
- **Actual:** Error thrown.
- **HTTP Status:** 400 Bad Request



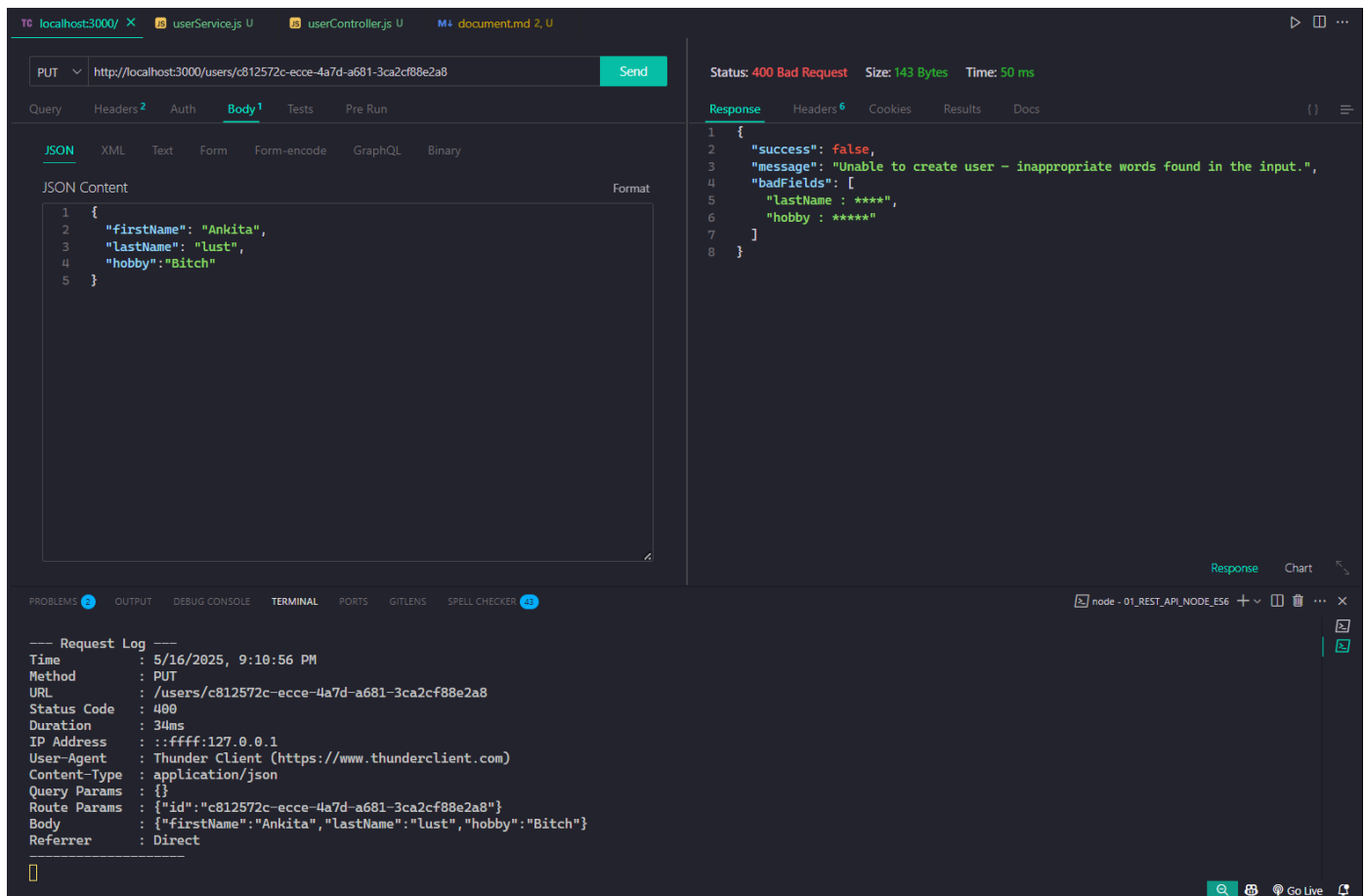
(3) When API URL is `http://localhost:port/users/id` — Valid ID but some fields are empty

- **Scenario:** Request body has some empty fields.
- **Expected:** Server returns validation error.
- **Actual:** Error thrown.
- **HTTP Status:** 400 Bad Request



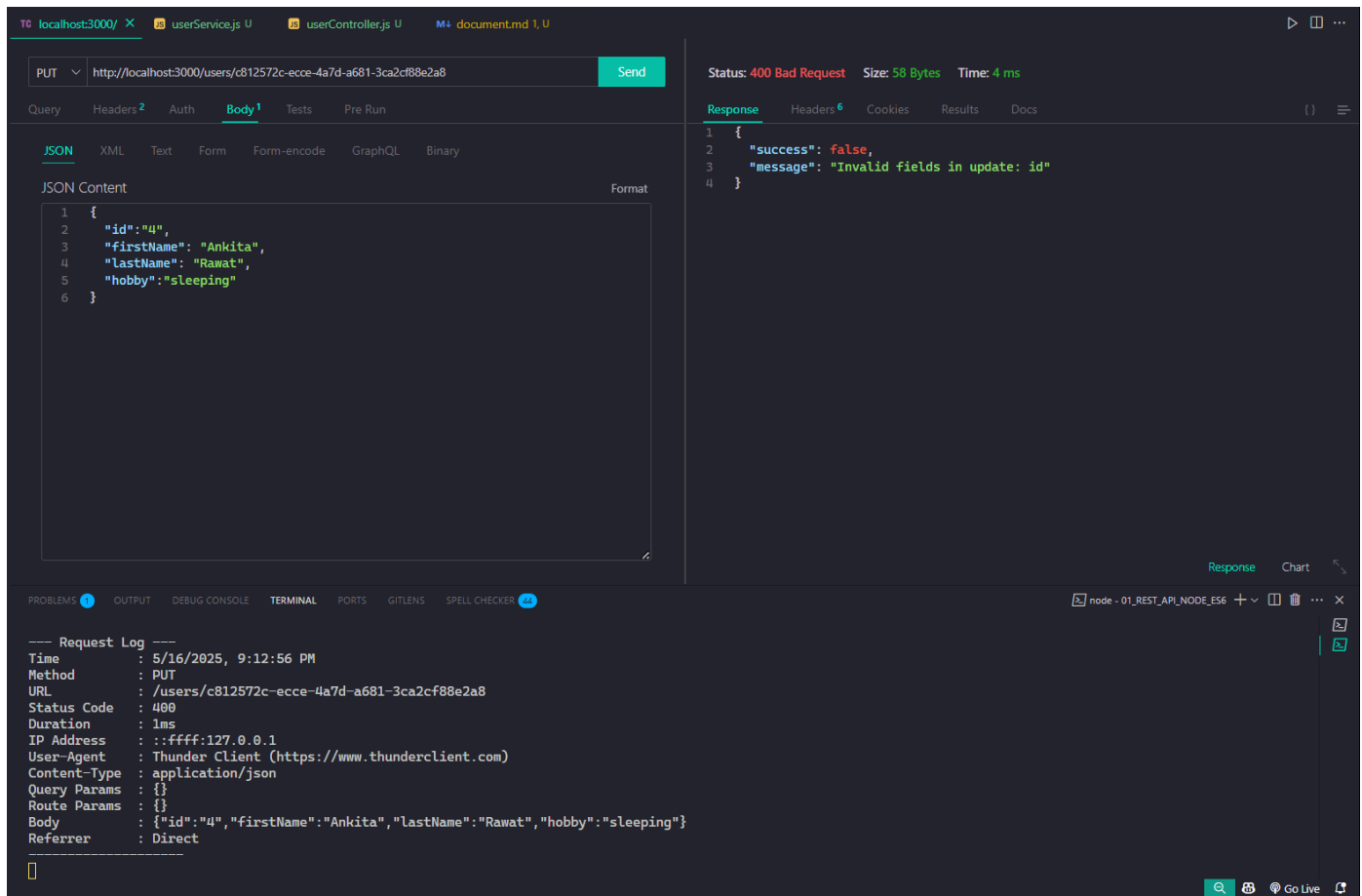
(4) When API URL is `http://localhost:port/users/id` — Valid ID but some fields contain bad words

- **Scenario:** Request body contains inappropriate content.
- **Expected:** Server rejects update due to bad content.
- **Actual:** Error thrown.
- **HTTP Status:** 400 Bad Request



(5) When API URL is http://localhost:port/users/id — Valid ID but extra invalid fields included

- **Scenario:** Request body contains fields not allowed to be updated.
- **Expected:** Server ignores or throws error for invalid fields.
- **Actual:** Error thrown or ignored.
- **HTTP Status:** 400 Bad Request



(6) When API URL is `http://localhost:port/users/id` — Valid ID and valid request body

- **Scenario:** All fields are valid and allowed.
- **Expected:** User updated successfully.
- **Actual:** Success response.
- **HTTP Status:** 200 OK

The screenshot displays the VS Code interface with a REST client request and response. The request is a PUT to `http://localhost:3000/users/c812572c-ecce-4a7d-a681-3ca2cf88e2a8` with a JSON body. The response is a 200 OK status with a JSON body indicating success.

Request:

```
PUT http://localhost:3000/users/c812572c-ecce-4a7d-a681-3ca2cf88e2a8
```

JSON Content:

```
{
  "firstName": "Ankita",
  "lastName": "Rawat",
  "hobby": "sleeping"
}
```

Response:

```
{
  "sucess": true,
  "message": "congratualations user info updated sucessfully"
}
```

Request Log:

```
Time : 5/16/2025, 9:14:34 PM
Method : PUT
URL : /users/c812572c-ecce-4a7d-a681-3ca2cf88e2a8
Status Code : 200
Duration : 2ms
IP Address : ::ffff:127.0.0.1
User-Agent : Thunder Client (https://www.thunderclient.com)
Content-Type : application/json
Query Params : {}
Route Params : {"id":"c812572c-ecce-4a7d-a681-3ca2cf88e2a8"}
Body : {"firstName":"Ankita","lastName":"Rawat","hobby":"sleeping"}
Referrer : Direct
```