

# Quick Demo

## User SignUp

The screenshot shows the Postman application interface. On the left, the 'Body' tab is selected, displaying a JSON payload for user sign-up:

```
1 {
2   "name": "Ankit Chahal",
3   "email": "abcd13@gmail.com",
4   "password": "12345"
5 }
```

On the right, the response section shows a status of 200 OK, size of 0 bytes, and time of 25 ms. The response body contains a single key '1':

```
1
```

## UserLogin

The screenshot shows the Postman application interface. On the left, the 'Body' tab is selected, displaying a JSON payload for user login:

```
1 {
2   "email": "abcd13@gmail.com",
3   "password": "12345"
4 }
```

On the right, the response section shows a status of 200 OK, size of 0 bytes, and time of 21 ms. The response body contains a single key '1' with a value 'sid':

```
1 {
2   "sid": "9e844032-e0c3-4847-8aa3-e0e31ed8553b"
3 }
```

## Create Note

POST  http://localhost:8080/v1/notes

| Query                | Headers <small>2</small> | Auth                 | <b>Body <small>1</small></b> | Tests                       | Pre Run                 |                        |
|----------------------|--------------------------|----------------------|------------------------------|-----------------------------|-------------------------|------------------------|
| <a href="#">JSON</a> | <a href="#">XML</a>      | <a href="#">Text</a> | <a href="#">Form</a>         | <a href="#">Form-encode</a> | <a href="#">GraphQL</a> | <a href="#">Binary</a> |

JSON Content

```

1  {
2    "sid": "9e844032-e0c3-4847-8aa3-e0e31ed8553b",
3    "note": "my note is this part 2"
4  }

```

Status: **200 OK** Size: **10 Bytes** Time: **21 ms**

**Response** Headers 4 Cookies Results Docs { }

```

1  {
2    "id": "9"
3  }

```

## Get Note

GET  http://localhost:8080/v1/notes

| Query                | Headers <small>2</small> | Auth                 | <b>Body <small>1</small></b> | Tests                       | Pre Run                 |                        |
|----------------------|--------------------------|----------------------|------------------------------|-----------------------------|-------------------------|------------------------|
| <a href="#">JSON</a> | <a href="#">XML</a>      | <a href="#">Text</a> | <a href="#">Form</a>         | <a href="#">Form-encode</a> | <a href="#">GraphQL</a> | <a href="#">Binary</a> |

JSON Content

```

1  {
2    "sid": "9e844032-e0c3-4847-8aa3-e0e31ed8553b"
3  }

```

Status: **200 OK** Size: **223 Bytes** Time: **19 ms**

**Response** Headers 4 Cookies Results Docs { }

```

1  [
2    {
3      "id": "1",
4      "note": "hello",
5    },
6    {
7      "id": "2",
8      "note": "hello",
9    },
10   {
11     "id": "5",
12     "note": "helllo",
13   },
14   {
15     "id": "6",
16     "note": "helllo",
17   },
18   {
19     "id": "7",
20     "note": "helllo",
21   },
22   {
23     "id": "8",
24     "note": "my note is this",
25   },
26   {
27     "id": "9",
28     "note": "my note is this part 2",
29   }
30 ]
31 ]
32 ]

```

## Delete Note with id 9

The screenshot shows the Postman interface. In the top left, there's a dropdown menu with 'DELETE' selected, followed by a URL field containing 'http://localhost:8080/v1/notes'. To the right of the URL is a blue 'Send' button. Below the URL field, there are tabs for 'Query', 'Headers 2', 'Auth', 'Body 1', 'Tests', and 'Pre Run'. The 'Body 1' tab is active and has a sub-tab 'JSON' selected. The JSON content is a single-line object:

```
1 {  
2   "sid": "f72e5d86-ac65-4a0e-91a8-f23eee0361de",  
3   "id": "9"  
4 }
```

In the top right corner, the status bar shows 'Status: 200 OK' in green, 'Size: 0 Bytes', and 'Time: 23 ms'. Below the status bar, there are tabs for 'Response', 'Headers 3', 'Cookies', 'Results', and 'Docs'. The 'Response' tab is active, showing the number '1'.

On calling the Get Note API, we are seeing note with id 9 as it was deleted in previous step

The screenshot shows the Postman interface. In the top left, there's a dropdown menu with 'GET' selected, followed by a URL field containing 'http://localhost:8080/v1/notes'. To the right of the URL is a blue 'Send' button. Below the URL field, there are tabs for 'Query', 'Headers 2', 'Auth', 'Body 1', 'Tests', and 'Pre Run'. The 'Body 1' tab is active and has a sub-tab 'JSON' selected. The JSON content is a single-line object:

```
1 {  
2   "sid": "f72e5d86-ac65-4a0e-91a8-f23eee0361de"  
3 }
```

In the top right corner, the status bar shows 'Status: 200 OK' in green, 'Size: 180 Bytes', and 'Time: 19 ms'. Below the status bar, there are tabs for 'Response', 'Headers 4', 'Cookies', 'Results', and 'Docs'. The 'Response' tab is active, showing a large JSON object representing a list of notes. A 'Copy' button is visible next to the response content. The JSON content is as follows:

```
1 {  
2   "notes": [  
3     {  
4       "note": "hello",  
5       "id": "1"  
6     },  
7     {  
8       "note": "hello",  
9       "id": "2"  
10    },  
11    {  
12      "note": "helllo",  
13      "id": "5"  
14    },  
15    {  
16      "note": "helllo",  
17      "id": "6"  
18    },  
19    {  
20      "note": "helllo",  
21      "id": "7"  
22    },  
23    {  
24      "note": "my note is this",  
25      "id": "8"  
26    }  
27  ]  
28 }
```

Calling the Delete API once, session ID is expired.

DELETE ▼ http://localhost:8080/v1/notes Send

| Query   | Headers <span style="color: #007bff;">2</span> | Auth                 | Body <span style="color: #007bff;">1</span> | Tests                       | Pre Run                 |                        |
|---|--|----------------------|---|-----------------------------|-------------------------|------------------------|
| <a href="#">JSON</a>  | <a href="#">XML</a>                            | <a href="#">Text</a> | <a href="#">Form</a>                        | <a href="#">Form-encode</a> | <a href="#">GraphQL</a> | <a href="#">Binary</a> |
| JSON Content  |  |                      | Format                                      |                             |                         |                        |
| <pre> 1  { 2    "sid": "f72e5d86-ac65-4a0e-91a8-f23eee0361de", 3    "id": "6" 4  } </pre> |  |                      |   |                             |                         |                        |

Status: **401 Unauthorized** Size: **120 Bytes** Time: **18 ms**

| Response  | Headers <span style="color: #007bff;">4</span> | Cookies | Results | Docs | {} ▾ |
|---|--|---------|---------|------|------|
| <pre> 1  { 2    "code": 401, 3    "message": "Session ID is invalid, user is not 4      authenticated", 5    "trace": "51abdeed-1080-4038-be81-678b80df3cd0" </pre> |  |         |         |      |      |

## Logs

```

2023-07-24T21:52:52.163+05:00 INFO service/service.go:120 received request for user sign-up, txid : 0ef5120a-87af-4677-b9b9-2f48a4d7af3c
2023-07-24T21:52:52.164+05:00 INFO service/service.go:123 user request is unmarshalled successfully for user sign-up, txid : 0ef5120a-87af-4677-b9b9-2f48a4d7af3c
2023-07-24T21:52:52.170+05:00 INFO service/service.go:132 user sign up is successful, txid : 0ef5120a-87af-4677-b9b9-2f48a4d7af3c
2023-07-24T21:53:24.715+05:00 INFO service/service.go:38 received request for user login, txid : a9c8baef-a869-4979-9720-b887589595a5
2023-07-24T21:53:24.715+05:00 INFO service/service.go:41 user request is unmarshalled successfully, txid : a9c8baef-a869-4979-9720-b887589595a5
2023-07-24T21:53:24.715+05:00 INFO service/service.go:44 calling db layer for user login, txid : a9c8baef-a869-4979-9720-b887589595a5
2023-07-24T21:53:24.715+05:00 INFO service/service.go:79 user login is successful, txid : a9c8baef-a869-4979-9720-b887589595a5
notes : (7e77910-0e09-a4bf-ab5d-5b10e644991d) my note is this )
2023-07-24T21:53:49.802+05:00 INFO service/service.go:178 received request for note creation, txid : 14487795-9800-40dc-a8bb-d0a8a05dc046
2023-07-24T21:53:49.802+05:00 INFO service/service.go:180 user request is unmarshalled successfully for note creation, txid : 14487795-9800-40dc-a8bb-d0a8a05dc046
2023-07-24T21:53:49.802+05:00 INFO service/service.go:128 receiving request for note creation, txid : d8a9a89dc9c046
2023-07-24T21:53:49.802+05:00 INFO db/notes.go:30 successfully added notes entry in db, txid : 14487795-9800-40dc-a8bb-d0a8a05dc046
notes : (7e77910-0e09-a4bf-ab5d-5b10e644991d)
2023-07-24T21:54:18.801+05:00 INFO service/service.go:245 received request for fetching all the notes, txid : 436df4d2-5aca-4edb-a62-931b44148e2a
2023-07-24T21:54:18.801+05:00 INFO service/service.go:248 user request unmarshalled successfully for fetching all the notes, txid : 436df4d2-5aca-4edb-a62-931b44148e2a
2023-07-24T21:54:18.801+05:00 INFO db/notes.go:80 successfully fetched all the notes from db, txid : 436df4d2-5aca-4edb-a62-931b44148e2a
2023-07-24T21:54:18.801+05:00 INFO service/service.go:123 user request has successfully fetched all the notes, txid : 436df4d2-5aca-4edb-a62-931b44148e2a
2023-07-24T21:55:01.555+05:00 INFO service/service.go:128 receiving request for user sign-up, txid : d8161936-fb86-4f46-a407-5dc1d9d4ca78
2023-07-24T21:55:01.555+05:00 INFO service/service.go:123 user request is unmarshalled successfully for user sign-up, txid : d8161936-fb86-4f46-a407-5dc1d9d4ca78
2023-07-24T21:55:01.555+05:00 INFO service/service.go:162 calling db layer for user sign-up, txid : d8161936-fb86-4f46-a407-5dc1d9d4ca78
2023-07-24T21:55:01.555+05:00 INFO service/service.go:211 calling db layer for note creation, txid : 14487795-9800-40dc-a8bb-d0a8a05dc046
2023-07-24T21:55:01.555+05:00 INFO db/notes.go:38 receiving request for note creation, txid : 14487795-9800-40dc-a8bb-d0a8a05dc046
notes : (5a7b794ec-e7b6-4b8d-b983-888b81d305b)
2023-07-24T21:55:19.145+05:00 INFO service/service.go:41 user request is unmarshalled successfully for user login, txid : 5a7b794ec-e7b6-4b8d-b983-888b81d305b
2023-07-24T21:55:19.145+05:00 INFO service/service.go:99 calling db layer for user login, txid : 5a7b794ec-e7b6-4b8d-b983-888b81d305b
notes : (988a4b32-e033-4847-8aa3-8e031ed5d533)
2023-07-24T21:55:19.145+05:00 INFO service/service.go:78 user login is successful, txid : 5a7b794ec-e7b6-4b8d-b983-888b81d305b
notes : (7f23ee031ed5d533)
2023-07-24T21:55:48.291+05:00 INFO service/service.go:178 receiving request for note creation, txid : c96952b3-7fca-429c-8426-8de8d0cb12ab
2023-07-24T21:55:48.291+05:00 INFO service/service.go:180 user request for note creation is unmarshalled successfully, txid : c96952b3-7fca-429c-8426-8de8d0cb12ab
2023-07-24T21:55:48.291+05:00 INFO service/service.go:211 calling db layer for note creation, txid : c96952b3-7fca-429c-8426-8de8d0cb12ab
2023-07-24T21:55:48.291+05:00 INFO db/notes.go:39 successfully added notes entry in db, txid : c96952b3-7fca-429c-8426-8de8d0cb12ab
notes : (988a4b32-e033-4847-8aa3-8e031ed5d533)
2023-07-24T21:55:48.757+05:00 INFO service/service.go:245 received request for fetching all the notes, txid : 3462bf4c-3fb0-4e3b-962a-b8eac2c1c9fc
2023-07-24T21:55:48.757+05:00 INFO service/service.go:248 user request unmarshalled successfully for fetching all the notes, txid : 3462bf4c-3fb0-4e3b-962a-b8eac2c1c9fc
2023-07-24T21:55:48.757+05:00 INFO db/notes.go:80 calling db layer for note creation, txid : 3462bf4c-3fb0-4e3b-962a-b8eac2c1c9fc
2023-07-24T21:55:48.757+05:00 INFO db/notes.go:38 receiving request for note creation, txid : 3462bf4c-3fb0-4e3b-962a-b8eac2c1c9fc
2023-07-24T21:55:48.757+05:00 INFO db/notes.go:80 successfully fetched all the notes from db, txid : 3462bf4c-3fb0-4e3b-962a-b8eac2c1c9fc
notes : (7f23ee031ed5d533)
2023-07-24T21:55:59.759+05:00 INFO service/service.go:284 user has successfully fetched all the notes, txid : 3462bf4c-3fb0-4e3b-962a-b8eac2c1c9fc
notes : (7f23ee031ed5d533)
2023-07-24T21:56:42.120+05:00 INFO service/service.go:222 received request for deleting a note, txid : b92d58ff-f5b1-4ed9-9243-2d0d9b645f5b
2023-07-24T21:56:42.120+05:00 INFO service/service.go:222 user request unmarshalled successfully for deleting a note, txid : b92d58ff-f5b1-4ed9-9243-2d0d9b645f5b
2023-07-24T21:56:42.120+05:00 INFO service/service.go:258 calling db layer for deleting a note, txid : b92d58ff-f5b1-4ed9-9243-2d0d9b645f5b
2023-07-24T21:56:42.120+05:00 INFO db/notes.go:80 successfully deleted notes entry from db, txid : 48894259-a3f1-a4e7-9a55-d48494a398c2
2023-07-24T21:56:42.120+05:00 INFO service/service.go:235 user has successfully deleted a note, txid : 48894259-a3f1-a4e7-9a55-d48494a398c2
notes : (f72e5d86-ac65-4a0e-91a8-f23eee0361de)
2023-07-24T21:57:24.208+05:00 INFO service/service.go:265 received request for fetching all the notes, txid : 6cb6db44-922a-4fae-8c49-d6bf6aaa296f

```

## DB

```
select * from notes;
```

Disconnect

Table Output Messages Notifications



| <b>id</b><br>[PK] integer | <b>note</b><br>character varying |
|---------------------------|----------------------------------|
| 1                         | hello                            |
| 2                         | hello                            |
| 5                         | hell1o                           |
| 6                         | hell1o                           |
| 7                         | hell1o                           |
| 8                         | my note is this                  |

```
select * from users;
```

Disconnect

ta Output    Messages    Notifications

The screenshot shows a database interface with a query editor at the top containing the SQL command `select * from users;`. Below the editor is a toolbar with various icons. The main area displays a table with three columns: `emailid`, `name`, and `password`. The data in the table is as follows:

| emailid                                   | name         | password |
|---|--------------|----------|
| [PK] character varying<br>abcd9@gmail.com | Ankit Chahal | 12345    |
| abcd10@gmail.com                          | Ankit Chahal | 12345    |
| abcd11@gmail.com                          | Ankit Chahal | 12345    |
| abcd12@gmail.com                          | Ankit Chahal | 12345    |
| abcd13@gmail.com                          | Ankit Chahal | 12345    |

Neagтив scenario

EmailId is missing while user is trying to login

POST  http://localhost:8080/v1/login

Query Headers [2](#) Auth [Body \*\*1\*\*](#) Tests Pre Run

[JSON](#) XML Text Form Form-encode GraphQL Binary

JSON Content Format

```

1 {
2   "password": "12345"
3 }
```

Status: **400 Bad Request** Size: **90 Bytes** Time: **21 ms**

[Response](#) Headers [4](#) Cookies Results Docs { } [☰](#)

```

1 {
2   "code": 400,
3   "message": "EmailId is missing",
4   "trace": "1522d652-cd88-4ceb-b9bd-30ad36ee72d9"
5 }
```

Name is missing while user is trying to call sign-up api

POST  http://localhost:8080/v1/signup

Query Headers [2](#) Auth [Body \*\*1\*\*](#) Tests Pre Run

[JSON](#) XML Text Form Form-encode GraphQL Binary

JSON Content Format

```

1 {
2   "email": "abcd13@gmail.com",
3   "password": "12345"
4 }
```

Status: **400 Bad Request** Size: **87 Bytes** Time: **28 ms**

[Response](#) Headers [4](#) Cookies Results Docs { } [☰](#)

```

1 {
2   "code": 400,
3   "message": "name is missing",
4   "trace": "6bd8cdd6-deec-4d54-a37c-5f6a4f21fc63"
5 }
```

Note is missing while trying to call the create Note API

POST  http://localhost:8080/v1/notes

Query Headers [2](#) Auth [Body \*\*1\*\*](#) Tests Pre Run

[JSON](#) XML Text Form Form-encode GraphQL Binary

JSON Content Format

```

1 {
2   "sid": "6a9fb0df-71fb-47b6-9b5c-41ed8aafc595"
3 }
```

Status: **400 Bad Request** Size: **85 Bytes** Time: **22 ms**

[Response](#) Headers [4](#) Cookies Results Docs { } [☰](#)

```

1 {
2   "code": 400,
3   "message": "note is empty",
4   "trace": "33c040da-cd59-4b8e-84dc-7cc78b17c5af"
5 }
```

Id is missing while trying to call Delete Note API

The screenshot shows a REST client interface with the following details:

- Request URL:** `http://localhost:8080/v1/notes`
- Method:** `DELETE`
- Body Content:**

```
1 {
2   "sid": "6a9fb0df-71fb-47b6-9b5c-41ed8aafc595"
3 }
```
- Response Status:** `400 Bad Request`
- Response Headers:** `Content-Type: application/json`
- Response Body:**

```
1 {
2   "code": 400,
3   "message": "id is missing",
4   "trace": "6caad275-1358-4610-bbdb-7936b8fc325"
5 }
```

And similarly there will be other scenarios also.