

1) Create a sample table in the Database in MYSQL using queries.

The screenshot shows a MySQL IDE interface. The main editor window displays the following SQL queries:

```
1 • CREATE database db3;  
2  
3 • USE db3;  
4  
5 • CREATE TABLE userinfo (  
6     id INT AUTO_INCREMENT PRIMARY KEY,  
7     name VARCHAR(255) NOT NULL,  
8     age INT NOT NULL  
9 );  
10
```

On the right side, there is a help message: "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help."

At the bottom, the "Output" panel is visible, showing the "Action Output" tab. It contains a table with the execution results of the queries:

#	Time	Action	Message	Duration / Fetch
✗ 1	13:13:41	USE db3	Error Code: 1049. Unknown database 'db3'	0.000 sec
✓ 2	13:14:17	CREATE database db3	1 row(s) affected	0.015 sec
✓ 3	13:14:17	USE db3	0 row(s) affected	0.000 sec
✓ 4	13:14:17	CREATE TABLE userinfo ( id INT AUTO_INCRE...	0 row(s) affected	0.047 sec

2) Create sample API for Post, Put, Get, and Delete methods.

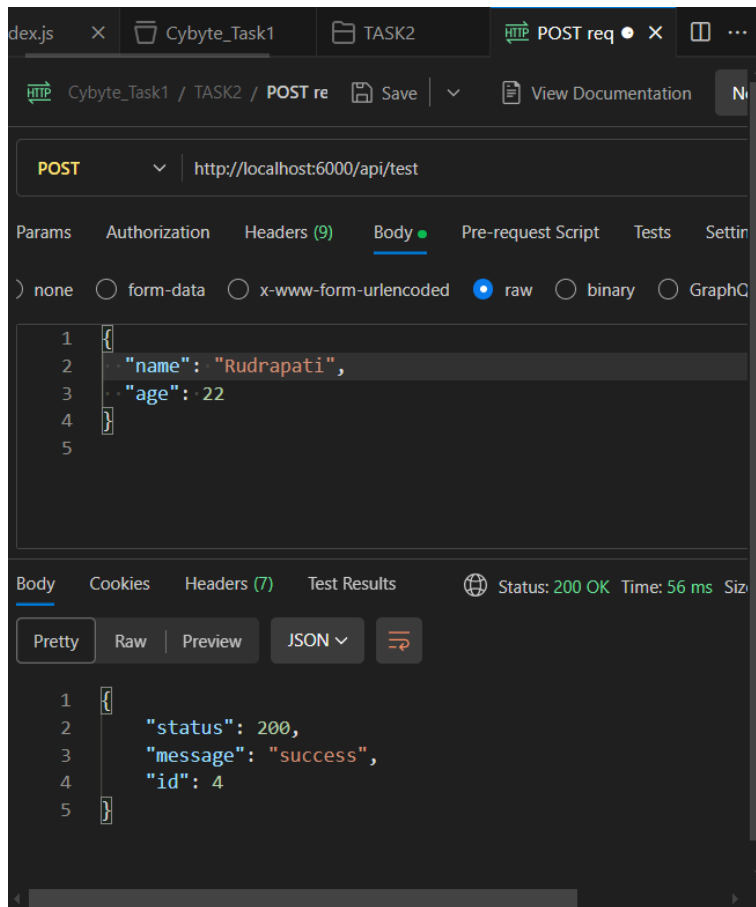
3) Test them in the Postman and verify the data.

4) Write the required queries / store procedures in MYSQL to insert, update, get, and delete.

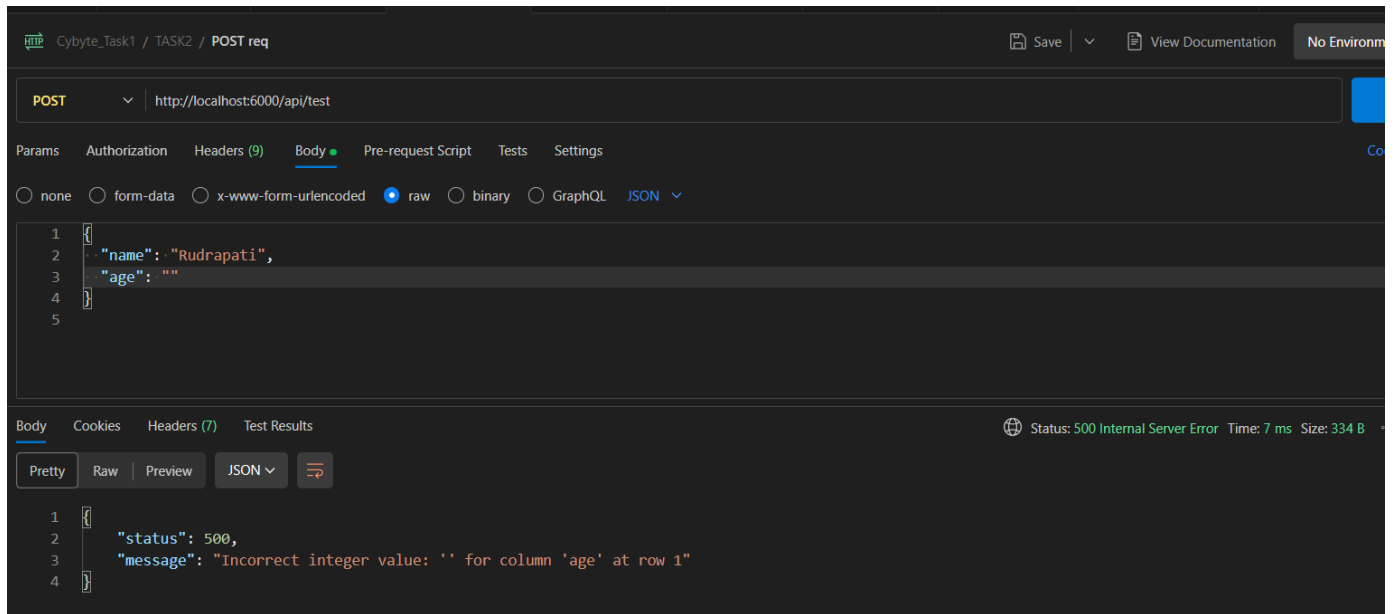
Github: [https://github.com/rudrapatirohith/Cybyte\\_task/blob/main/index.js](https://github.com/rudrapatirohith/Cybyte_task/blob/main/index.js)

5) Send us the method name and body and the result of it as below:

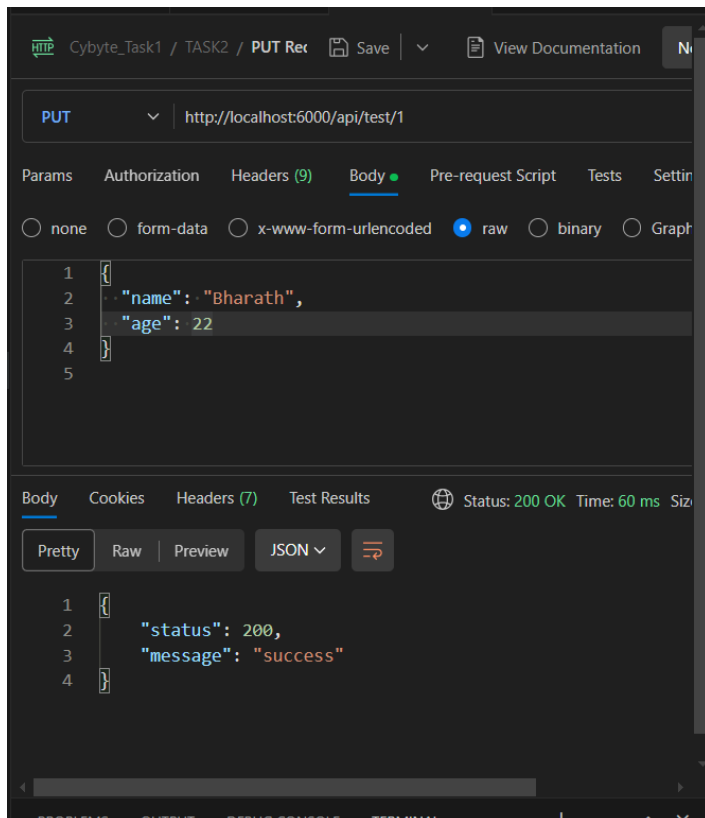
POST METHOD:



if I don't give age



PUT METHOD:



if we give id which doesn't exist, we will get this

The screenshot shows a REST client interface for a PUT request to `http://localhost:6000/api/test/6`. The request body is a JSON object: `{ "name": "Vinay", "age": 22 }`. The response status is `404 Not Found` with a message `"Record not found"`.

```
PUT http://localhost:6000/api/test/6
```

```
{
  "name": "Vinay",
  "age": 22
}
```

```
{
  "status": 404,
  "message": "Record not found"
}
```

if we don't give age

The screenshot shows a REST client interface for a PUT request to `http://localhost:6000/api/test/4`. The request body is a JSON object: `{ "name": "Vinay", "age": "" }`. The response status is `500 Internal Server Error` with a message `"Incorrect integer value: '' for column 'age' at row 1"`.

```
PUT http://localhost:6000/api/test/4
```

```
{
  "name": "Vinay",
  "age": ""
}
```

```
{
  "status": 500,
  "message": "Incorrect integer value: '' for column 'age' at row 1"
}
```

## GET METHOD:

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

- Method:** GET
- URL:** http://localhost:6000/api/test/2
- Query Params:** A table with 2 columns: Key, Value. It contains one row with 'Key' and 'Value'.
- Body:** Pretty, Raw, Preview, JSON (selected). The JSON response is: 

```
{  "status": 200,  "data": {    "id": 2,    "name": "Rohith",    "age": 22  }}
```
- Status:** 200 OK, Time: 50 ms, Size: 100 B

## If I give wrong id

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

- Method:** GET
- URL:** http://localhost:6000/api/test/8
- Query Params:** A table with 2 columns: Key, Value. It contains one row with 'Key' and 'Value'.
- Body:** Pretty, Raw, Preview, JSON (selected). The JSON response is: 

```
{  "status": 404,  "message": "Record not found"}
```
- Status:** 404 Not Found, Time: 1 ms, Size: 100 B

## Get Method to get all the data

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

- URL:** `http://localhost:6000/api/test`
- Method:** `GET`
- Body:** This request does not have a body.
- Response:** Status: 200 OK, Time: 14 ms, Size: 367 B. The response body is a JSON array of three user objects.

```
3  "data": [  
4    {  
5      "id": 2,  
6      "name": "Rohith",  
7      "age": 22  
8    },  
9    {  
10     "id": 3,  
11     "name": "Rudrapati",  
12     "age": 24  
13   },  
14   {  
15     "id": 4,  
16     "name": "Rudrapati",  
17     "age": 22  
18   }  
19 ]
```

## DELETE METHOD:

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

- URL:** `http://localhost:6000/api/test/1`
- Method:** `DELETE`
- Body:** No body is present.
- Response:** Status: 200 OK, Time: 66 ms, Size: 269 B. The response body is a JSON object.

```
1  {  
2    "status": 200,  
3    "message": "success"  
4  }
```

If I give wrong id

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

- Header:** HTTP Cybyte\_Task1 / TASK2 / Delete Req. Save button.
- Request Bar:** DELETE method, URL: http://localhost:6000/api/test/9.
- Request Tabs:** Params, Authorization, Headers (7), Body (selected), Pre-request Script, Tests, Settings.
- Request Body:** none (selected), form-data, x-www-form-urlencoded, raw, binary, GraphQL. A message states: "This request does not have a body".
- Response Tabs:** Body (selected), Cookies, Headers (7), Test Results.
- Response Body:** Pretty, Raw, Preview, JSON (selected). The response is a JSON object:

```
1 {
2   "status": 404,
3   "message": "Record not found"
4 }
```