

## Lesson 01 Demo 04

### Performing CRUD Methods in JSON DB

**Objective:** To perform create, read, update, and delete operations on the JSON database using different HTTP methods and endpoints via curl

**Tools required:** Visual Studio Code, json-server, and curl

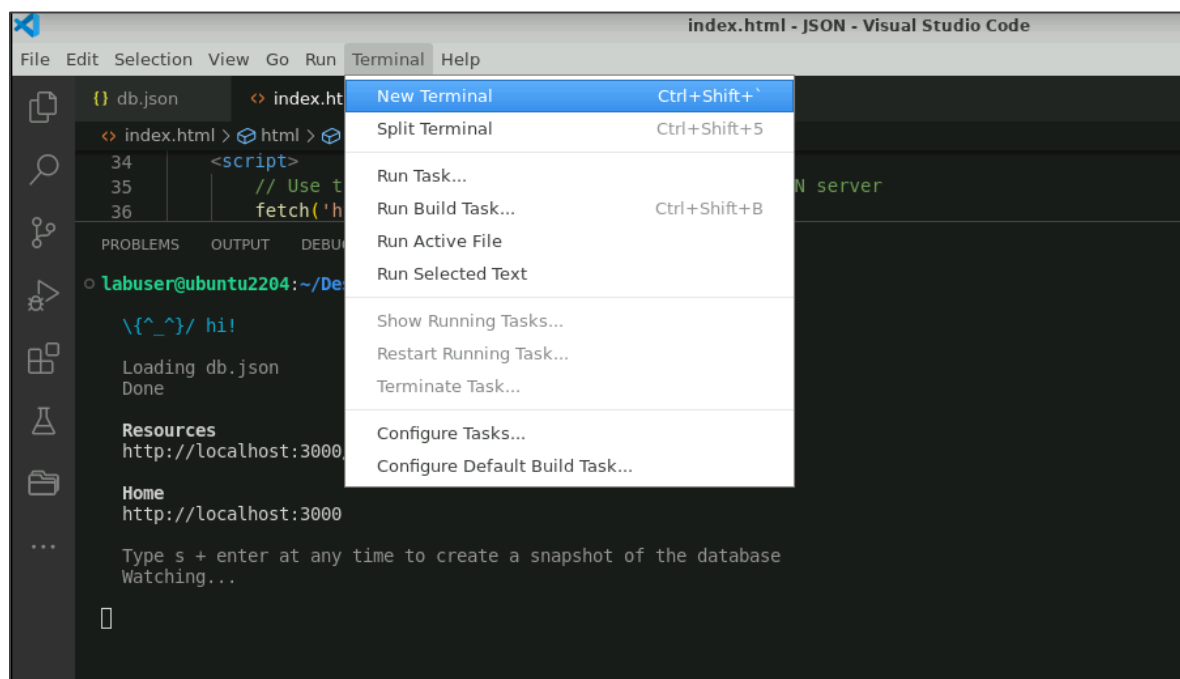
**Prerequisites:** Ensure **db.json** is active and running on **localhost:3000** port

Steps to be followed:

1. Create a new user in the users collection
2. Read data of a specific user (by their ID)
3. Update data of a specific user (by their ID)
4. Delete a specific user (by their ID)

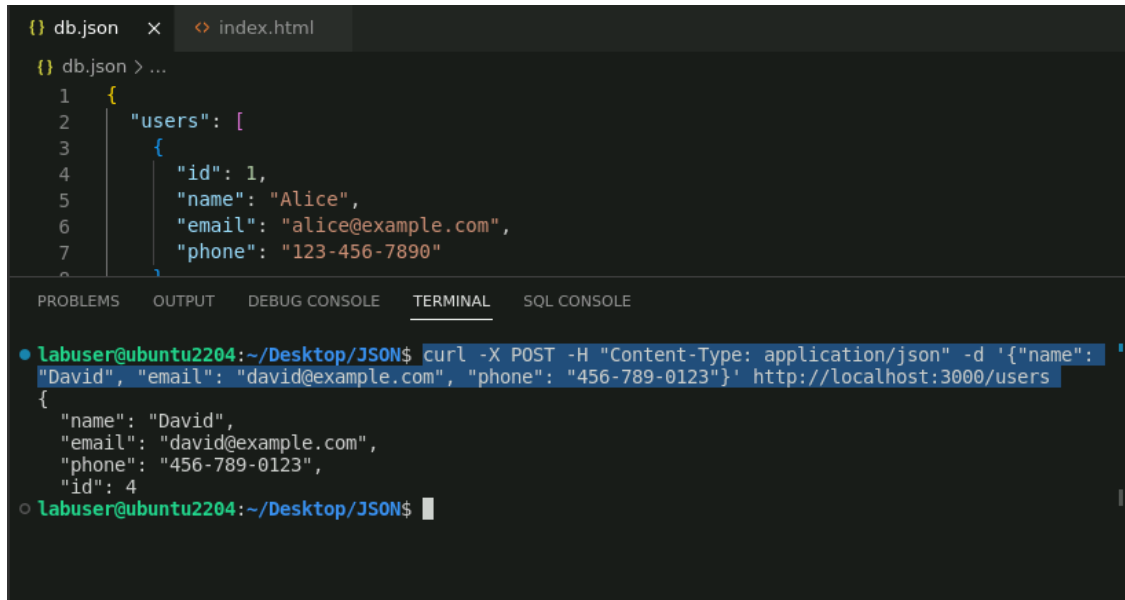
#### Step 1: Create a new user in the users collection

##### 1.1 Open a terminal or command prompt



1.2 Use **curl** with the **POST** method to create a new user:

```
curl -X POST -H "Content-Type: application/json" -d '{"name": "David", "email": "david@example.com", "phone": "456-789-0123"}' http://localhost:3000/users
```

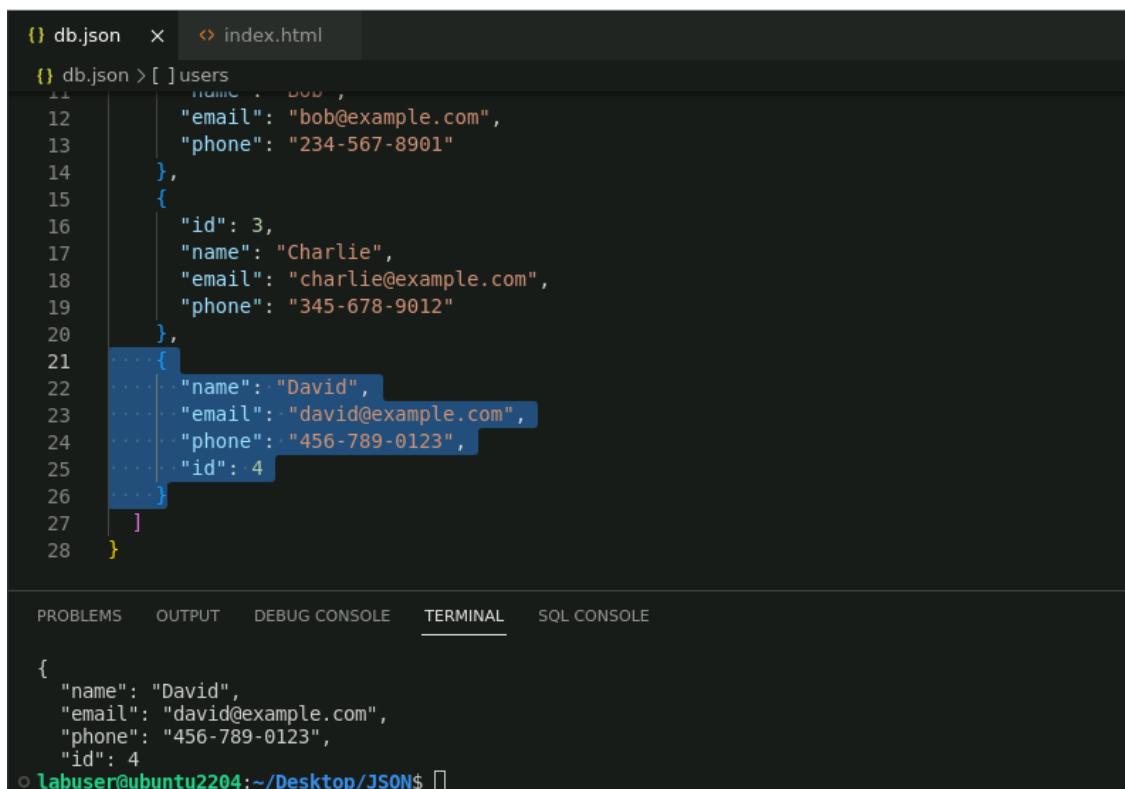


The screenshot shows a VS Code editor with two tabs: `db.json` and `index.html`. The `db.json` file contains a JSON array of users, with the first user being Alice. Below the editor, the **TERMINAL** panel is active, showing the execution of the `curl` command. The output of the command is a JSON object representing the newly created user David.

```
db.json > ...
{
  "users": [
    {
      "id": 1,
      "name": "Alice",
      "email": "alice@example.com",
      "phone": "123-456-7890"
    }
  ]
}

labuser@ubuntu2204:~/Desktop/JSON$ curl -X POST -H "Content-Type: application/json" -d '{"name": "David", "email": "david@example.com", "phone": "456-789-0123"}' http://localhost:3000/users
{
  "name": "David",
  "email": "david@example.com",
  "phone": "456-789-0123",
  "id": 4
}
labuser@ubuntu2204:~/Desktop/JSON$
```

1.3 Open the **db.json** file to verify that the new user **David** has been added



The screenshot shows the `db.json` file in VS Code after the new user David has been added. The JSON array now contains three users: Alice, Bob, and Charlie, with David added at the end. The **TERMINAL** panel at the bottom shows the JSON output of the `curl` command, confirming that David was successfully created.

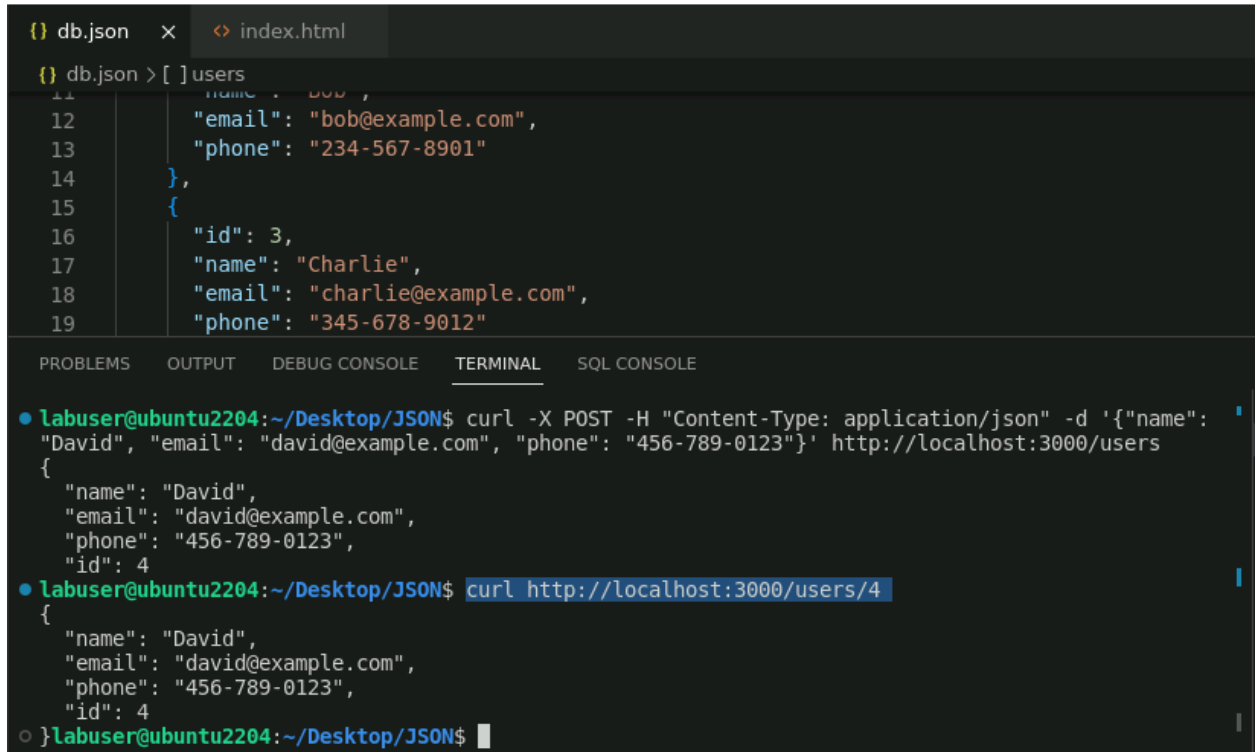
```
db.json > [ ] users
{
  "name": "Bob",
  "email": "bob@example.com",
  "phone": "234-567-8901"
},
{
  "id": 3,
  "name": "Charlie",
  "email": "charlie@example.com",
  "phone": "345-678-9012"
},
{
  "name": "David",
  "email": "david@example.com",
  "phone": "456-789-0123",
  "id": 4
}
]
}

labuser@ubuntu2204:~/Desktop/JSON$
```

## Step 2: Read data of a specific user (by their ID)

2.1 Use **curl** with the **GET** method to read a user's data:

**curl http://localhost:3000/users/4**



The screenshot shows a code editor with a file named `db.json` containing a JSON array of user objects. The first user is Bob (ID 1) and the second is Charlie (ID 3). Below the editor is a terminal window showing the execution of a `curl` command to create a new user (David, ID 4) and then a `curl` command to retrieve the data for user ID 4.

```
{
  "id": 1,
  "name": "Bob",
  "email": "bob@example.com",
  "phone": "234-567-8901"
},
{
  "id": 3,
  "name": "Charlie",
  "email": "charlie@example.com",
  "phone": "345-678-9012"
}
]

```

```
labuser@ubuntu2204:~/Desktop/JSON$ curl -X POST -H "Content-Type: application/json" -d '{"name": "David", "email": "david@example.com", "phone": "456-789-0123"}' http://localhost:3000/users
{"name": "David", "email": "david@example.com", "phone": "456-789-0123", "id": 4}
labuser@ubuntu2204:~/Desktop/JSON$ curl http://localhost:3000/users/4
{"name": "David", "email": "david@example.com", "phone": "456-789-0123", "id": 4}
labuser@ubuntu2204:~/Desktop/JSON$

```

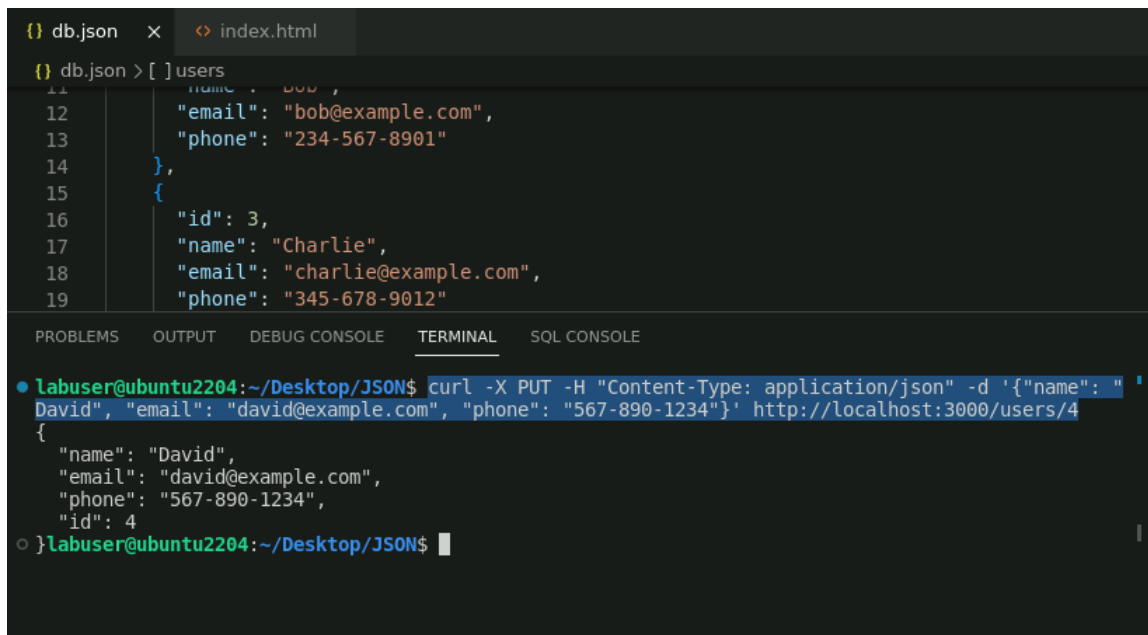
**Note:** Replace 4 with the user's ID

Compare the output in the terminal with the corresponding user data in the **db.json** file

### Step 3: Update data of a specific user (by their ID)

3.1 Use **curl** with the **PUT** or **PATCH** method to update a user's data:

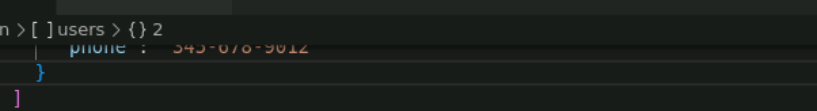
```
curl -X PUT -H "Content-Type: application/json" -d '{"name": "David", "email": "david@example.com", "phone": "567-890-1234"}' http://localhost:3000/users/4
```



The screenshot shows a VS Code editor with two tabs: `db.json` and `index.html`. The `db.json` file contains a JSON array of users. The terminal window shows the execution of a `curl` command to update a user's data.

```
{
  "users": [
    {
      "id": 1,
      "name": "Bob",
      "email": "bob@example.com",
      "phone": "234-567-8901"
    },
    {
      "id": 3,
      "name": "Charlie",
      "email": "charlie@example.com",
      "phone": "345-678-9012"
    }
  ]
}
```

```
labuser@ubuntu2204:~/Desktop/JSON$ curl -X PUT -H "Content-Type: application/json" -d '{"name": "David", "email": "david@example.com", "phone": "567-890-1234"}' http://localhost:3000/users/4
{"name": "David", "email": "david@example.com", "phone": "567-890-1234", "id": 4}
labuser@ubuntu2204:~/Desktop/JSON$
```

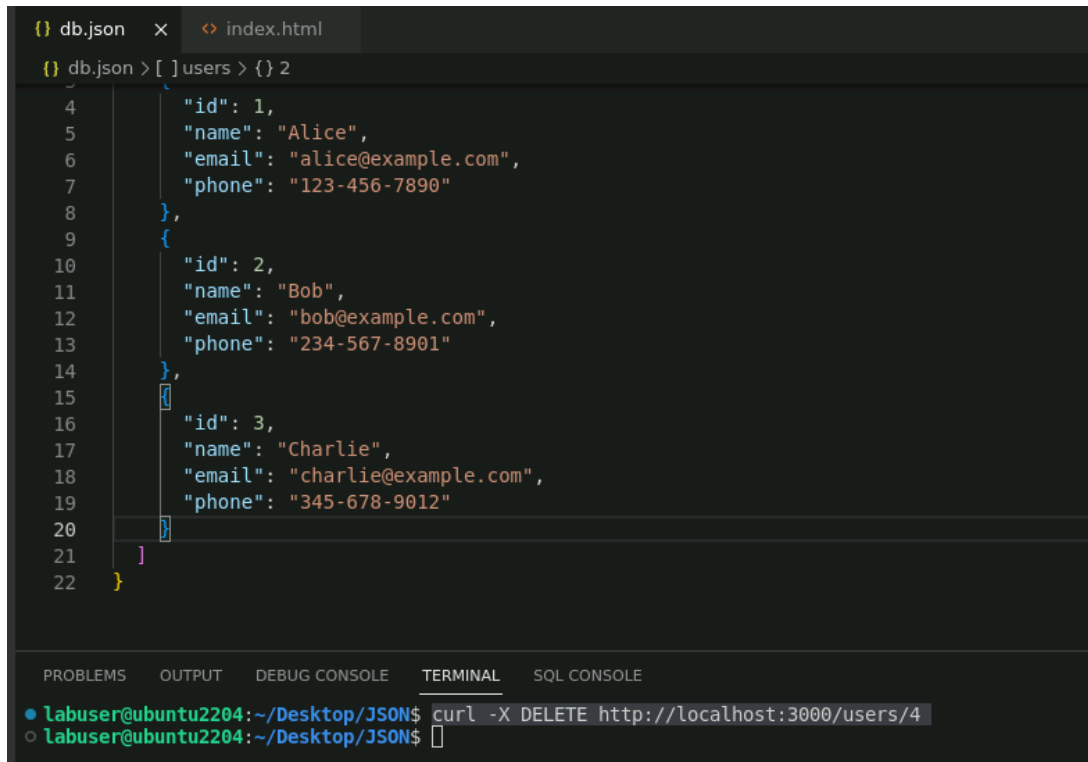


The screenshot shows the VS Code editor interface. At the top, there are two tabs: 'db.json' and 'index.html'. The 'db.json' tab is active, displaying a JSON array of two user objects. The first object has a 'name' property with the value 'John' and a 'phone' property with the value '345-676-9012'. The second object has a 'name' property with the value 'Jane'. The terminal window at the bottom shows the command prompt for a user named 'labuser' on an 'ubuntu2204' system. The command 'curl -X DELETE http://localhost:3000/users/4' has been entered and executed, resulting in a response of '{ }'.

```
{
  "name": "John",
  "phone": "345-676-9012"
},
{
  "name": "Jane"
}
]
```

```
labuser@ubuntu2204:~/Desktop/JSON$ curl -X DELETE http://localhost:3000/users/4
{ }
```

#### 4.2 Open the **db.json** file to confirm that the user with ID **4** has been removed



The screenshot shows a code editor with two tabs: `db.json` and `index.html`. The `db.json` file is open, displaying a JSON array of three user objects. The first user is Alice (ID 1), the second is Bob (ID 2), and the third is Charlie (ID 3). The user with ID 4 has been removed. Below the code editor, a terminal window is open, showing the command `curl -X DELETE http://localhost:3000/users/4` being executed. The terminal output shows the command was successful.

```
{
  "users": [
    {
      "id": 1,
      "name": "Alice",
      "email": "alice@example.com",
      "phone": "123-456-7890"
    },
    {
      "id": 2,
      "name": "Bob",
      "email": "bob@example.com",
      "phone": "234-567-8901"
    },
    {
      "id": 3,
      "name": "Charlie",
      "email": "charlie@example.com",
      "phone": "345-678-9012"
    }
  ]
}
```

```
labuser@ubuntu2204:~/Desktop/JSON$ curl -X DELETE http://localhost:3000/users/4
labuser@ubuntu2204:~/Desktop/JSON$
```

By following these steps, you have successfully performed CRUD operations on the JSON DB using **curl**. This process demonstrates a thorough approach to interacting with a REST API.