

Lesson 01 Demo 01

Creating and Using a JSON Server File

Objective: To create a JSON file with sample data and use **json-server** and **curl** to retrieve and display the user data in the console

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

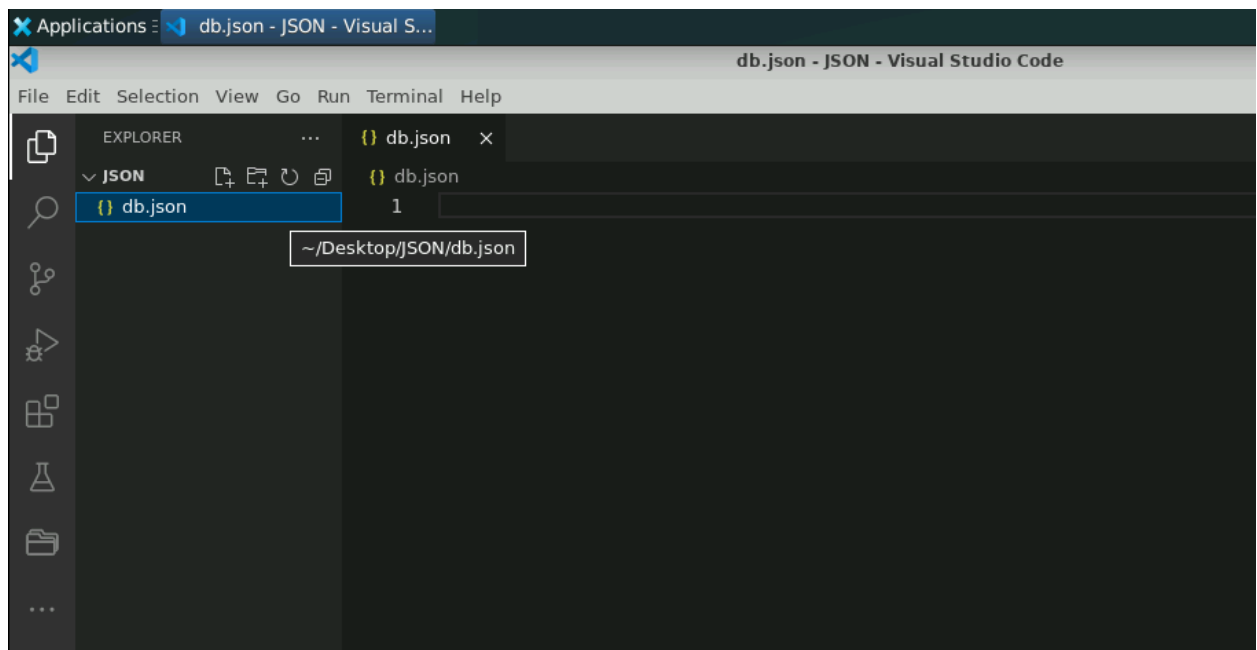
Prerequisites: None

Steps to be followed:

1. Create a **db.json** file with user data
2. Install and run the **json-server** using the **db.json** file
3. Use **curl** to retrieve and print the user data on the console

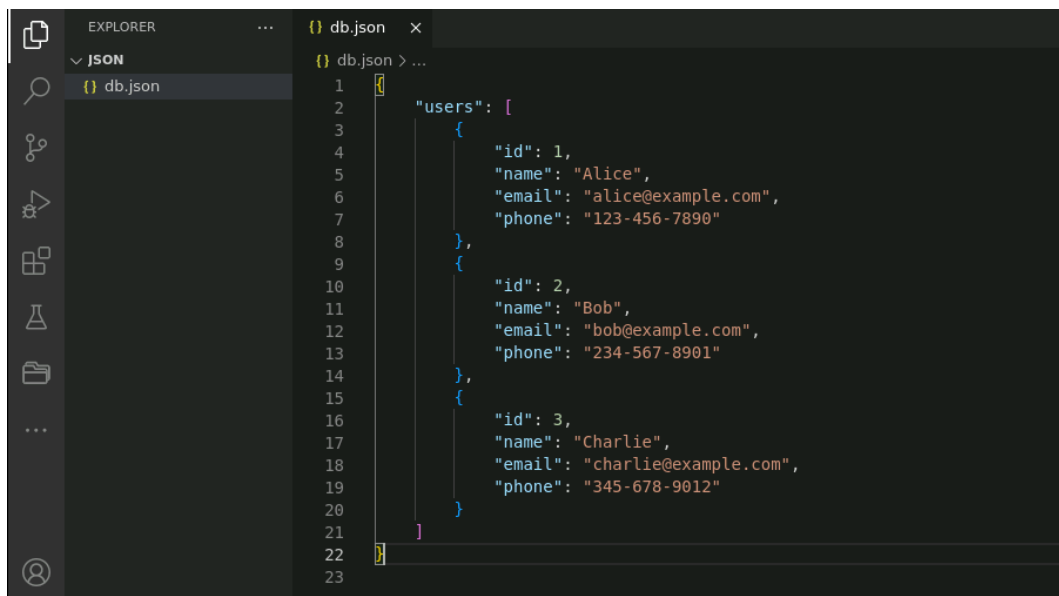
Step 1: Create a db.json file with user data

- 1.1 Open the Visual Studio Code editor and create a file named **db.json**



1.2 Add the following JSON data about users to the **db.json** file:

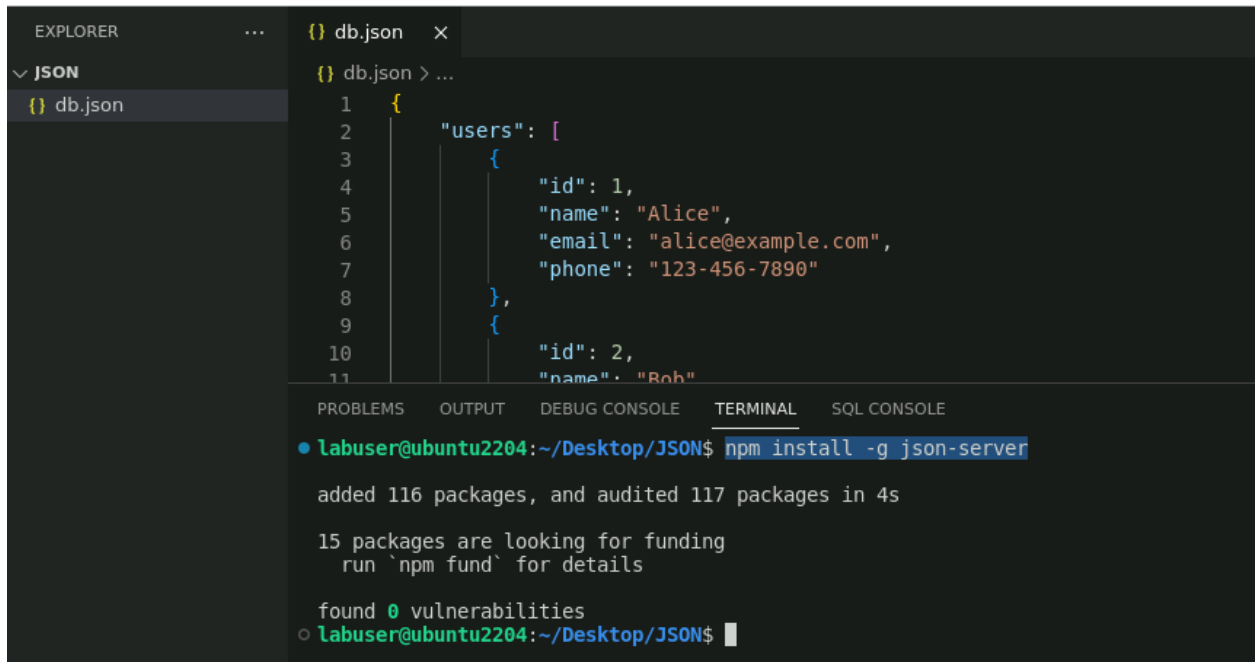
```
{
  "users": [
    {
      "id": 1,
      "name": "Alice",
      "email": "alice@example.com",
      "phone": "123-456-7890"
    },
    {
      "id": 2,
      "name": "Bob",
      "email": "bob@example.com",
      "phone": "234-567-8901"
    },
    {
      "id": 3,
      "name": "Charlie",
     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

A screenshot of the Visual Studio Code editor interface. The Explorer sidebar on the left shows a project named 'JSON' with a file named 'db.json'. The main editor window displays the content of 'db.json', which is a JSON array of three user objects. The code is syntax-highlighted, with strings in quotes, numbers in blue, and JSON structure symbols in purple and blue. Line numbers 1 through 23 are visible on the left side of the editor.

```
1 {
2   "users": [
3     {
4       "id": 1,
5       "name": "Alice",
6       "email": "alice@example.com",
7       "phone": "123-456-7890"
8     },
9     {
10      "id": 2,
11      "name": "Bob",
12      "email": "bob@example.com",
13      "phone": "234-567-8901"
14    },
15    {
16      "id": 3,
17      "name": "Charlie",
18      "email": "charlie@example.com",
19      "phone": "345-678-9012"
20    }
21  ]
22 }
23
```

Step 2: Install and run the json-server using the db.json file

2.1 Open the terminal and run the following command to install **json-server** using **npm**:
npm install -g json-server



```

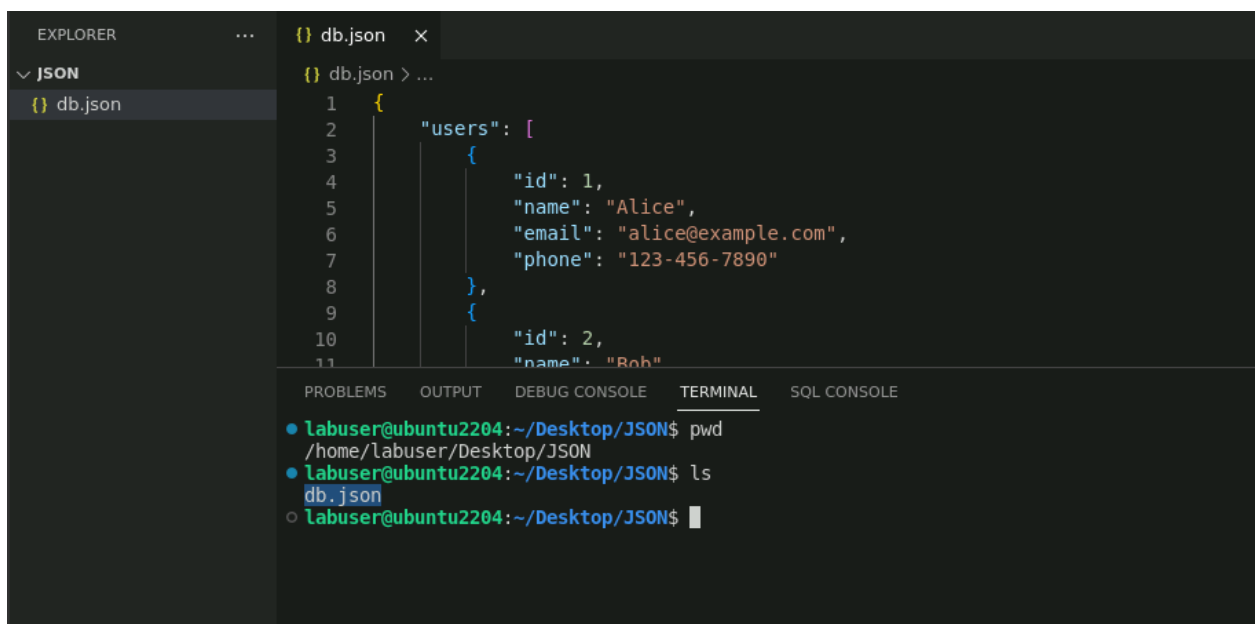
EXPLORER
  JSON
  db.json

db.json
1 {
2   "users": [
3     {
4       "id": 1,
5       "name": "Alice",
6       "email": "alice@example.com",
7       "phone": "123-456-7890"
8     },
9     {
10      "id": 2,
11      "name": "Bob"
  }
}

TERMINAL
labuser@ubuntu2204:~/Desktop/JSON$ npm install -g json-server
added 116 packages, and audited 117 packages in 4s
15 packages are looking for funding
run 'npm fund' for details
found 0 vulnerabilities
labuser@ubuntu2204:~/Desktop/JSON$

```

2.2 Navigate to the directory where **db.json** is saved



```

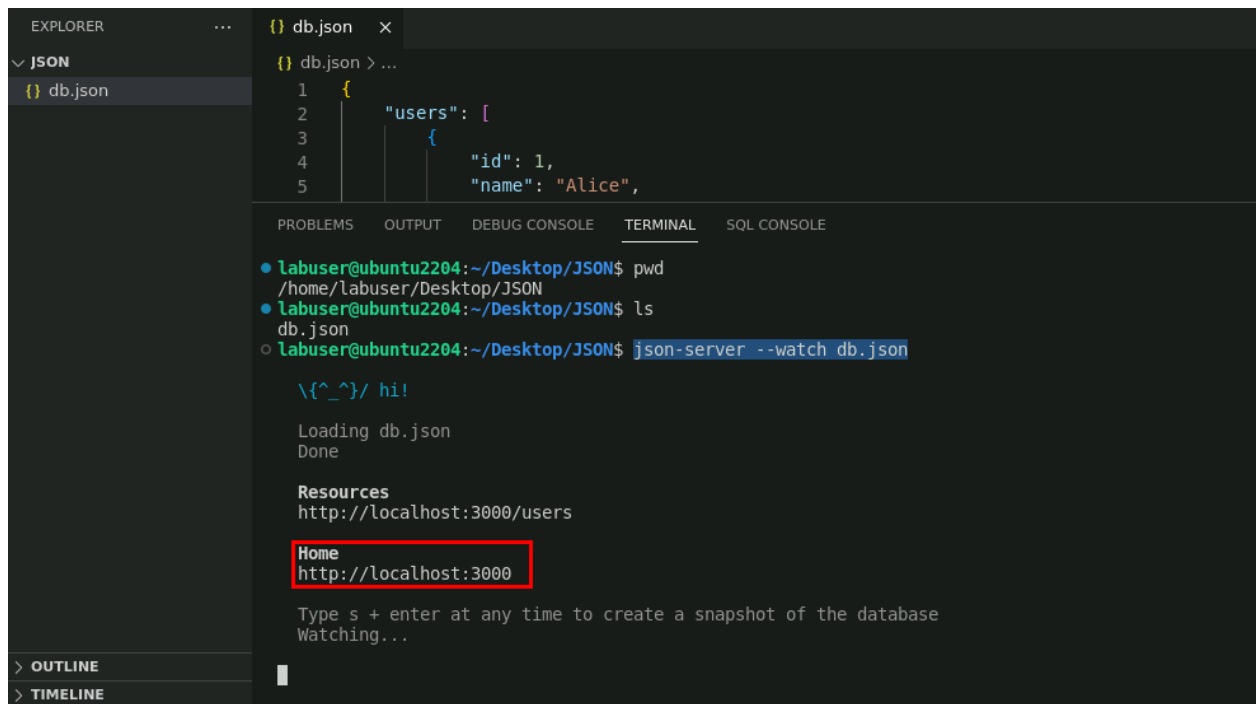
EXPLORER
  JSON
  db.json

db.json
1 {
2   "users": [
3     {
4       "id": 1,
5       "name": "Alice",
6       "email": "alice@example.com",
7       "phone": "123-456-7890"
8     },
9     {
10      "id": 2,
11      "name": "Bob"
  }
}

TERMINAL
labuser@ubuntu2204:~/Desktop/JSON$ pwd
/home/labuser/Desktop/JSON
labuser@ubuntu2204:~/Desktop/JSON$ ls
db.json
labuser@ubuntu2204:~/Desktop/JSON$

```

2.3 Start **json-server** using the following command:
json-server --watch db.json



The screenshot shows a VS Code editor with a file named `db.json` open. The file contains a JSON object with a `users` array. The terminal window shows the execution of `json-server --watch db.json`, which outputs a greeting, loads the database, and provides the resource URL `http://localhost:3000/users`. The `Home` link for this resource is highlighted with a red box.

```
EXPLORER
  JSON
    db.json

db.json
1 {
2   "users": [
3     {
4       "id": 1,
5       "name": "Alice",

TERMINAL
labuser@ubuntu2204:~/Desktop/JSON$ pwd
/home/labuser/Desktop/JSON
labuser@ubuntu2204:~/Desktop/JSON$ ls
db.json
labuser@ubuntu2204:~/Desktop/JSON$ json-server --watch db.json

\{^_^}/ hi!

Loading db.json
Done

Resources
http://localhost:3000/users
Home
http://localhost:3000

Type s + enter at any time to create a snapshot of the database
Watching...
```

Note: The **db.json** is accessible at **localhost:3000**.

Step 3: Use curl to retrieve and print the user data on the console

3.1 Open another terminal or command prompt window

The screenshot shows the Visual Studio Code interface. On the left, a file named `db.json` is open, displaying a JSON array of users. The `users` array contains one object with `id: 1` and `name: "Alice"`. On the right, a terminal window titled `Tilix: labuser@ubuntu2204: ~` is open. It shows the following commands and output:

```
labuser@ubuntu2204:~/Desktop/JSON$ pwd
/home/labuser/Desktop/JSON
labuser@ubuntu2204:~/Desktop/JSON$ ls
db.json
labuser@ubuntu2204:~/Desktop/JSON$ json-server --watch db.json
{(^_^)/ hi!
Loading db.json
Done
Resources
http://localhost:3000/users
Home
http://localhost:3000
Type s + enter at any time to create a snapshot of the database
Watching...
```

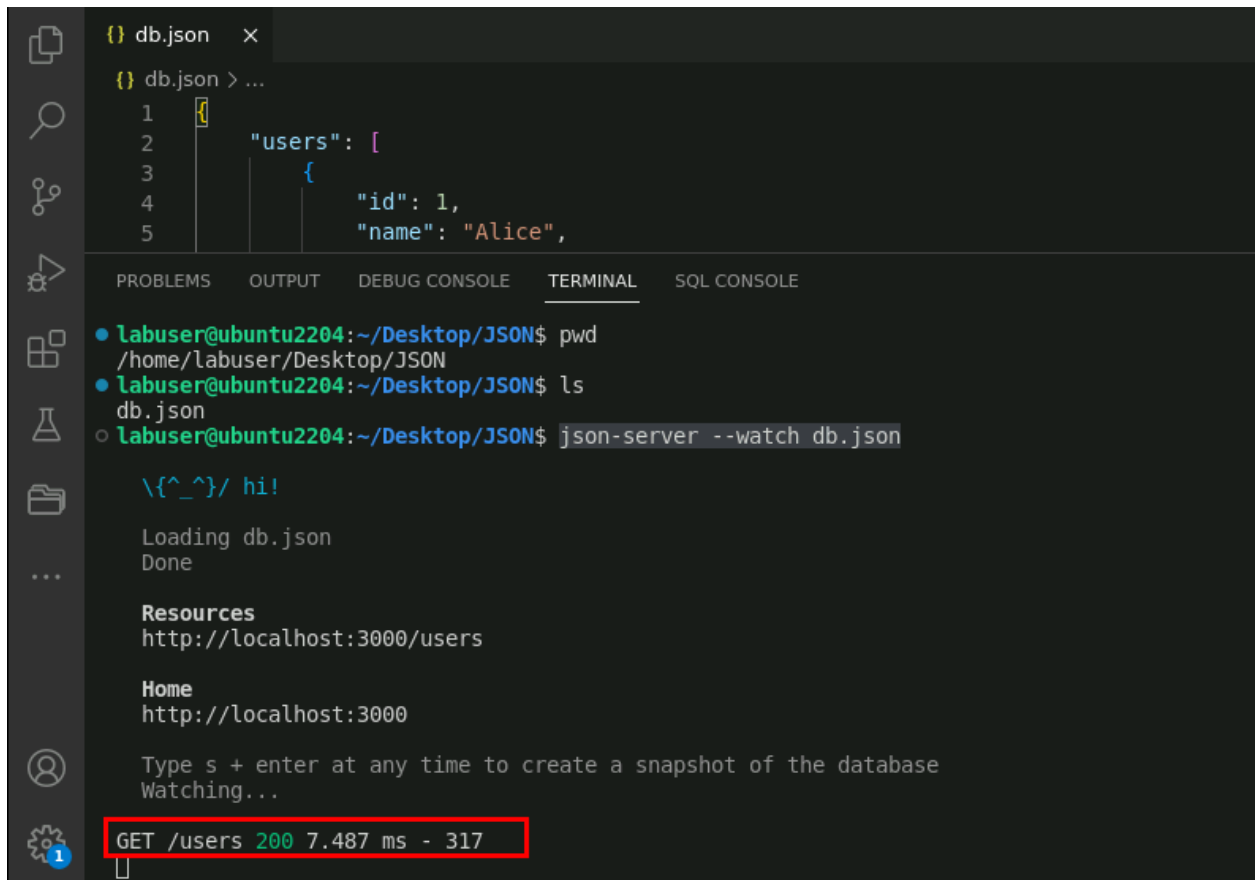
3.2 Use curl to make a GET request to the server:

curl http://localhost:3000/users

This screenshot shows the same Visual Studio Code environment as before, but now the terminal window has the `curl` command executed. The output of the command is displayed in the terminal, showing a JSON array of three users. A red box highlights the output:

```
labuser@ubuntu2204:~/Desktop/JSON$ curl http://localhost:3000/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"}]
```

The user data from **db.json** will display on the console.



The screenshot shows a VS Code editor with a file named `db.json` containing a JSON array of users. Below the editor, the `TERMINAL` tab is active, showing the execution of `json-server` and the results of a REST client request.

```
{
  "users": [
    {
      "id": 1,
      "name": "Alice",
    }
  ]
}
```

```
labuser@ubuntu2204:~/Desktop/JSON$ pwd
/home/labuser/Desktop/JSON
labuser@ubuntu2204:~/Desktop/JSON$ ls
db.json
labuser@ubuntu2204:~/Desktop/JSON$ json-server --watch db.json

\{^_^\}/ hi!

Loading db.json
Done

Resources
http://localhost:3000/users

Home
http://localhost:3000

Type s + enter at any time to create a snapshot of the database
Watching...

GET /users 200 7.487 ms - 317
```

You can confirm the same in the terminal where **db.json** was executed. A GET request on the **/users** endpoint should return a **200** response.

By completing these steps, you will have successfully used **json-server** to serve your JSON data and **curl** to retrieve and display it in the console. This demonstrates a simple way to test and interact with a REST API using local JSON data.