

# Fetch API by axios (get, post, delete and update)

Create a new folder (serverAPI) and Install JSON Server

```
npm i --save json-server
```

Create a db.json file with some data

```
{
  "users": [
    {"id": 1, "name": "name1", "username": "username1", "email": "email1@gmail.com"},
    {"id": 2, "name": "name2", "username": "username2", "email": "email2@gmail.com"},
    {"id": 3, "name": "name3", "username": "username3", "email": "email3@gmail.com"}
  ]
}
```

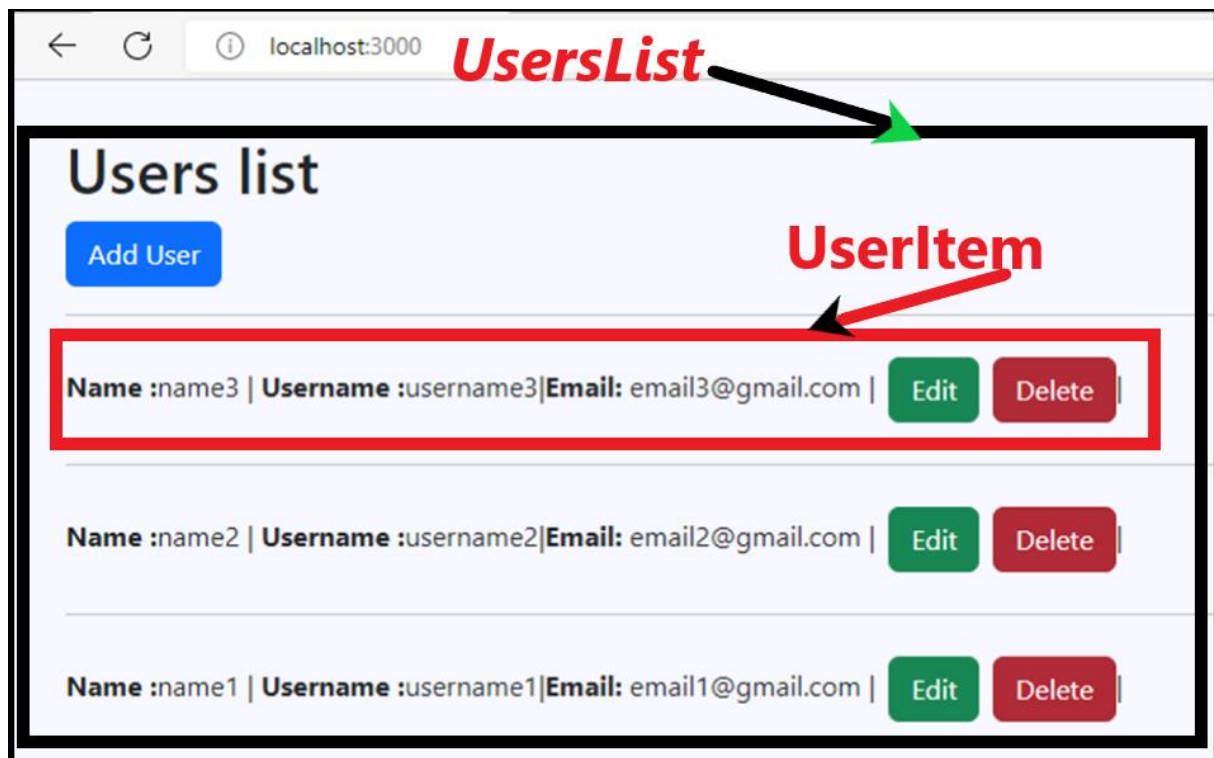
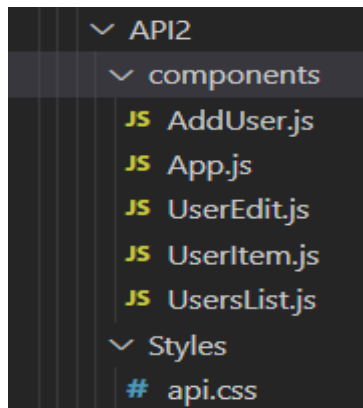
Start JSON Server (package.json)

```
"scripts": {
  "start": "json-server -p 3006 -w db.json"
},
```

Now if you go to <http://localhost:3006/users/1>, you'll get

```
{
  "id": 1,
  "name": "name1",
  "username": "username1",
  "email": "email1@gmail.com"
}
```

Create the following components:



Code : App.js

```
export default function App(){  
  
  return <BrowserRouter>  
  
    <Routes>  
      <Route path="/" element={<UsersList/>}/>  
      <Route path="/userEdit/:id" element={<UserEdit/>}/>  
      <Route path="/addUser" element={<AddUser/>}/>  
    </Routes>  
  </BrowserRouter>  
}
```

### 1. Get Users :

```
const [users, setUsers]=useState([])
const API=axios.create({
  baseURL:"http://localhost:3006/"
})
const getUsers =async() => {
  try{
    const res = await API.get("/users");

    setUsers(res.data)
  }
  catch(err)
  {
    console.log(err)
  }
};
```

### 2. Post User

```
const [user, setUser]=useState({name:'user1',email:"email23@gmail.com",
username:"username254"})
const API=axios.create({
  baseURL:"http://localhost:3006/"
})
const addUser=async()=>{
  try {
    await API.post(`/users/`,user)
  } catch (error) {
    console.log(error)}};
```

← ↻ ⓘ localhost:3000/addUser

## Add User :

Name :

Username :

Email :

Ok

back

### 3. Delete User

```
const deleteUser=async(id)=>{
  try{
    const resp= await API.delete(`/users/${id}`)
  }
  catch(err)
  {
    console.log(err)
  }
};
```

### 4. Edit User

```
const [user,setUser]=useState({name:'aa',email:"aa",username:"aa"})
const API=axios.create({
  baseUrl:"http://localhost:3006/"
})
const id=2 // update user that have id equal to 2

const updateUser=async()=>{
  try {
    await API.put(`/users/${id}`,user)
  } catch (error) {
```

```
console.log(error)
}}
```

A screenshot of a web browser window showing a user edit form. The browser's address bar displays "localhost:3000/userEdit/3". The form is titled "Edit User :". It contains three input fields: "Name" with the value "name3", "Username" with the value "username3", and "Email" with the value "email3@gmail.com". Below the input fields are two buttons: a green "Update" button and a blue "back" button.

← ↻ ⓘ localhost:3000/userEdit/3

## Edit User :

Name :

Username :

Email :

Update

back