

# Lecture 6. REST API using Express js with Nodejs

10 April 2025 19:39

Complete Process:-

## Revision Summary: Express.js CRUD API

### Dependencies

```
js
const express = require("express");
const users = require("./MOCK_DATA.json");
```

- express: Web framework to create server and routes.
- MOCK\_DATA.json: Simulated database containing user details.

### Server Setup

```
js
const app = express();
const PORT = 8000;
```

- Initializes Express app on port 8000.

### Routes Overview

#### GET /users – Renders User List in HTML

```
js
app.get("/users", (req, res) => {
  const html = `
    <ul>
      ${users.map((user) => `<li>${user.first_name}</li>`).join("")}
    </ul>`;
  res.send(html);
});
```

- Displays a basic HTML list of user **first names**.

#### GET /api/users – Fetch All Users as JSON

```
js
app.get("/api/users", (req, res) => {
  return res.json(users);
});
```

- Sends full user data in JSON format.
- Useful for frontend or API testing.

#### GET /api/users/:id – Get Single User by ID

```
js
app.route("/api/users/:id").get((req, res) => {
  const id = Number(req.params.id);
  const user = users.find(user => user.id === id);
});
```

```
    return res.json(user);
  })
  • Parses id from route and finds the matching user.
```

## PATCH /api/users/:id – Edit User (Stub)

```
js
.patch((req, res) => {
  return res.json({ status: "Pending" });
})
  • Placeholder for future logic to edit user data.
```

## DELETE /api/users/:id – Delete User (Stub)

```
js
.delete((req, res) => {
  return res.json({ status: "Pending" });
});
  • Placeholder to delete a user by ID.
```

## POST /api/users – Create a New User (Stub)

```
js
app.post("/api/users", (req, res) => {
  res.json({ status: "pending" });
});
  • Placeholder to add a new user.
```

## Server Listener

```
js
Copy code
app.listen(PORT, () => console.log(`Server started at Port: ${PORT}`));
  • Starts the server on port 8000.
```

Complete Code:-

```
const express = require("express");
const users = require("./MOCK_DATA.json");

const app = express();
const PORT = 8000;

//Routes
app.get("/users", (req, res) => {
  //HTML document rendering
  const html = `
    <ul>
      ${users.map((user) => `<li> ${user.first_name}</li>`).join("")}
    </ul>
  `;
  res.send(html);
})
//REST API POINTS
app.get("/api/users", (req, res) => {
```

```

        return res.json(users);
    });

    app
    .route("/api/users/:id").get((req, res)=>{
        const id = Number(req.params.id);
        const user = users.find(user => user.id === id);
        return res.json(user);
    })
    .patch((req, res) => {
        //edit user id
        return res.json({status:"Pending"})
    })
    .delete((req, res) => {
        //delete user id
        return res.json({status:"Pending"})
    });

    app.post('/api/users', (req, res) => {
        //TODO: create a new user
        res.json({ status: "pending"});
    })

    app.listen(PORT, () => console.log(`Server started at Port :${PORT}`));

```