REST

REST :-

- Representational State Transfer
- REST is an architectural style that defines a set of constraints to be used for creating web services.

CRUD Operations :-

↳ Create, Read, Update & Delete

- GET retrieves resources.
- POST submits new data to the server
- PUT updates existing data
- PATCH update existing data partially
- DELETE removes data.

Creating RESTful APIs :-

• GET	/posts	to get data for all posts	Index (main) Route
• POST	/posts	to add a new post	CREATE Route
• GET	/posts/:id	to get one post (using id)	VIEW "
• PATCH	/posts/:id	to update specific post	UPDATE "
• DELETE	/posts/:id	to delete specific post	DESTROY "

Implement : GET /posts :-

↙

Index Route

→

REST

File structure

REST_CLASS
- node_modules
- public
  - style.css
- views
  - index.ejs
- index.js
- package.json
- package-lock.json

index.js

```
const express = require("express");
const app = express();
const port = 8080;
const path = require("path");

app.use(express.urlencoded({extended: true}));
app.set("view engine", "ejs");
app.set("views", path.join(__dirname, "views"));
app.use(express.static(path.join(__dirname, "public")));

let posts = [
    {
        username: "apnacollege",
        content: "I love coding",
    },
```

```
    {
      username: "shradhakhapra",
      content: "hardwork pays off",
    },
    {
      username: "aayushjha",
      content: "Awesome work",
    },
];

app.get("/posts", (req, res) => {
    res.render("index.ejs", {posts});
});

app.listen(port, () => {
    console.log(`listening to port ${port}`);
});
```

index.ejs

```
<html>
    <head> - - </head>
    <body>
        <h1> Quora Posts </h1>
        <% for(post of posts) { %>
            <div class="post">
                <h3 class="user">@<%= post.username %></h3>
                <h4 class="content">@<%= post.content %></h4>
            </div>
</html></body><% } %>
```

style.css:-

```
h1{
    color: maroon;
}

.post{
    background-color: blanchedalmond;
}

.user{
    font-style: italic;
}
```

Implement: POST/posts:-

Create Route

2 Routes

- Serve the form GET /posts/new
- Add the new post POST /posts

In the last file structure, inside views directory, create a file new.ejs.

Now, add the following codes in the files:-

```
app.get("/posts/new", (req, res) => {
    res.render("new.ejs");
});
```

```
app.post("/posts", (req, res) => {
    let {username, content} = req.body;
    posts.push({username, content});
    res.redirect("/posts");
});
```

redirects to the specified page

index.ejs :→

```
<html>
    <head> ---- </head>
    <body>
        <div>
        </div>
        <br>
        <br>
        <a href="http://localhost:8080/posts/new">Create
            New Post </a>
    </body>
</html>
```

new.ejs :→

```
<html>
    <head> ---- </head>
    <body>
        <form method="post" action="/posts">
            <input placeholder="enter username"
                name="username">
```

<link rel="stylesheet" href="/style.css">

```
<br>
<br>
<textarea placeholder="write your post" name=
    "content"> </textarea>
<br>
<br>
<button> Submit Post </button>
</form>
</body>
</html>
```

Implement: GET /posts/:id :→

Show Route

In the last file structure, inside views directory, create a file show.ejs.

Keep new.ejs, style.css as it was earlier:→

~~<form>~~

~~</form>~~

index.js:→

```
let posts = [
    {
        id: "1a"
        username: "  ";
        content: "  ";
    },
```

→ Add it in posts.

```
{
    id: "2b",
    username: "  ",
    content: "  ",
},
{
    id: "3c",
    username: "  ",
    content: "  ",
},
];
```

// Add this →

```
app.get("/posts/:id", (req, res) => {
    let {id} = req.params;
    let post = posts.find((p) => id === p.id);
    res.render("show.ejs", {post});
});
```

index.ejs :-

```
<html>
    <head> ---- </head>
    <body>
        <h1> ---- </h1>
        <div ---- >
            // After h3 & h4, add :-
```

```
    <br>
    <a href="http://localhost:8080/posts/<%= post.id %>">
        See in Detail</a>
</div>

  —  —

  —  —

  —  -  -

</body>
</html>
```

show.ejs:-

```
<html>
    <head>
        - - -
        - - -
        <link rel="stylesheet" href="/style.css">
    </head>
    <body>
        <h2> Here is your post in detail </h2>
        <p> Post id: <%= post.id %> </p>
        <div class="post">
            <h3 class="user">@<%= post.username %></h3>
            <h4><%= post.content %></h4>
        </div>
    </body>
</html>
```

→

Creating Ids (UUID):-

Create id for Posts

UUID Package

→ Universally unique identifier

npm install uuid

→ Keep everything as earlier:-

Just make index.js as:-

```
const express= require("express");
const app= express();
const port = 8080;
const path = require("path");
const {v4: uuidv4} = require('uuid');

app.use(express.urlencoded({extended:true}));
app.set("view engine", "ejs");
app.set("views", path.join(__dirname,"views"));
app.use(express.static(path.join(__dirname,"public")));

let posts=[
    {
        id: uuidv4(),
        username: "apnacollege",
        content: "I love coding",
    },
```

```
{
    id: uuidv4(),
    username: "shradhakhapra",
    content: "hardwork pays off",
},
{
    id: uuidv4(),
    username: "aayushjha",
    content: "Awesome work",
},
];

app.get("/posts", (req, res) => {
    res.render("index.ejs", { posts });
});

app.get("/posts/new", (req, res) => {
    res.render("new.ejs");
});

app.post("/posts", (req, res) => {
    let {username, content} = req.body;
    let id = uuidv4();
    posts.push({id, username, content});
    res.redirect("/posts");
});

app.get("/posts/:id", (req, res) => {
    let {id} = req.params;
    let post = posts.find((p) => id === p.id);
```

```
    res.render("show.ejs", {post});
});

app.listen(port, ()=>{
    console.log('listening to port ${port}');
});
```

(#) Implement: PATCH /posts/:id :→

Update Route

In index.js :→

```
app.use(express.json());        }→ At top

app.patch("/posts/:id", (req, res)=>{
    let {id} = req.params;
    let newContent = req.body.content;
    let post = posts.find((p)=> id === p.id);
    post.content = newContent;
    console.log(post);
    res.send("patch request working");
});
```

(##) Create form for Update :→

Edit Route

Serve the edit form GET /posts/:id/edit

In views dir, create edit.ejs :→

edit.ejs :→

```
<html>
    <head> - - - </head>
    <body>
        <h2> Edit your post </h2>
        <p> Post id: <%= post.id %> </p>
        <p> Post username: @<%= post.username %> </p>
        <form method="post" action="/posts/<%= post.id %>
            ?_method=PATCH">
            <textarea rows="10" cols="35" name="content">
                <%= post.content %> </textarea>
            <button> Submit </button>
        </form>
    </body>
</html>
```

index.js :→

```
const methodOverride = require("method-override");

app.use(methodOverride("_method"));

//In app.patch do this :→

app.patch("/posts/:id", (req, res) => {
    let {id} = req.params;
    let newContent = req.body.content;
    let post = posts.find((p) => id === p.id);
```

```
    post.content = newContent;
    console.log(post);
    res.redirect("/posts");
});
```

// Also :→

```
app.get("/posts/:id/edit", (req, res) => {
    let {id} = req.params;
    let post = posts.find((p) => id === p.id);
    res.render("edit.ejs", {post});
});
```

Implement : /posts/:id :→

Destroy Route

DELETE /posts/:id to delete specific post

index.js :→

```
app.delete("/posts/:id", (req, res) => {
    let {id} = req.params;
    posts = posts.filter((p) => id !== p.id);
    res.redirect("/posts");
});
```

index.ejs :→

```
<html>
    <head> ——— <link rel="stylesheet" href="/styles
    </head>
```

```
<body>
    <h1 ___ </h1>
    <% for (post of posts) { %>
    <div>
        <h3 class= ___ - -
        <h4 . ___ ___
        <br>
        <a href="http://localhost:8080/posts/
            <%= post.id %>"> See in Detail </a>
        <a href="http://localhost:8080/posts/
            <%= post.id %> /edit"> Edit </a>
        <form method="post" action="/posts/ <%= post.id
        %> ?_method=DELETE">
            <button> Delete Post </button>
        </form>
    </div>
    <% } %>
    <br>
    <br>
    <a href="http://localhost:8080/posts/new">. create
        New Post </a>

</body>
</html>
```