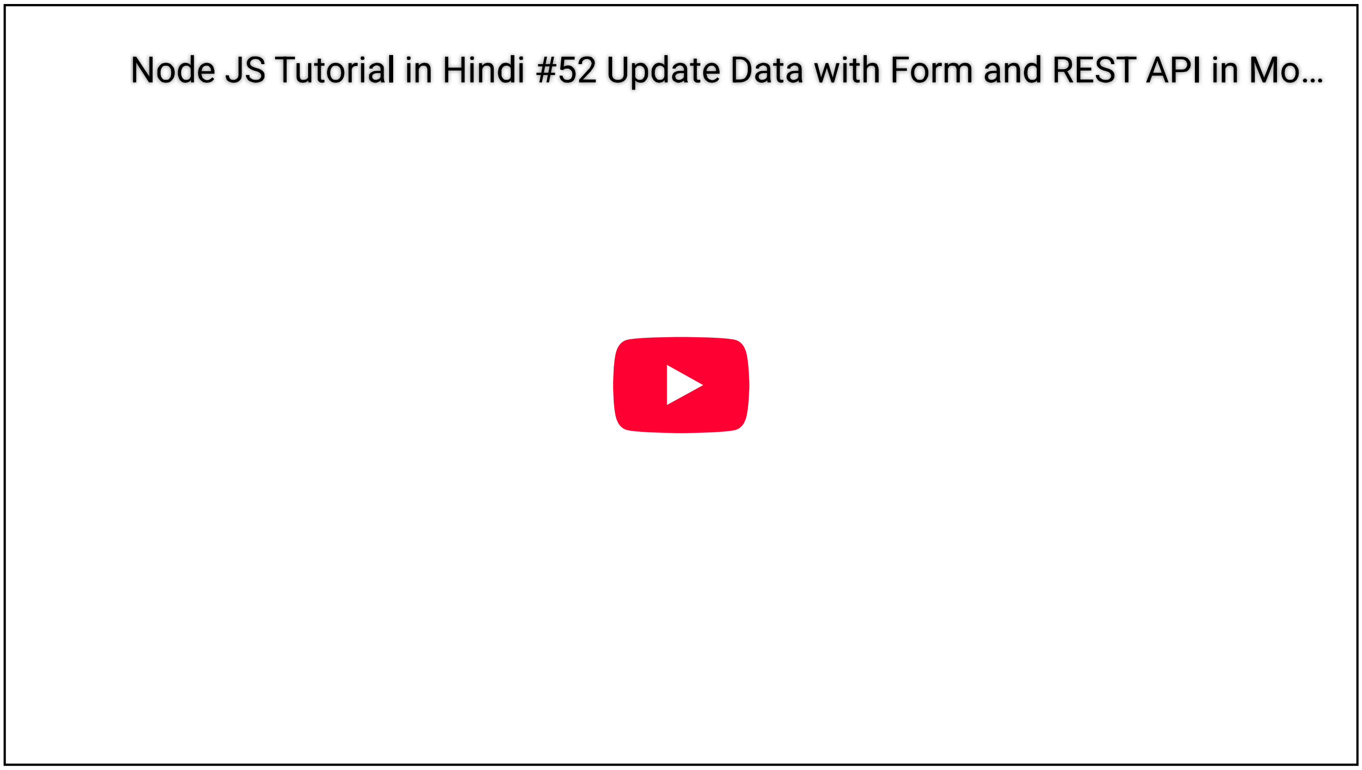
Node JS Tutorial in Hindi #52 Update Data with Form and REST API in MongoDB



Node JS Tutorial in Hindi #52 Update Data with Form and REST API in Mo...

Update data with form and API

Make route to handle post request

Send data to post route

Update data with UI

Update data with API

Test Flow and DB

Interview Questions

Notes, Code and playlist.

Index.js file code

```
import express from 'express'
import { MongoClient, ObjectId } from 'mongodb';

const dbName = "school"
const url = "mongodb://localhost:27017"
const client = new MongoClient(url)



const app = express();
app.use(express.urlencoded({ extended: true }))
app.use(express.json());

app.set("view engine", 'ejs')
client.connect().then((connection) => {
    const db = connection.db(dbName);

    app.get("/api", async (req, resp) => {
        const collection = db.collection("students")
        const students = await collection.find().toArray();
        resp.send(students)
    })
    app.get("/ui", async (req, resp) => {
        const collection = db.collection("students")
        const students = await collection.find().toArray();
        resp.render('students', { students })
    })

    app.get('/add', (req, resp) => {
        resp.render('add-student')
    })

    app.post("/add-student", async (req, resp) => {
        // console.log(req.body);

        const collection = db.collection("students")
        const result = await collection.insertOne(req.body)
        console.log(result);

        // const students= await collection.find().toArray();
        resp.send("data saved")
    })

    app.post("/add-student-api", async (req, resp) => {
        console.log(req.body);
        const { name, age, email } = req.body;
        if (!name || !age || !email) {
            resp.send({ message: "operation failed", success: false })
            return false
        }
        const collection = db.collection("students");
        const result = await collection.insertOne(req.body)
        resp.send({ message: "data stored", success: true, result: 
result })
    })

    app.delete("/delete/:id", async (req, resp) => {
        console.log(req.params.id);
        const collection = db.collection("students")
        const result = await collection.deleteOne({ _id: new 
ObjectId(req.params.id) })
        if (result) {
            resp.send({
                message: "student data deleted",
                success: true
            })
        } else {
            resp.send({
                message: "student data not deleted, try after 
sometime",
                success: false
            })
        }
    })

    app.get("/ui/delete/:id", async (req, resp) => {
        console.log(req.params.id);
        const collection = db.collection("students")
        const result = await collection.deleteOne({ _id: new
ObjectId(req.params.id) })
        if (result) {
            resp.send("<h1>Student record deleted<h1>")
        } else {
            resp.send("<h1>Student record  not deleted<h1>")
        }
    })

    app.get("/ui/student/:id", async (req, resp) => {
        const id = req.params.id;
        console.log(id);
        const collection = db.collection("students")
        const result = await collection.findOne({ _id: new
ObjectId(req.params.id) })
        resp.render('update-student', { result })
    })

    app.get("/student/:id", async (req, resp) => {
        const id = req.params.id;
        console.log(id);
        const collection = db.collection("students")
        const result = await collection.findOne({ _id: new
ObjectId(req.params.id) })
        resp.send({
            message: 'data fetched',
            success: true,
            result: result
        })
    })

    app.post("/ui/update/:id", (req, resp) => {
        console.log(req.body);
        console.log(req.params.id);

        const collection = db.collection("students");
        const filter = { _id: new ObjectId(req.params.id) };
        const update = { $set: req.body }
        const result = collection.updateOne(filter,update)

        if (result) {
            resp.send("data updated")
        } else {
            resp.send("data not updated")
        }

    })



        app.put("/update/:id", (req, resp) => {
        console.log(req.body);
        console.log(req.params.id);

        const collection = db.collection("students");
        const filter = { _id: new ObjectId(req.params.id) };
        const update = { $set: req.body }
        const result = collection.updateOne(filter,update)

        if (result) {
              resp.send({
            message: 'data updated',
            success: true,
            result: req.body
        })
        } else {
             resp.send({
            message: 'data not updated',
            success: false,
            result: null
        })
        }

    })

})

// app.set("view engine",'ejs')
```

```
app.listen(3200)
```

views/students.ejs

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Document</title>
</head>
<body>
    <h1>Data will be here</h1>

    <table border="1">
        <tr>
            <td>Name</td>
            <td>Age</td>
            <td>Email</td>
            <td>Operation</td>

        </tr>

<% students.forEach(student=>{ %>
 <tr>
            <td><%=  student.name %></td>
             <td><%=  student.age %></td>
             <td><%=  student.email %></td>
             <td><a href="/ui/delete/<%=  student._id %>">Delete</a>
</td>
<td><a href="/ui/student/<%=  student._id %>">Update</a></td>
```

```
        </tr>
<% }) %>

    </table>
</body>
</html>
```

views/update-student.ejs

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Add Students</title>
</head>
<body>
    <form method="post" action="/ui/update/<%= result._id %>">
            <input type="text" name="name" value="<%= result.name %>"
placeholder="enter student name" />
            <br/><br/>
            <input type="text" name="email" value="<%= result.email
%>" placeholder="enter student email" />
            <br/><br/>
            <input type="text" name="age" value="<%= result.age %>"
placeholder="enter student age" />
            <br/><br/>
            <button>Update  Student</button>
            </form>
</body>
</html>
```

Home About Contact