Connecting **MongoDB** with **Node.js** is simple using the official **MongoDB Node.js driver**:-

Option 1: Using the Official MongoDB Driver

1. Install the MongoDB driver

Run this command in your project folder:

mkdir my-node-server

cd my-node-server

npm init –y

npm install express

npm install mongodb

Create db1.js file code:-

```
const { MongoClient } = require("mongodb");

// Replace with your MongoDB connection string
const uri = "mongodb://localhost:27017"; // or use your Atlas URL

const client = new MongoClient(uri);

async function connectDB() {
  try {
    await client.connect();
    console.log("✅ Connected to MongoDB!");
    const db = client.db("myrecord"); // choose your DB name
    return db;
  } catch (err) {
    console.error("❌ MongoDB connection failed:", err);
  }
}

module.exports = connectDB;
```

Create file server.js file:-

```
const express = require("express");
const connectDB = require("./db1");
const { ObjectId } = require("mongodb");
const app = express();
app.use(express.json());

// 🟢 Fetch all users
app.get("/", async (req, res) => {
  try {
    const db = await connectDB();
    const users = await db.collection("users").find().toArray();
    res.json(users);
  } catch (err) {
    res.status(500).json({ error: err.message });
  }
});

// 🟡 Insert user
app.post("/user-insert", async (req, res) => {
  try {
    const db = await connectDB();
    const result = await db.collection("users").insertOne(req.body);
    res.status(201).json({
      message: "✅ User inserted successfully!",
      insertedId: result.insertedId, // <- will now work
    });
  } catch (err) {
    res.status(400).json({ error: err.message });
  }
});
```

```
app.put("/user-update/:id", async (req, res) => {
  try {
    const db = await connectDB();
    const { id } = req.params;

    // Convert string id to ObjectId
    const result = await db.collection("users").updateOne(
      { _id: new ObjectId(id) },
      { $set: req.body }
    );

    if (result.matchedCount === 0) {
      return res.status(404).json({ message: "User not found" });
    }

    res.json({ message: "✅ User updated successfully!" });
  } catch (err) {
    res.status(400).json({ error: err.message });
  }
});
```

```
app.delete("/user-delete/:id", async (req, res) => {
  try {
    const db = await connectDB();
    const { id } = req.params;

    const result = await db.collection("users").deleteOne({ _id: new ObjectId(id) });

    if (result.deletedCount === 0) {
      return res.status(404).json({ message: "User not found" });
    }

    res.json({ message: "✅ User deleted successfully!" });
  } catch (err) {
    res.status(400).json({ error: err.message });
  }
});
```

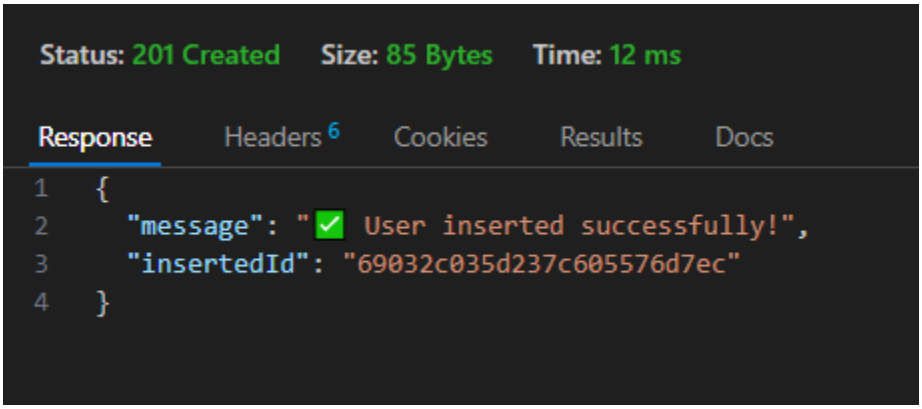
```
app.listen(3000, () => console.log("🚀 Server running on port 3000"));
```

Run it:-

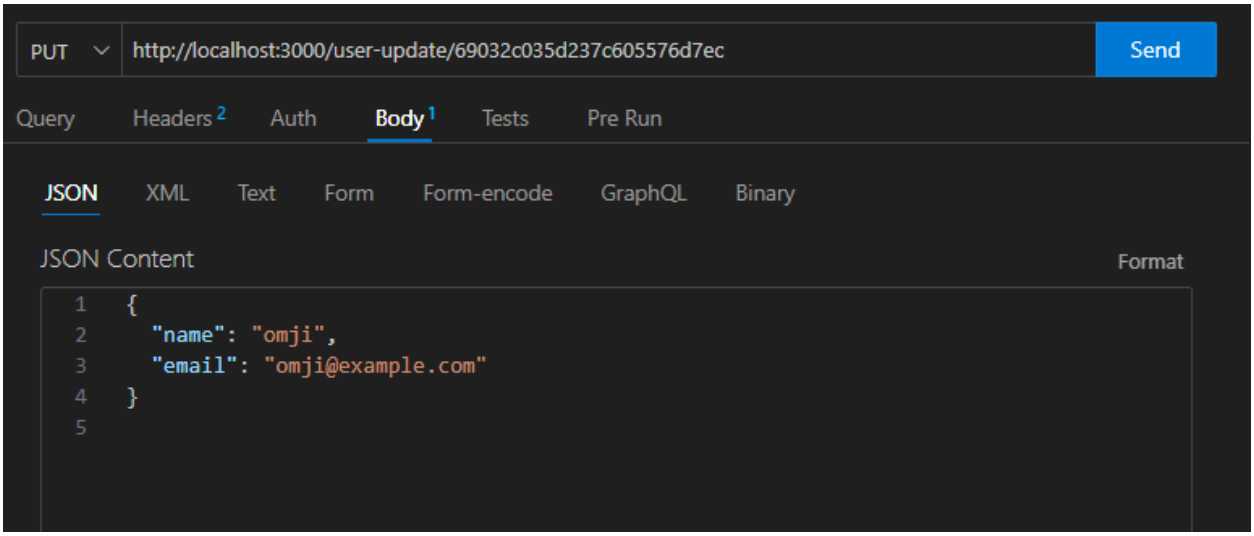
node server.js

for insert record :-

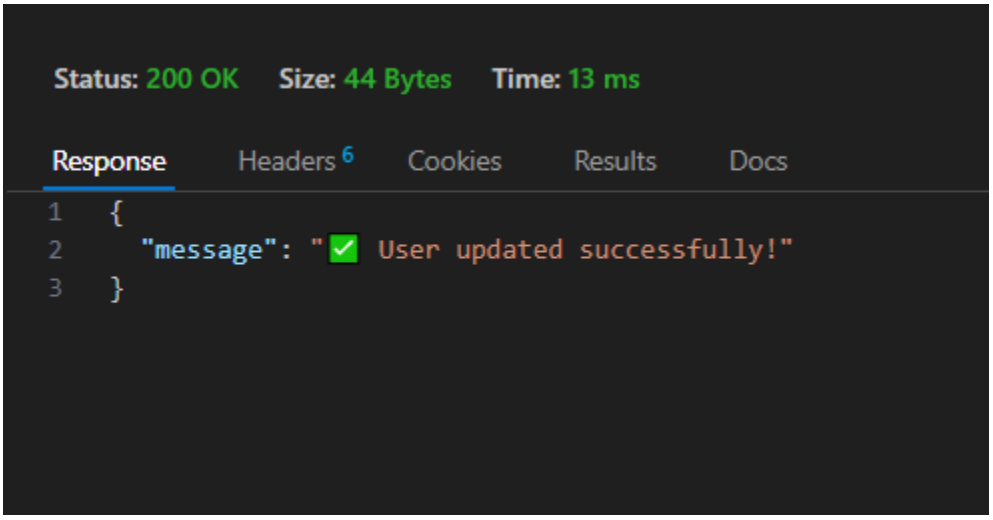
And you will see response:-



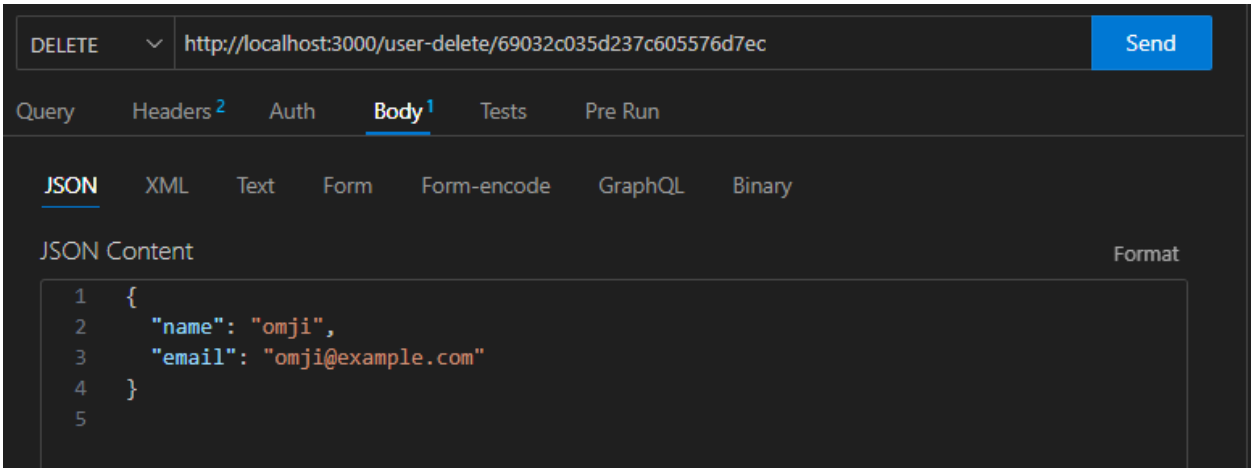
For update user record :-



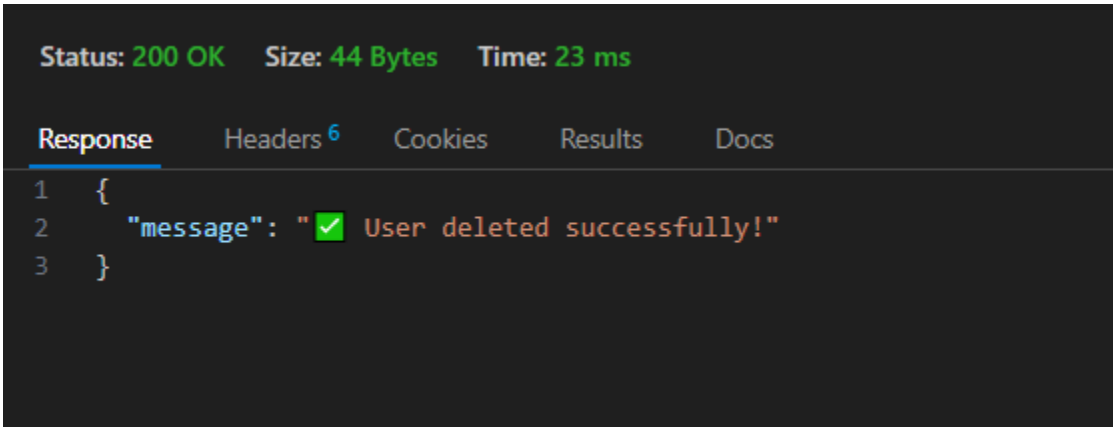
Response :-



for delete in vs code thunder client make new request as



Response:-


Status: 200 OK
Size: 44 Bytes
Time: 23 ms
Response
Headers 6
Cookies
Results
Docs
1 {
2 "message": "✅ User deleted successfully!"
3 }