Don Bosco Institute of Technology

Kurla (W), Mumbai 400 070

Web Lab

Name: Afzal Siddiquie

Class: IT

Roll no.: 53

Date: 17/04/2024

Experiment no.10

Title: Create Typescript REST API using MongoDB

Code:

basic example of how to create a TypeScript REST API using Express.js and MongoDB as the database.

First, make sure you have Node.js and npm installed on your system. Then, you can follow these steps:

1. Initialize your project:

```
mkdir ts-rest-api
cd ts-rest-api
npm init -y
```

2. Install necessary dependencies:

```
npm install express mongoose body-parser
npm install --s
```

3. Create a `tsconfig.json` file in the root directory:

```
{
  "compilerOptions": {
    "target": "es6",
```

```
    "module": "commonjs",
    "outDir": "./dist",
    "strict": true
  },
  "include": ["src/**/*.ts"],
  "exclude": ["node_modules"
```

4. Create a folder named **src**, and within it, create a TypeScript file **server.ts**:

```
import express, { Request, Response } from 'express';
import mongoose from 'mongoose';
import bodyParser from 'body-parser';

const app = express();
const PORT = process.env.PORT || 3000;

// Middleware
app.use(bodyParser.json());

// MongoDB setup
mongoose.connect('mongodb://localhost:27017/mydb', {
  useNewUrlParser: true,
  useUnifiedTopology: true,
});

const db = mongoose.connection;
db.on('error', console.error.bind(console, 'MongoDB connection error:'));

// Define schema and model
const Schema = mongoose.Schema;
const UserSchema = new Schema({
  name: String,
  email: String,
});

const UserModel = mongoose.model('User', UserSchema);

// Routes
app.get('/users', async (req: Request, res: Response) => {
  try {
    const users = await UserModel.find();
    res.json(users);
  } catch (err) {
    res.status(500).json({ message: err.message });
  }
});

app.post('/users', async (req: Request, res: Response) => {
  const user = new UserModel({
    name: req.body.name,
```

```
    email: req.body.email,
  });

  try {
    const newUser = await user.save();
    res.status(201).json(newUser);
  } catch (err) {
    res.status(400).json({ message: err.message });
  }
});

app.listen(PORT, () => {
  console.log(`Server is listening on port ${PORT}`);
});
```

5. Run TypeScript compiler to compile the code:

```
npx tsc
```

6. Start the server:

```
node dist/server.js
```

This sets up a basic Express.js server with TypeScript that connects to a MongoDB database and provides REST API endpoints for creating and fetching users. Make sure to replace the MongoDB connection URL (`'mongodb://localhost:27017/mydb'`) with your actual MongoDB database URL.

You can test the API using tools like Postman or by making HTTP requests from your frontend application.

References

[1] https://www.mongodb.com/compatibility/using-typescript-with-mongodb-tutorial

[2] https://tomanagle.medium.com/build-a-rest-api-with-node-js-typescript-mongodb