

NAME: MUSUNURU VARSHA

TRACK:FULL STACK

COLLEGE:Vignan's Nirula Institute of Technology and Science for
Women(20NN1A0590)

Phone: 7093688917

ASSIGNMENT-3

Restful API using express.js to perform CRUD operations on database with MongoDB

CODE:

```
const express = require('express');
const { MongoClient, ObjectId } = require('mongodb');

const app = express();
const PORT = process.env.PORT || 3000;
const MONGODB_URI = 'mongodb://localhost:27017/your_database_name';

app.use(express.json());

// Connect to MongoDB
let db;
MongoClient.connect(MONGODB_URI, { useUnifiedTopology: true })
  .then(client => {
    console.log('Connected to MongoDB');
    db = client.db();
  })
  .catch(err => console.error('Error connecting to MongoDB:', err));

// Routes
app.get('/api/resource', async (req, res) => {
  try {
    const resources = await db.collection('resources').find().toArray();
    res.json(resources);
  } catch (err) {
    console.error('Error fetching resources:', err);
    res.status(500).json({ error: 'Internal server error' });
  }
})
```

```

});

app.post('/api/resource', async (req, res) => {
  try {
    const result = await db.collection('resources').insertOne(req.body);
    res.json(result.ops[0]);
  } catch (err) {
    console.error('Error creating resource:', err);
    res.status(500).json({ error: 'Internal server error' });
  }
});

app.put('/api/resource/:id', async (req, res) => {
  try {
    const { id } = req.params;
    const updatedResource = await db.collection('resources').findOneAndUpdate(
      { _id: ObjectId(id) },
      { $set: req.body },
      { returnOriginal: false }
    );
    res.json(updatedResource.value);
  } catch (err) {
    console.error('Error updating resource:', err);
    res.status(500).json({ error: 'Internal server error' });
  }
});

app.delete('/api/resource/:id', async (req, res) => {
  try {
    const { id } = req.params;
    await db.collection('resources').deleteOne({ _id: ObjectId(id) });
    res.json({ message: 'Resource deleted successfully' });
  } catch (err) {
    console.error('Error deleting resource:', err);
    res.status(500).json({ error: 'Internal server error' });
  }
});

// Start the server
app.listen(PORT, () => {
  console.log(`Server is running on http://localhost:${PORT}`);
});

```

README

- This project is a simple RESTful API built using Express.js and MongoDB, allowing users to perform CRUD (Create, Read, Update, Delete) operations on a resource.
- Make sure MongoDB is installed and running locally on the default port
- The server will start running on <http://localhost:3000> by default.

API Endpoints

GET /api/resource

-Description: Retrieve all resources from the database.

-Example Response:

```
{  
  "_id": "6092d5e1f49fa84ac8b862a8",  
  "name": "Resource 1",  
  "description": "Description of Resource 1"  
},  
{  
  "_id": "6092d5f7f49fa84ac8b862a9",  
  "name": "Resource 2",  
  "description": "Description of Resource 2"
```

```
}
```

####POST /api/resource

Description: Create a new resource in the database.

Request Body Example:

```
{  
  "name": "New Resource",  
  "description": "Description of the new resource"  
}
```

Example Response:

```
{  
  "_id": "6092d682f49fa84ac8b862aa",  
  "name": "New Resource",  
  "description": "Description of the new resource"  
}
```

####PUT /api/resource/:id

Description: Update a specific resource identified by its ID.

Request Body Example:

```
{  
  "name": "Updated Resource",  
  "description": "Updated description of the resource"
```

```
}
```

Example Response:

```
{  
  "_id": "6092d682f49fa84ac8b862aa",  
  "name": "Updated Resource",  
  "description": "Updated description of the resource"  
}
```

###DELETE /api/resource/:id

Description: Delete a specific resource identified by its ID.

Example Response:

```
{  
  "message": "Resource deleted successfully"  
}
```

Explanation of Routes: each route and its functionality

/api/resource (GET): Retrieve all resources from the database.

/api/resource (POST): Create a new resource in the database.

/api/resource/:id (PUT): Update a specific resource identified by its ID.

/api/resource/:id (DELETE): Delete a specific resource identified by its ID.