

app.js

```
const express = require('express');

const mongoose = require('mongoose');

const app = express();

const port = 3001;

// Connect to MongoDB

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

const db = mongoose.connection;

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

db.once('open', () => {

  console.log('Connected to MongoDB');

});

// Define a simple model

const Item = mongoose.model('Item', {

  id:Number,

  name: String,

  description: String,

});

// Middleware for parsing JSON

app.use(express.json());

// Routes

app.get('/api/items', async (req, res) => {

  try {

    const items = await Item.find();

    res.json(items);

  } catch (err) {

    res.status(500).json({ error: err.message });

  }

});
```

```

});

app.post('/api/items', async (req, res) => {
  const { name, description } = req.body
  try {
    const newItem = new Item({ name, description });
    await newItem.save();
    res.status(201).json(newItem);
  } catch (err) {
    res.status(500).json({ error: err.message });
  }
});

app.put('/api/items/:id', async (req, res) => {
  const { name, description } = req.body;
  const { id } = req.params;

  try {
    const updatedItem = await Item.findByIdAndUpdate(id, { name, description }, { new: true });
    res.json(updatedItem);
  } catch (err) {
    res.status(500).json({ error: err.message });
  }
});

app.delete('/api/items/:id', async (req, res) => {
  const { id } = req.params;

  try {
    await Item.findByIdAndDelete(id);
    res.json({ message: 'Item deleted successfully' });
  } catch (err) {
    res.status(500).json({ error: err.message });
  }
}

```

```
});

// noStart the server

app.listen(port, () => {

  console.log(Server is running on http://localhost:${port});

});
```

GET METHOD:

The screenshot shows the Thunder Client interface with a GET request to `http://localhost:3001/api/items` successfully executed. The response is a JSON array of two items.

Request Details:

- Method: GET
- URL: `http://localhost:3001/api/items`
- Status: 200 OK
- Size: 521 Bytes
- Time: 5 ms

Response Body:

```
[
  {
    "_id": "65f46fda99e02d0daf1f7232",
    "name": "pavani",
    "description": "hi i am pursuing mca",
    "_v": 0
  },
  {
    "_id": "65f46fef99e02d0daf1f7234",
    "name": "sneha",
    "description": "hi i am pursuing mca",
    "_v": 0
  }
]
```

Terminal Output:

```
Connected to MongoDB
PS F:\OneDrive\Desktop\restapi\api> ^C
PS F:\OneDrive\Desktop\restapi\api> ^C
PS F:\OneDrive\Desktop\restapi\api> node route.js
(node:24136) [MONGODB DRIVER] Warning: useNewUrlParser is a deprecated option: useNewUrlParser has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
(Use 'node --trace-warnings ...' to show where the warning was created)
(node:24136) [MONGODB DRIVER] Warning: useUnifiedTopology is a deprecated option: useUnifiedTopology has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
Server is running on http://localhost:3001
Connected to MongoDB
```

PUT METHOD:

The screenshot shows the Thunder Client interface with a PUT request to `http://localhost:3001/api/items/65f46fef99e02d0daf1f7234`. The request body is a JSON object: `{ "name": "ammulu", "description": "hi i am pursuing mca" }`. The response is a 200 OK status with a size of 95 Bytes and a time of 11 ms. The response body is a JSON object: `{ "_id": "65f46fef99e02d0daf1f7234", "name": "ammulu", "description": "hi i am pursuing mca", "_v": 0 }`. The terminal shows the command `node route.js` being executed, with warnings about deprecated options and a message indicating the server is running on `http://localhost:3001`.

POST METHOD:

The screenshot shows the Thunder Client interface with a POST request to `http://localhost:3001/api/items`. The request body is a JSON object: `{ "name": "sneha", "description": "hi i am pursuing mca" }`. The response is a 201 Created status with a size of 94 Bytes and a time of 8 ms. The response body is a JSON object: `{ "name": "sneha", "description": "hi i am pursuing mca", "_id": "65f46fef99e02d0daf1f7234", "_v": 0 }`. The terminal shows the command `node route.js` being executed, with warnings about deprecated options and a message indicating the server is running on `http://localhost:3001`.

DELETE METHOD:

The screenshot displays the Thunder Client interface with a DELETE request configured. The URL is `http://localhost:3001/api/items/65f46fef99e02d0daf17234`. The request body is a JSON object: `{ "name": "ammulu", "description": "hi i am pursuing mca" }`. The response is a 200 OK status with a JSON body: `{ "message": "Item deleted successfully" }`. The terminal shows the server running on `http://localhost:3001`.

Request Details:

- Method: DELETE
- URL: `http://localhost:3001/api/items/65f46fef99e02d0daf17234`
- Body (JSON):

```
1 {
2   {
3     "name": "ammulu",
4     "description": "hi i am pursuing mca"
5   }
6 }
```

Response Details:

- Status: 200 OK
- Size: 39 Bytes
- Time: 6 ms
- Body (JSON):

```
1 {
2   "message": "Item deleted successfully"
3 }
```

Terminal Output:

```
Connected to MongoDB
PS F:\OneDrive\Desktop\restapi\api> ^C
PS F:\OneDrive\Desktop\restapi\api> ^C
PS F:\OneDrive\Desktop\restapi\api> node route.js
(node:24136) [MONGODB DRIVER] Warning: useNewUrlParser is a deprecated option: useNewUrlParser has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
(Use 'node --trace-warnings ...' to show where the warning was created)
(node:24136) [MONGODB DRIVER] Warning: useUnifiedTopology is a deprecated option: useUnifiedTopology has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
Server is running on http://localhost:3001
Connected to MongoDB
[]
```