

SOURCE CODE server.js

```
Const express = require('express'); const mongoose = require('mongoose'); const cors = require('cors');  
const todoRoutes = require('./routes/todoRoutes');
```

```
Const app = express();
```

```
// Middleware app.use(cors()); app.use(express.json());
```

```
// MongoDB connection (optional — skip if using local array)  
mongoose.connect('mongodb://localhost/todoapp')
```

```
.then(() => console.log('✅ MongoDB Connected'))
```

```
.catch(err => console.log(err));
```

```
// Routes
```

```
App.use('/api/todos', todoRoutes);
```

```
// Start server const PORT = 5000; app.listen(PORT, () => console.log(`🚀 Server running on port ${PORT}`));
```

Routes/todoRoutes.js

```
Const express = require('express'); const router = express.Router(); const { getTodos, addTodo, deleteTodo } = require('../controllers/todoController');
```

```
// Define routes router.get('/', getTodos); router.post('/', addTodo); router.delete('/:id', deleteTodo);
```

```
Module.exports = router;
```

Controllers/todoController.js

```
Const Todo = require('../models/ToDo');
```

```
// Get all todos exports.getTodos = async (req, res) => { const todos = await Todo.find();  
res.json(todos);
```

```
};
```

```
// Add a new todo exports.addTodo = async (req, res) => { const newTodo = new Todo({ task:  
req.body.task, completed: false
```

```
});
```

```
Await newTodo.save(); res.json(newTodo);
```

```
};
```

```
// Delete a todo exports.deleteTodo = async (req, res) => { await  
Todo.findByIdAndDelete(req.params.id); res.json({ message: 'Todo deleted' });
```

```
};
```

Models/ToDo.js

```
Const mongoose = require('mongoose');
```

```
Const todoSchema = new mongoose.Schema({  
  
  Task: String,  completed: { type: Boolean, default: false }  
  
});
```

```
Module.exports = mongoose.model('Todo', todoSchema);
```

Package.json (main parts)

```
{  
  
  "name": "todo-backend",  
  
  "version": "1.0.0",  
  
  "main": "server.js",  
  
  "scripts": {  
  
    "start": "node server.js",  
  
    "dev": "nodemon server.js"
```

```
},
```

```
  "dependencies": {
```

```
    "cors": "^2.8.5",
```

```
    "express": "^4.19.2",
```

```
    "mongoose": "^8.0.3"
```

```
  },
```

```
  "devDependencies": {
```

```
    "nodemon": "^3.1.0"
```

```
  }
```

```
}
```