Node.js + Express + MongoDB Project Structure

This document provides a professional overview of the file structure, sequence of file creation, and their purpose when building a Node.js + Express application with MongoDB integration. The document covers the purpose of each file and explains why the sequence of creation matters for the scalability and maintainability of the project.

# File Sequence and Purpose

|  |  |  |
| --- | --- | --- |
| File Name | Purpose | Explanation |
| server.js | Main entry point for the application | The `server.js` file is the entry point of the Node.js application. It initializes the app and imports the main Express app from `src/app.js`. This file starts the server and listens on a specified port. |
| src/app.js | Express app setup | The `app.js` file sets up the Express application, configures middlewares, and imports routes. It acts as the central hub for defining how requests are handled throughout the application. |
| src/routes/userRoutes.js | Route handling | The `userRoutes.js` file contains the actual routing logic for user-related operations. It maps incoming requests to the appropriate controller actions based on the route. |
| src/controllers/userController.js | Business logic | The `userController.js` file defines the business logic or actions that should be performed when certain routes are hit. This could include operations like creating users or fetching user data. |
| .env | Environment variables | The `.env` file is used for storing sensitive data such as the MongoDB connection string, server port, and other environment-specific configuration settings that should not be hard-coded in the application. |
| .gitignore | Git ignore configuration | The `.gitignore` file tells Git which files or directories to ignore. This includes the `node\_modules/` directory and `.env` file to keep sensitive data and dependencies out of the version control system. |

# Conclusion

This document outlines the file structure and sequence of creating a Node.js + Express app. Following this structure ensures that your project remains maintainable and scalable as it grows. The sequence from server setup to route and controller connections is designed to ensure clarity in how requests are handled, making the project more modular and easier to extend in the future.