

User Guide for Node.js Template Project

Introduction

Welcome to the Node.js Template Project! This template provides a modular, production-ready foundation for building scalable web applications or APIs. This guide will walk you through the setup, customization, and usage of the template so you can get started building your project efficiently.

1. Requirements

Before you begin, make sure you have the following installed:

- **Node.js** (v14 or later)
 - **MongoDB** (or any database of your choice, with easy configuration)
 - **Git** (for cloning the repository)
-

2. Getting Started

Installing Dependencies

Once you've cloned the project, you need to install the necessary dependencies.

1. In the project directory, run:
2. `npm install`

This will install all the required Node.js packages listed in the `package.json` file.

3. Configuration

Environment Variables

The project uses environment variables for sensitive information like database connection strings and secret keys.

1. In the root directory of your project, create a `.env` file.
2. Add the following configurations (or adapt them to your needs):
3. `DB_URI=mongodb://your-database-uri`
4. `JWT_SECRET=your-secret-key`
5. `PORT=3000`

Ensure that your database URI and secret keys are correctly set.

4. Setting Up the Database

The template is configured to work with MongoDB by default. To set up your database:

1. Install MongoDB locally or use a cloud database provider (e.g., MongoDB Atlas).
2. If you're using a database other than MongoDB, modify the `config/database.js` file to suit your database configuration.

5. Running the Application

Once you've configured your environment, you can run the application in development mode.

1. In the project directory, run:
2. `npm start`

This will start the server on the port you specified in the `.env` file (default is 3000).

6. API Endpoints

The project follows a RESTful API architecture. You can modify or add new routes as needed.

- **Authentication:** The authentication routes are handled under the `auth` module. Use JWT tokens for secure access.
- **User Model:** The default User model is located in `models/User.js`. You can extend this model with additional fields as required by your application.
- **API Routes:** Routes for the API are located in the `routes` directory. You can easily add new routes or modify existing ones.

7. Logging and Error Handling

The template includes basic logging and global error handling:

- **Logging:** Use the `utils/logger.js` file to set up logging with your preferred logging library (e.g., Winston, Bunyan).
- **Error Handling:** The global error handler is located in `middleware/errorHandler.js`. Customize the error messages and response format to suit your application's needs.

8. Customization

Adding New Features:

To extend your project, simply add new modules by creating separate directories and files within the `src` folder.

For example, to add a new "notifications" feature:

1. Create a new folder called notifications.
2. Inside, create routes/notifications.js, controllers/notificationsController.js, and models/notification.js.

Integrating Front-End:

This template doesn't include a front-end, but it can be easily integrated with a front-end framework (e.g., React, Angular, Vue) or used as a back-end API for your existing application.

9. Testing

To ensure your application works as expected, it's important to write unit and integration tests. You can set up a testing framework like **Jest** or **Mocha**.

The basic test structure is provided under the test directory. To run tests, you can use the following command:

```
npm test
```

10. Deployment

When you're ready to deploy the application, you can host it on platforms like **Heroku**, **AWS**, or **DigitalOcean**.

1. Make sure all environment variables are configured properly in the hosting provider's settings.
 2. For production, run the following to build the application:
 3. `npm run build`
 4. Deploy your application following your hosting provider's deployment guidelines.
-

11. Troubleshooting

If you encounter any issues, consider the following:

- **Database Connection Issues:** Check the .env file for correct database URIs.
- **Missing Dependencies:** Ensure all dependencies are installed with `npm install`.
- **Server Errors:** Check the server logs and middleware/errorHandler.js for possible issues.

For further support or troubleshooting, feel free to reach out through the support channels provided.

12. License and Terms

This project is licensed under THECODECRAFTSMEN and can be used, modified, and redistributed with proper attribution.