

1. Create Nestjs project by using CLI

-run cmd: 'npm i -g @nestjs/cli'

-run cmd: 'nest new project', this is the 'messages' project in the folder

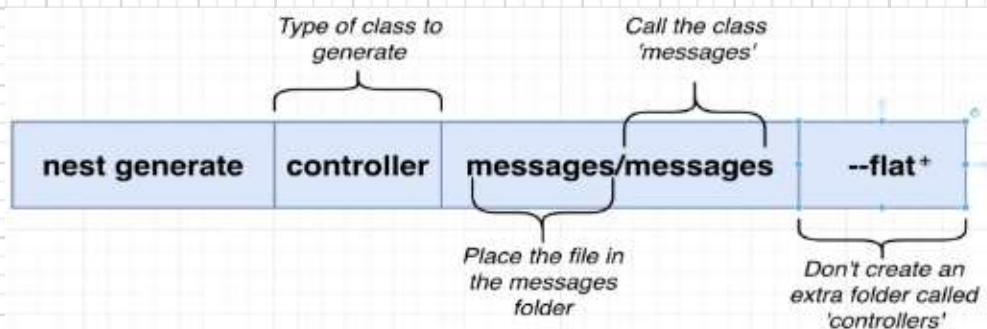
2. After creating the project, use cmd 'npm run start:dev' to run the app in watch mode

3. You can delete the auto-generated files inside the 'src' folder except the 'main.ts' if you want to build your own files

4. Then run cmd 'nest generate module projectname' to generate your own files

5. A new folder with projectname will be created inside 'src' with a 'projectname.module.ts' file

6. Create a controller by using cmd: 'nest generate controller projectname/projectname --flat'



(optional) Use VSCode REST Client Extension to test APIs:

-Install the extension in VSCode

-Created a request.html inside root directory and check the content in example folder

Validate request data with pipes:

-Import ValidationPipe in main.ts

-Add corresponding syntax to the 'app'

Setting Up Automatic Validation

- 1 Tell Nest to use global validation
- 2 Create a class that describes the different properties that the request body should have
Data transfer object. Dto
- 3 Add validation rules to the class
- 4 Apply that class to the request handler

Create dtos folders under 'src/messages'

Implement a repository

-Create 'projectname.repository.ts' and 'projectname.service.ts'

-Look at the exmaple folder for content in these files

Create repo → Create Service → Create controller

Dependency Injection: Inversion control

DI Container Flow

