NestJS | 3

Bootcamp

Discord | Official Documentation

Kushagra Acharya

Disclaimer

- This is an optional course and will not effect your academic credit
- If you're not interested and cannot fullfill any requirement or class rules you will be resulted for class dropout.

General Rules

- Having a laptop and a separate notebook is compulsory

- Faliure to answer at least 3 viva question will result in dissmissal.

- Faliure to complete homework/classwork without any valid result will be unacceptable.

Prerequisite

- Separate notebook/copy for notes

- NVM with Node Installed

- PC with VS Code Installed

- Stable Internet Connection
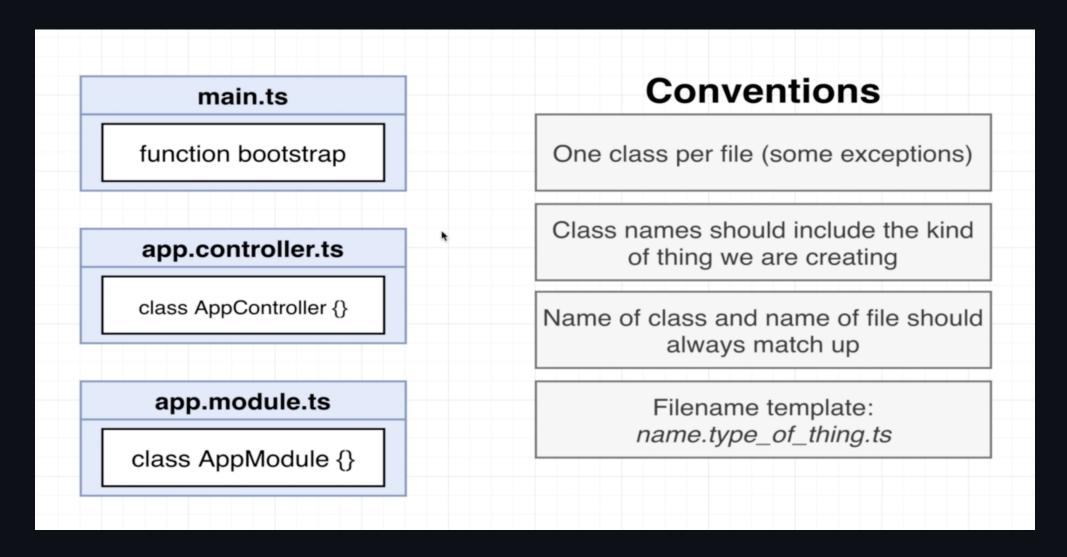
Setup and Modification

Extensions

Install the following in VSCode

- NestJS

- Prettier

Structuring

- Currently we have all our classes in a single file

- This is not a very good approach for development

- We will now extract the app controller and the app module to a separate file

Before that: Conventions

main.ts
function bootstrap

app.controller.ts
class AppController {}

app.module.ts
class AppModule {}

Conventions

One class per file (some exceptions)

Class names should include the kind of thing we are creating

Name of class and name of file should always match up

Filename template:
name.type_of_thing.ts

Creating files

- Create `app.controller.ts` file

- Move your `AppController class` to this file

- Fix any imports if necessary

- Do the same for `AppModule class` in `app.module.ts` file

main.ts

- The `main.ts` file should now only contain the `bootstrap()`
- But you will need to fix the import for `AppModule` in *main.ts* as it is being used as `NestFactory.create(AppModule)`

Run the application

- `npx ts-node-dev src/main.ts`

Discussion

- Importance of naming conventions

- What is separation of concern

- Help in project structuring

- Project scalability

Congratulations

Level Completed!

- Structuring for files and naming