**✅ tsconfig.json**

**📂 What is tsconfig.json?**

This file defines the **TypeScript compiler options** and **project structure**. It ensures consistency across the development team and optimizes how TypeScript code is compiled into JavaScript.

NestJS uses TypeScript by default, so a properly configured tsconfig.json is vital for development and production builds.

**📄 Production-Ready tsconfig.json Example for NestJS**

{

"compilerOptions": {

"module": "CommonJS", // Required for Node.js compatibility

"declaration": true, // Generate .d.ts type definition files

"removeComments": true, // Removes comments from transpiled JS

"emitDecoratorMetadata": true, // Enables metadata reflection for decorators (used in NestJS)

"experimentalDecorators": true, // Enables usage of decorators (required for NestJS)

"allowSyntheticDefaultImports": true, // Allows default imports from modules with no default export

"target": "ES2021", // Sets the JS output version

"sourceMap": true, // Generates source maps for easier debugging

"outDir": "./dist", // Where compiled code goes

"baseUrl": "./", // Base path for non-relative module names

"incremental": true, // Speed up compilation using incremental builds

"strict": true, // Enables all strict type-checking options

"noImplicitAny": true, // Disallow variables with type `any` unless explicitly specified

"paths": {

"@app/\*": ["src/app/\*"],

"@common/\*": ["src/common/\*"],

"@config/\*": ["src/config/\*"],

"@modules/\*": ["src/modules/\*"]

},

"esModuleInterop": true, // Enable `import x from 'y'` style

"skipLibCheck": true // Skips type checking of declaration files

},

"include": ["src/\*\*/\*"],

"exclude": ["node\_modules", "dist"]

}

**🧠 Explanation of Key Options**

| **Option** | **Description** |
| --- | --- |
| emitDecoratorMetadata + experimentalDecorators | Required for NestJS to work with decorators |
| strict, noImplicitAny | Ensures strong typing across your code |
| baseUrl, paths | Helps with cleaner imports like @modules/auth instead of ../../../modules/auth |
| declaration | Helps in packaging your module as a library (optional) |
| outDir | Clean separation of compiled JS code (dist) from source code |

**📁 Folder Structure Affected**

/

├── src/

│ └── modules/

├── dist/

├── tsconfig.json

The compiler will:

* take all .ts files from src/
* transpile them to .js inside dist/

**📦 Usage in NestJS Scripts**

Your package.json should include:

"scripts": {

"start": "nest start",

"start:dev": "nest start --watch",

"build": "tsc -p tsconfig.json",

"start:prod": "node dist/main.js"

}

**🛠 Tips**

* If you’re using monorepo with NestJS (like Nx), tsconfig.base.json will be the root, and each app/module can extend from it.
* You can also define a tsconfig.build.json for build-specific overrides (e.g., exclude test files).