

Nest js :-

NestJS is a **backend framework for building server-side applications (APIs, web apps, microservices) using Node.js**, mainly with **TypeScript**.

Here's a **complete beginner-friendly NestJS example** using all 3 parts together:

- ☐ Controller (handles request)
- ☐ Service (logic)
- ☐ Module (connects everything)

This will create a simple **/users API**.

☐ Step 1: Create project

```
npm i -g @nestjs/cli
nest new my-nest-app
cd my-nest-app
npm run start
```

☐ Step 2: Create Users feature

Run:

```
nest generate module users
nest generate controller users
nest generate service users
```

```
D:\nestjs2026>cd my-nest-app

D:\nestjs2026\my-nest-app>nest generate module users
CREATE src/users/users.module.ts (86 bytes)
UPDATE src/app.module.ts (322 bytes)

D:\nestjs2026\my-nest-app>nest generate controller users
CREATE src/users/users.controller.ts (103 bytes)
CREATE src/users/users.controller.spec.ts (503 bytes)
UPDATE src/users/users.module.ts (174 bytes)

D:\nestjs2026\my-nest-app>nest generate service users
CREATE src/users/users.service.ts (93 bytes)
CREATE src/users/users.service.spec.ts (471 bytes)
UPDATE src/users/users.module.ts (251 bytes)
```

This creates:

```
src/users/users.module.ts
src/users/users.controller.ts
src/users/users.service.ts
```

□ Step 3: Add Service (logic)

```
□ src/users/users.service.ts

import { Injectable } from '@nestjs/common';

@Injectable()
export class UsersService {

  private users = ['Alice', 'Bob', 'Charlie'];

  getAllUsers() {
    return this.users;
  }
}
```

□ Step 4: Add Controller (API routes)

```
□ src/users/users.controller.ts

import { Controller, Get } from '@nestjs/common';
import { UsersService } from './users.service';

@Controller('users')
export class UsersController {

  constructor(private readonly usersService: UsersService) {}

  @Get()
  getUsers() {
    return this.usersService.getAllUsers();
  }
}
```

□ Step 5: Connect everything in Module

□ src/users/users.module.ts

```
import { Module } from '@nestjs/common';
import { UsersController } from '../users.controller';
import { UsersService } from '../users.service';

@Module({
  controllers: [UsersController],
  providers: [UsersService],
})
export class UsersModule {}
```

□ Step 6: Add module to main app

□ src/app.module.ts

```
import { Module } from '@nestjs/common';
import { UsersModule } from '../users/users.module';

@Module({
  imports: [UsersModule],
})
export class AppModule {}
```

▶ □ Step 7: Run project

npm run start

□ Step 8: Test in browser

Open:

<http://localhost:3000/users>

You will see:

```
["Alice", "Bob", "Charlie"]
```

☐ What you just learned

Flow of request:

Browser → Controller → Service → Controller → Response

Roles:

- Controller → receives `/users`
 - Service → returns user data
 - Module → connects both
-

☐ One-line summary

- ☐ Controller handles API
- ☐ Service handles logic
- ☐ Module connects them