

# Nestjs 中的控制器、路由、Get、Post、方法参数装饰器

主讲教师：（大地）

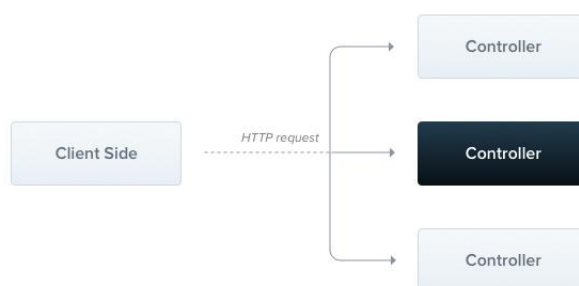
合作网站：[www.itying.com](http://www.itying.com) （IT 营）

我的专栏：<https://www.itying.com/category-79-b0.html>

1、 NestJS 中的控制器.....	1
2、 NestJS 中的路由.....	2
3、 NestJS 中的 Get Post 以及通过方法参数装饰器获取传值.....	2
4、 NestJS 中的动态路由.....	4

## 1、NestJS 中的控制器

Nest 中的控制器层负责处理传入的请求，并返回对客户端的响应。



控制器的目的是接收应用的特定请求。路由机制控制哪个控制器接收哪些请求。通常，每个控制器有多个路由，不同的路由可以执行不同的操作。

可以通过 NestCli 创建控制器:

```
nest g controller news
```

## 2、NestJS 中的路由

Nestjs 中没有单独配置路由的地方。定义好控制器后 nestjs 会自动给我们配置对应的路由。

下面代码定义了一个新闻控制器。装饰器为@Controller('article')，装饰器参数里面的'article'就是我们的路由。

如果我们要返回 index 方法里面的内容我们在浏览器输入 <http://localhost:3000/article>

如果我们要返回 add 方法里面的内容我们在浏览器输入 <http://localhost:3000/article/add>

```
import { Controller, Get } from '@nestjs/common';

@Controller('article')
export class ArticleController {
  @Get()
  index(): string {
    return '这是 article 里面的 index';
  }
  @Get('add')
  add(): string {
    return '这是 article 里面的 index';
  }
}
```

关于 nest 的 return: 当请求处理程序返回 JavaScript 对象或数组时，它将自动序列化为 JSON。但是，当它返回一个字符串时，Nest 将只发送一个字符串而不是序列化它。这使响应处理变得简单：只需要返回值，Nest 负责其余部分。

## 3、NestJS 中的 Get Post 以及通过方法参数装饰器获取传值

1、首先看一个 Get Post 请求的例子，注意我们使用 nestjs 内置装饰器的时候必须得在 @nestjs/common 模块下面引入对应的装饰器。

```
import { Controller, Get, Post } from '@nestjs/common';

@Controller('cats') export class CatsController {
  @Post()
  create(): string {
    return 'This action adds a new cat';
  }

  @Get()
  findAll(): string {
    return 'This action returns all cats';
  }
}
```

Nestjs 也提供了其他 HTTP 请求方法的装饰器 @Put() 、 @Delete()、@Patch()、@Options()、 @Head() 和 @All()

## 2、Nestjs 中获取请求参数

在 Nestjs 中获取 Get 传值或者 Post 提交的数据的话我们可以使用 Nestjs 中的装饰器来获取。

@Request() @Req()	req
@Response() @Res()	res
@Next()	next
@Session()	req.session
@Param(key?: string)	req.params / req.params[key]
@Body(key?: string)	req.body / req.body[key]
@Query(key?: string)	req.query / req.query[key]
@Headers(name?: string)	req.headers / req.headers[name]
@Ip()	req.ip
@HostParam()	req.hosts

```
import { Controller, Get, Post } from '@nestjs/common';
@Controller('news')
export class NewsController {
  @Get()
  getAbout(@Query() query): string {
    console.log(query); //这里获取的就是所有的 Get 传值
    return '这是 about'
  }
  @Get('list')
```

```
getNews(@Query('id') query):string {
    console.log(query);    //这里获取的就是 Get 传值里面的 Id 的值
    return '这是新闻'
}
@Post('doAdd')
async addNews(@Body() newsData){
    console.log(newsData);
    return '增加新闻'
}
}
```

## 4、NestJS 中的动态路由

当您需要接受动态数据作为请求的一部分时，具有静态路径的路由将不起作用（例如，GET /cats/1)获取具有 id 的 cat 1）。为了定义带参数的路由，我们可以在路由中添加路由参数标记，以捕获请求 URL 中该位置的动态值。@Get() 下面的装饰器示例中的路由参数标记演示了此用法。可以使用 @Param() 装饰器访问以这种方式声明的路由参数，该装饰器应添加到函数签名中。

```
@Get(':id')
findOne(@Param() params): string {
    console.log(params.id);
    return `This action returns a #${params.id} cat`;
}
```