

Nestjs 中使用模板引擎和配置静态资源

主讲教师：（大地）

合作网站：www.itying.com （IT 营）

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

- 1、 NestJS 中配置静态资源..... 1
- 2、 NestJS 中配置模板引擎..... 2
- 3、 NestJS 中模板引擎结合 Post 演示、以及路由跳转..... 4

1、NestJS 中配置静态资源

官方文档：<https://docs.nestjs.com/techniques/mvc>

```
app.useStaticAssets('public');
```

完整代码：

```
import { NestFactory } from '@nestjs/core';
import { AppModule } from './app.module';
import { NestExpressApplication } from '@nestjs/platform-express';
async function bootstrap() {
  const app = await NestFactory.create<NestExpressApplication>(AppModule);
  app.useStaticAssets('public');
  await app.listen(3000);
}
bootstrap();
```

在根目录新建 public 目录，然后在目录里面保存一个图片比如 1.jpg，这样就可以通过 <http://localhost:3000/1.jpg> 来访问图片。

我们也可以配置虚拟目录，比如我们想通过 <http://localhost:3000/static/1.jpg> 来访问 public 目录里面的文件，这时候代码如下：

```
app.useStaticAssets(join(__dirname, '..', 'public'), {  
  prefix: '/static/', //设置虚拟路径  
});
```

```
import { NestFactory } from '@nestjs/core';  
import { AppModule } from './app.module';  
import { NestExpressApplication } from '@nestjs/platform-express';  
import { join } from 'path';  
  
async function bootstrap() {  
  const app = await NestFactory.create<NestExpressApplication>(AppModule);  
  // app.useStaticAssets('public');  
  app.useStaticAssets(join(__dirname, '..', 'public'), {  
    prefix: '/static/', //设置虚拟路径  
  });  
  await app.listen(3000);  
}  
bootstrap();
```

2、NestJS 中配置模板引擎

官方文档: <https://docs.nestjs.com/techniques/mvc>

1、安装对应的模板引擎 比如 ejs

```
cnpm i ejs --save
```

2、配置模板引擎

```
app.setBaseViewsDir(join(__dirname, '..', 'views')) // 放视图的文件  
app.setViewEngine('ejs');
```

完整代码:

```
import { NestFactory } from '@nestjs/core';  
import { AppModule } from './app.module';  
import { NestExpressApplication } from '@nestjs/platform-express';
```

```
import {join} from 'path';

async function bootstrap() {
  const app = await NestFactory.create<NestExpressApplication>(AppModule);
  // app.useStaticAssets('public');
  app.useStaticAssets(join(__dirname, '..', 'public'),{
    prefix: '/static/',    //设置虚拟路径
  });

  app.setBaseViewsDir(join(__dirname, '..', 'views')) // 放视图的文件
  app.setViewEngine('ejs');

  await app.listen(3000);
}
bootstrap();
```

3、渲染页面

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

@Controller()
export class AppController {
  @Get()
  @Render('index')
  root() {
    return { message: 'Hello world!' };
  }
}
```

4、项目根目录新建 views 目录然后新建 index.ejs

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
</head>
<body>
```

```
这是 ejs 演示代码  
<br>  
<%=message%>  
</body>  
</html>
```

3、NestJS 中模板引擎结合 Post 演示、以及路由跳转

看教程演示

```
import { Controller, Get, Post, Body, Response, Render } from '@nestjs/common';  
  
@Controller('user')  
export class UserController {  
  
  @Get()  
  @Render('default/user')  
  index(){  
    return {"name":"张三"};  
  }  
  @Post('doAdd')  
  doAdd(@Body() body,@Response() res){  
  
    console.log(body);  
  
    res.redirect('/user'); //路由跳转  
  
  }  
}
```

模板

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
  <meta charset="UTF-8">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title>Document</title>
<link rel="stylesheet" href="/static/base.css">
</head>
<body>



<br>
<form action="/user/doAdd" method="post">

    <input type="text" name="username" placeholder="请输入用户名" />

    <br>
    <br>

    <input type="text" name="age" placeholder="年龄" />

    <br>
    <br>
    <input type="submit" value="提交">
</form>
</body>
</html>
```