



链滴


Nodejs + Nestjs 极速开发后端 API

作者: [Gakkiyomi2019](#)

原文链接: <https://ld246.com/article/1619443079468>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

  <https://b3logfile.com/bing/20210417.jpg?imageView2/1/w/960/h/540/interlace/1/q/100>


Nodejs + Nestjs 极速开发后端 API

作为一名 Java 后端程序员，一直想学习前端进行全栈开发，而前端的重中之重就是 Javascript，以学习 NodeJs 是有必要的，毕竟 NodeJs 也是使用 Js 语法，而且后端语言的思想也比较熟悉，容易入门。由于是门外汉对 js 和 ts 根本不了解，只是凭借经验尝试，讲错的地方请大家指正。

NestJs

什么是 NestJs?

Nest (NestJS) 是用于构建高效，可扩展的 Node.js 服务器端应用程序的框架。内置并完全支持 TypeScript (但仍使开发人员能够使用纯 JavaScript 进行编码)

NestJs 对于 Java 程序员来说简直不要太简单，其中的模块思想和 MVC 思想以及依赖注入的思想与 Spring 不谋而合。并且通过注解可以使 Spring 使用者感觉回家一样  。

TypeScript

在没有接触 TypeScript 之前，听到这个名字感觉恐怖如斯，js 都不太会用，ts 就出来了，太卷太卷了 :flushed:，在了解后才发现 ts 的出现大大的降低了后端程序员进入前端的门槛，众所周知 js 一门没有静态类型检查的语言，也是动态类型的语言，这使得 js 难以构建大型的复杂系统，而 typescript 的出现改变了这一现状，typescript 是 javascript 的超集，在 javascript 的基础上增加了类型系统使得一些在运行时出现的错误，被提前到编译期间产生，而且 typescript 能够提升了 IDE 的功能，包括代码补全、接口提示、跳转到定义、代码重构等，这在很大程度上提高了开发效率。而且 TypeScript 有近百个[编译选项]，如果你认为类型检查过于严格，那么可以通过修改编译选项来降低类型检查标准。

所以不必深入研究 typescript，仅仅凭借 Js 的知识和一些后端语言的 OOP 面向对象的思想就可以使用 Typescript 来进行开发。如果是后端的同学使用 typescript 的话更是事半功倍。

快速开始

本文既然是介绍 nestjs 作为服务端 API 接口开发，自然是会介绍使用方式。

所以 demo 会使用 nestjs 框架连接 mongodb 数据库来进行增删改查，并且集成 swagger 来生成接口文档。swagger 永远滴神

安装 nestjs

```
$ npm i -g @nestjs/cli
```


查看是否安装成功

 <https://b3logfile.com/file/2021/04/image-b727ec8c.png?imageView2/2/interlace/1/format/jpg>

快速启动


ps: 对于 js 项目复杂的项目结构无法 hold 住? 没关系 nestjs 一键生成

```
nest new nest-blog-api
```

 <https://b3logfile.com/file/2021/04/image-575a82e6.png?imageView2/2/interlace/1/format/jpg>

可以选择使用 npm 还是 yarn 进行安装,都可以

使用 vscode 打开 nest-blog-api 文件夹

 <https://b3logfile.com/file/2021/04/image-abf65ec5.png?imageView2/2/interlace/1/format/jpg>

<p>项目目录就已经生成好了</p>

<p>启动一下项目</p>

<p><code>npm run start:dev</code></p>

<p></p>

<p>在浏览器输入 <code>http://127.0.0.1:3000</code></p>

<p></p>

<h3 id="项目结构">项目结构</h3>

<p>项目自动生成了 helloworld 的示例代码，我们来分析一下</p>

<p></p>

<p><code>main.ts</code> 可以看出是启动的入口，监听了 3000 端口，而 <code>app.controller.ts</code>,<code>app.module.ts</code>,<code>app.service.ts</code> 更是 java 程序员喜闻乐见的 mvc 结构</p>

<p><code>app.service.ts</code> 写了一个类 <code>AppService</code>，里面有个 getHello 的方法，并且使用 <code>@Injectable()</code> 表示这个类是可以被注入的</p>

<p></p>

<p><code>app.controller.ts</code> 中注入了 <code>AppService</code> 并且定义了一个 get 请求接口，返回 helloworld,看到了熟悉的注解，根本不需任何 nest 文档也能凭借 java 开发的经验读代码。</p>

<p></p>

<p><code>app.module.ts</code> 定义了一个模块，模块是 nestjs 里面非常重要的一个概念，不同业务分成不同模块，可以清楚的将代码分类，提高可读性，也便于管理</p>

<p></p>

<p><code>@Module</code> 注解提供了 4 个属性来修饰这个模块，也就是定义这个模块的具内容</p>

<p><code>imports</code> 代表将其他引用进来</p>

<p><code>controllers</code> 代表这个模块的控制器，也就是 API 接口</p>

<p><code>providers</code> 代表注入的实例，可以在模块中共享</p>

<p><code>exports</code> 表示其他模块引用自己时，提供给其他模块访问的实例</p>

<h2 id="集成swagger">集成 swagger</h2>

```
<pre><code class="language-bash highlight-chroma"><span class="highlight-line"><span class="highlight-cl">npm install --save @nestjs/swagger swagger-ui-express</span></span></code></pre>
```

<p>然后在 <code>main.ts</code> 中写上 swagger 的配置</p>

<p></p>

<p>在浏览器打开 <code>http://127.0.0.1:3000/api-docs</code></p>

<p></p>

<h2 id="集成mongodb">集成 mongodb</h2>

<p>我们使用 mongodb 进行增删改查操作</p>

```
<pre><code class="language-bash highlight-chroma"><span class="highlight-line"><span class="highlight-cl">yarn add mongoose nestjs-typegoose @typegoose/typegoose</span></span></code></pre>
```

<p>安装完后我们先新建一个模块 <code>posts</code> 来写增删改查</p>

```
<pre><code class="language-bash highlight-chroma"><span class="highlight-line"><span class="highlight-cl">nest create mo posts //新建posts模块</span></span><span class="highlight-line"><span class="highlight-cl">nest create co pos s //新建controller</span></span></code></pre>
```

<p>在 <code>app.module.ts</code> 中导入 post 模块和 mongo 连接信息</p>

```
<p></p>
```

<p>定义好模型</p>

<p>用于数据库数据和对象 orm 映射</p>

```
<p></p>
```

<p>然后在 <code>app.controller.ts</code> 中写好接口</p>

```
<pre><code class="language-ts highlight-chroma"><span class="highlight-line"><span class="highlight-cl"><span class="highlight-kr">import</span><span class="highlight-p">{</span></span><span class="highlight-nx">Body</span><span class="highlight-p">,</span><span class="highlight-nx">Controller</span><span class="highlight-p">,</span><span class="highlight-nx">Delete</span><span class="highlight-p">,</span><span class="highlight-nx">Get</span><span class="highlight-p">,</span><span class="highlight-nx">Param</span><span class="highlight-p">,</span><span class="highlight-nx">Post</span><span class="highlight-p">,</span><span class="highlight-nx">Put</span><span class="highlight-p">}</span><span class="highlight-kr">from</span><span class="highlight-s1">'@nestjs/common'</span><span class="highlight-p">;</span></span></span><span class="highlight-line"><span class="highlight-cl"><span class="highlight-kr">import</span><span class="highlight-p">{</span><span class="highlight-nx">ApiOperation</span><span class="highlight-p">,</span><span class="highlight-nx">ApiProperty</span><span class="highlight-p">,</span><span class="highlight-nx">ApiTags</span><span class="highlight-p">}</span><span class="highlight-kr">from</span><span class="highlight-s1">'@nestjs/swagger'</span><span class="highlight-p">;</span></span></span><span class="highlight-line"><span class="highlight-cl"><span class="highlight-kr">import</span><span class="highlight-p">{</span><span class="highlight-nx">ModelType</span><span class="highlight-p">}</span><span class="highlight-kr">from</span><span class="highlight-s1">'@typegoose/typegoose/lib/types'</span><span class="highlight-p">;</span></span></span><span class="highlight-line"><span class="highlight-cl"><span class="highlight-kr">import</span><span class="highlight-p">{</span><span class="highlight-nx">InjectModel</span><span class="highlight-p">}</span><span class="highlight-kr">from</span><span class="highlight-s1">'@nestjs/typegoose'</span><span class="highlight-p">;</span></span></span><span class="highlight-line"><span class="highlight-cl"><span class="highlight-kr">import</span><span class="highlight-p">{</span><span class="highlight-nx">Post</span><span class="highlight-p">,</span><span class="highlight-kr">as</span><span class="highlight-nx">PostSchema</span><span class="highlight-p">}</span><span class="highlight-kr">from</span><span class="highlight-s1">'./post.model'</span><span class="highlight-p">;</span></span></span><span class="highlight-line"><span class="highlight-cl"><span class="highlight-kr">class</span><span class="highlight-nx">CreatePostDTO</span><span class="highlight-p">{</span></span></span>
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class="h
ghlight-kd">@ApiModelProperty</span><span class="highlight-p">({</span>
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class=
highlight-nx">description</span><span class="highlight-o">:</span><span class="highligh
-s1">'标题'</span>
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class="h
ghlight-p">})</span>
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class="h
ghlight-nx">title</span>:<span class="highlight-kt">string</span>
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class="h
ghlight-kd">@ApiModelProperty</span><span class="highlight-p">({</span>
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class=
highlight-nx">description</span><span class="highlight-o">:</span><span class="highligh
-s1">'正文'</span><span class="highlight-p">,</span>
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class="h
ghlight-p">})</span>
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class="h
ghlight-nx">content</span>:<span class="highlight-kt">string</span>
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"><span class="high
ight-p">}</span>
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"><span class="high
ight-kd">@Controller</span><span class="highlight-p">(</span><span class="highlight-s
">'posts'</span><span class="highlight-p">)</span>
</span></span><span class="highlight-line"><span class="highlight-cl"><span class="high
ight-kd">@ApiTags</span><span class="highlight-p">(</span><span class="highlight-s2"
"帖子"</span><span class="highlight-p">)</span>
</span></span><span class="highlight-line"><span class="highlight-cl"><span class="high
ight-kr">export</span> <span class="highlight-kr">class</span> <span class="highlight-nx
">PostsController</span> <span class="highlight-p">{</span>
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class="h
ghlight-kr">constructor</span><span class="highlight-p">(</span>
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class=
highlight-kd">@InjectModel</span><span class="highlight-p">(</span><span class="highl
ght-nx">PostSchema</span><span class="highlight-p">)</span> <span class="highlight-kr
">private</span> <span class="highlight-kr">readonly</span> <span class="highlight-nx">
ostModel</span>:<span class="highlight-kt">ModelType</span><span class="highlight-p"
">&lt;</span><span class="highlight-nt">PostSchema</span><span class="highlight-p">&gt;
</span>
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class="h
ghlight-p">})</span>
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class="h
ghlight-kd">@Get</span><span class="highlight-p">()</span>
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class="h
ghlight-kd">@ApiOperation</span><span class="highlight-p">({</span>
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class=
highlight-nx">summary</span><span class="highlight-o">:</span><span class="highlight-
2">"帖子列表"</span><span class="highlight-p">,</span>
</span></span></pre>
```



```

highlight-p">}</span>
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class="h
ghlight-p">}</span>
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class="h
ghlight-kd">@Delete</span><span class="highlight-p">(</span><span class="highlight-s1
>:id'</span><span class="highlight-p">)</span>
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class="h
ghlight-kd">@ApiOperation</span><span class="highlight-p">{</span>
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class=
highlight-nx">summary</span><span class="highlight-o">:</span><span class="highlight-
2">"删除帖子"</span><span class="highlight-p">,</span>
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class=
highlight-nx">description</span><span class="highlight-o">:</span><span class="highligh
-s2">"删除帖子"</span>
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class="h
ghlight-p">}</span>
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class="h
ghlight-kr">async</span> <span class="highlight-nx">remove</span><span class="highlig
t-p">(</span><span class="highlight-kd">@Param</span><span class="highlight-p">(</s
an><span class="highlight-s1">'id'</span><span class="highlight-p">)</span> <span clas
="highlight-nx">id</span>:<span class="highlight-kt">string</span><span class="highligh
-p">){</span>
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class=
highlight-k">await</span> <span class="highlight-k">this</span><span class="highlight-p
>.</span><span class="highlight-nx">postModel</span><span class="highlight-p">.</spa
><span class="highlight-nx">findByIdAndDelete</span><span class="highlight-p">(</spa
><span class="highlight-nx">id</span><span class="highlight-p">)</span>
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class=
highlight-k">return</span> <span class="highlight-p">{</span>
</span></span><span class="highlight-line"><span class="highlight-cl"> <span clas
="highlight-nx">success</span>:<span class="highlight-kt">>true</span>
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class=
highlight-p">}</span>
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class="h
ghlight-p">}</span>
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class="h
ghlight-p">}</span>
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class="high
ight-p">}</span>
</span></span></code></pre>
<h3 id="启动">启动</h3>
<pre><code class="language-bash highlight-chroma"><span class="highlight-line"><span c
ass="highlight-cl">npm run start:dev
</span></span></code></pre>
<p></
>
<p>这样一个后端 API 服务就搭建完成了。</p>
<h2 id="源码">源码</h2>
<p><a href="https://ld246.com/forward?goto=https%3A%2F%2Fgithub.com%2Fgakkiyomi
2Fnestjs-demo" target="_blank" rel="nofollow ugc">源码</a></p>

```