**码上飞应用部署操作手册**

**建议直接阅读在线部署手册，内容更新更及时哦！**

1. 基于源码的手册：

<https://vvx03gck2p.feishu.cn/docx/MoCSd6YhTopgYlxcz4qcuneNnYc>

1. 基于一键安装包的手册：

<https://vvx03gck2p.feishu.cn/docx/FbF5dPAAZod4zWxE1mUc767knrd>

**环境准备**

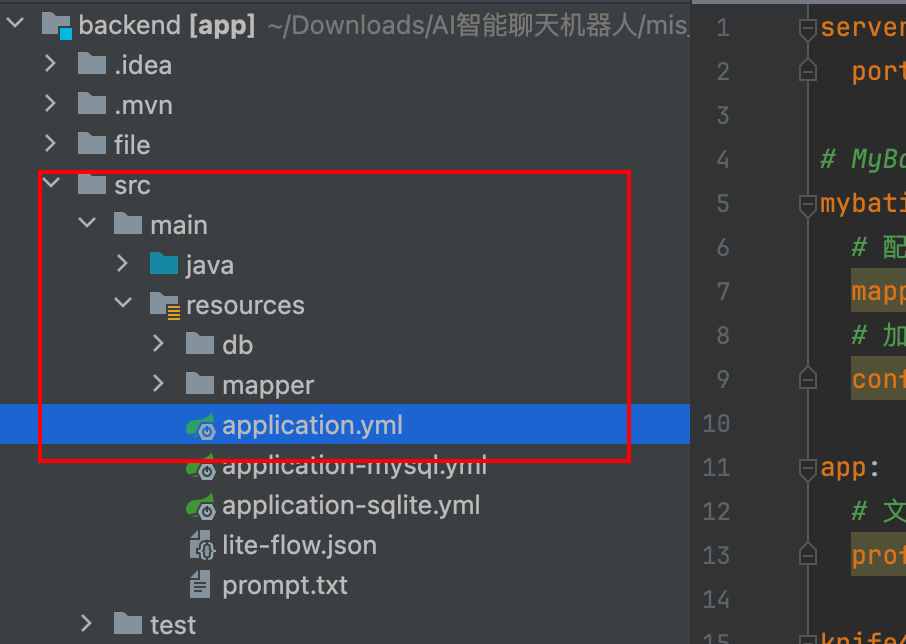
|  |
| --- |
| Shell Java 环境配置 JDK下载：https://cloud.tencent.com/document/product/1149/38537 wget https://github.com/Tencent/TencentKona-8/releases/download/8.0.9-GA/TencentKona8.0.9.b1\_jdk\_linux-x86\_64\_8u322.tar.gz maven下载：https://maven.apache.org/download.cgi wget https://dlcdn.apache.org/maven/maven-3/3.9.7/binaries/apache-maven-3.9.7-bin.tar.gz 下载完成后解压到： /usr/local/TencentKona-8.0.9-322 tar -zxvf TencentKona8.0.9.b1\_jdk\_linux-x86\_64\_8u322.tar.gz -C /usr/local /usr/local/apache-maven-3.9.7 tar -zxvf apache-maven-3.9.7-bin.tar.gz -C /usr/local/ 修改替换原来0000的mvn源 vim /usr/local/apache-maven-3.9.7/conf/settings.xml  <mirror>  <id>aliyunmaven</id>  <mirrorOf>\*</mirrorOf>  <name>阿里云公共仓库</name>  <url>https://maven.aliyun.com/repository/public</url>  </mirror> 配置环境变量： vim ~/.bash\_profile JAVA\_HOME=/usr/local/TencentKona-8.0.9-322 CLASSPATH=${JAVA\_HOME}/lib M2\_HOME=/usr/local/apache-maven-3.9.7 M2=$M2\_HOME/bin MAVEN\_OPTS="-Xms256m -Xmx512m" PATH=$PATH:$HOME/bin:${JAVA\_HOME}/bin:$M2 export PATH export JAVA\_HOME export CLASSPATH export M2\_HOME export M2 export MAVEN\_OPTS  source ~/.bash\_profile  安装 nginx yum install nginx -y |

Nginx 配置文件目录：/etc/nginx/nginx.conf

**后端**

1）找到后端配置文件 application.yml，修改 enable 配置，改为 true

|  |
| --- |
| ⚠️注意：需要登录验证的改为 true，不需要登录验证的不需要修改 |



|  |
| --- |
| YAML login:  enable: ${LOGIN\_ENABLE:true}  table: UserInfo  select:  table: UserInfo  column: phone\_number  openid:  table: UserInfo  column: wechat\_login\_id |

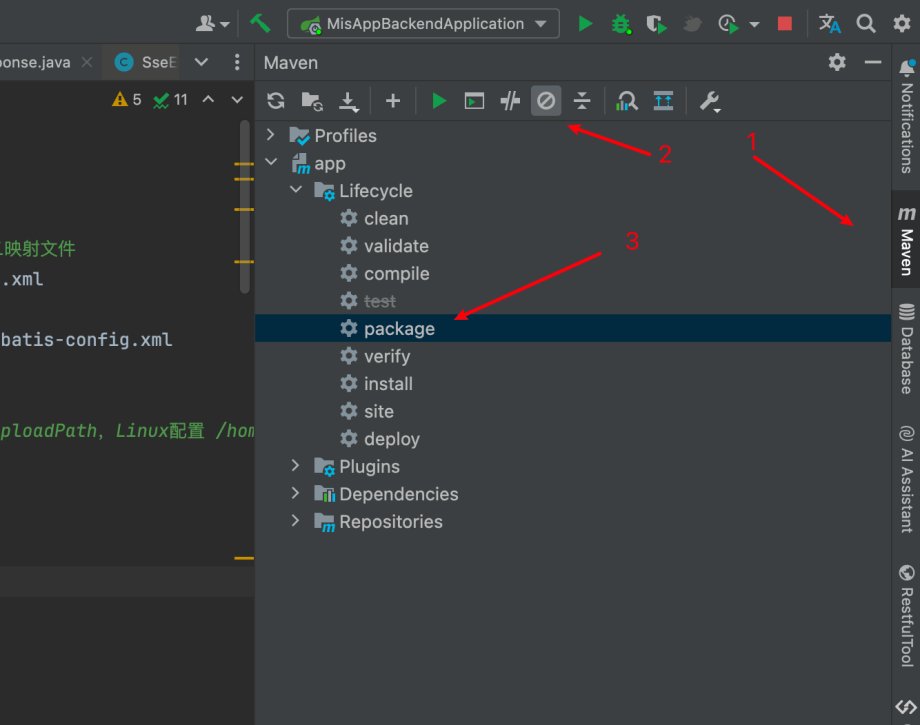
如果端口号有冲突需要修改端口号

|  |
| --- |
| YAML server:  port: ${PORT:8083} |

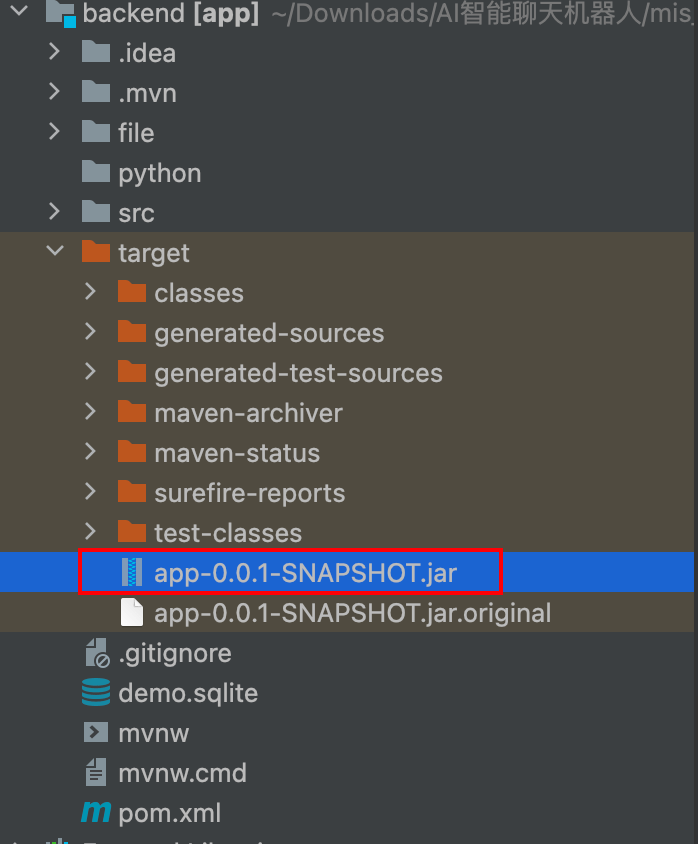
2）在项目目录打开终端进行打包或直接使用 IDEA 进行打包

|  |
| --- |
| Shell mvn install -Dmaven.test.skip=true |

或者

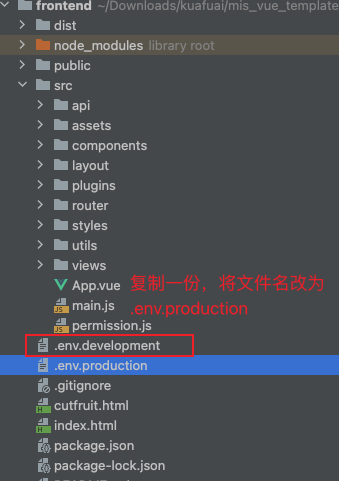


打包成功后，项目包放在 target 目录下面



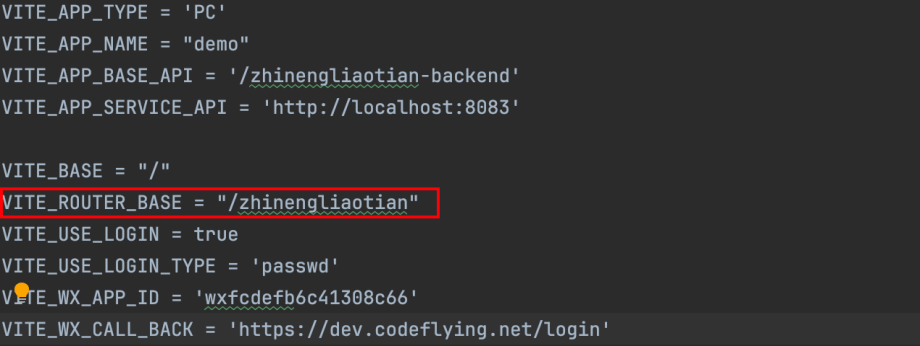
**前端**

1）创建.env.prodution文件

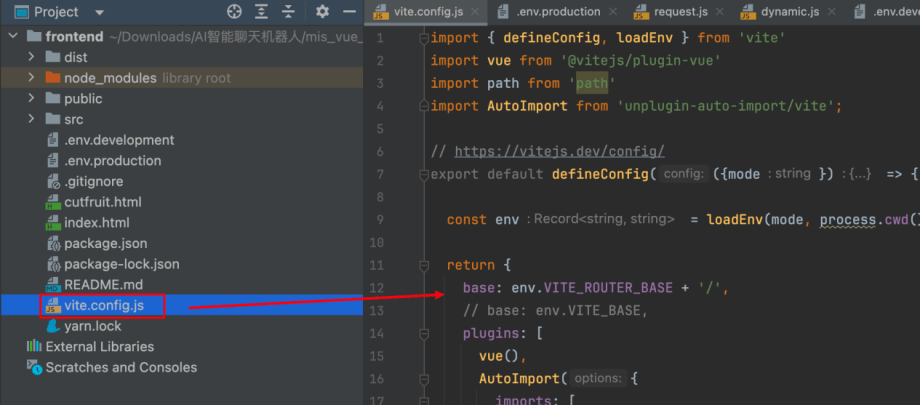


2）在文件中添加一行内容VITE\_ROUTER\_BASE = "/路径名称"比如/xiangmuguanli

并将 VITE\_APP\_BASE\_API中的内容改为/路径名称-backend比如/zhinengliaotian-backend

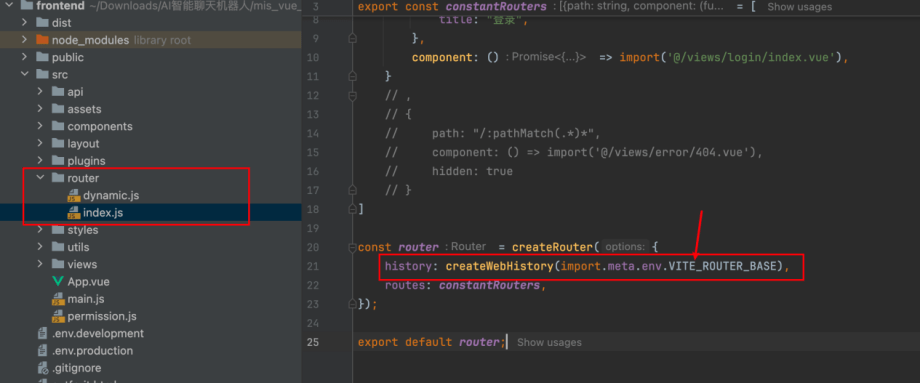


3）修改此行配置，改为刚才设置的 env.VITE\_ROUTER\_BASE + "/"，注意添加 "/"



4）找到项目中 router 目录下的 index.js 然后修改如下内容

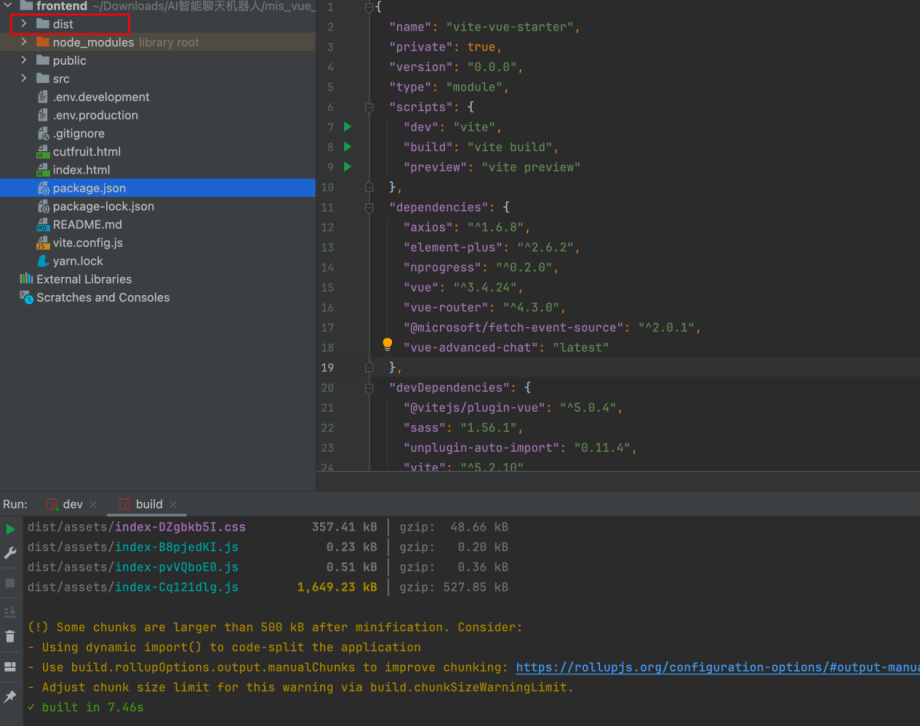
把之前的内容改为 VITE\_ROUTER\_BASE



5）找到 package.json 执行 build 命令，会将项目打包成 dist 目录



位置如下图所示



**部署**

至此，我们已经把前端、后端项目打包好了

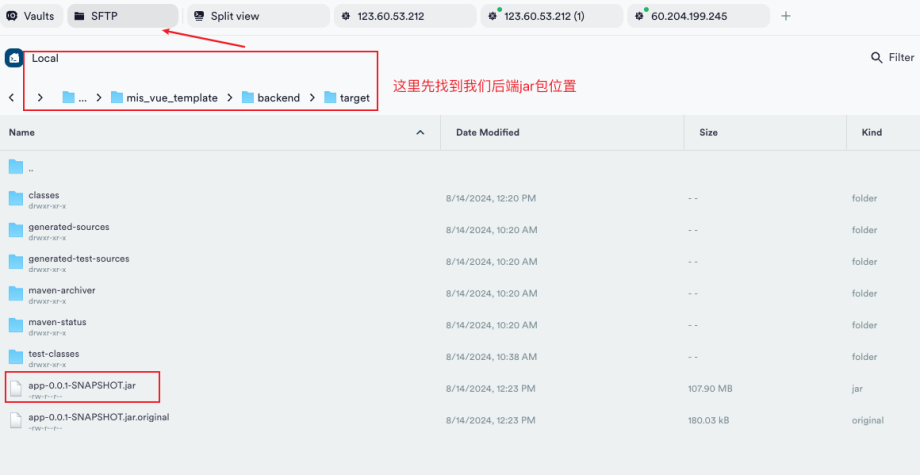
* 前端：dist 目录
* 后端：app-0.0.1-SNAPSHOT.jar（jar包）

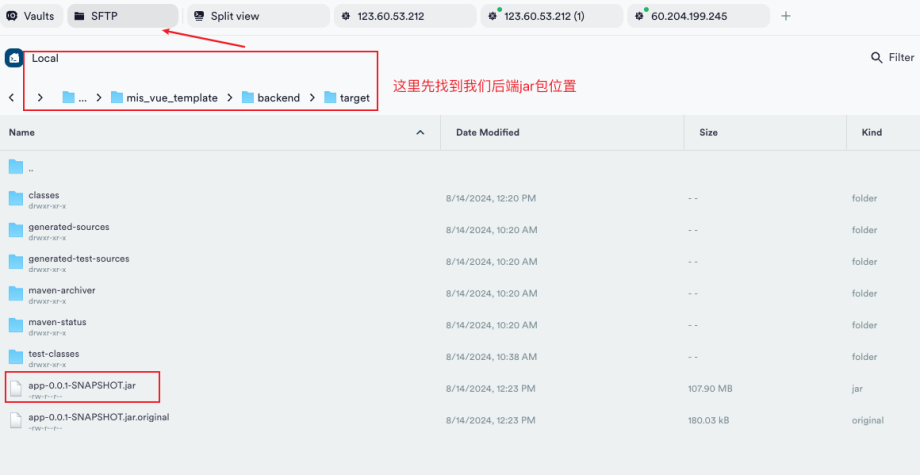
⚠️：以下操作均使用 Terminus 软件进行演示

1）首先，连接上我们需要部署的服务器，创建一个用于存放我们打包好文件的目录

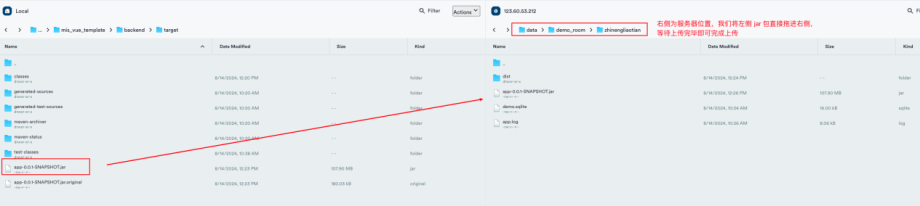
我放在 /data/demo\_room/zhinengliaotian目录下面

2）点击 SFTP，然后左侧为我们本机的路径





3）在右侧选择服务器路径，为这里放的位置是/data/demo\_room/zhinengliaotian



前端打包好的 dist 目录也是如此，左侧找到本地 dist 目录所在位置，拖动到右侧即可

4）找到 nginx 配置文件，然后进行编辑

Nginx 配置文件目录：/etc/nginx/nginx.conf

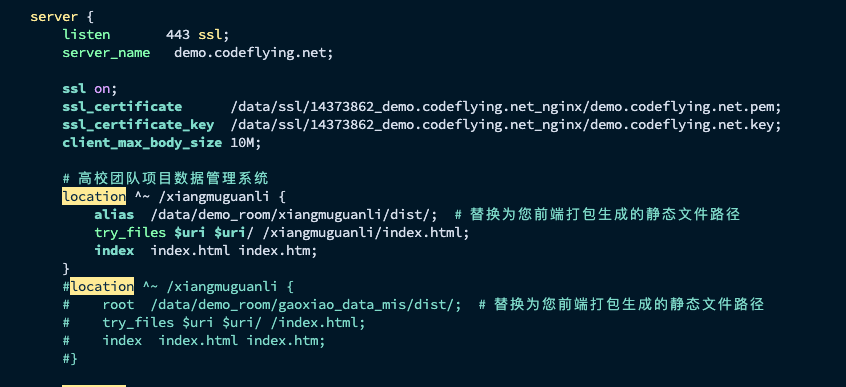
编辑命令

|  |
| --- |
| Bash 进入 /etc/nginx 目录 cd /etc/nginx 编辑内容 vim nginx.conf |

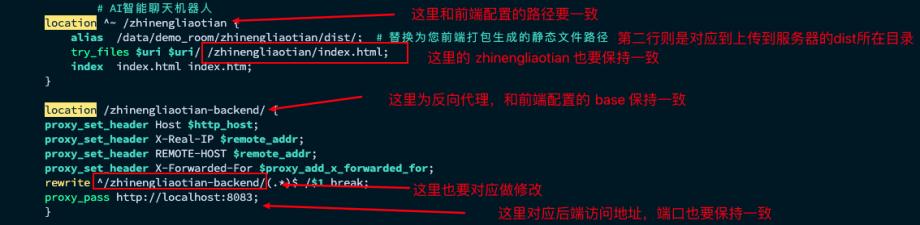
或者直接执行

|  |
| --- |
| Shell vim /etc/nginx/nginx.conf |

然后进到了 nginx.conf 文件内部，切换英文输入法输入按i即可在鼠标光标位置进行编辑



⚠️注意：要在上图 server 括号内部进行编辑，比如你可以放到client\_max\_body\_size这行的下面



|  |
| --- |
| Nginx AI智能聊天机器人 location ^~ /zhinengliaotian {  alias /data/demo\_room/zhinengliaotian/dist/; # 替换为您前端打包生成的静态文件路径  try\_files  $uri $ uri/ /zhinengliaotian/index.html;  index index.html index.htm; } location /zhinengliaotian-backend/ { proxy\_set\_header Host $http\_host; proxy\_set\_header X-Real-IP $remote\_addr; proxy\_set\_header REMOTE-HOST $remote\_addr; proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for; rewrite ^/zhinengliaotian-backend/(.\*) $ /$ 1 break; proxy\_pass http://localhost:8083; } |

编辑完成之后按ESC然后按:wq保存退出

5）退出之后重启 nginx 服务，输入如下命令

|  |
| --- |
| Nginx service nginx reload |

6）启动后端服务，设置 JVM 参数，执行如下命令

|  |
| --- |
| Java nohup java -jar -Xms256m -Xmx500m app-0.0.1-SNAPSHOT.jar > app.log 2>&1 & |

* 这里的app-0.0.1-SNAPSHOT.jar就是你要启动的 jar包名称
* 将程序执行日志写入app.log文件中
* -Xms256m：设置最小堆大小为 256MB
* -Xmx500m：设置最大堆大小为 500MB

执行完上述命令后，可以执行如下命令来实时查看日志输出

|  |
| --- |
| Shell tail -f app.log |

7）访问页面，看看是否部署成功，访问格式为：[https://demo.codeflying.net/](https://demo.codeflying.net/zhinengliaotian/)路径名称/

比如 <https://demo.codeflying.net/zhinengliaotian/>

**Windows 部署文档**

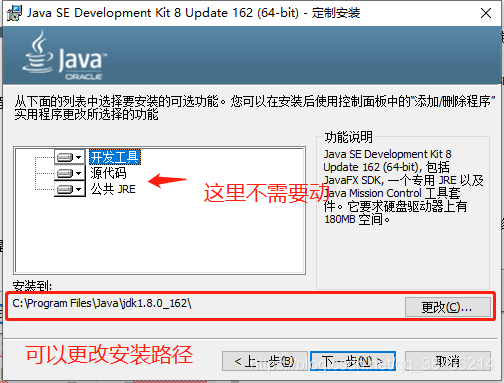
**JDK安装与配置**

**下载**

下载地址：https://www.oracle.com/java/technologies/downloads/#java8

**安装**

双击下载好的程序，直接运行。一直下一步

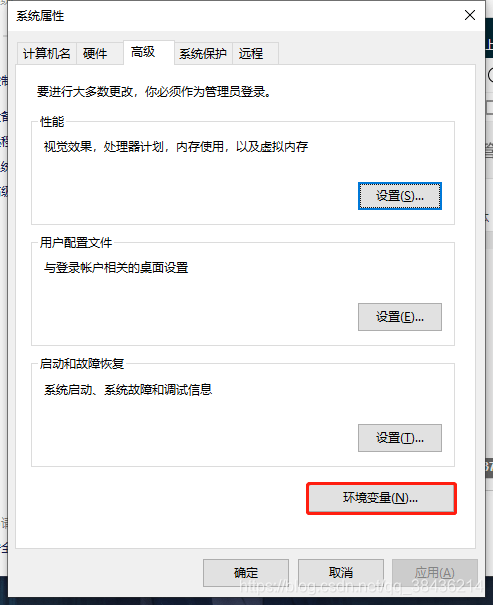


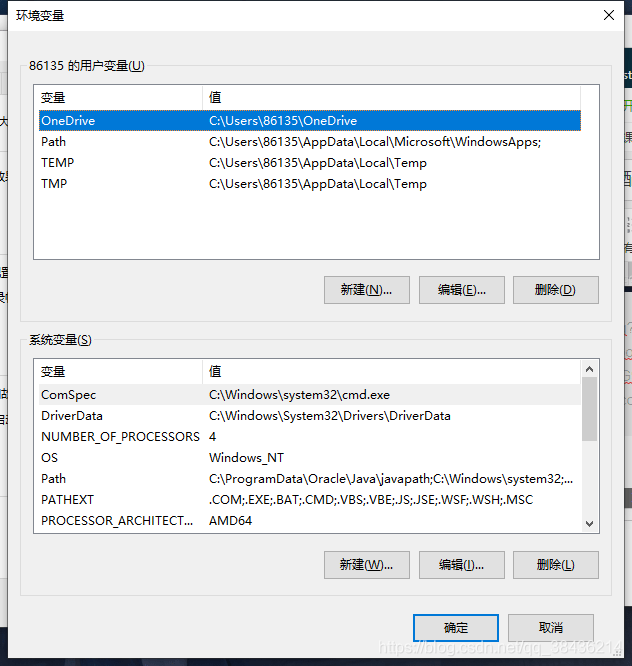
如图注意事项。

**系统环境变量配置**

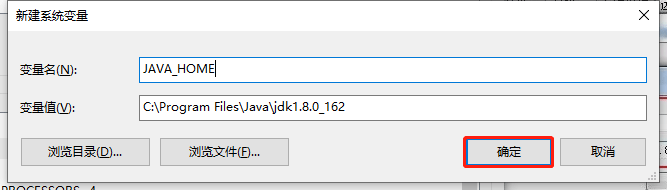
1. 可以在设置里面搜索，编辑系统环境变量。
2. 也可以在 “此电脑” -> 高级系统设置.

如图：

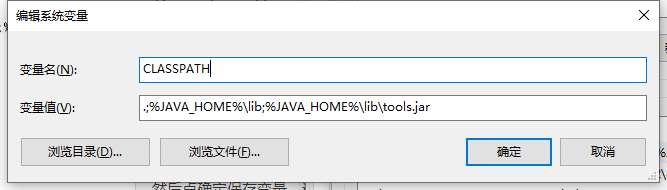




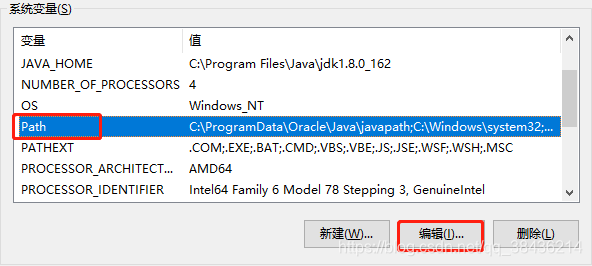
1. 点击新建

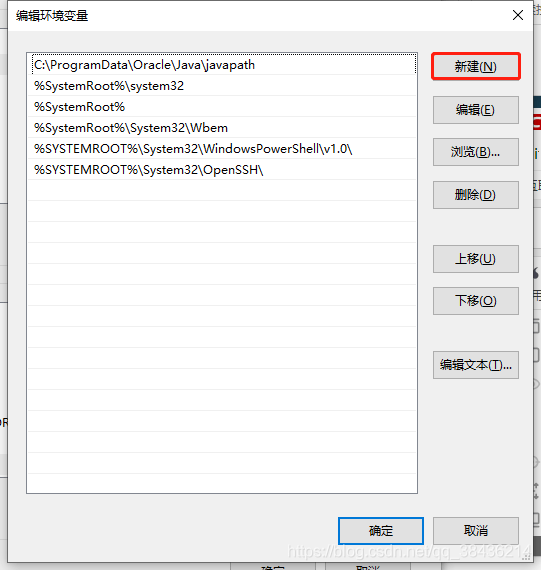


变量值，就是 安装好的路径。

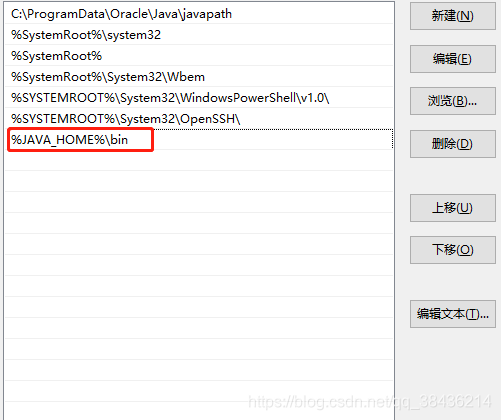


配置Path，找到path 双击。





点击新建。



**验证JDK**

快捷键 “win + R” 进入命令行。输入 java 回车。

**Maven安装与配置**

**下载**

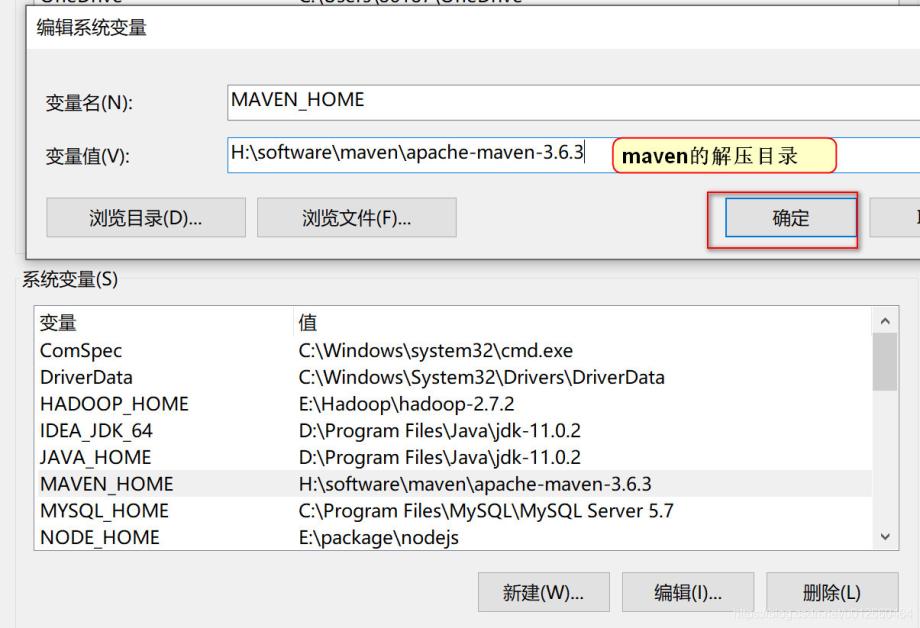
下载地址：https://maven.apache.org/download.cgi

**安装**

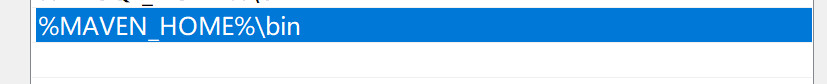
下载下来的事个zip包。直接解压即可。请记住解压目录

**系统环境变量配置**

与JAVA配置一样，进入到系统环境配置界面，



再将MAVEN\_HOME 添加到Path里。



**验证MAVEN**

进入命令行。输入mvn -v

**Node 安装与配置**

**下载**

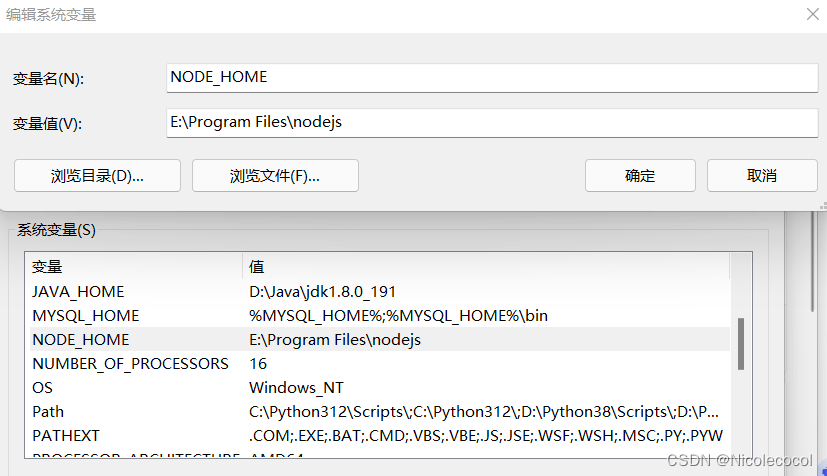
下载地址：https://nodejs.org/zh-cn

**安装**

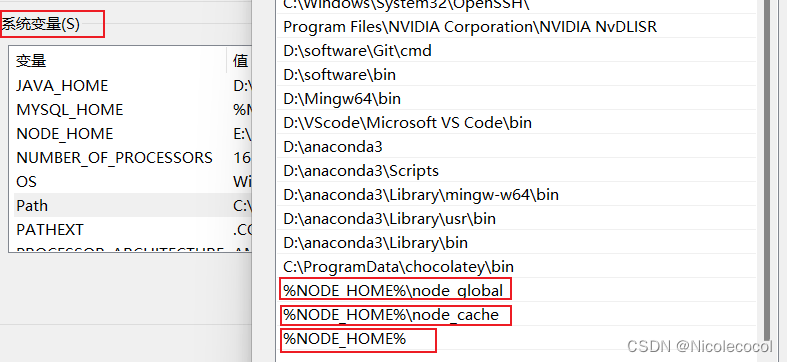
下载好的一个应用程序，双击运行。直接下一步下一步。选择安装目录。记住这个安装目录。

**系统环境变量配置**

和上面一样。进入到系统环境变量界面。



在Path里面新建



**验证Node**

进入命令行。输入 node -v ,npm -v

**本地运行**

进入命令行，进入到源代码目录。分别是backend和frontend目录。

backend目录下执行：mvn spring-boot:run

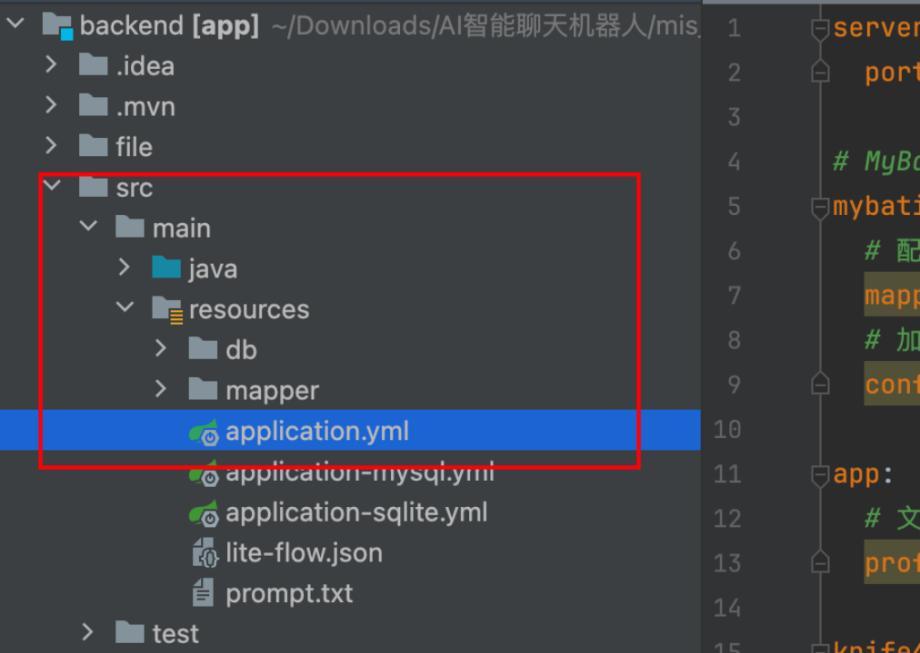
frontend目录下执行：npm install 成功之后在运行 npm run dev

**打包**

**后端**

1）找到后端配置文件 application.yml，修改 enable 配置，改为 true

|  |
| --- |
| ⚠️注意：需要登录验证的改为 true，不需要登录验证的不需要修改 |



|  |
| --- |
| YAML login:  enable: ${LOGIN\_ENABLE:true}  table: UserInfo  select:  table: UserInfo  column: phone\_number  openid:  table: UserInfo  column: wechat\_login\_id |

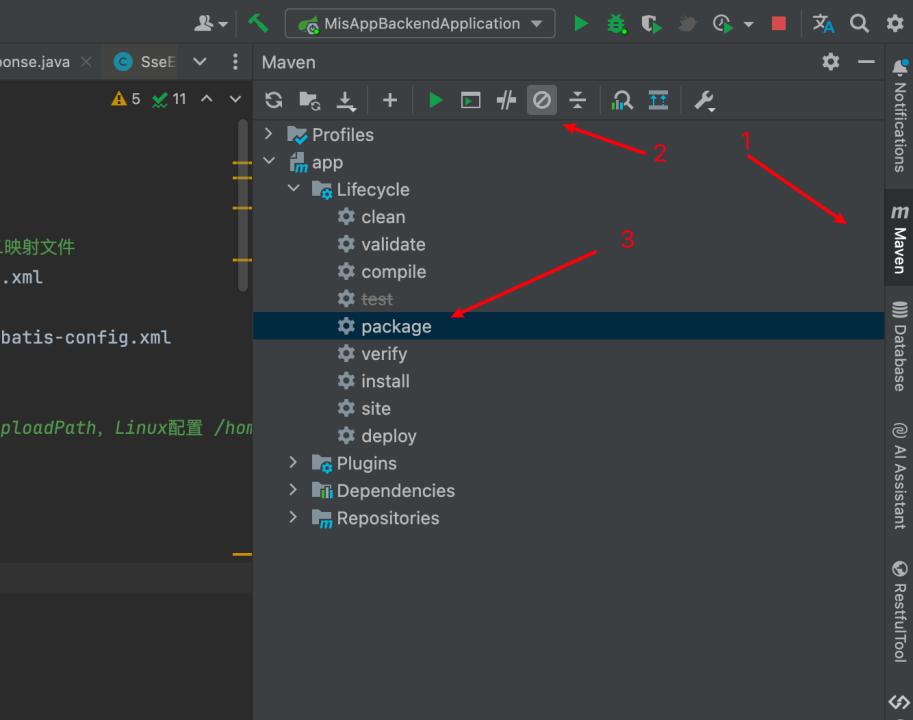
如果端口号有冲突需要修改端口号

|  |
| --- |
| YAML server:  port: ${PORT:8083} |

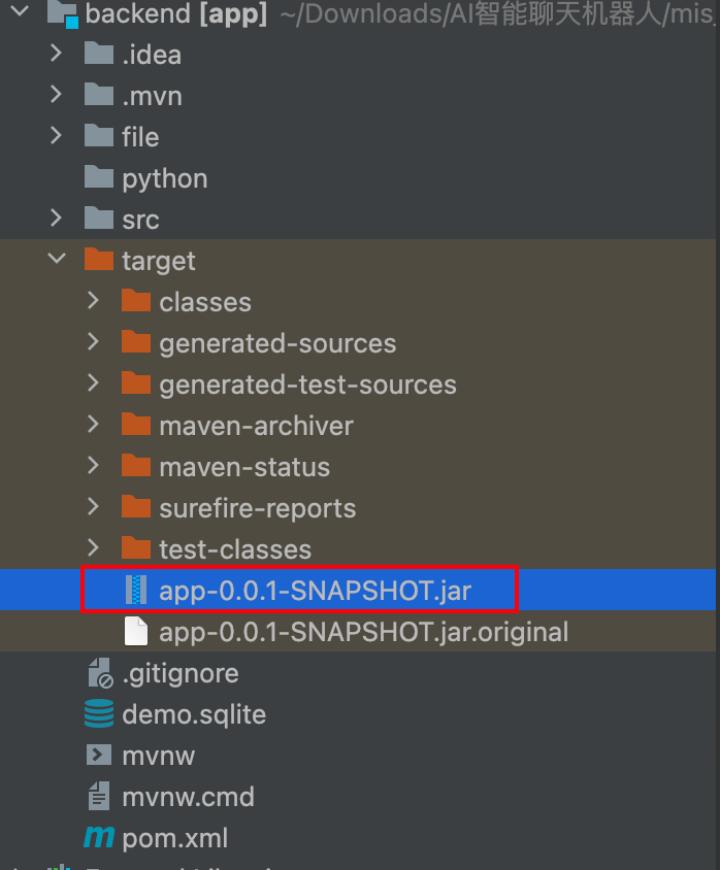
2）在项目目录打开终端进行打包或直接使用 IDEA 进行打包

|  |
| --- |
| Shell mvn install -Dmaven.test.skip=true |

或者

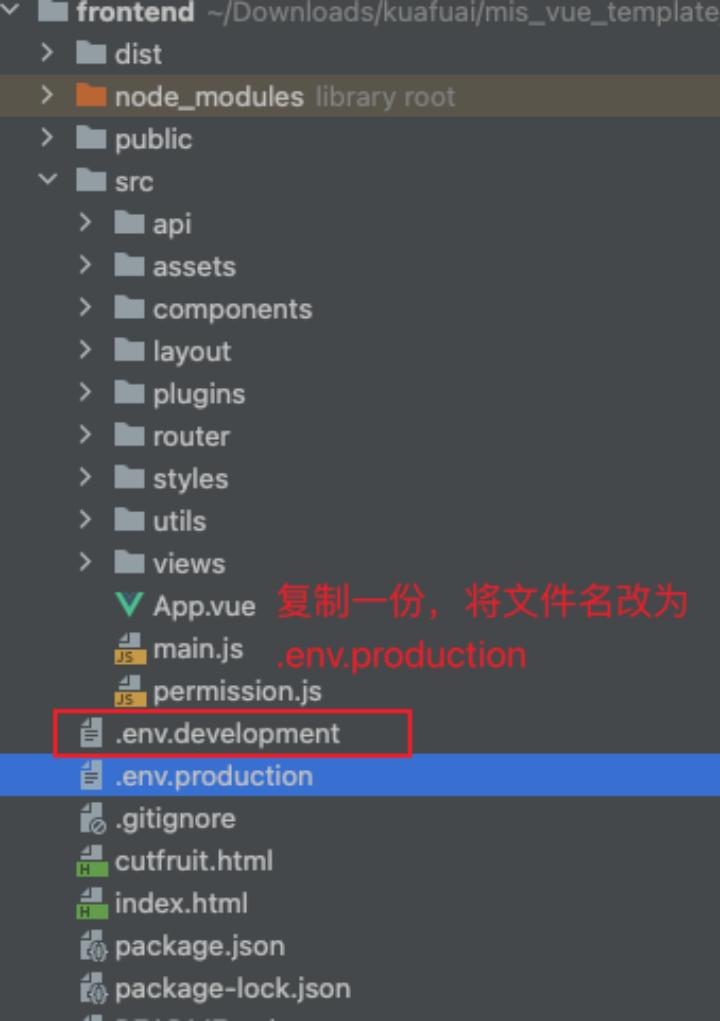


打包成功后，项目包放在 target 目录下面



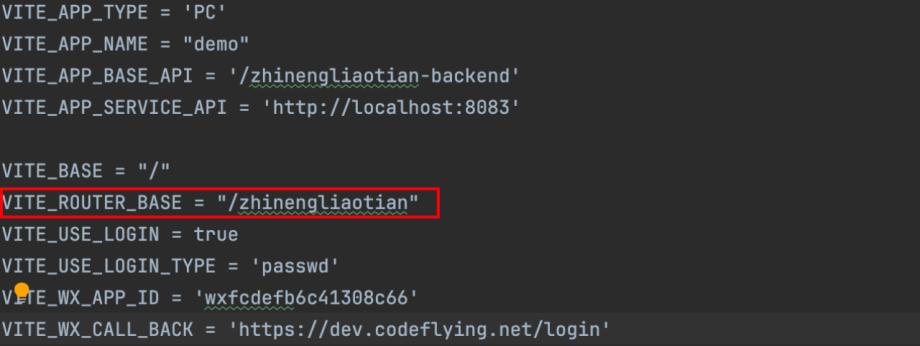
**前端**

1）创建.env.prodution文件

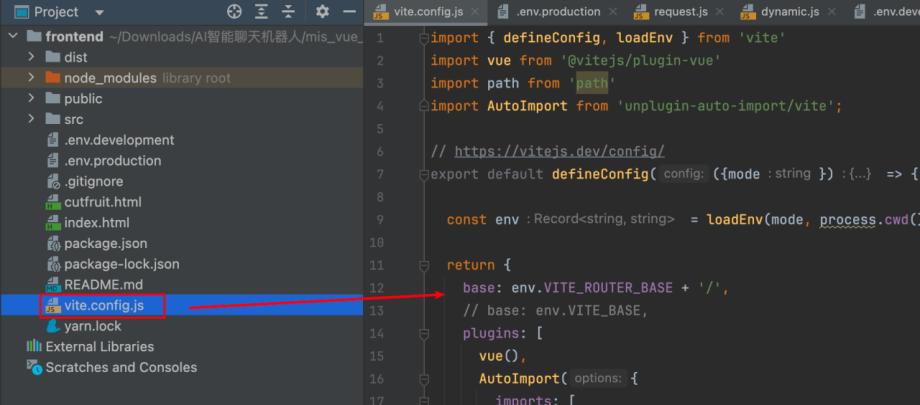


2）在文件中添加一行内容VITE\_ROUTER\_BASE = "/路径名称"比如/xiangmuguanli

并将 VITE\_APP\_BASE\_API中的内容改为/路径名称-backend比如/zhinengliaotian-backend

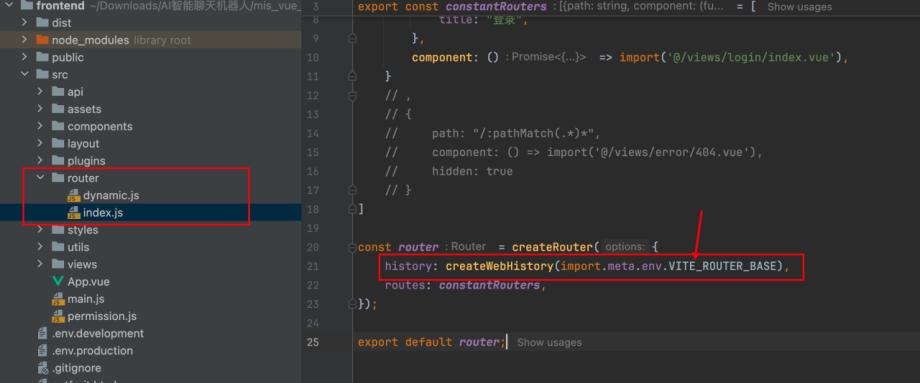


3）修改此行配置，改为刚才设置的 env.VITE\_ROUTER\_BASE + "/"，注意添加 "/"

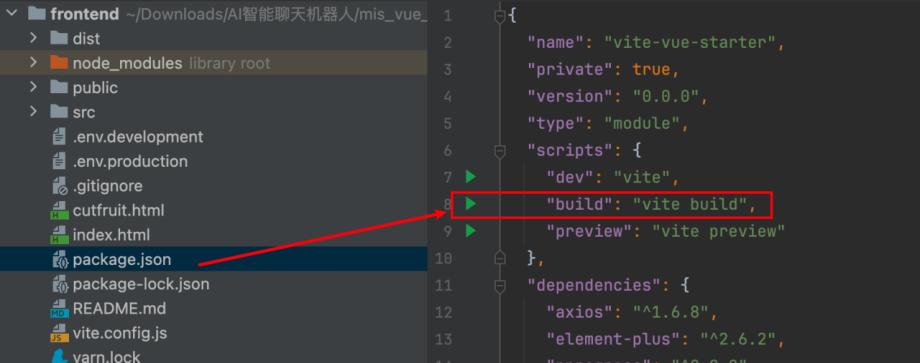


4）找到项目中 router 目录下的 index.js 然后修改如下内容

把之前的内容改为 VITE\_ROUTER\_BASE



5）找到 package.json 执行 build 命令，会将项目打包成 dist 目录



位置如下图所示

