

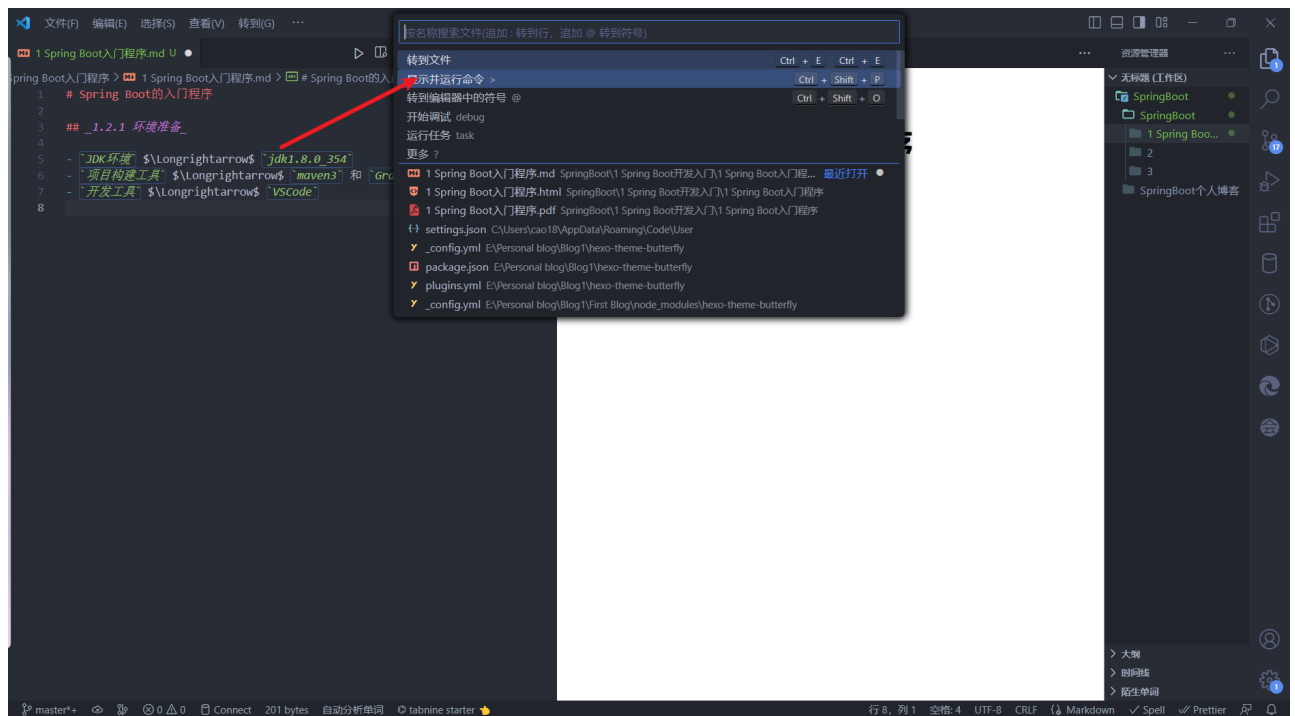
# Spring Boot的入门程序

## 1.2.1 环境准备

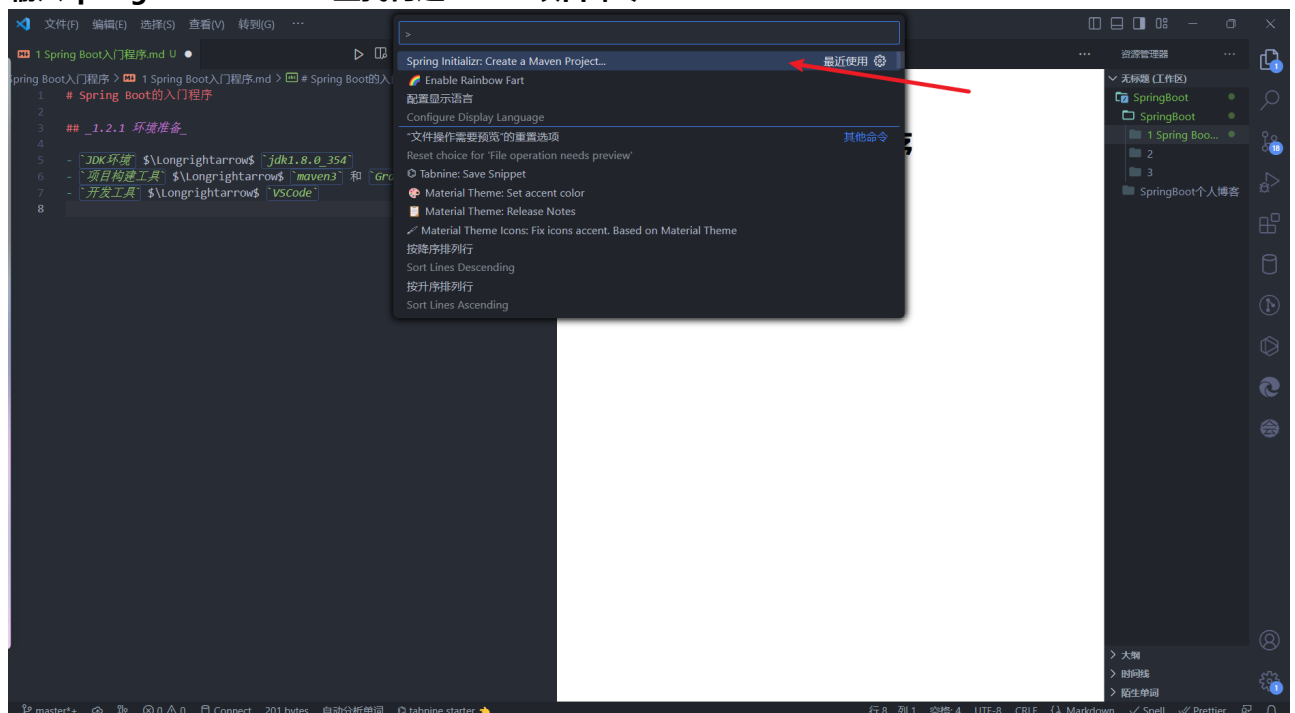
- **JDK环境**  $\rightarrow$  `jdk1.8.0_354`
- **项目构建工具**  $\rightarrow$  `maven3` 和 `Gradle`
- **开发工具**  $\rightarrow$  `VSCode`

## 1.2.2 Spring Boot的项目构建

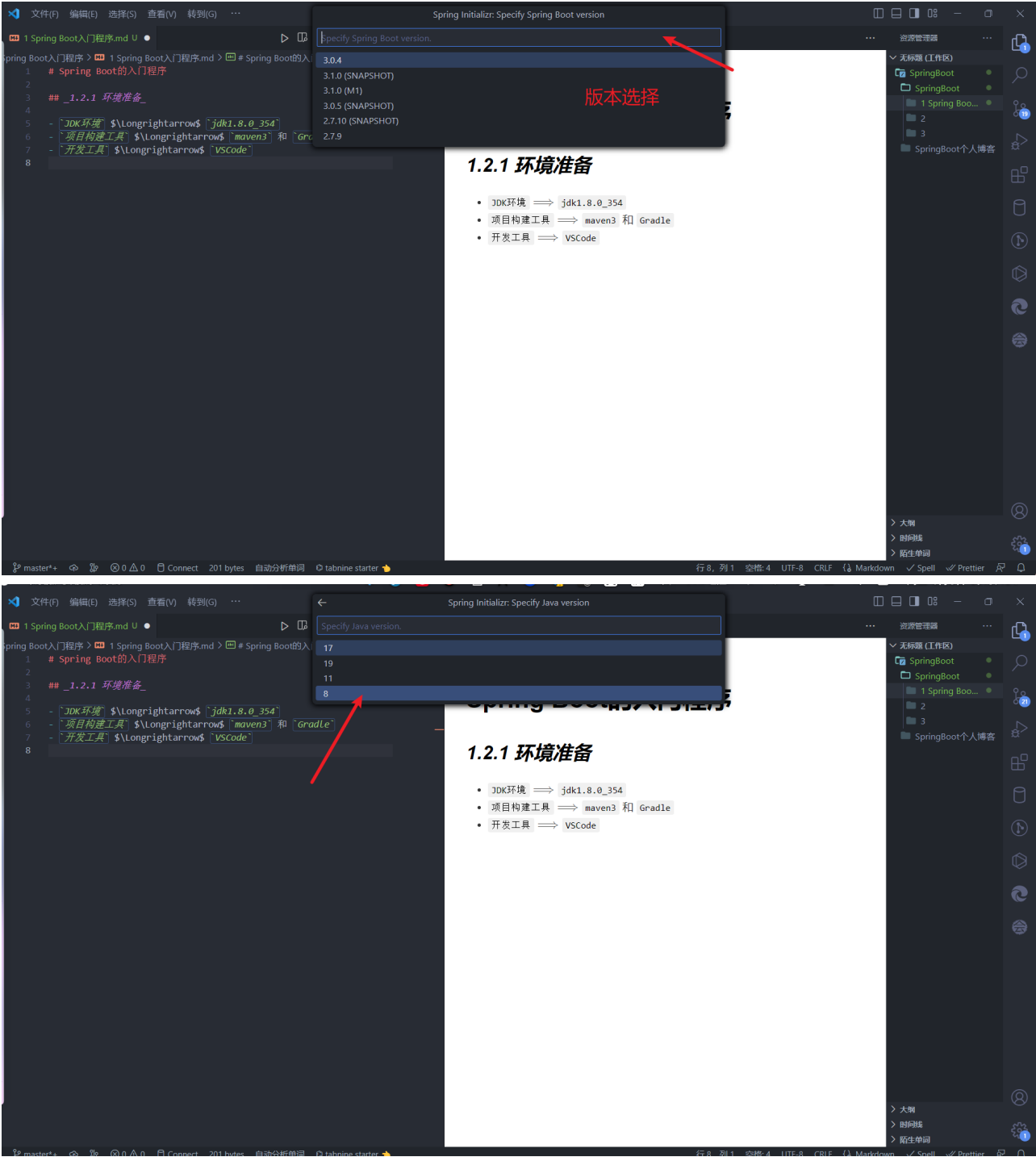
### 1. 选择命令框

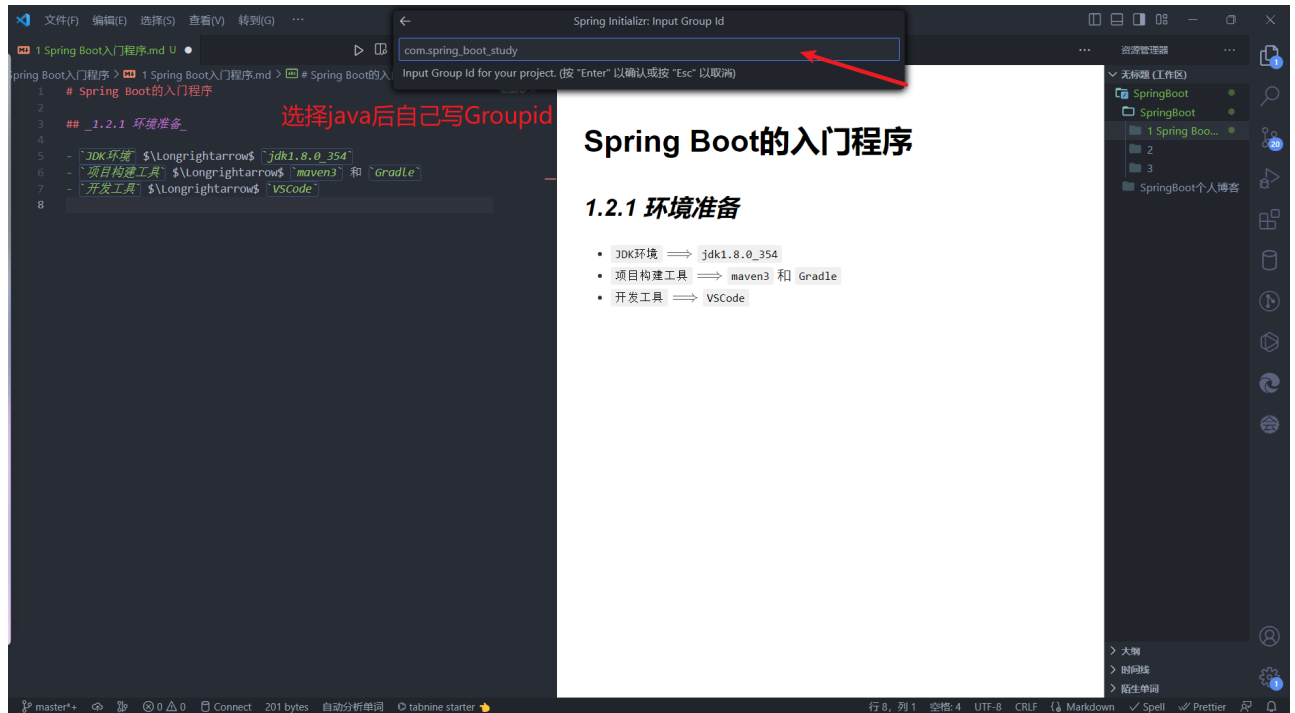


### 2. 输入spring boot initializr查找构建maven项目命令



3. 选择maven版本和jdk版本，自己写组织名

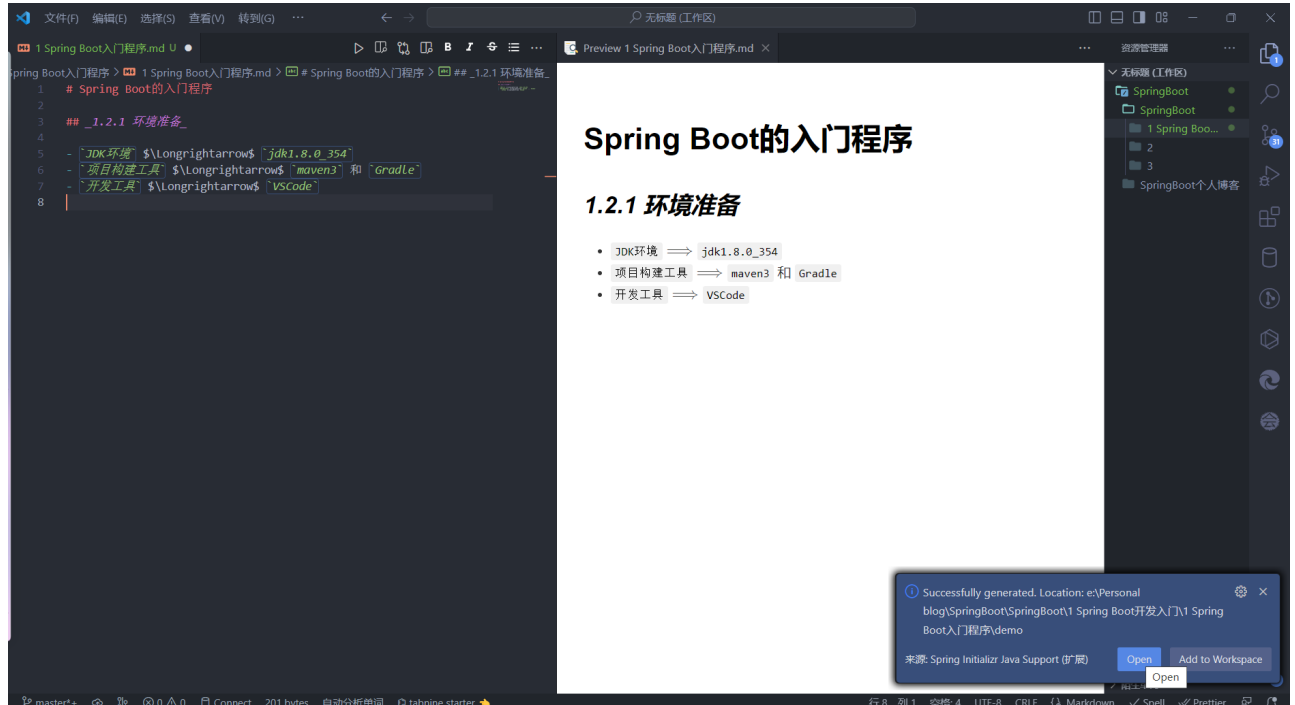


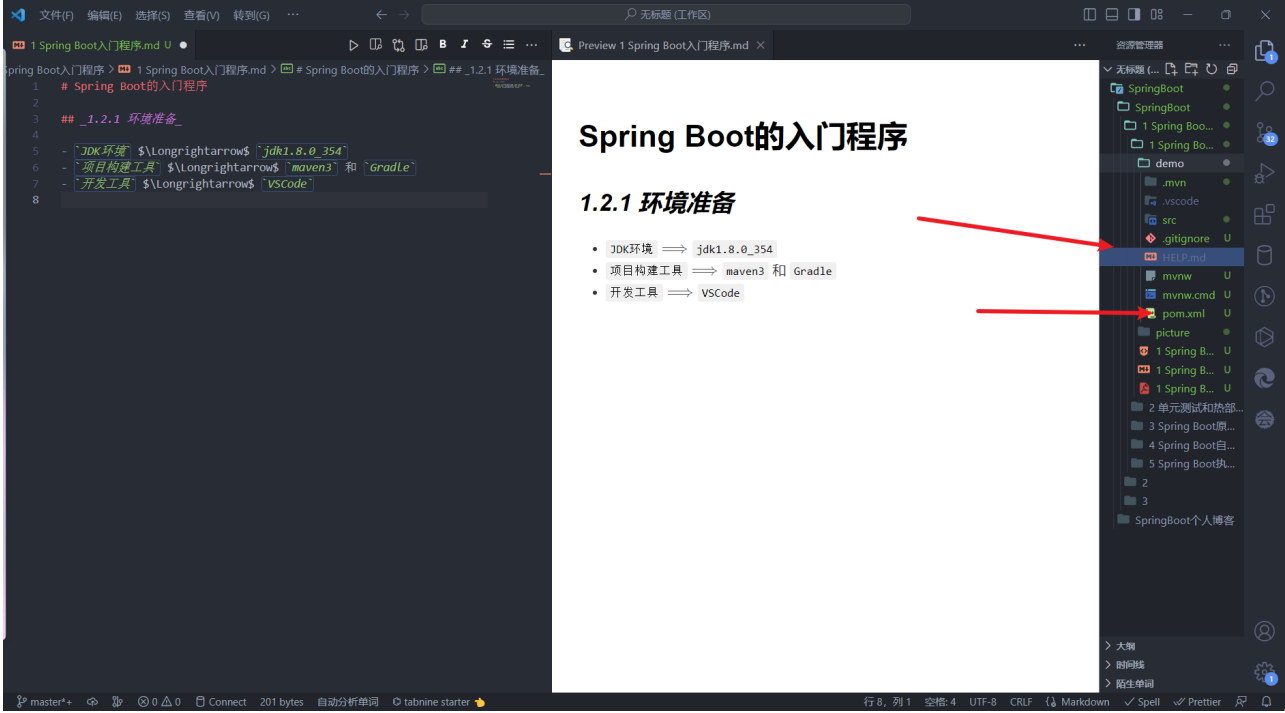


**什么是groupid和artifactId?** groupid和artifactId被统称为“坐标” 是为了保证项目唯一性而提出的，如果你要把你项目弄到maven本地仓库去，你想要找到你的项目就必须根据这两个id去查找。

groupid一般分为多个段，第一段为域，第二段为公司名称。域又分为org、com、cn等等许多，其中org为非营利组织，com为商业组织。举个apache公司的tomcat项目例子：这个项目的groupid是org.apache，它的域是org（因为tomcat是非营利项目），公司名称是apache，artifactId是tomcat。

#### 4. 项目创建结果





5. pom.xml依赖选择

```
1 Spring Boot入门程序.md U pom.xml U
SpringBoot > SpringBoot > 1 Spring Boot开发入门 > 1 Spring Boot入门程序 > demo > pom.xml
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/200
3   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/
4   <modelVersion>4.0.0</modelVersion>
5   <!-- 引入Spring Boot依赖 -->
6   <parent>
7     <groupId>org.springframework.boot</groupId>
8     <artifactId>spring-boot-starter-parent</artifactId>
9     <version>3.0.4</version>
10    <relativePath/> <!-- lookup parent from repository -->
11  </parent>
12  <groupId>com.spring_boot_study</groupId>
13  <artifactId>demo</artifactId>
14  <version>0.0.1-SNAPSHOT</version>
15  <name>demo</name>
16  <description>Demo project for Spring Boot</description>
17  <properties>
18    <java.version>17</java.version>
19  </properties>
20  <dependencies>
21    <dependency>
22      <groupId>org.springframework.boot</groupId>
23      <artifactId>spring-boot-starter-data-rest</artifactId>
24    </dependency>
25    <dependency>
26      <groupId>org.springframework.boot</groupId>
27      <artifactId>spring-boot-starter-graphql</artifactId>
28    </dependency>
29    <dependency>
30      <groupId>org.springframework.boot</groupId>
31      <artifactId>spring-boot-starter-hateoas</artifactId>
32    </dependency>
33    <dependency>
34      <groupId>org.springframework.boot</groupId>
35      <artifactId>spring-boot-starter-security</artifactId>
36    </dependency>
37    <dependency>
```

```
SpringBoot > SpringBoot > 1 Spring Boot开发入门 > 1 Spring Boot入门程序 > demo > pom.xml
24 </dependency>
25 <dependency>
26   <groupId>org.springframework.boot</groupId>
27   <artifactId>spring-boot-starter-graphql</artifactId>
28 </dependency>
29 <dependency>
30   <groupId>org.springframework.boot</groupId>
31   <artifactId>spring-boot-starter-hateoas</artifactId>
32 </dependency>
33 <dependency>
34   <groupId>org.springframework.boot</groupId>
35   <artifactId>spring-boot-starter-security</artifactId>
36 </dependency>
37 <dependency>
38   <!-- 引入Web场景依赖启动器 -->
39   <groupId>org.springframework.boot</groupId>
40   <artifactId>spring-boot-starter-web</artifactId>
41 </dependency>
```

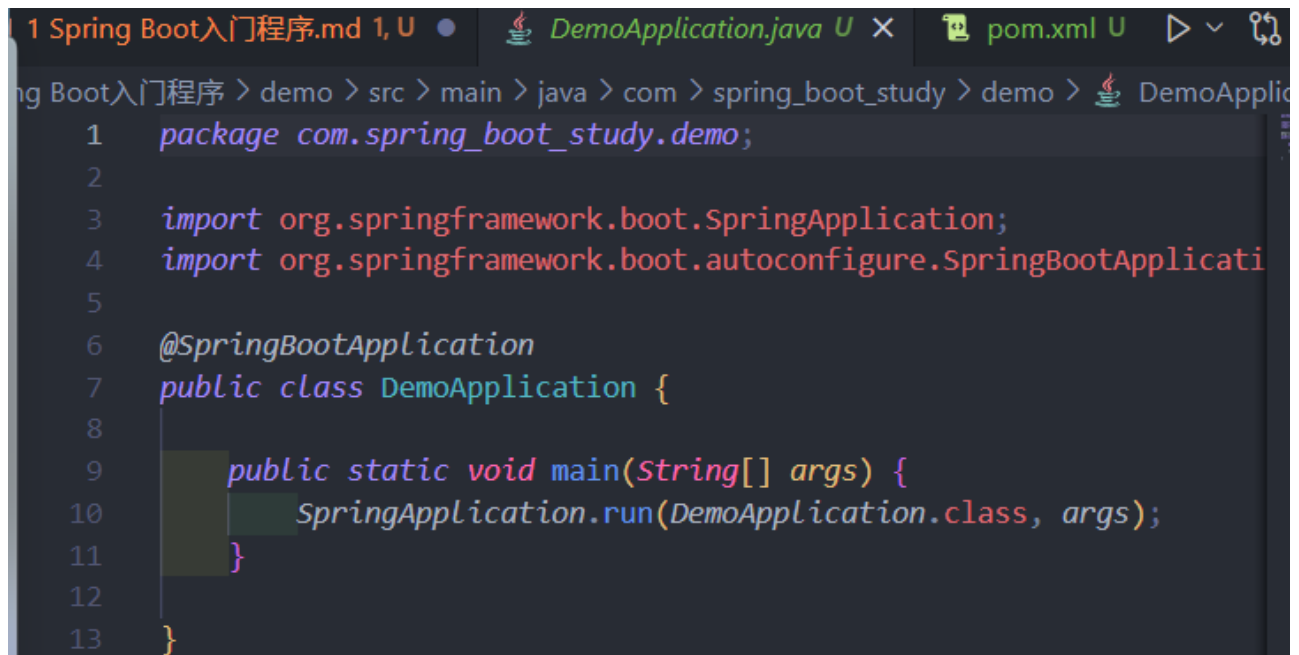
```
42     <dependency>
43         <groupId>org.springframework.boot</groupId>
44         <artifactId>spring-boot-starter-web-services</artifactId>
45     </dependency>
46     <dependency>
47         <groupId>org.springframework.boot</groupId>
48         <artifactId>spring-boot-starter-webflux</artifactId>
49     </dependency>
50     <dependency>
51         <groupId>org.springframework.data</groupId>
52         <artifactId>spring-data-rest-hal-explorer</artifactId>
53     </dependency>
54     <dependency>
55         <groupId>org.springframework.session</groupId>
56         <artifactId>spring-session-core</artifactId>
57     </dependency>
58
59     <dependency>
60         <groupId>org.springframework.boot</groupId>
```

SpringBoot > SpringBoot > 1 Spring Boot开发入门 > 1 Spring Boot入门程序 > demo > pom.xml

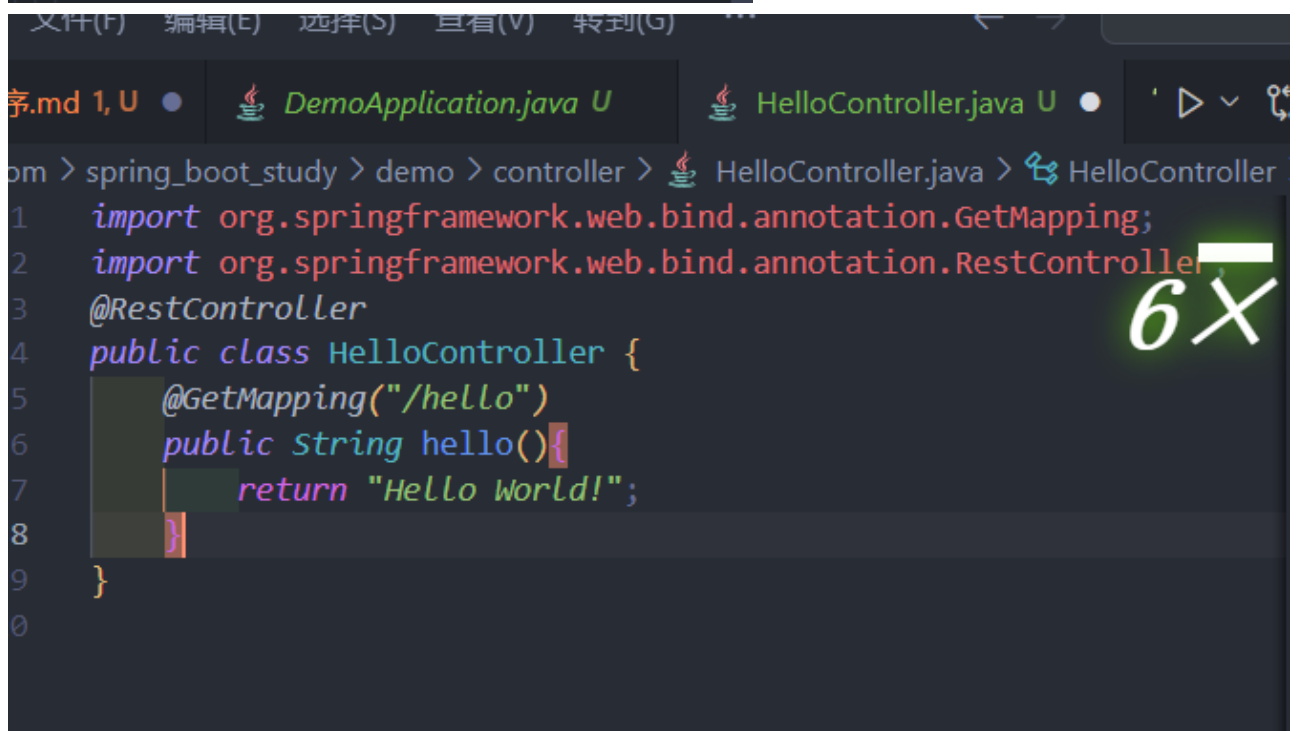
```
24     </dependency>
25     <dependency>
26         <groupId>org.springframework.boot</groupId>
27         <artifactId>spring-boot-starter-graphql</artifactId>
28     </dependency>
29     <dependency>
30         <groupId>org.springframework.boot</groupId>
31         <artifactId>spring-boot-starter-hateoas</artifactId>
32     </dependency>
33     <dependency>
34         <groupId>org.springframework.boot</groupId>
35         <artifactId>spring-boot-starter-security</artifactId>
36     </dependency>
37     <dependency>
38         <!-- 引入Web场景依赖启动器 -->
39         <groupId>org.springframework.boot</groupId>
40         <artifactId>spring-boot-starter-web</artifactId>
41     </dependency>
42     <dependency>
43         <groupId>org.springframework.boot</groupId>
44         <artifactId>spring-boot-starter-web-services</artifactId>
45     </dependency>
46     <dependency>
47         <groupId>org.springframework.boot</groupId>
48         <artifactId>spring-boot-starter-webflux</artifactId>
49     </dependency>
50     <dependency>
51         <groupId>org.springframework.data</groupId>
52         <artifactId>spring-data-rest-hal-explorer</artifactId>
53     </dependency>
54     <dependency>
55         <groupId>org.springframework.session</groupId>
56         <artifactId>spring-session-core</artifactId>
57     </dependency>
58
59     <dependency>
60         <groupId>org.springframework.boot</groupId>
```

<parent> 标签中添加的spring-boot-starter-parent依赖是Spring Boot框架集成项目的统一父类管理依赖，只有添加它才可以使使用Spring Boot的相关特性。其他依赖的解释建议自己去查

## 6. 编写主程序以及控制类



```
1 package com.spring_boot_study.demo;
2
3 import org.springframework.boot.SpringApplication;
4 import org.springframework.boot.autoconfigure.SpringBootApplication;
5
6 @SpringBootApplication
7 public class DemoApplication {
8
9     public static void main(String[] args) {
10         SpringApplication.run(DemoApplication.class, args);
11     }
12
13 }
```

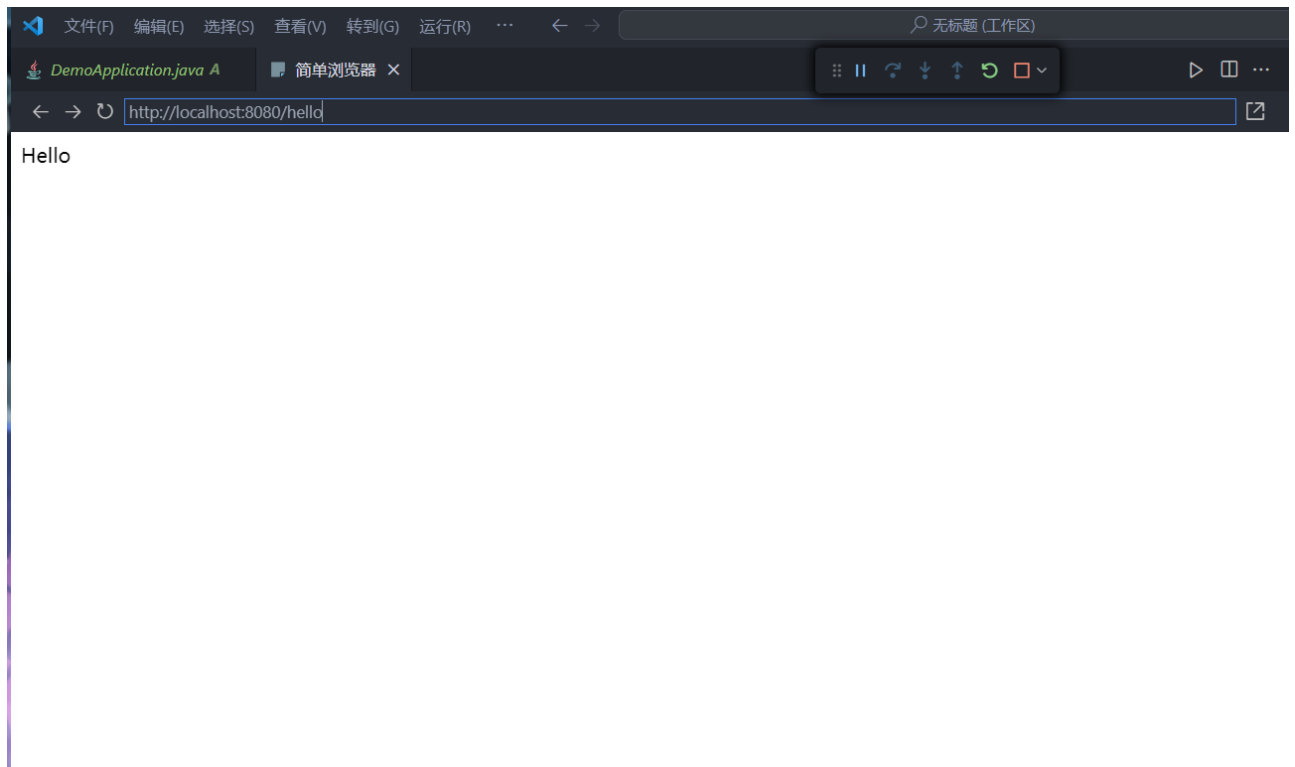


```
1 import org.springframework.web.bind.annotation.GetMapping;
2 import org.springframework.web.bind.annotation.RestController;
3 @RestController
4 public class HelloController {
5     @GetMapping("/hello")
6     public String hello(){
7         return "Hello World!";
8     }
9 }
10
```

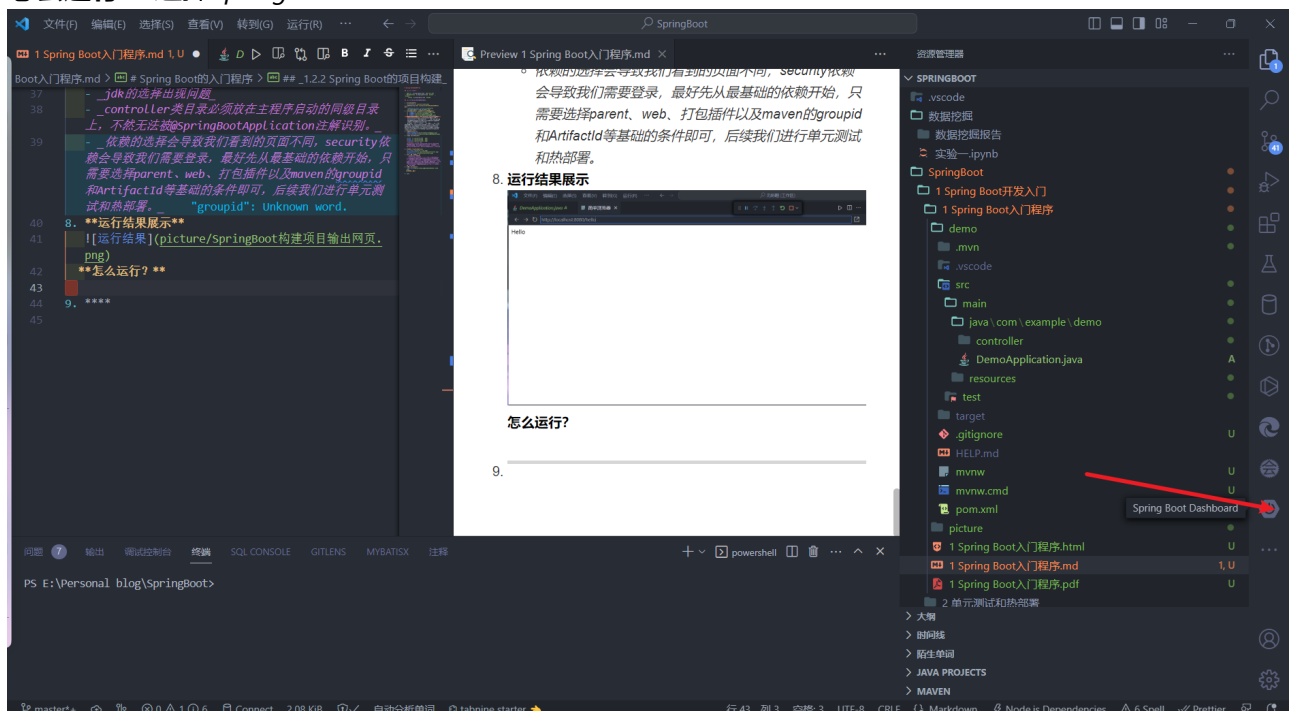
## 7. 关于运行时的建议

- jdk的选择出现问题
- controller类目录必须放在主程序启动的同级目录上，不然无法被@SpringBootApplication注解识别。
- 依赖的选择会导致我们看到的页面不同，security依赖会导致我们需要登录，最好先从最基础的依赖开始，只需要选择parent、web、打包插件以及maven的groupid和ArtifactId等基础的条件即可，后续我们进行单元测试和热部署。

## 8. 运行结果展示



## 怎么运行？ 选择Spring Boot Dashboard





选择运行按钮

