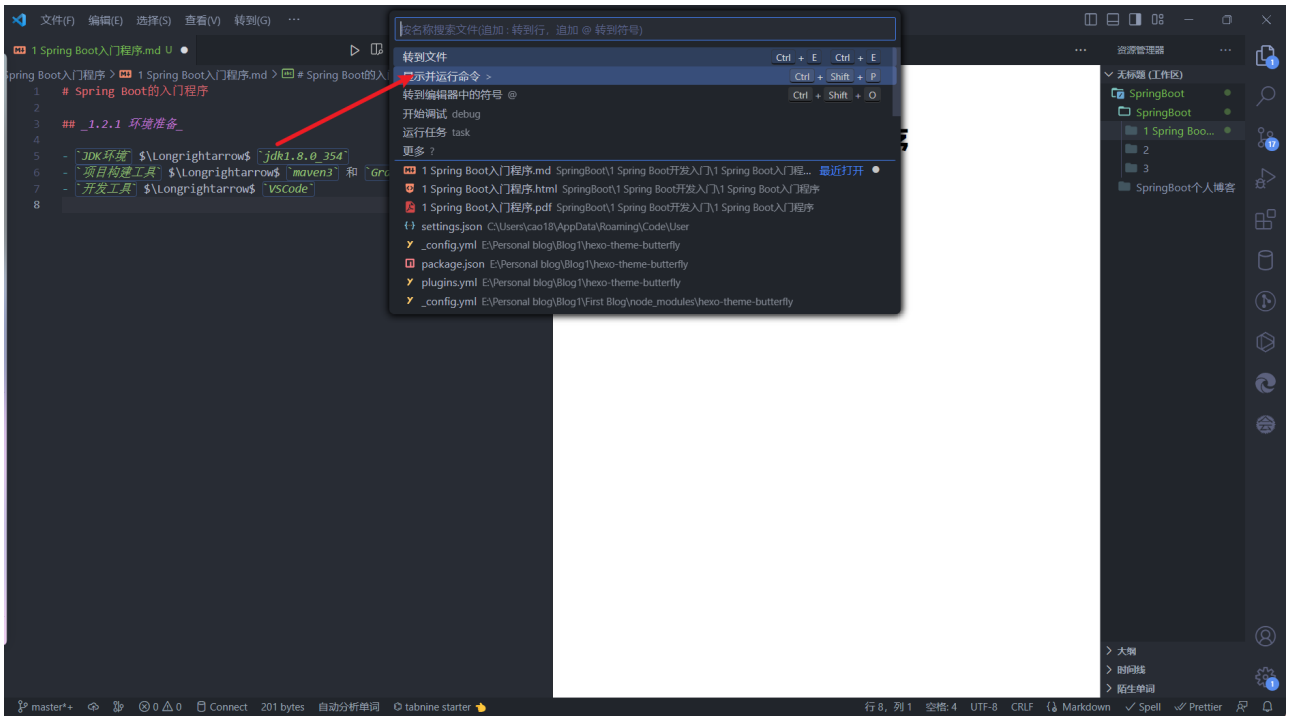
Spring Boot的入门程序

1.2.1 环境准备

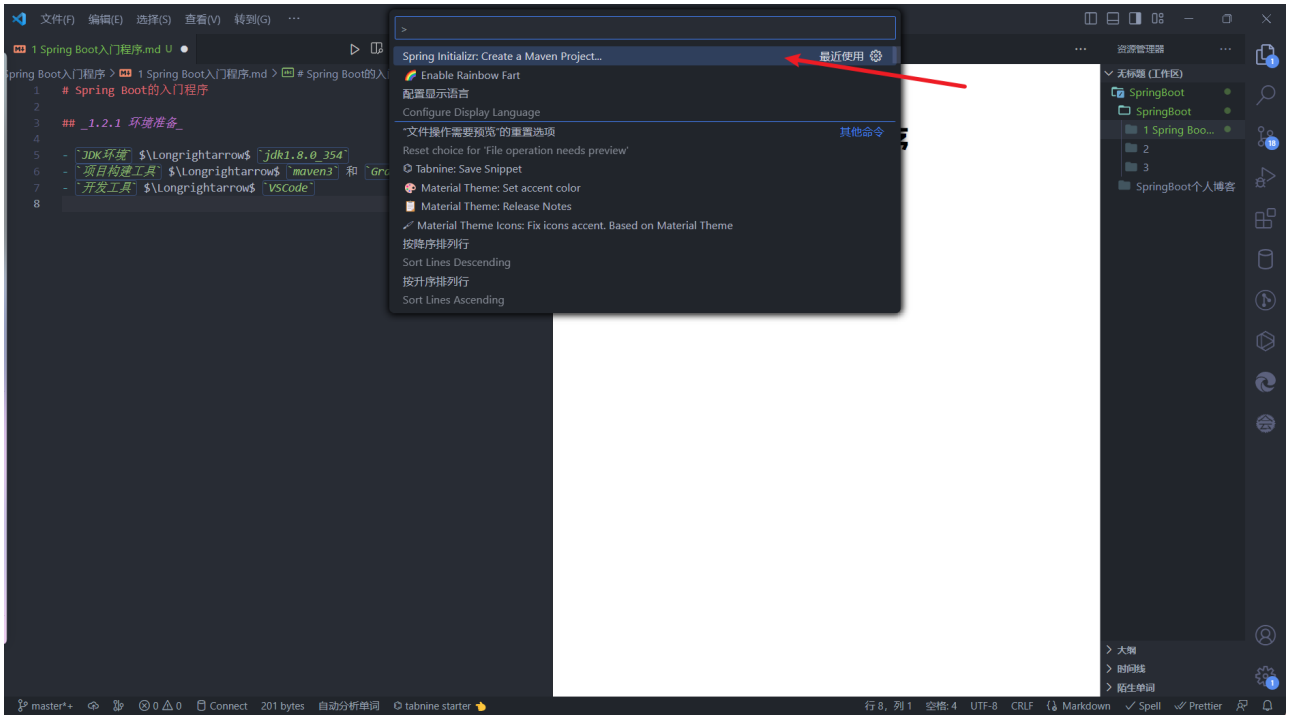
- JDK环境 $\Longrightarrow$ jdk1.8.0_354
- 项目构建工具 $\Longrightarrow$ maven3 和 Gradle
- 开发工具 $\Longrightarrow$ VSCode

1.2.2 Spring Boot的项目构建

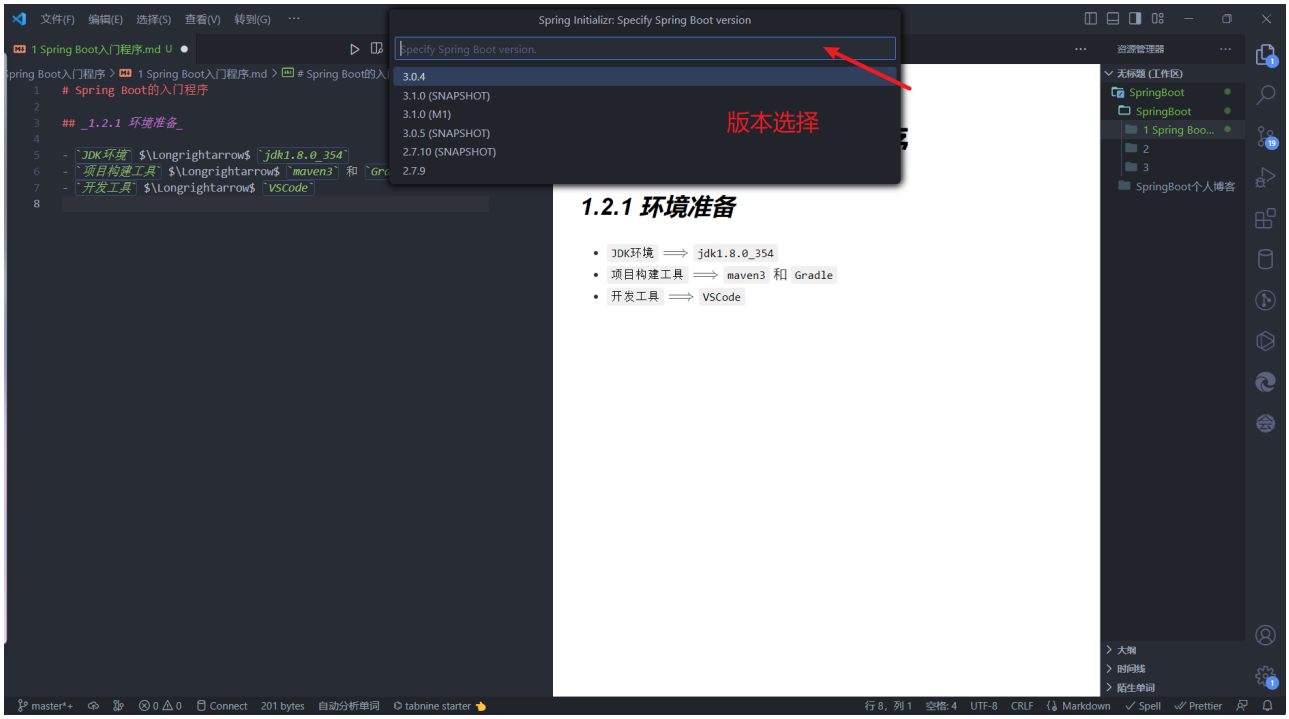
1. **选择命令框**

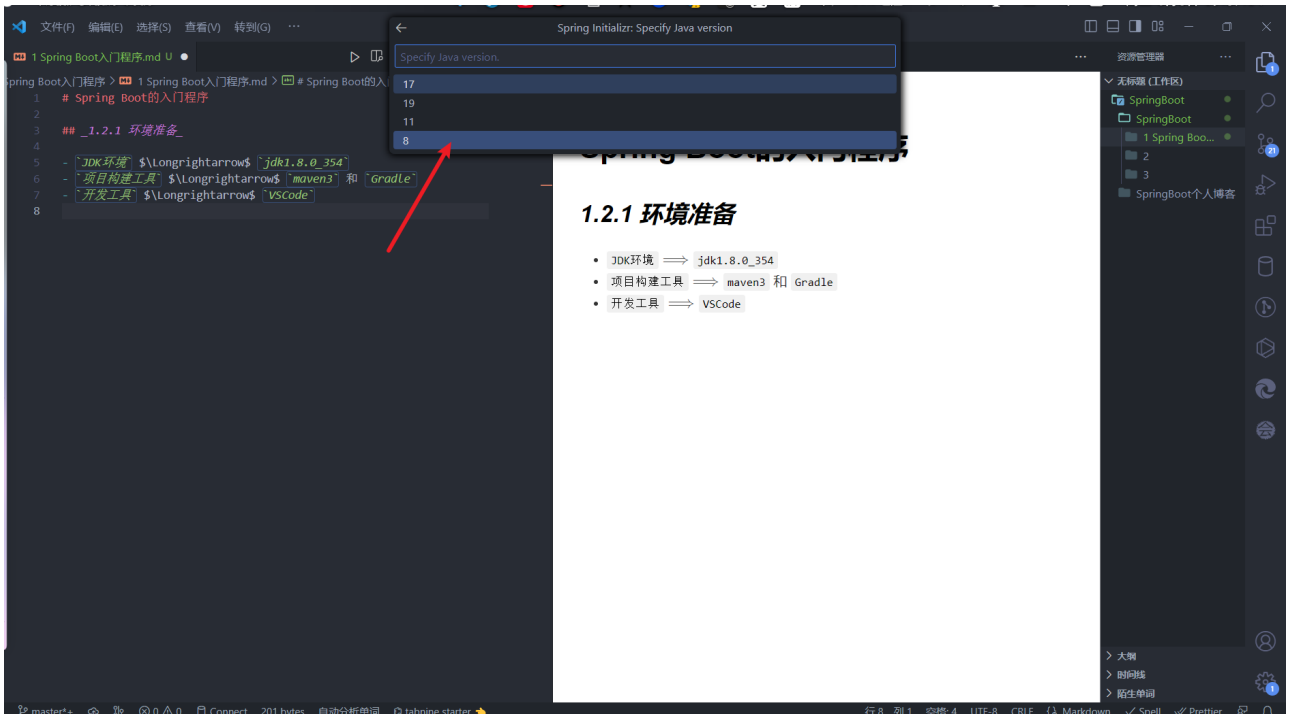


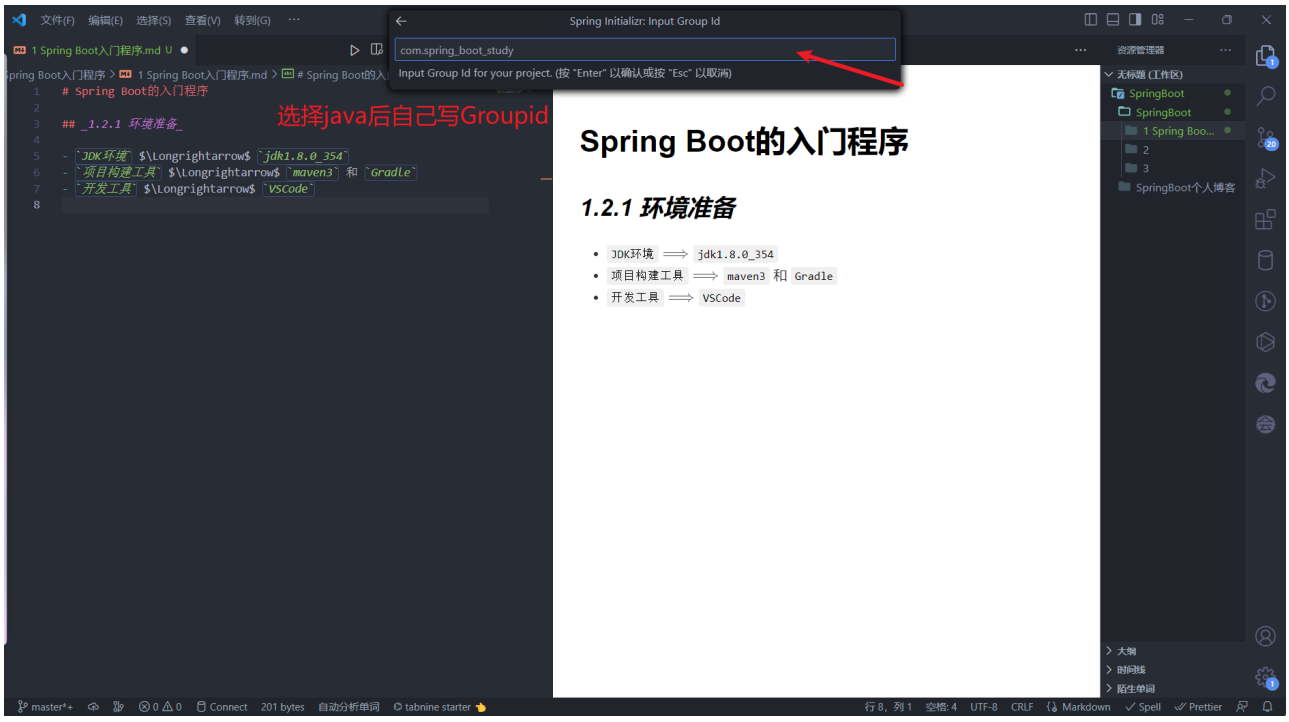
2. **输入spring boot initializr查找构建maven项目命令**



3. **选择maven版本和jdk版本，自己写组织名**



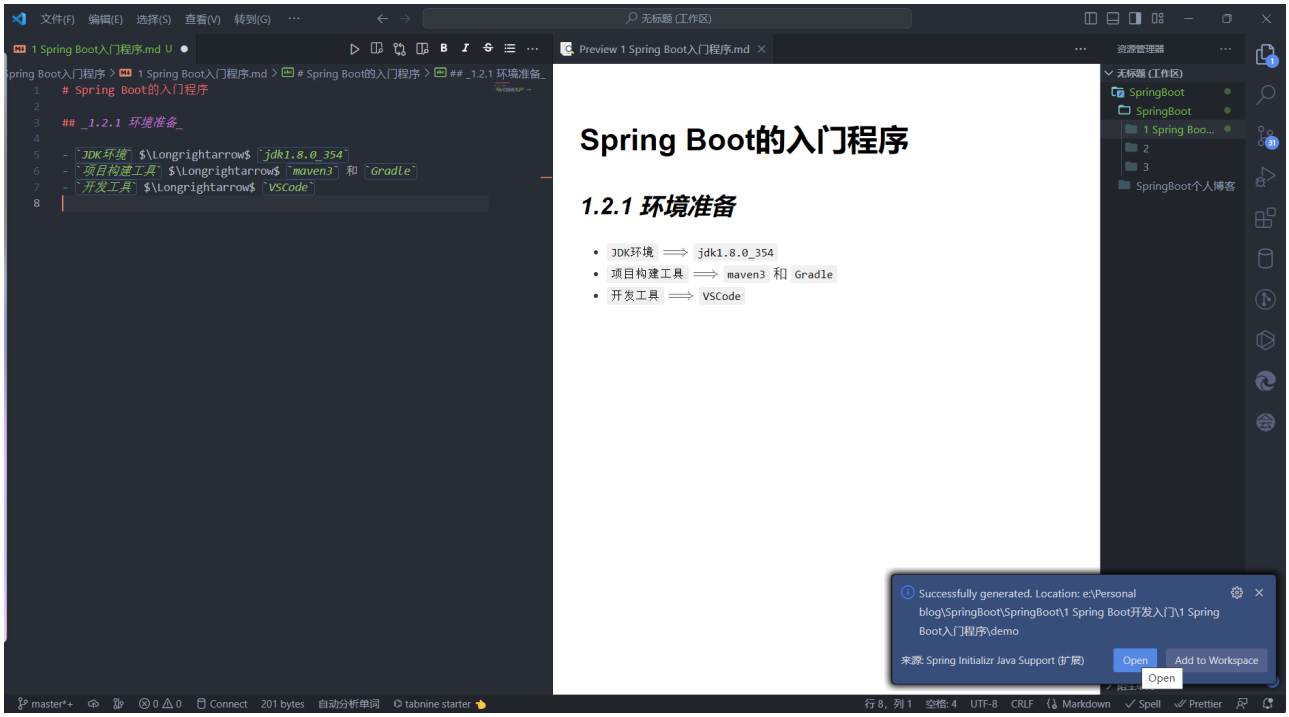


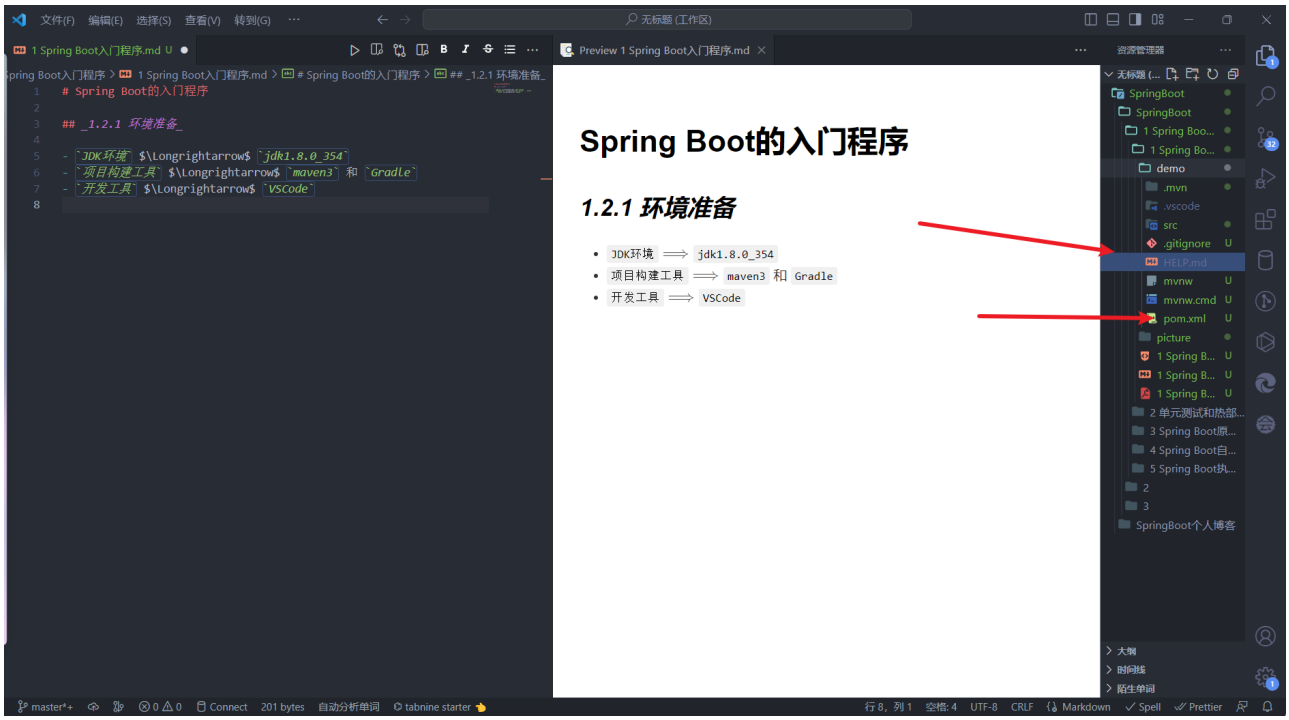


什么是groupid和artifactId? groupid和artifactId被统称为"坐标" 是为了保证项目唯一性而提出的，如果你要把你项目弄到maven本地仓库去，你想要找到你的项目就必须根据这两个id去查找。

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，artigactId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

```xml
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/200
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/
    <modelVersion>4.0.0</modelVersion>
    <!-- 引入Spring Boot依赖 -->
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>3.0.4</version>
        <relativePath/> <!-- lookup parent from repository -->
    </parent>
    <groupId>com.spring_boot_study</groupId>
    <artifactId>demo</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>demo</name>
    <description>Demo project for Spring Boot</description>
    <properties>
        <java.version>17</java.version>
    </properties>
    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-data-rest</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-graphql</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-hateoas</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-security</artifactId>
        </dependency>
        <dependency>
```

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

```xml
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-graphql</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-hateoas</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-security</artifactId>
        </dependency>
        <dependency>
            <!-- 引入Web场景依赖启动器 -->
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>
```

```
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web-services</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-webflux</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.data</groupId>
            <artifactId>spring-data-rest-hal-explorer</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.session</groupId>
            <artifactId>spring-session-core</artifactId>
        </dependency>

        <dependency>
            <groupId>org.springframework.boot</groupId>
```

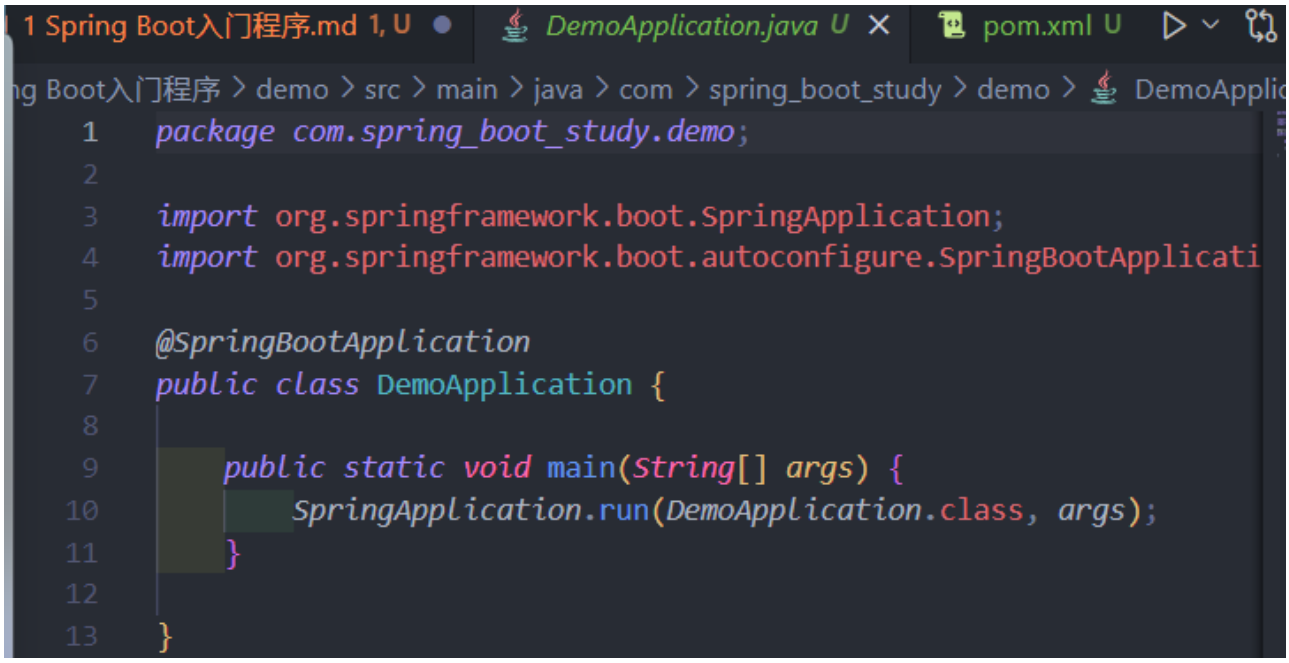
SpringBoot > SpringBoot > 1 Spring Boot开发入门 > 1 Spring Boot入门程序 > demo > pom.xml

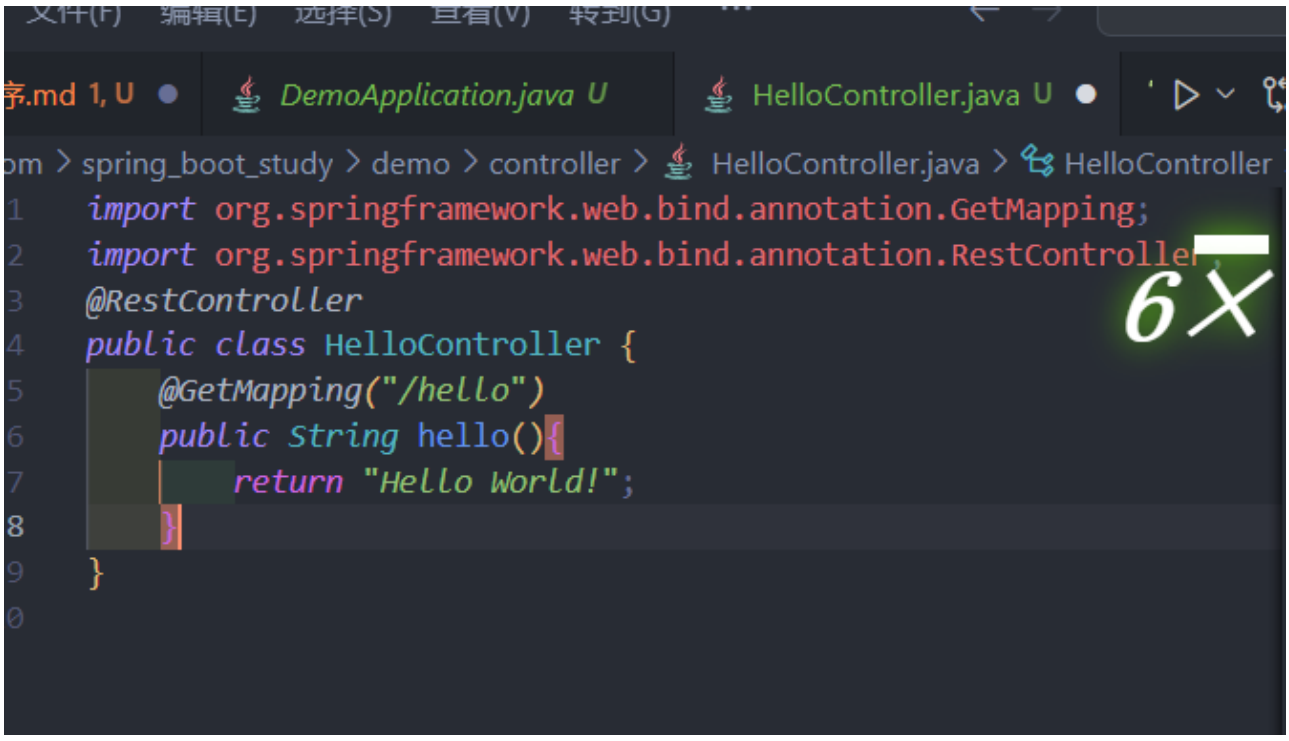
```
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-graphql</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-hateoas</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-security</artifactId>
        </dependency>
        <dependency>
            <!-- 引入web场景依赖启动器 -->
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web-services</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-webflux</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.data</groupId>
            <artifactId>spring-data-rest-hal-explorer</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.session</groupId>
            <artifactId>spring-session-core</artifactId>
        </dependency>

        <dependency>
            <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**

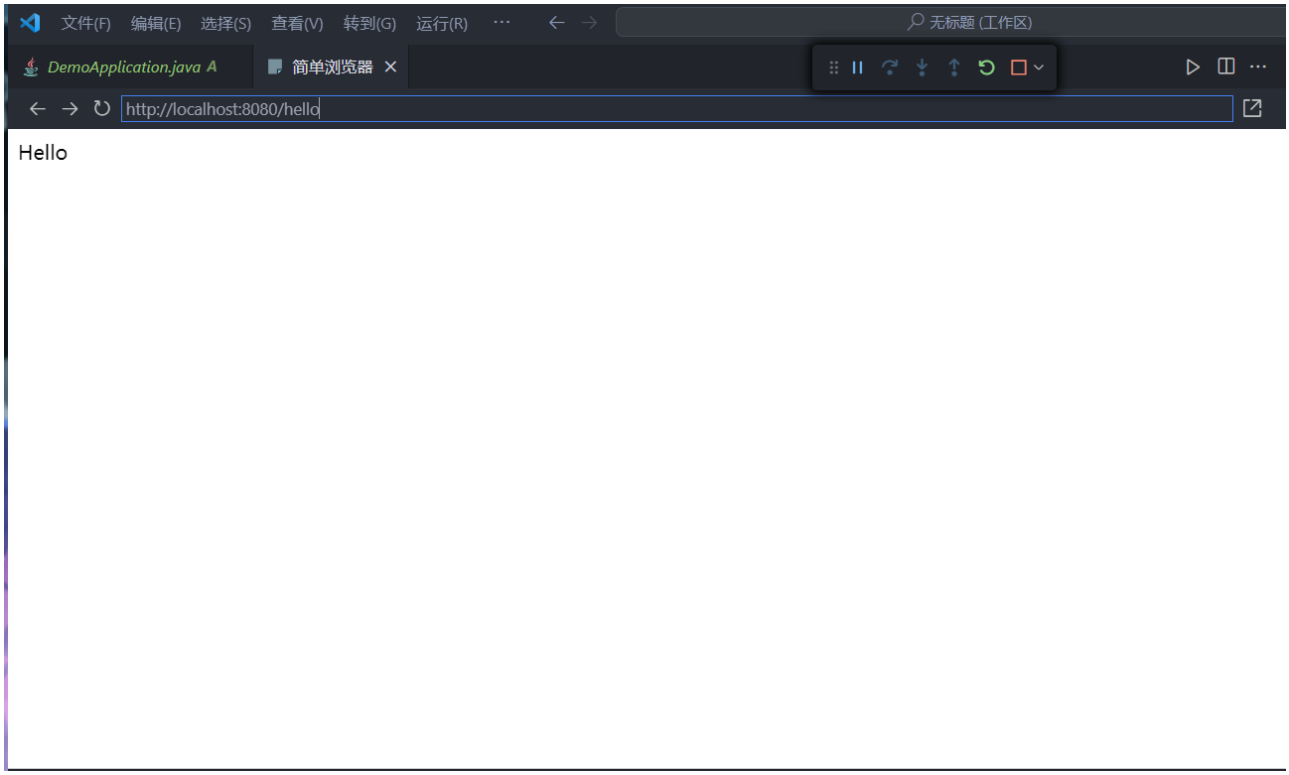




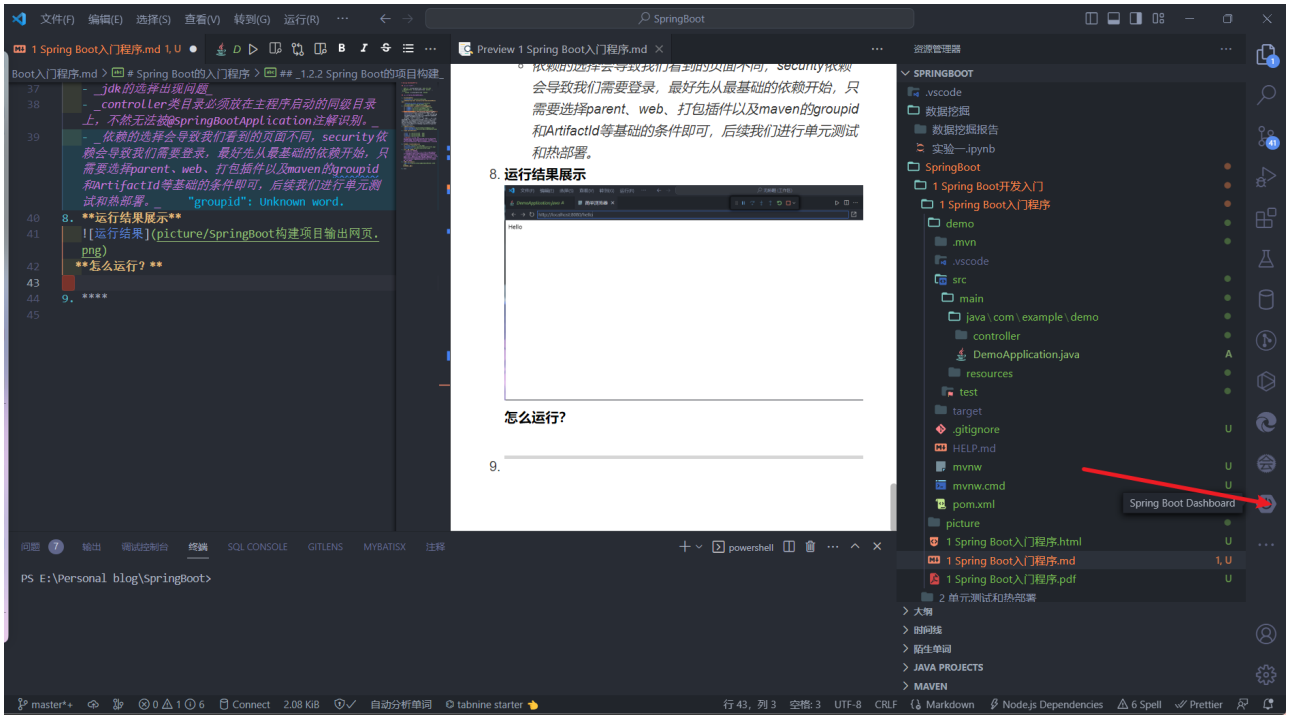
7. **关于运行时的建议**

   - *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。*

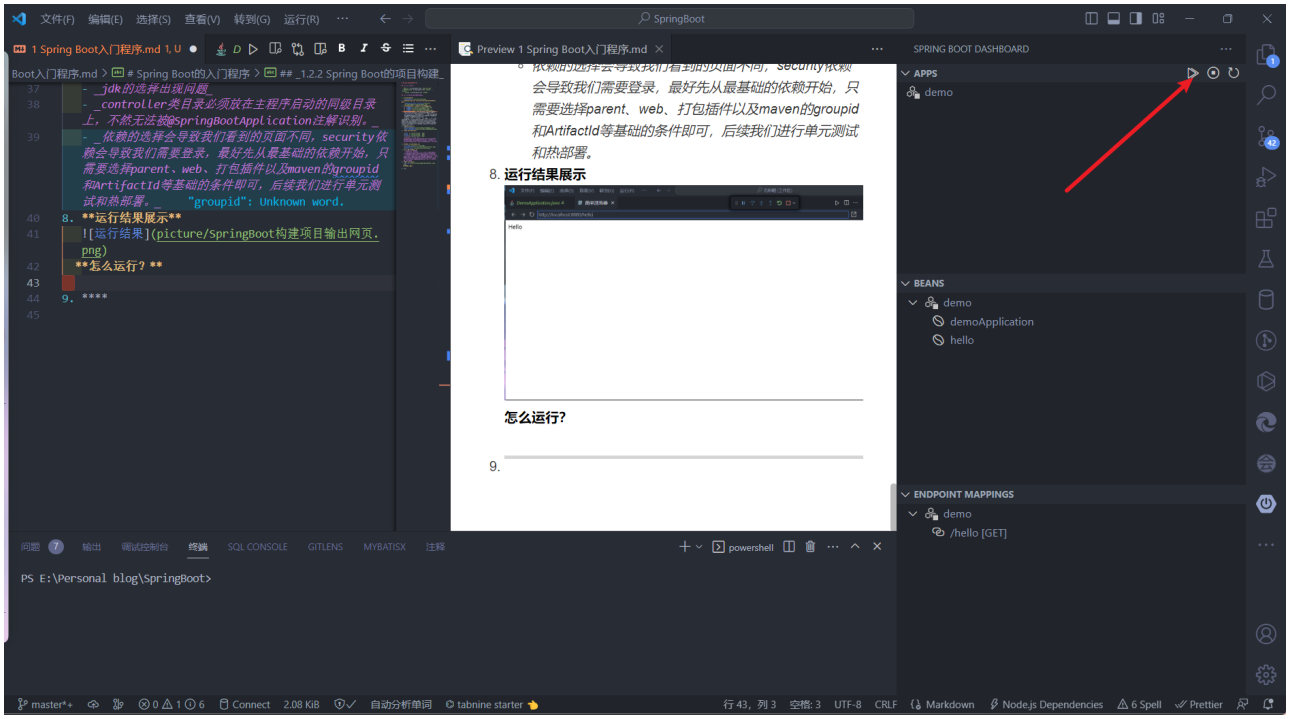
8. **运行结果展示**



怎么运行？ *选择Spring Boot DashBoard*



选择运行按钮


SPRING BOOT DASHBOARD
APPS
demo
BEANS
demo
demoApplication
hello
ENDPOINT MAPPINGS
demo
/hello [GET]
怎么运行?
PS E:\Personal blog\SpringBoot>