

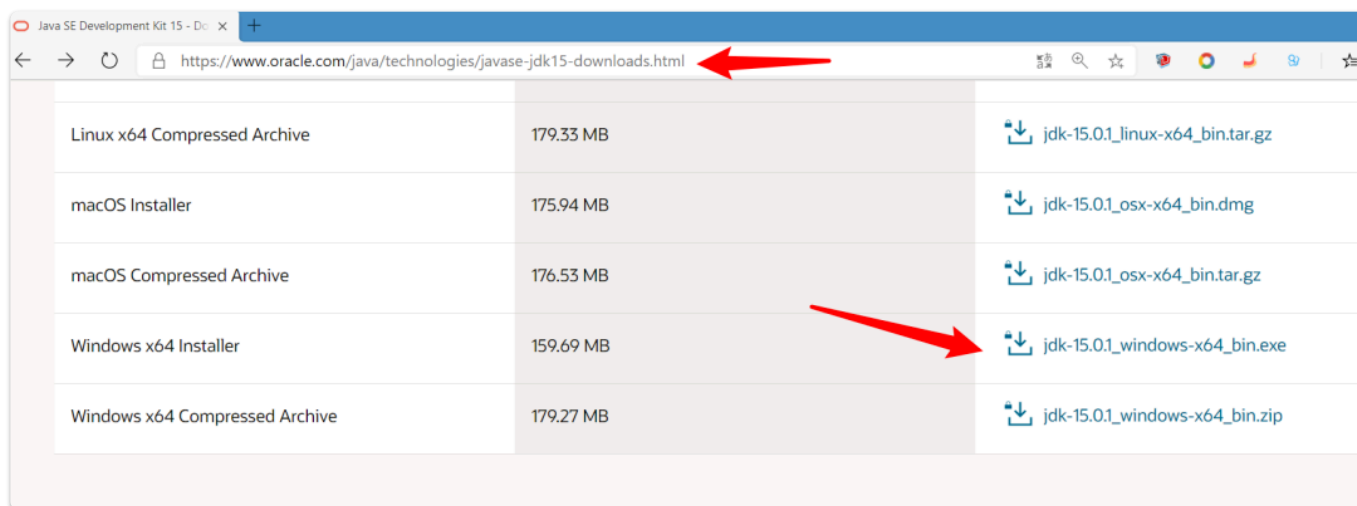
对于VS Code情有独钟。于是尝试了一下在Windows平台的VS Code中搭建Spring Boot的开发环境。

## 必要准备

Spring Boot属于Java框架，属于Java开发的必要条件，要先安装：

- 安装JDK

这一步，我们直接在Oracle官网下载最新的JDK安装即可：



然后双击安装，多次下一步之后，完成安装。

安装完成之后，记得配置一下环境变量（不知道为什么JDK的这个环境变量，一直不是自动配置的）：

# 关于

系统正在监控并保护你的电脑。

[在 Windows 安全中心中查看详细信息](#)

环境变量

twgww 的用户变量(U)

编辑系统变量

变量名(N):

JAVA\_HOME

变量值(V):

C:\Program Files\Java\jdk-15.0.1

浏览目录(D)...

浏览文件(F)...

确定

取消

系统变量(S)

变量	值
ComSpec	C:\Windows\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
JAVA_HOME	C:\Program Files\Java\jdk-15.0.1
NUMBER_OF_PROCESSORS	8
OS	Windows_NT
Path	C:\Program Files\Common Files\Oracle\Java\javapath;C:\Windo...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE	AMD64

新建(W)...

编辑(I)...

删除(L)

确定

取消

系统属性

计算机名 硬件 高级 系统保护 远程

要进行大多数更改，你必须作为管理员登录。

处理器计划，内存使用，以及虚拟内存

设置(S)...

设置(E)...

系统故障和调试信息

设置(T)...

环境变量(N)...

确定

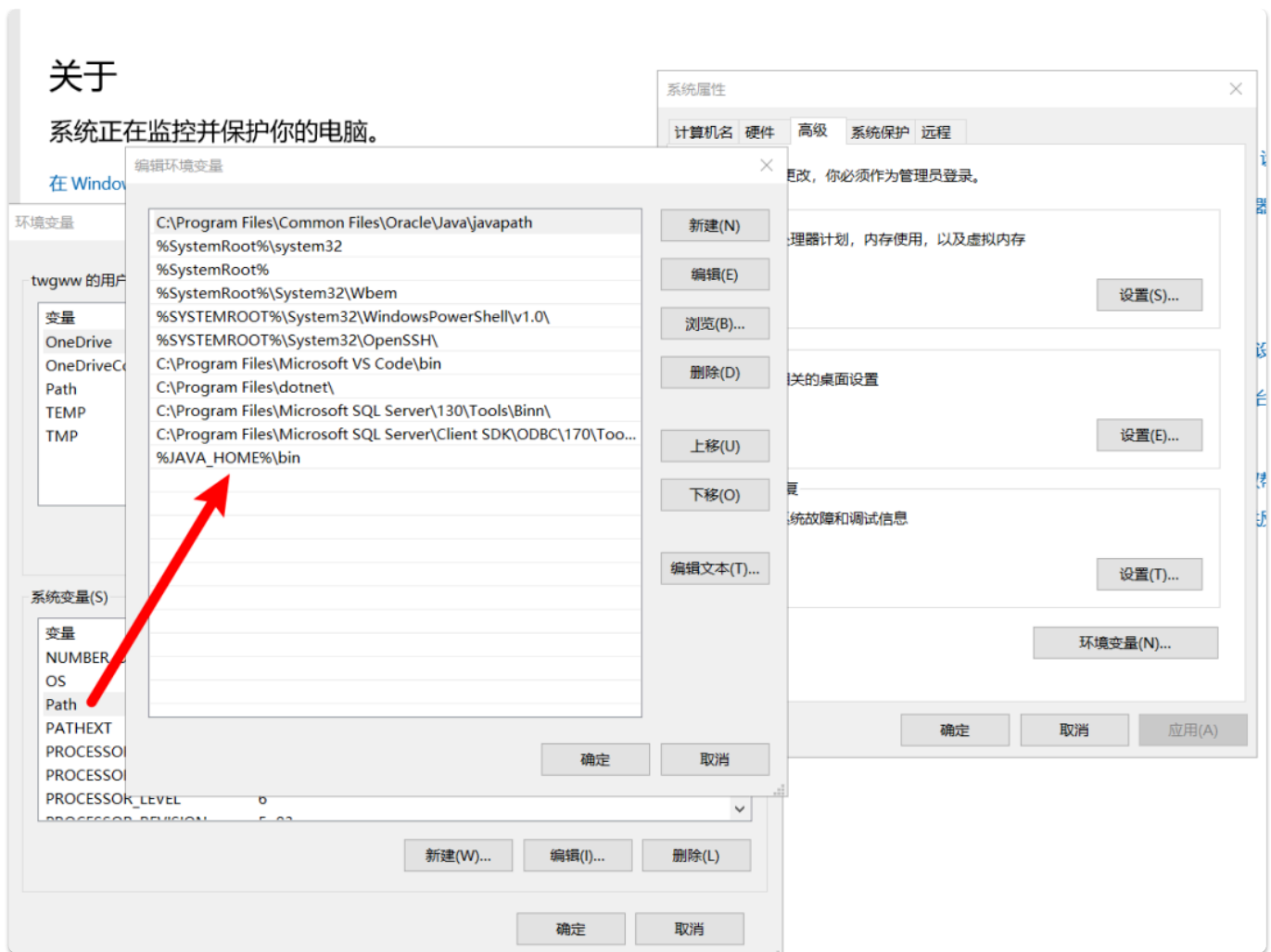
取消

应用(A)

设置  
器

设置  
台电脑

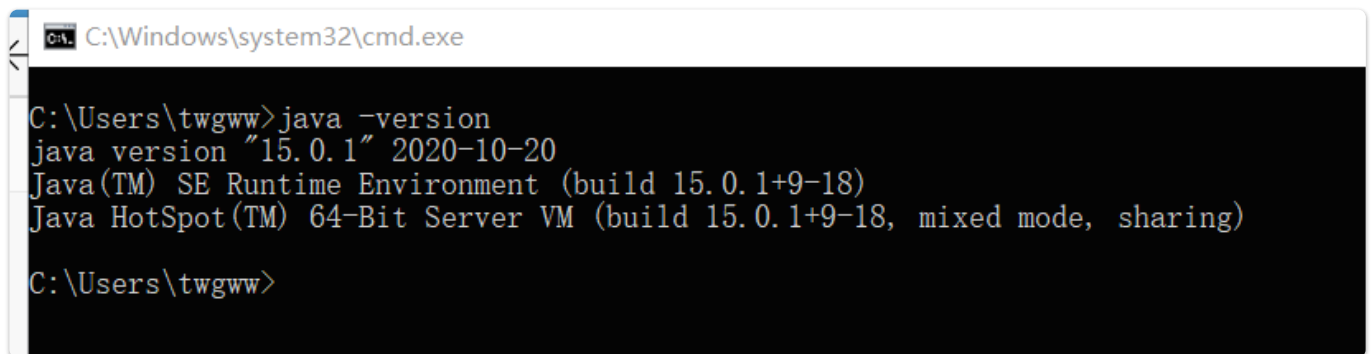
帮助  
反馈



当你的 VS Code 遇上 Spring Boot

配置好环境变量之后，在控制台中，执行命令来检查是否安装成功：

```
java -version
```

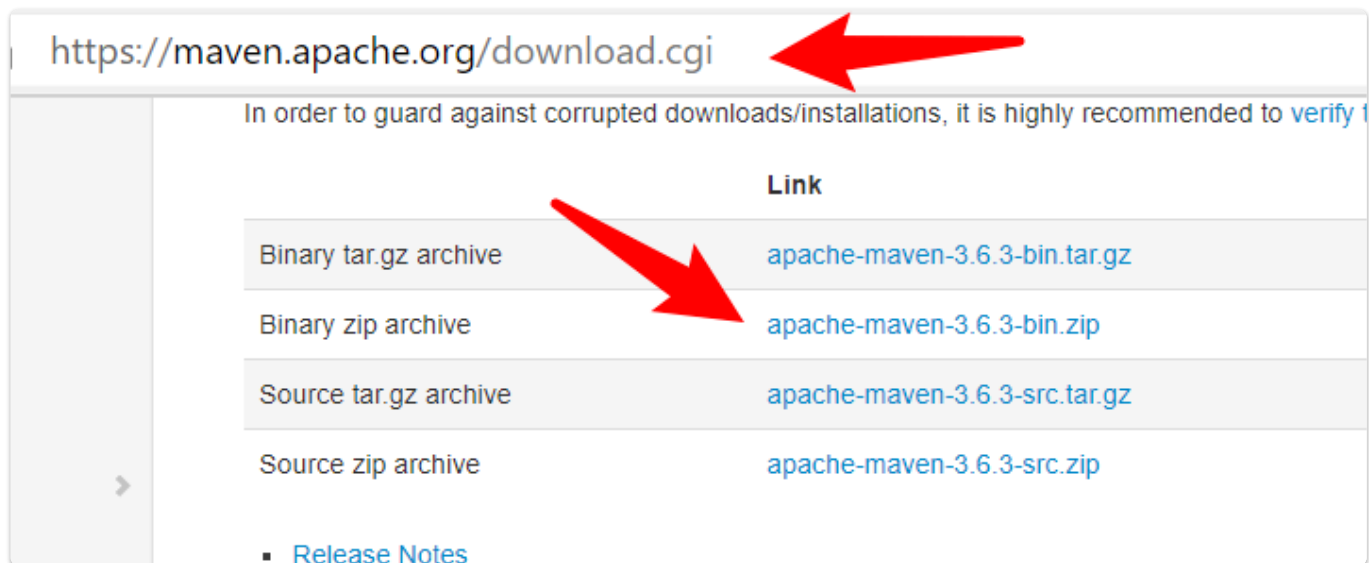


如果执行以上命令，成功看到Java的版本号，恭喜您，Java JDK安装成功。

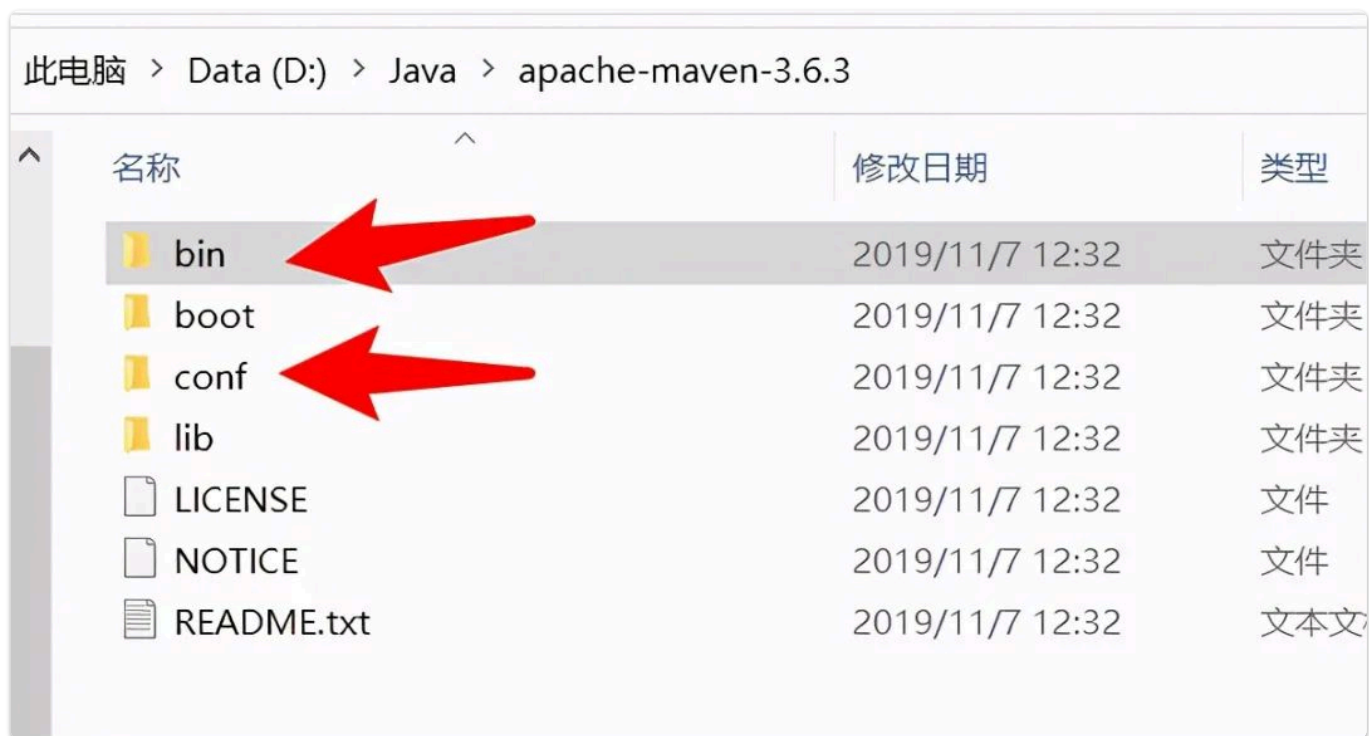
- 安装maven

maven作为Java界最流行的项目管理工具，在Spring Boot开发中，也是日常必须品，需要下载。

我们这里选择提前编译好的zip二进制包即可：

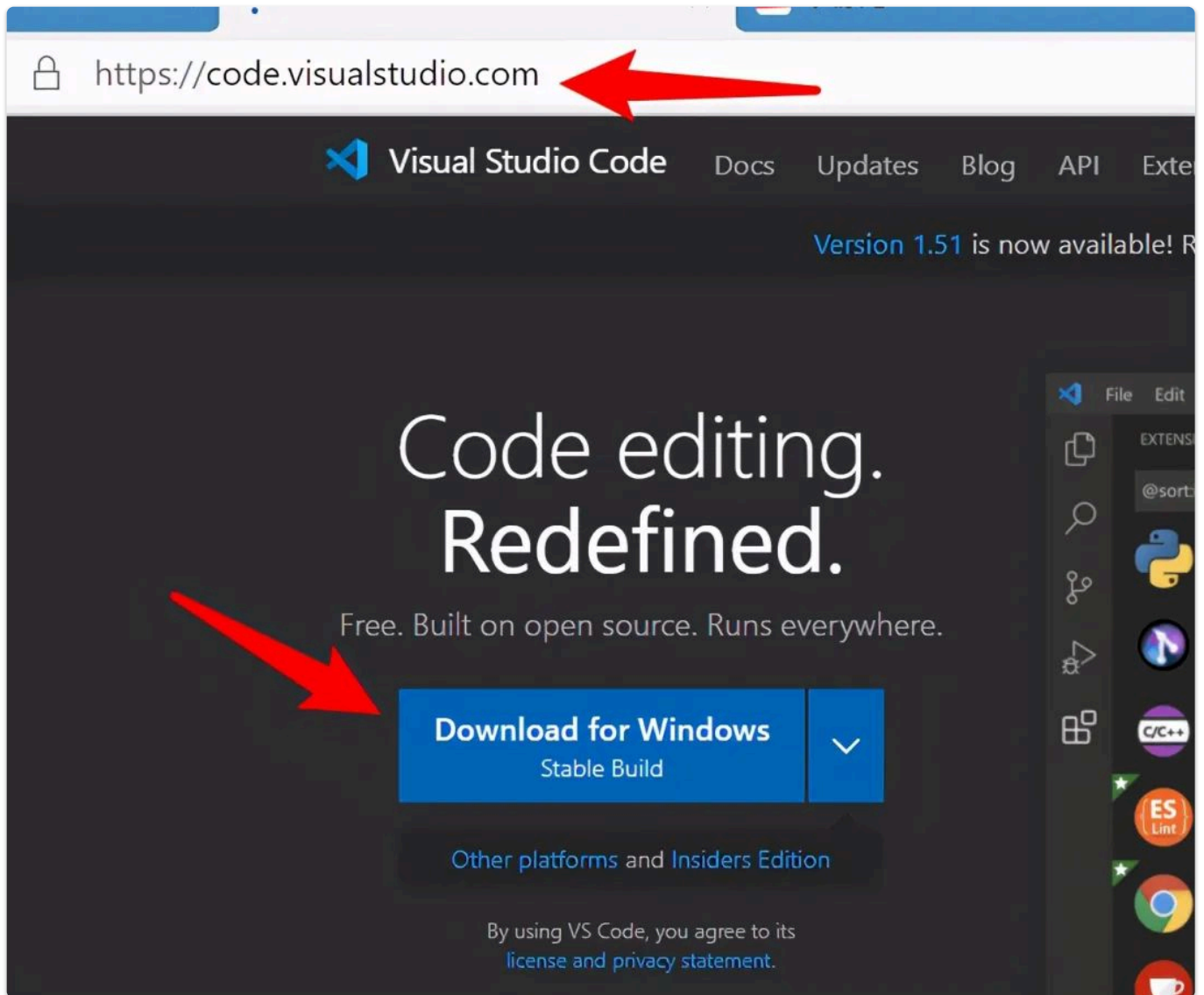


下载好maven在包之后，解压出来，找一个自己喜欢的位置，保存即可。记住您保存的位置，因为后面我们需要在VS Code中，配置maven相关的参数。



maven

- 安装VS Code

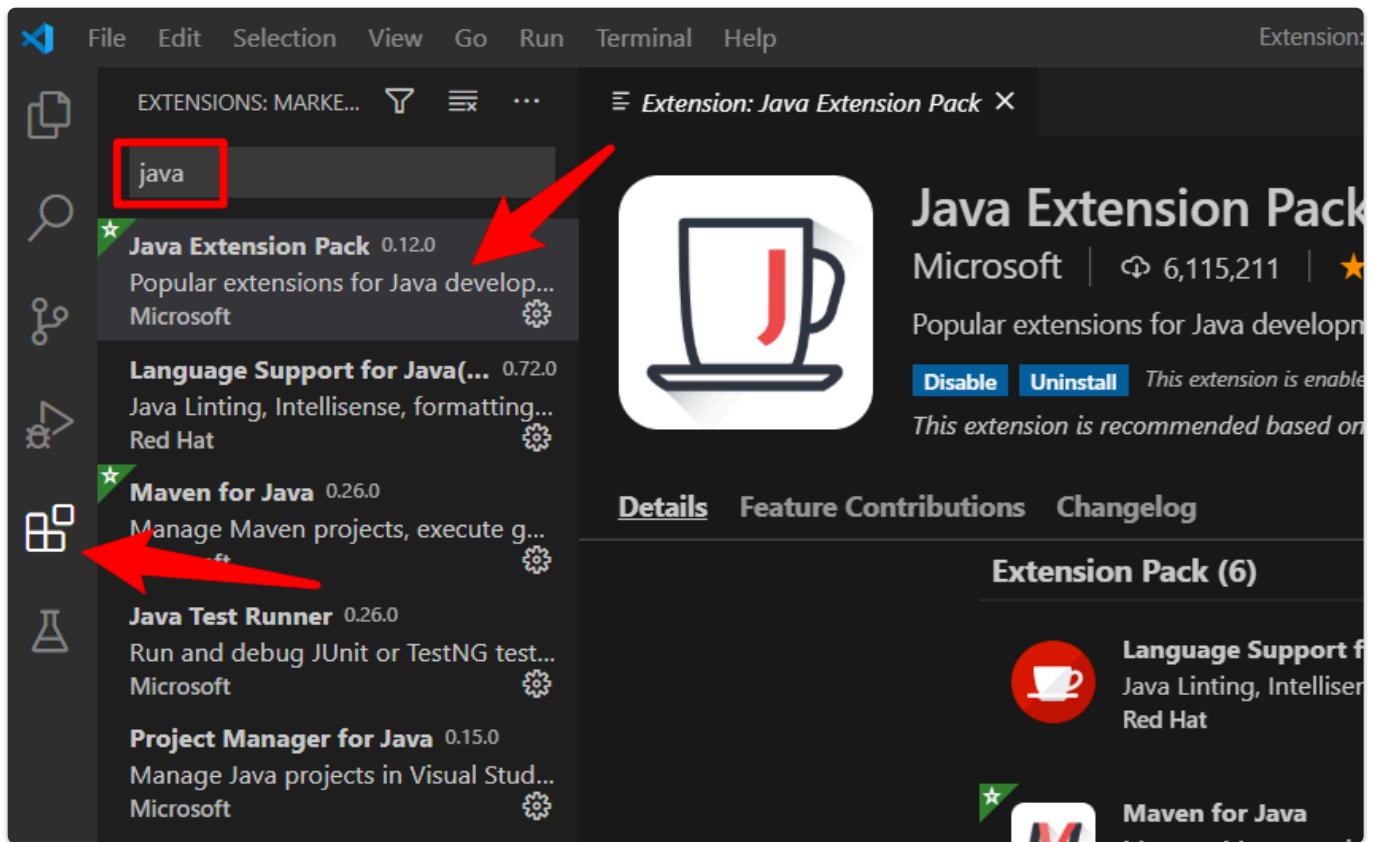


## 安装必要插件

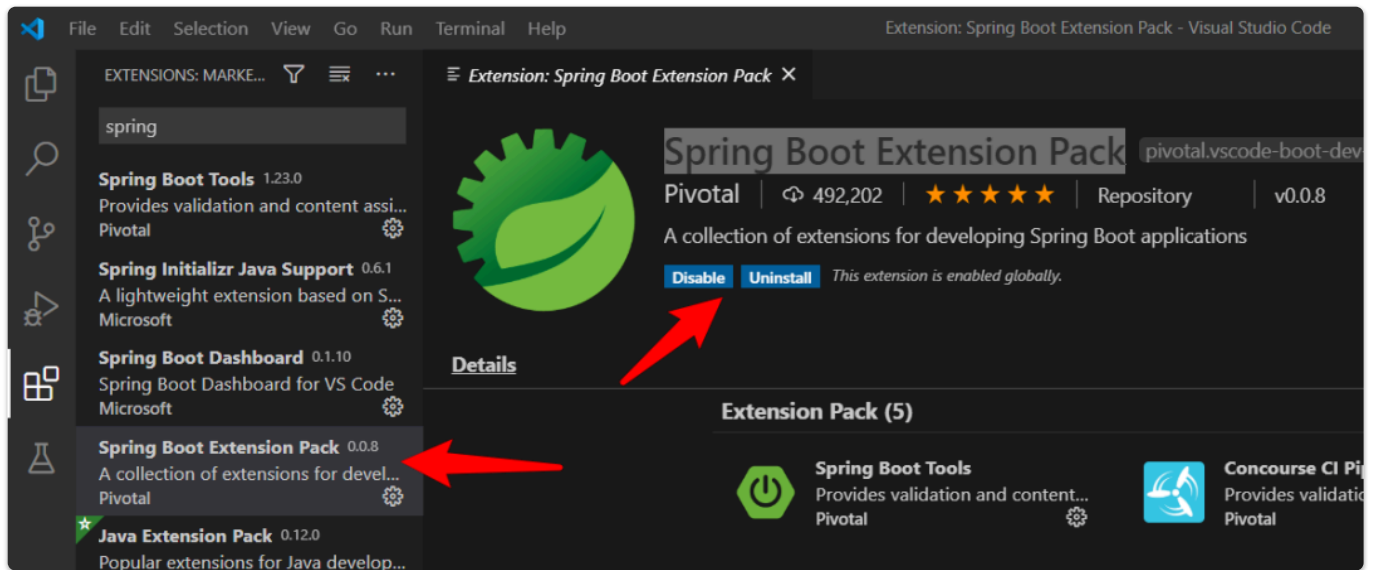
现在打开VS Code,安装下面两个插件:

Java Extension Pack

Spring Boot Extension Pack



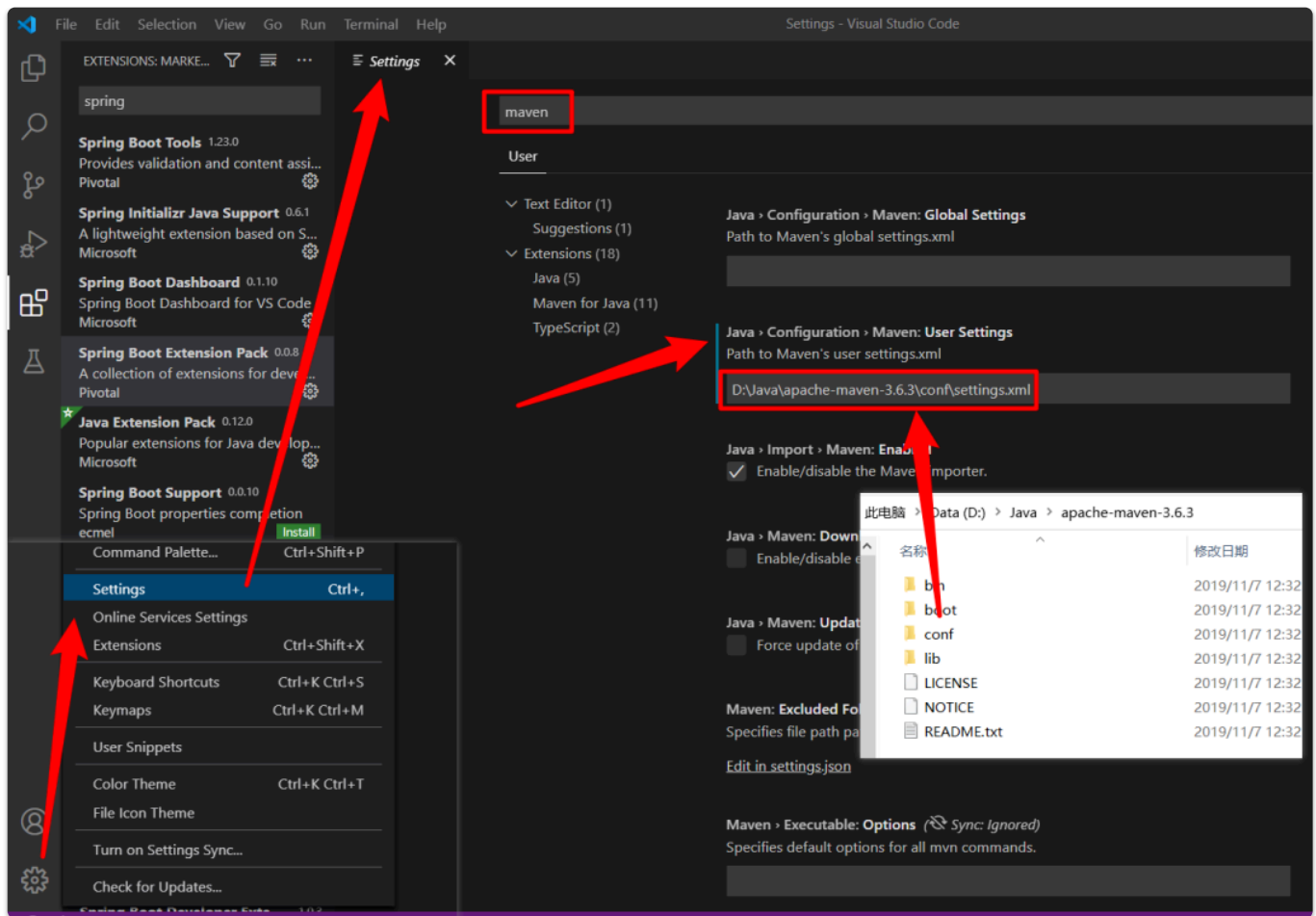
Java Extension PackJava Extension Pack 安装完成



Spring Boot Extension Pack 安装完成

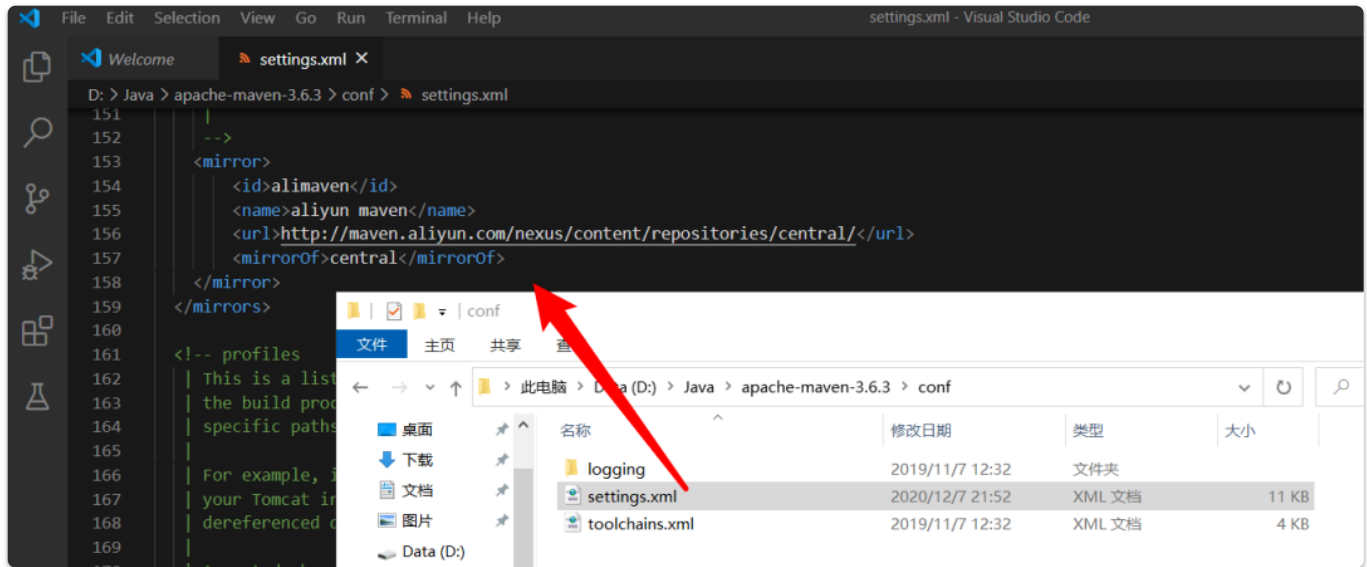
## 配置maven选项

现在还不能进行开发，还需要进行一些配置，不要着急：



配置maven的settings.xml路径

为了提升maven的访问速度，我们修改其源为 aliyun maven：



修改maven的镜像源为aliyun

maven的settings.xml中的局部代码如下：

```
<mirror>

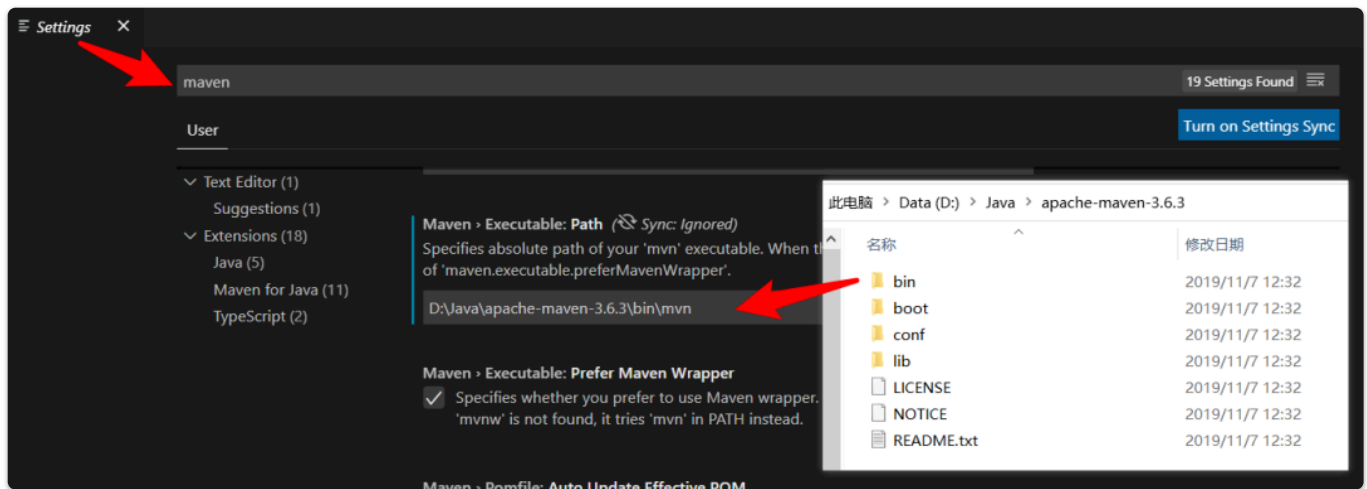
<id>alimaven</id>

<name>aliyun maven</name>

<url>http:

<mirrorOf>central</mirrorOf>

</mirror>
```

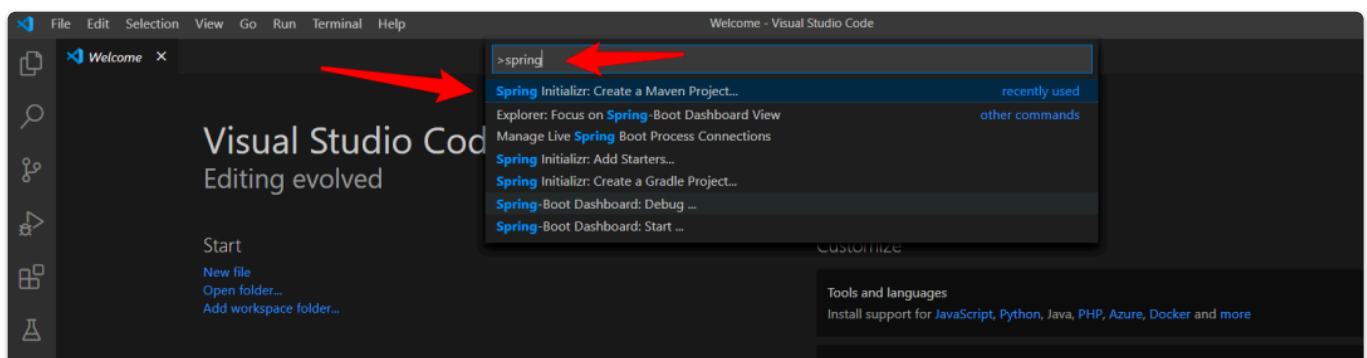


配置maven的执行路径

好了，到这里，我们的环境配置就完成了。重启VS Code，下一步就可以进行开发了！

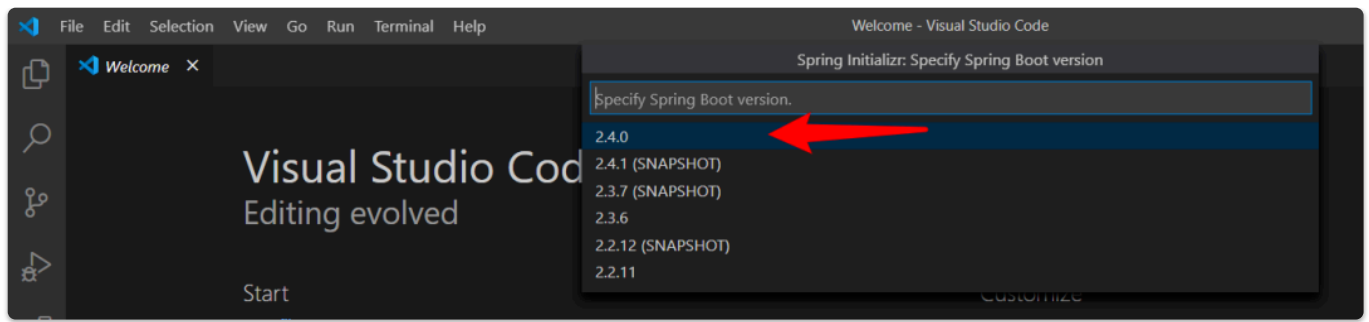
## 创建测试Demo项目

再次启动VS Code，按 Ctrl + Shift + P 快捷键，调出命令，输入spring，然后新建一个maven项目：

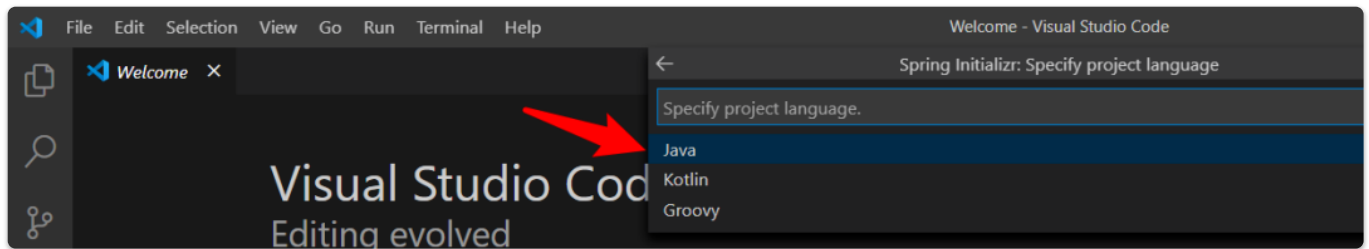


新建maven项目

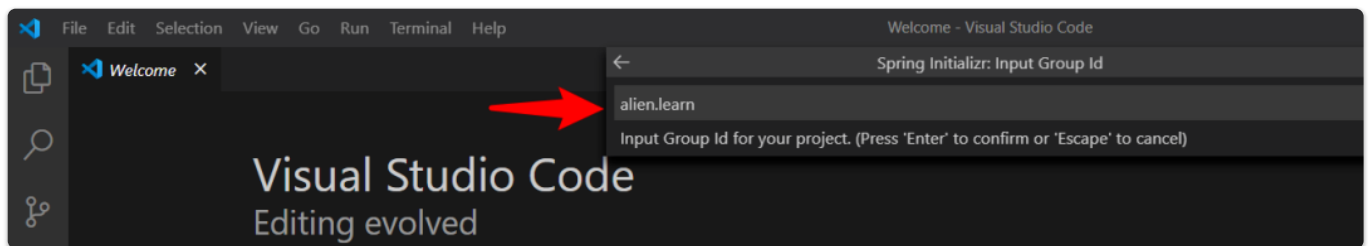




选择Spring Boot版本

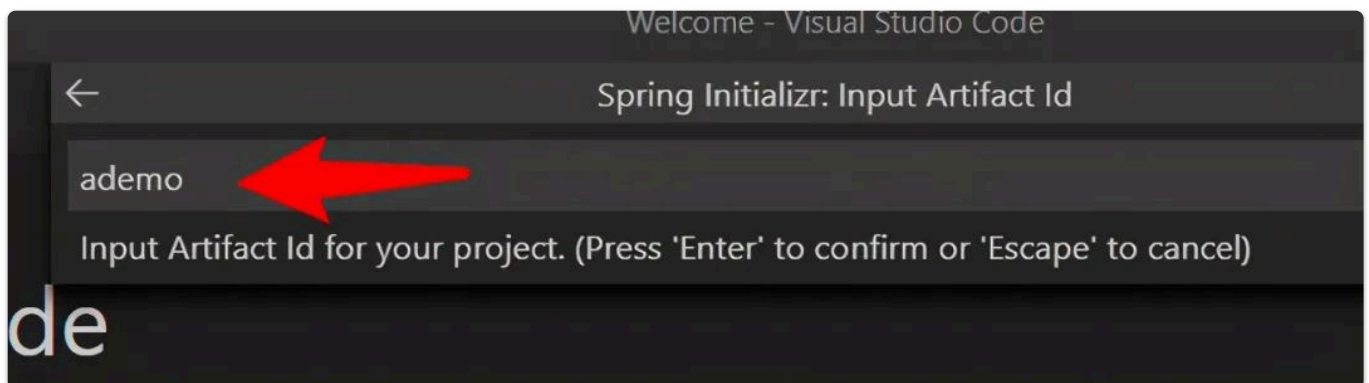


选择语言



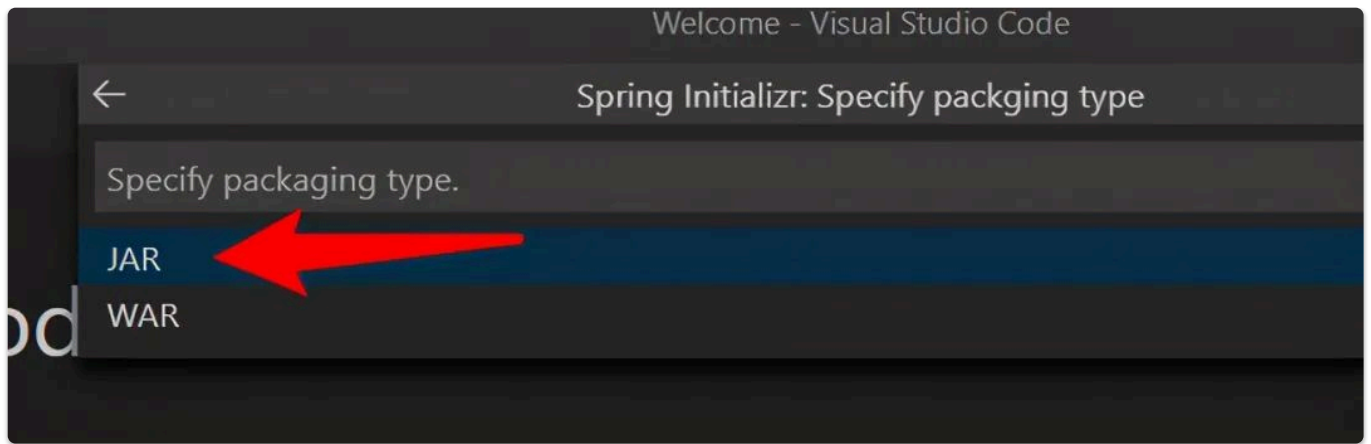
输入Group Id

注意：输入Group Id之后，敲回车键

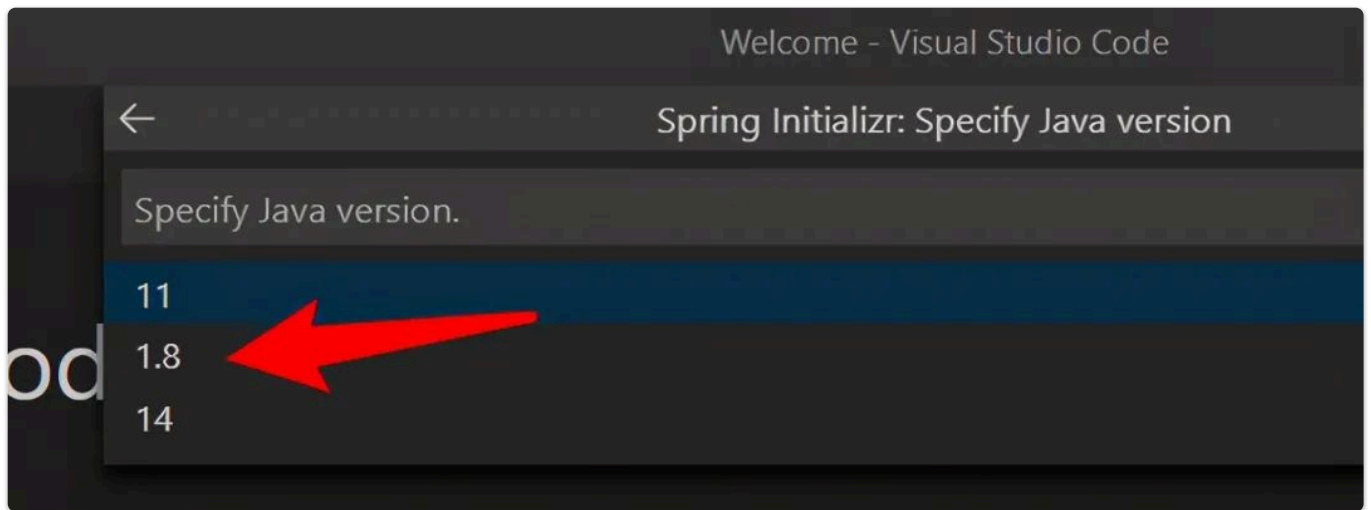


输入Artifact Id

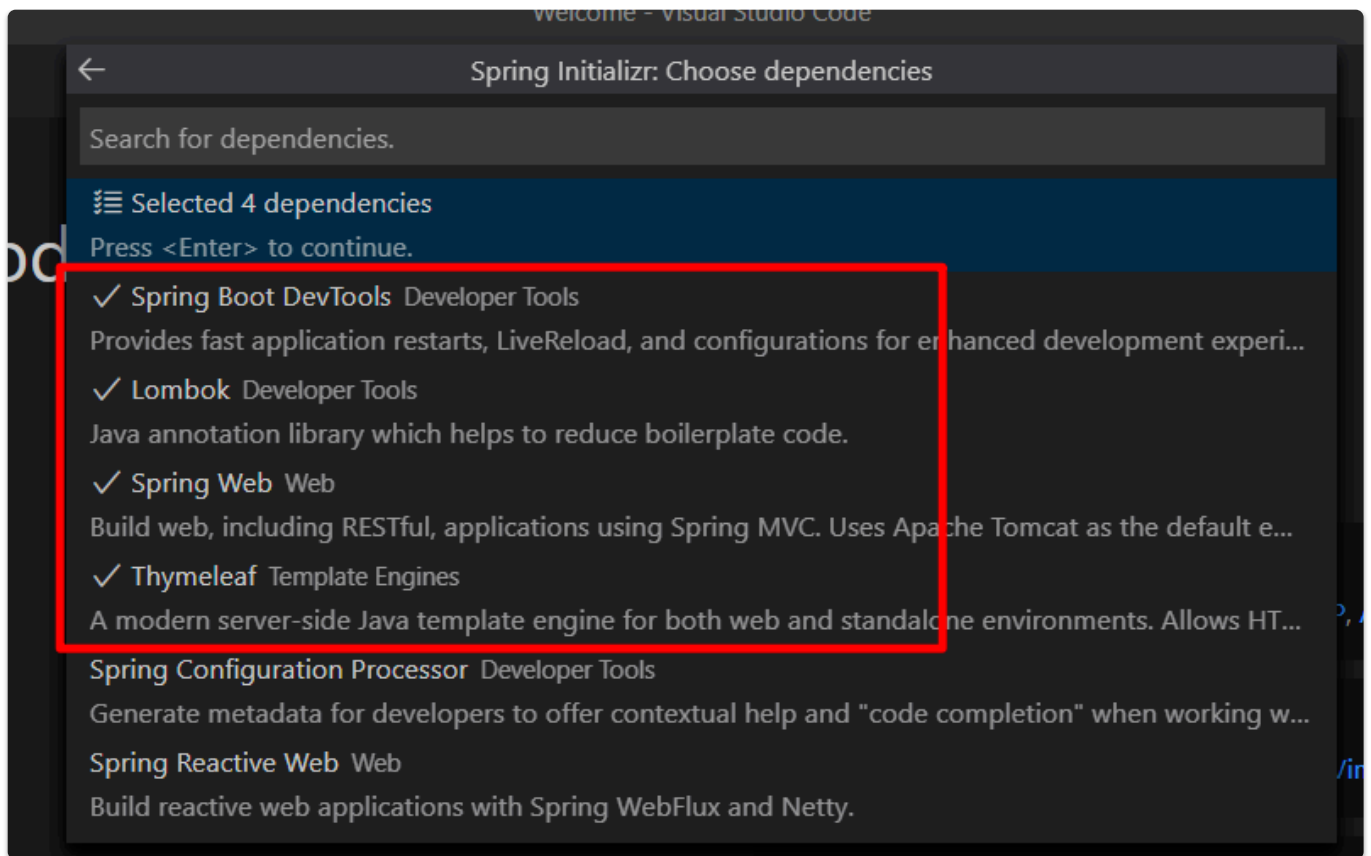
敲回车键



选择包类型



选择Java版本



选择常用依赖包

敲回车键

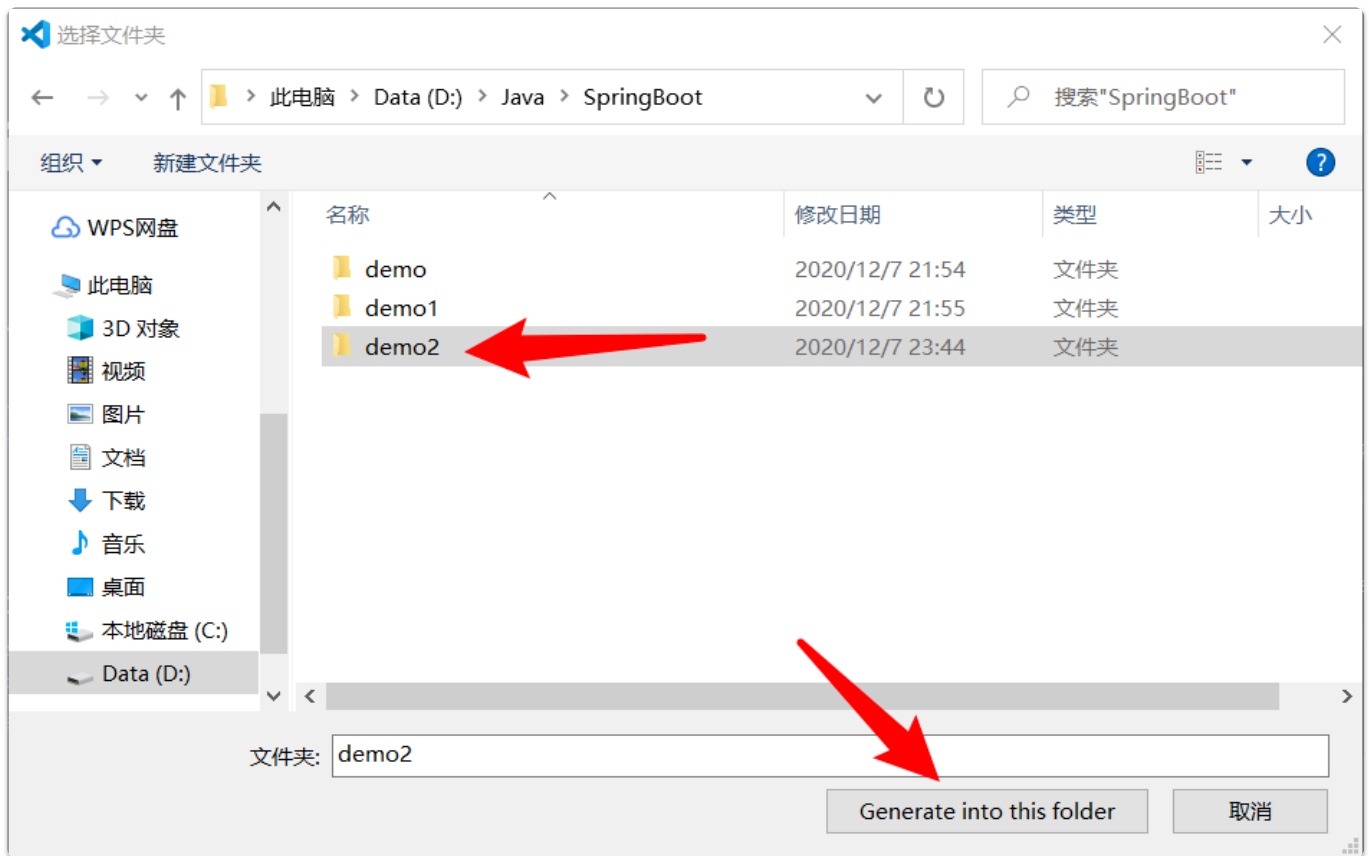
依赖包功能简介：

**Spring Boot DevTools:** 支持代码修改热更新，无需重启

**Spring Web:** 集成Tomcat SpringMVC

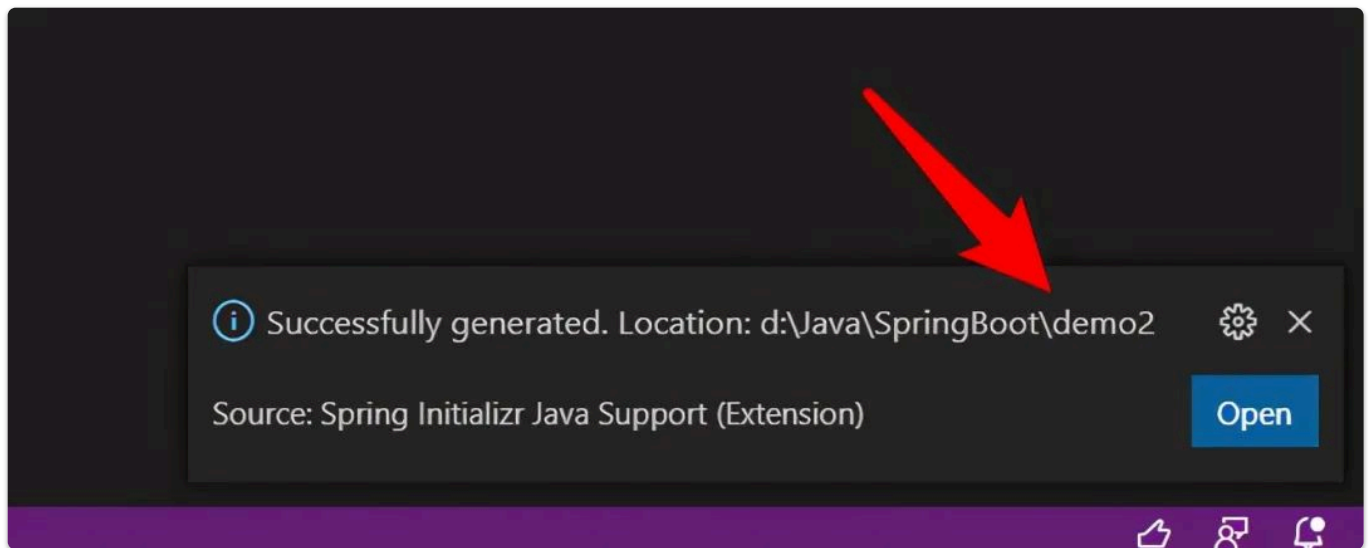
**Lombok:** 智能生成 setter getter toString等

**Thymeleaf:** 模板引擎

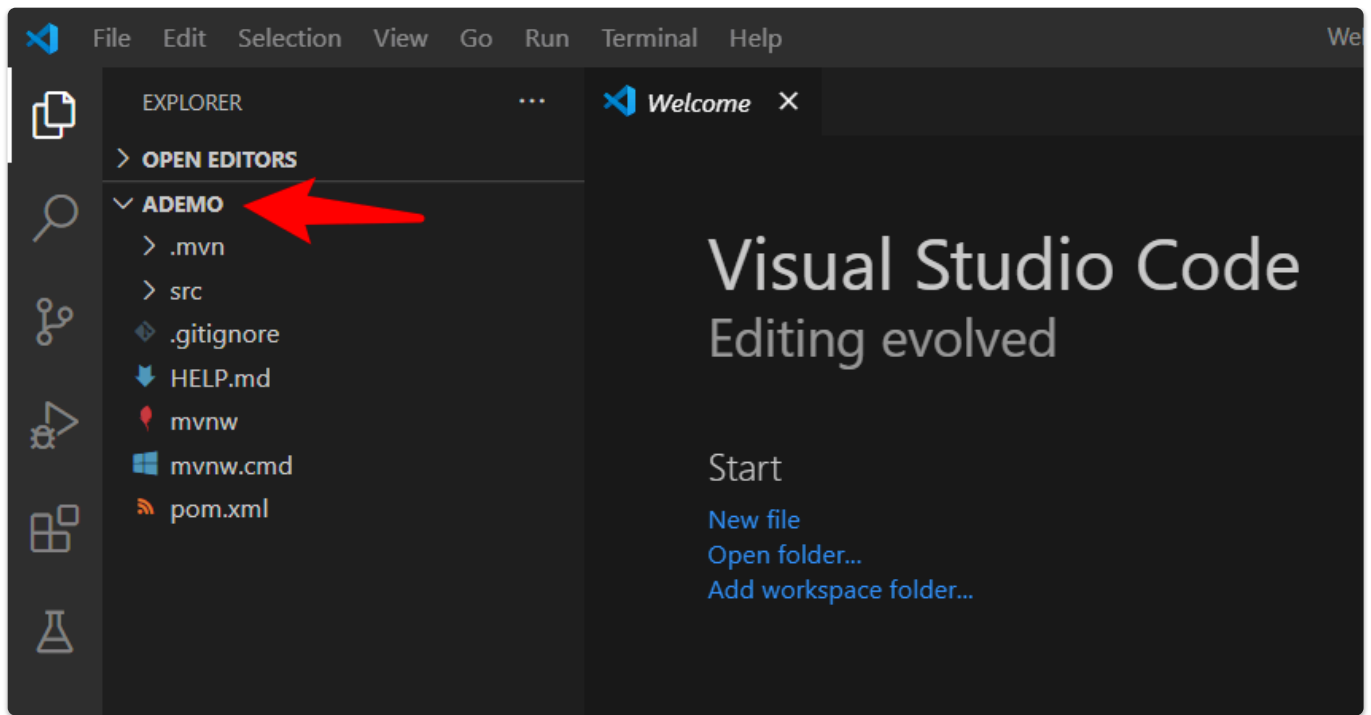


选择保存位置

等待项目创建完成，然后打开：



打开项目

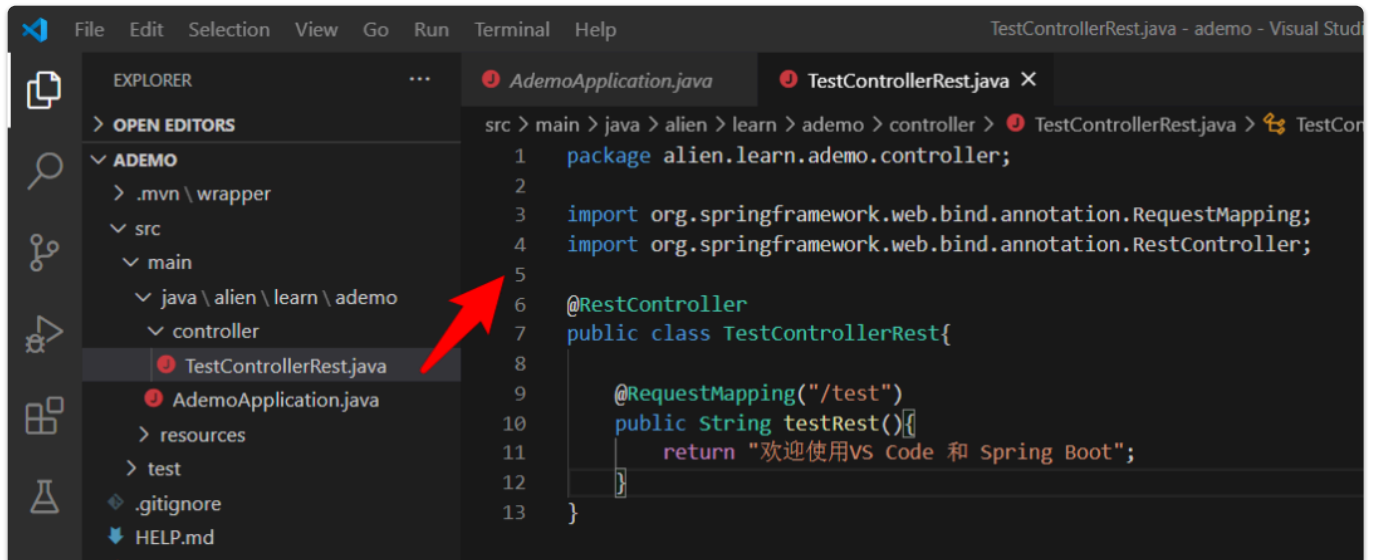


项目打开成功

如果看到上面的源代码结构，那么恭喜您，您成功了！

## 添加测试业务代码

项目创建成功，接下来，添加测试业务代码：



添加Rest

代码如下：

```

package alien.learn.ademo.controller;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RestController;

@RestController
public class TestControllerRest{

    @RequestMapping("/test")
    public String testRest(){

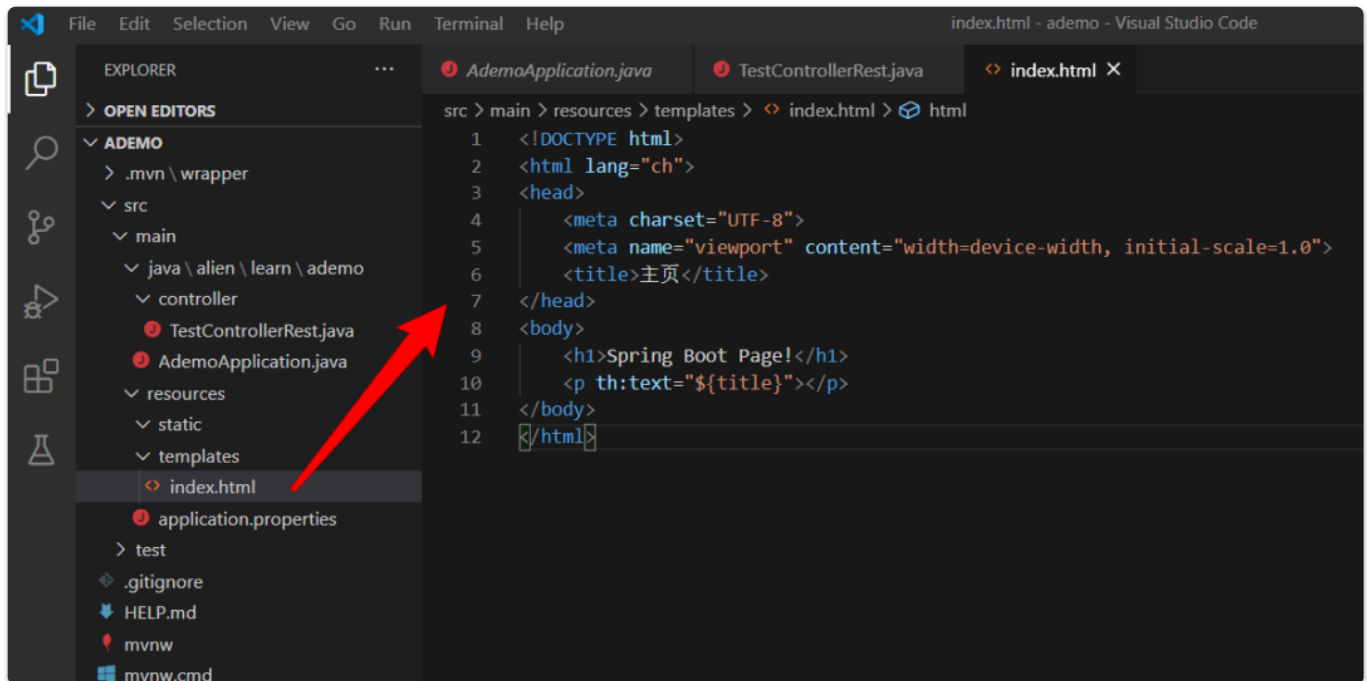
        return "欢迎使用VS Code 和 Spring Boot";

    }

}

```

再添加一个html模板：



代码：

```

<!DOCTYPE html>

<html lang="ch">

```

```
<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-
scale=1.0">

    <title>主页</title>

</head>

<body>

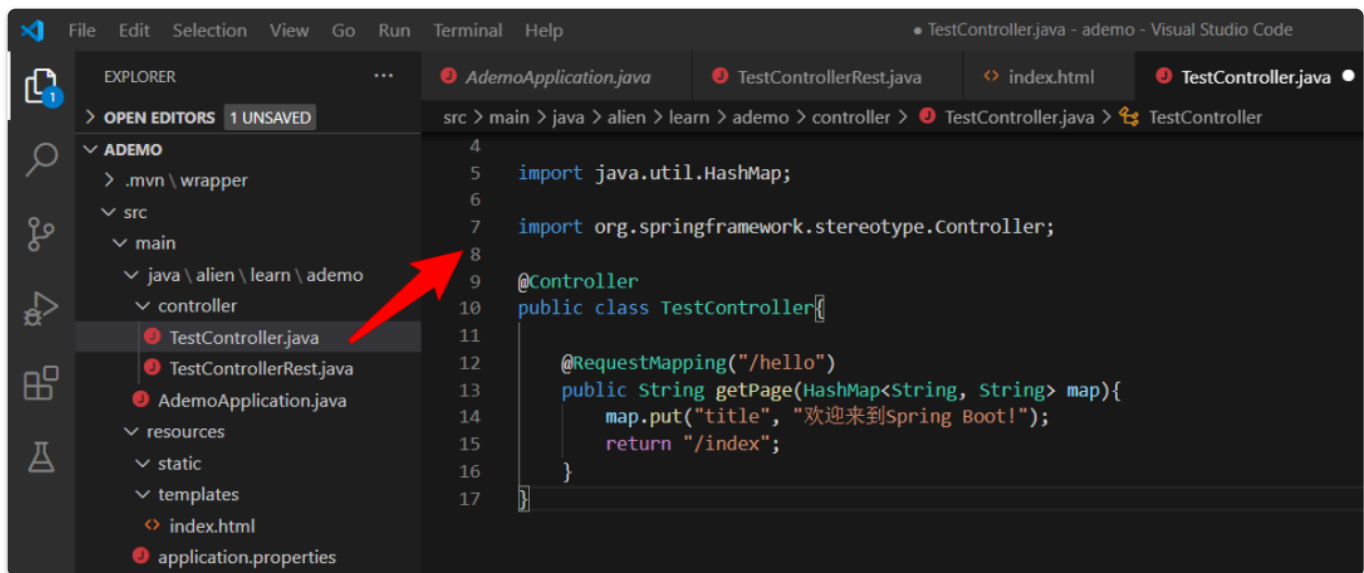
    <h1>Spring Boot Page!</h1>

    <p th:text="${title}"></p>

</body>

</html>
```

再添加一个Controller:



代码:

```
package alien.learn.ademo.controller;

import java.util.HashMap;
```

```

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.stereotype.Controller;

@Controller
public class TestController{

    @RequestMapping("/hello")
    public String getPage(HashMap<String, String> map){

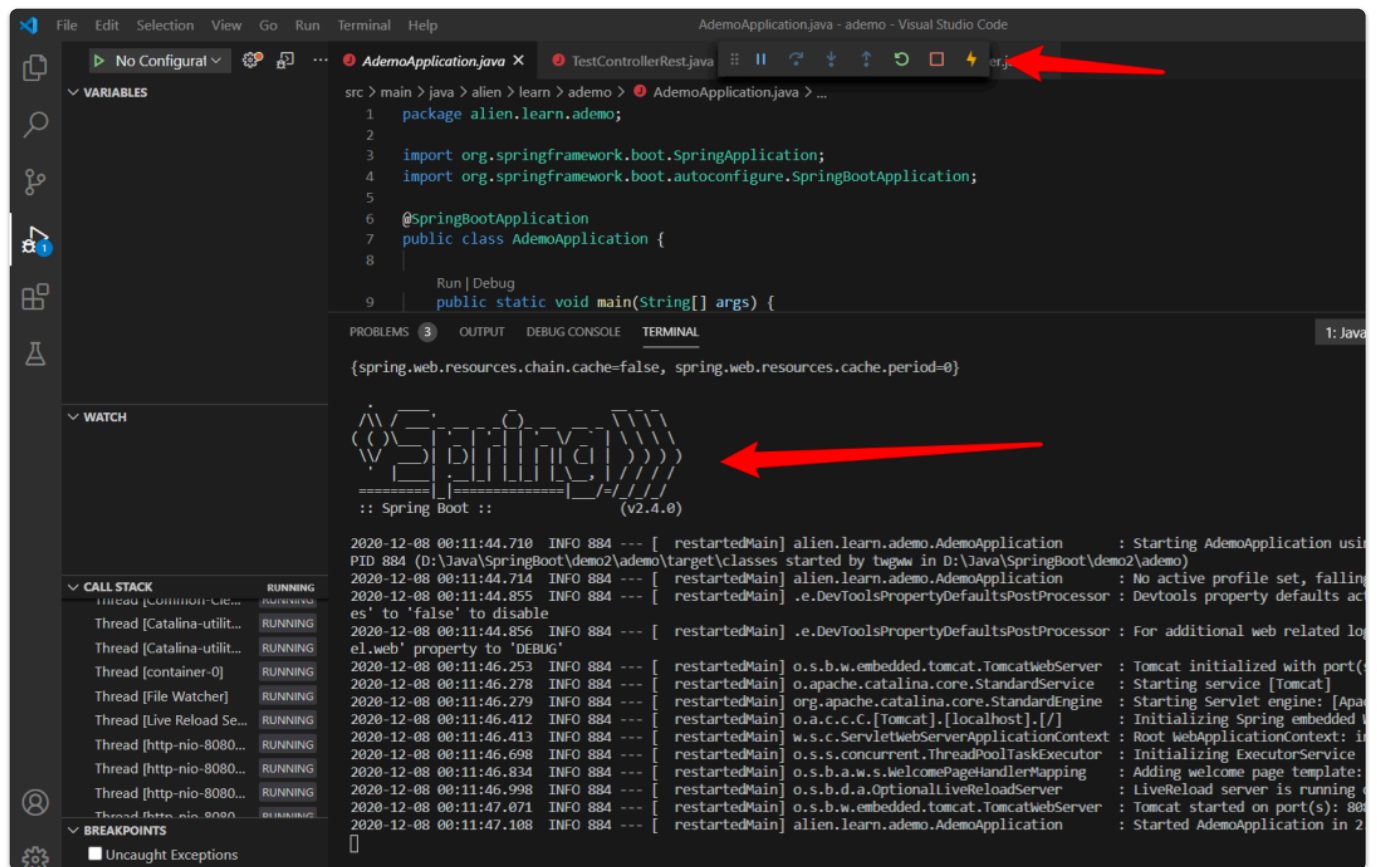
        map.put("title", "欢迎来到Spring Boot!");
        return "/index";

    }

}

```

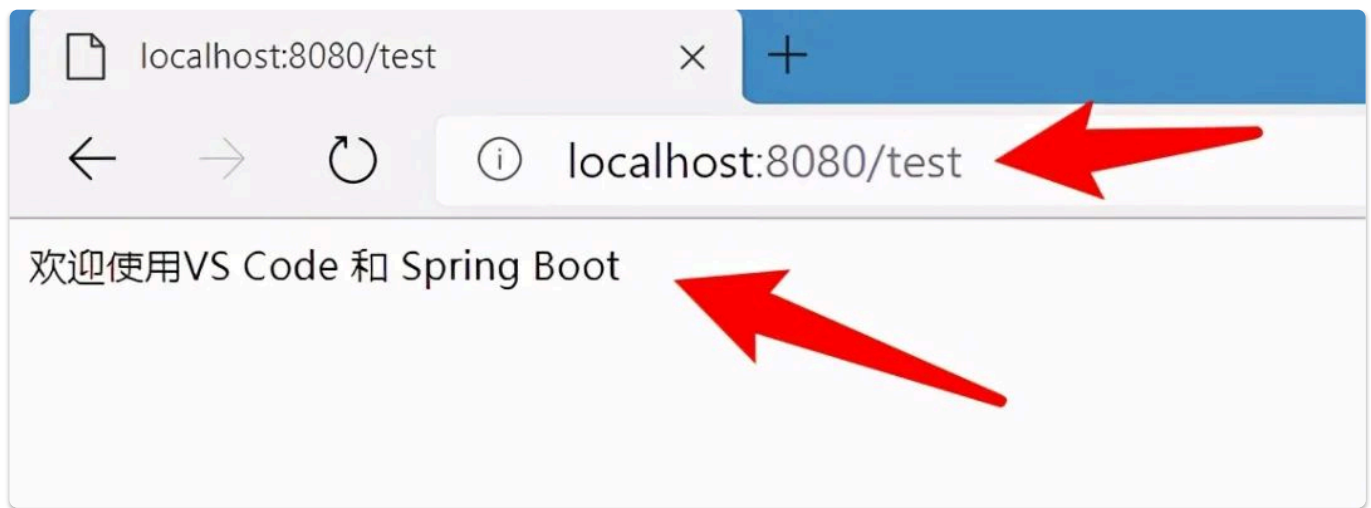
然后F5调试服务：



调试服务

然后在浏览器中尝试访问服务：





再访问一下hello:

