

一、扩展包说明

1 扩展列表里这些是什么？

你现在装的这些👉

- Language Support for Java (Red Hat)
- Maven for Java
- Debugger for Java
- Spring Boot Extension Pack
- Project Manager for Java

👉 它们的作用只有这些：

能做的	不能做的
代码提示	✗ 安装 JDK
语法检查	✗ 改系统 Java 版本
Maven 集成	✗ 设置 JAVA_HOME
Spring Boot 运行按钮	✗ 替换 /usr/bin/java

所以：

终端里显示的 JDK 11，不是你装的，是 Codespaces 镜像自带的

2 为什么系统偏偏是 JDK 11？

因为 GitHub Codespaces 默认基础镜像是：

- Ubuntu
- OpenJDK 11 (**历史兼容性最好**)

但问题来了：

Maven 4.x 不认 JDK 11

于是就炸了 💣

二、正确姿势：你应该“在系统里”装 JDK 17（不是扩展）

✓ 你现在唯一要做的事

忽略扩展，直接在终端装 JDK 17

下面这套是 Codespaces 专用、安全、不会搞坏环境的做法。

三、一步一步照着敲（重点来了）



第1步：安装 JDK 17

```
sudo apt update  
sudo apt install -y openjdk-17-jdk
```

第2步：确认现在系统里有两个 Java

```
ls /usr/lib/jvm
```

你大概率会看到类似：

```
java-11-openjdk-amd64  
java-17-openjdk-amd64
```

这一步很重要，说明安装是成功的。

第3步：告诉系统“以后用 JDK 17”

```
sudo update-alternatives --config java
```

会看到类似：

Selection	Path

* 0	/usr/lib/jvm/java-11-openjdk-amd64/bin/java
1	/usr/lib/jvm/java-17-openjdk-amd64/bin/java

👉 输入 1，回车。

第4步：设置 JAVA_HOME（90% 的坑都在这一步）

```
echo 'export JAVA_HOME=/usr/lib/jvm/java-17-openjdk-amd64' >> ~/.bashrc  
echo 'export PATH=$JAVA_HOME/bin:$PATH' >> ~/.bashrc  
source ~/.bashrc
```

第5步：三连验证（一定要看到对的）

```
java -version  
mvn -version  
echo $JAVA_HOME
```

你应该看到：

- Java 17

- Maven 使用 Java 17
 - JAVA_HOME 指向 `java-17-openjdk-amd64`
-

四、这时再跑你的 SSM / Spring Boot

```
mvn clean spring-boot:run
```

 **报错会消失