# spring boot学习笔记

## 第一章 spring boot环境的搭建

### 一、下载安装JDK，并配置java运行环境变量

参考：<https://www.w3cschool.cn/java/java-environment-setup.html>

### 二、安装eclipse

参考：<https://www.w3cschool.cn/eclipse/eclipse-install.html>

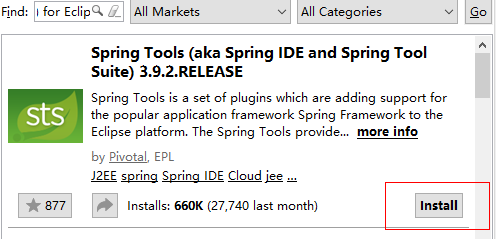
### 三、通过STS插件自动创建spring boot项目

步骤如下：

1**、Eclipse中安装STS插件：**

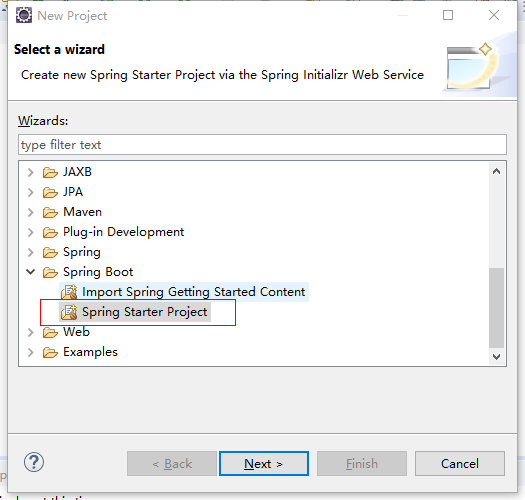
Help -> Eclipse Marketplace...

Search或选择“Popular”标签，选择Spring Tool Suite (STS) for Eclipse插件，选择让install，选择默认，安装完成即可：

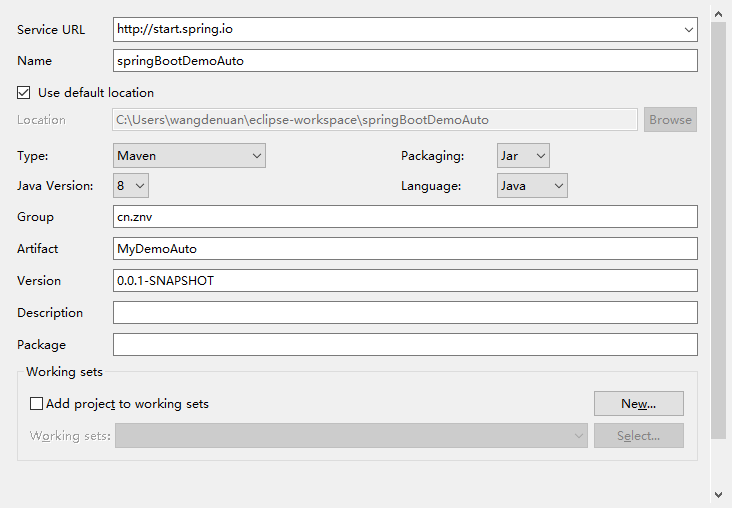


**2、New -> Project...**

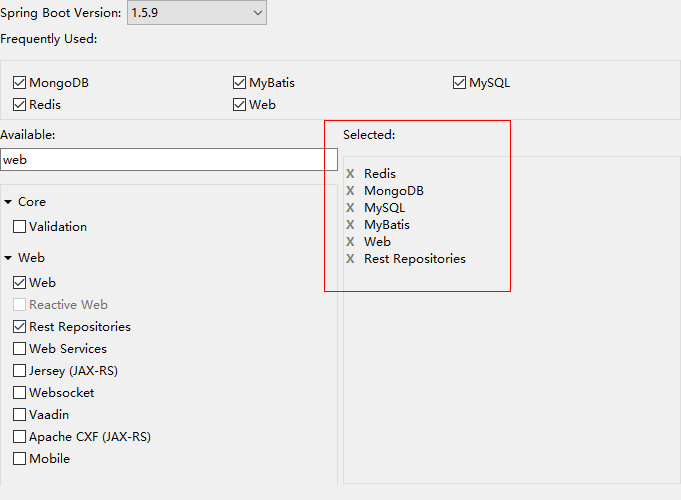
找到Spring目录，选择Spring Starter Project，Next



**3、填写跟项目相关的各种信息，然后Next：**



**4、选择需要的Dependency，然后Next：**



**5、Next，然后Finsh，新项目就创建好了，各个目录如下：**



**6、右键SpringBootDemoAutoApplication中的main方法，Run As -> Spring Boot App，项目就可以启动了。**

***package*** *cn.znv.MyDemoAuto;*

***import*** *org.springframework.boot.SpringApplication;*

***import*** *org.springframework.boot.autoconfigure.SpringBootApplication;*

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

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

*@RestController*

*@SpringBootApplication*

***public******class*** *SpringBootDemoAutoApplication {*

*@RequestMapping("/hello")*

***public*** *String ShowHello() {*

***return*** *"hello";*

*}*

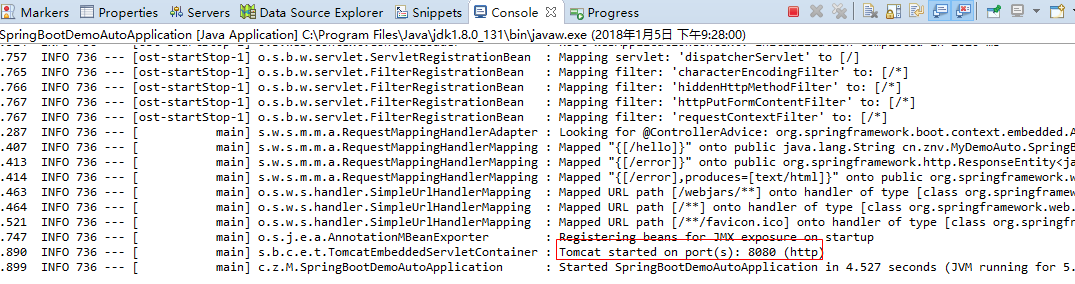
***public******static******void*** *main(String[] args) {*

*SpringApplication.run(SpringBootDemoAutoApplication.****class****, args);*

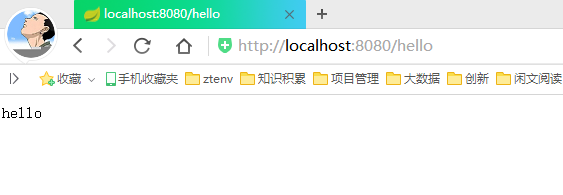
*}*

*}*

由于选择了web dependency默认启动一个Tomcat，8080端口监听



在IE中输入localhost:8080/hello



**7、在application.properties文件中，修改Tomcat的启动端口：**

server:

port: 8081

### 四、通过maven创建spring boot

通过上面的安装，我们已经简单实现了一个spring boot的web应用程序，springboot内置了tomcat，如果需要做发布版本，我们还需要一些步骤，后面在介绍，下面我们不通过插件的方式来进行springboot的环境搭建，采用maven的方式，有关maven的知识参考：

<https://www.w3cschool.cn/maven/j3x41ht2.html>

步骤如下：

1. 配置maven，由于maven默认下载源太慢，在pom中配置依赖jar包时，下载需要很久，这里介绍下eclipse如何加载maven自定义配置文件。

首先我们新建一个settings.xml文件，内容如下：

<?xml version="1.0"?>

<settings>

<localRepository>C:\Users\wangdenuan\.m2\repository</localRepository>

<!--需要改成自己的maven的本地仓库地址-->

<mirrors>

<mirror>

<id>alimaven</id>

<name>aliyun maven</name>

<url>http://maven.aliyun.com/nexus/content/groups/public/</url>

<mirrorOf>central</mirrorOf>

</mirror>

</mirrors>

<profiles>

<profile>

<id>nexus</id>

<repositories>

<repository>

<id>nexus</id>

<name>local private nexus</name>

<url>http://maven.oschina.net/content/groups/public/</url>

<releases>

<enabled>true</enabled>

</releases>

<snapshots>

<enabled>false</enabled>

</snapshots>

</repository>

</repositories>

<pluginRepositories>

<pluginRepository>

<id>nexus</id>

<name>local private nexus</name>

<url>http://maven.oschina.net/content/groups/public/</url>

<releases>

<enabled>true</enabled>

</releases>

<snapshots>

<enabled>false</enabled>

</snapshots>

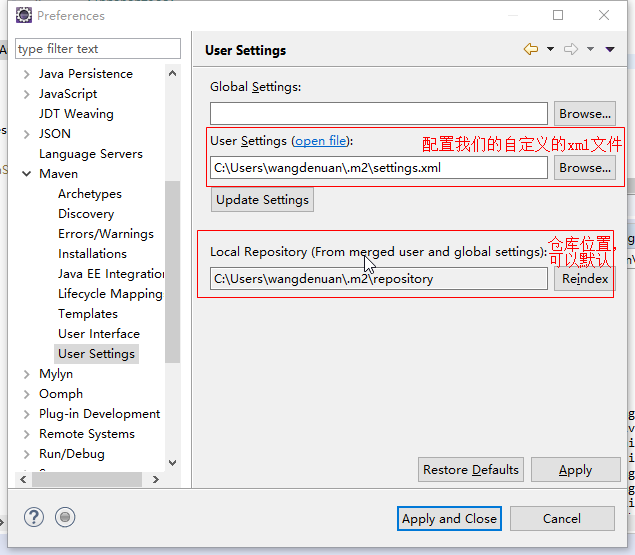
</pluginRepository>

</pluginRepositories>

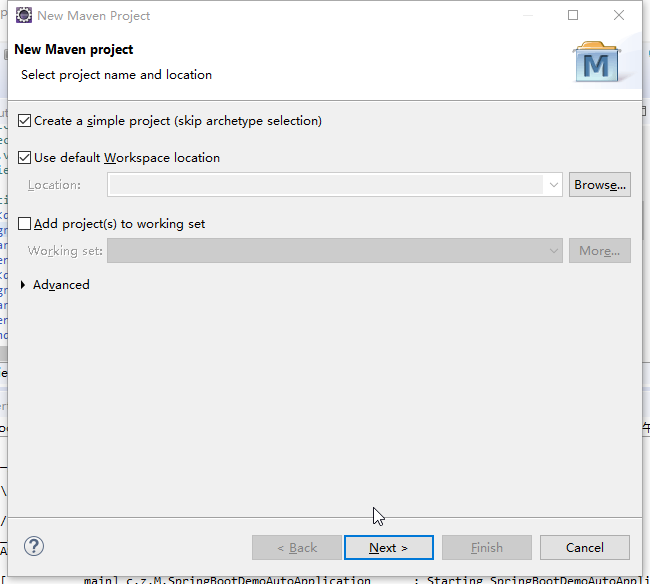
</profile></profiles>

</settings>

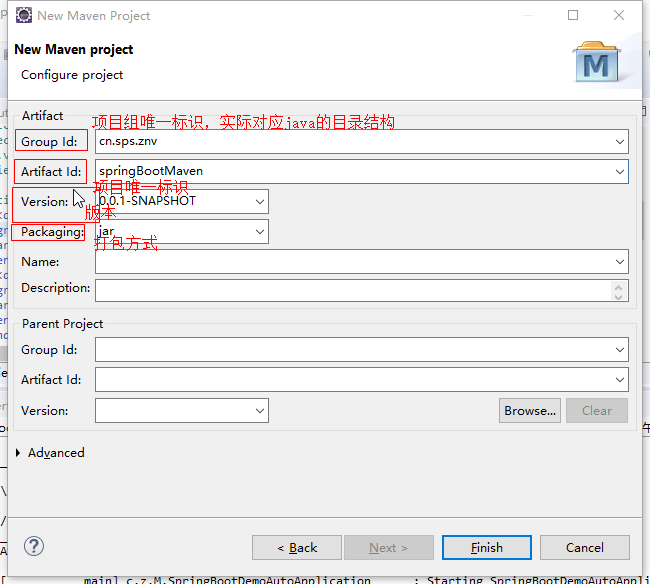
然后，打开eclipse，选择windows->preference,找到Maven->User Settings,配置如下：



1. 创建maven工程，选择file->New->Maven Project,打开New Maven Project对话框，配置如下，点击下一步：



1. 填写Maven工程相关的信息，如下图所示，点击finish

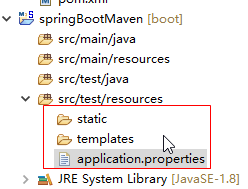


1. 配置pom.xml文件，导入SpringBoot的相关依赖，这里并没由自动产生需要在pom.xml中添加如下依赖：

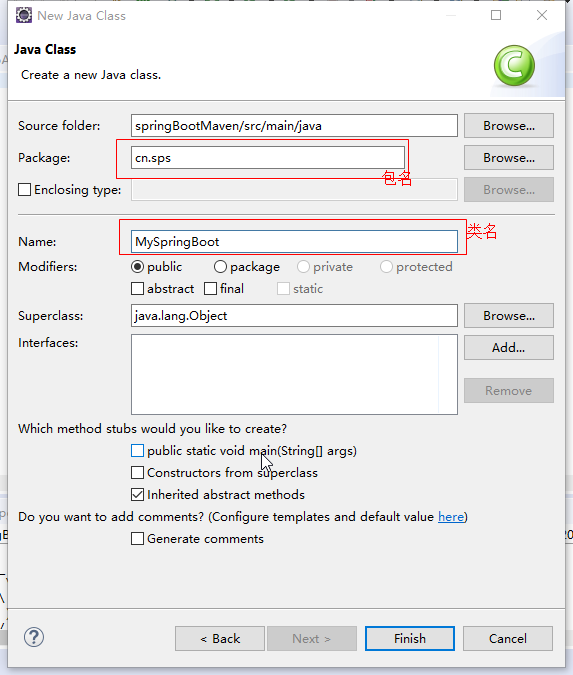
|  |
| --- |
| <project xmlns=*"http://maven.apache.org/POM/4.0.0"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*  xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"*>  <modelVersion>4.0.0</modelVersion>  <groupId>cn.sps.znv</groupId>  <artifactId>springBootMaven</artifactId>  <version>0.0.1-SNAPSHOT</version>  <parent>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-parent</artifactId>  <version>1.5.9.RELEASE</version>  <relativePath /> <!-- lookup parent from repository -->  </parent>  <properties>  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>  <java.version>1.8</java.version>  </properties>  <dependencies>  <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-web</artifactId>  </dependency>  </dependencies>  </project> |

这样就会自动加载springboot基础框架依赖的jar包了

1. 在资源文件夹下建立web应用所需要的资源文件夹，当然如果不需要页面，也可以不创建，这里创建三个目录，static、templates，application.propertys



6,创建应用程序入口类，在src/main/class上右击，new一个class，配置如下，点击finish：



在MySpringBoot.java中编写如下代码：

|  |
| --- |
| **package** cn.sps;  **import** org.springframework.boot.SpringApplication;  **import** org.springframework.boot.autoconfigure.SpringBootApplication;  //加入springboot注解,springboot通过注解找到入口  @SpringBootApplication  **public** **class** MySpringBoot {  //项目的入口  **public** **static** **void** main(String[] args) {  SpringApplication.*run*(MySpringBoot.**class**, args);    }  } |
|  |

此时就可以运行了，运行在终端窗口可以看到tomcat的启动信息，由于我们没有建立控制代码，所以还不能浏览器访问

### 五，创建用户控制代码