

# SpringBoot简介

# 课程目标

- SpringBoot简介
- Boot的Helloworld

# SpringBoot简介

# SpringBoot简介

- SpringBoot是由Pivotal团队在2013年开始研发、2014年4月发布第一个版本的全新开源的轻量级框架。
- 它基于Spring4.0设计，不仅继承了Spring框架原有的优秀特性，而且还通过简化配置来进一步简化了Spring应用的整个搭建和开发过程。
- 另外SpringBoot通过集成大量的框架使得依赖包的版本冲突，以及引用的不稳定性等问题得到了很好的解决。

□ **简单的说**SpringBoot并不是一个全新的框架,可以理解为Spring框架的一个优化升级,框架的运行和原理都是基于Spring的,所以在了解Spring前,最好能够系统的学习Spring框架.

# SpringBoot的特点

## □1) 独立运行的 Spring 项目

- Spring Boot 可以以 jar 包的形式独立运行，运行一个 Spring Boot 项目只需通过 `java-jar xx.jar` 来运行。

## □2) 内嵌 Servlet 容器

- Spring Boot 可选择内嵌 Tomcat、Jetty 或者 Undertow，这样我们无须以 war 包形式部署项目。

## □3) 提供 starter 简化 Maven 配置

- Spring 提供了一系列的 starter pom 来简化 Maven 的依赖加载，例如，当你使用了 `spring-boot-starter-web` 时，会自动加入如图 1 所示的依赖包。

## □4) 自动配置 Spring

- Spring Boot 会根据在类路径中的 jar 包、类，为 jar 包里的类自动配置 Bean，这样会极大地减少我们要使用的配置。当然，Spring Boot 只是考虑了大多数的开发场景，并不是所有的场景，若在实际开发中我们需要自动配置 Bean，而 Spring Boot 没有提供支持，则可以自定义自动配置。

## □5) 准生产的应用监控

- Spring Boot 提供基于 http、ssh、telnet 对运行时的项目进行监控。

## □6) 无代码生成和 xml 配置

- Spring Boot 的神奇的不是借助于代码生成来实现的，而是通过条件注解来实现的，这是 Spring 4.x 提供的新特性。Spring 4.x 提倡使用 Java 配置和注解配置组合，而 Spring Boot 不需要任何 xml 配置即可实现 Spring 的所有配置。

# 重要策略

□SpringBoot框架中还有两个非常重要的策略：

➤开箱即用

➤约定优于配置。



## □开箱即用:

- Outofbox, 是指在开发过程中, 通过在MAVEN项目的pom文件中添加相关依赖包, 然后使用对应注解来代替繁琐的XML配置文件以管理对象的生命周期。
- 这个特点使得开发人员摆脱了复杂的配置工作以及依赖的管理工作, 更加专注于业务逻辑

## □约定优于配置:

- Convention over configuration, 是一种由SpringBoot本身来配置目标结构, 由开发者在结构中添加信息的软件设计范式。

## □总结:

- 从最根本上来讲, Spring Boot就是一些类库的集合, 它能够被任意项目的构建系统所使用。
- Spring Boot 不是为了取代 Spring ,是为了让人们更容易的使用 Spring 。

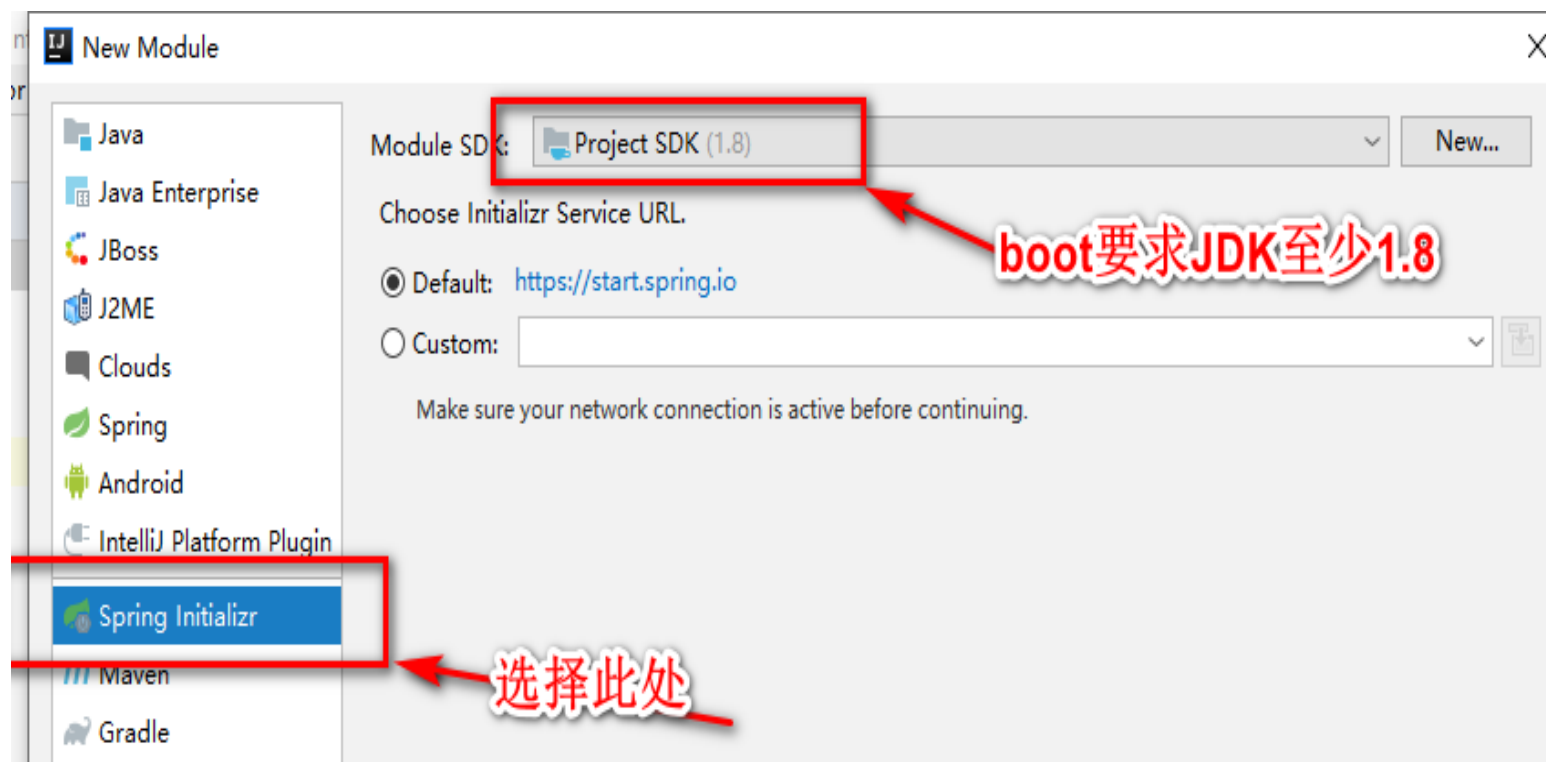
# Boot的Helloworld

# 环境

## □开发环境:

- IntelliJ IDEA 2018.3.4
- JDK1.8
- maven-4.0
- spring-boot-starter-parent 2.3.3

## □ 1. 创建Module



## New Module

### Project Metadata

Group:

mytest1

自定义

Artifact:

demo

Type:

Maven Project (Generate a Maven based project archive.)

Language:

Java

Packaging:

Jar

Java Version:

8

选择对应的java版本

Version:

0.0.1-SNAPSHOT

Name:

demo0


Description:

Demo project for Spring Boot

Package:

mytest1.demo0

## □ 选择如下:

 New Module ✕

**Dependencies**

Spring Boot 2.3.3 ▼

Developer Tools	<input checked="" type="checkbox"/> Spring Web
Web	<input type="checkbox"/> Spring Reactive Web
Template Engines	<input type="checkbox"/> Rest Repositories
Security	<input type="checkbox"/> Spring Session
SQL	<input type="checkbox"/> Rest Repositories HAL Explorer
NoSQL	<input type="checkbox"/> Rest Repositories HAL Browser
Messaging	<input type="checkbox"/> Spring HATEOAS


**Selected Dependencies**

Web

Spring Web ✕

Template Engines

Thymeleaf ✕

 New Module ✕

**Dependencies**

Spring Boot 2.3.3 ▼

Developer Tools	<input checked="" type="checkbox"/> Thymeleaf
Web	<input type="checkbox"/> Apache Freemarker
Template Engines	<input type="checkbox"/> Mustache
Security	<input type="checkbox"/> Groovy Templates
SQL	
NoSQL	
...	

**Selected Dependencies**

Web

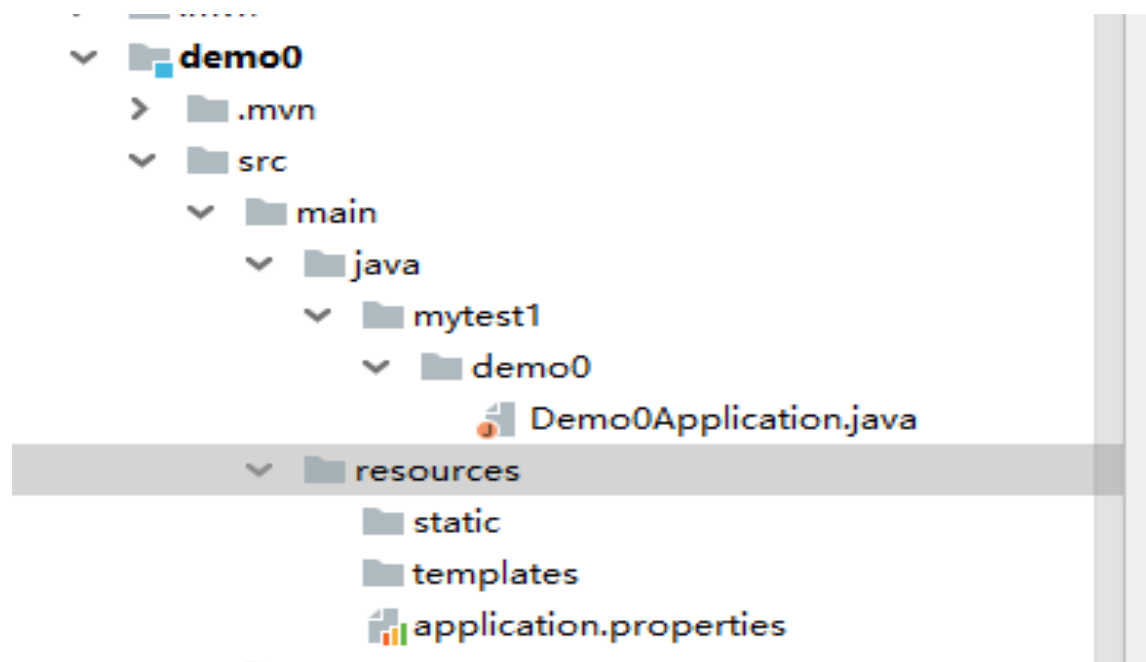
Spring Web ✕

Template Engines

Thymeleaf ✕

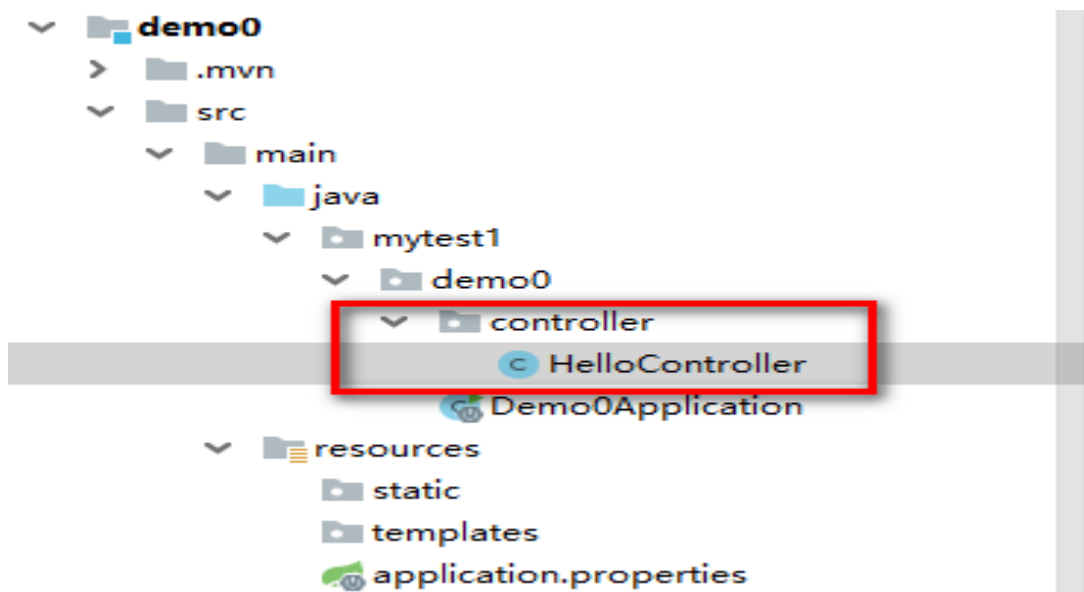


□创建完毕后结构如下:



## □2.添加package和class

➤注意:保证Demo0Application要和新添加的包在同一父目录中



### □3.在HelloController中输入如下内容

```
@RestController
public class HelloController {

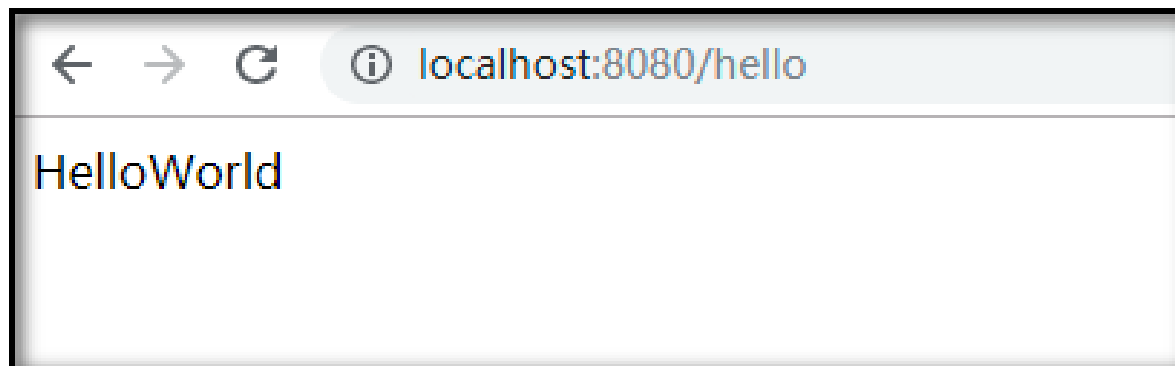
    @RequestMapping("hello")
    public String hello() {
        return "HelloWorld";
    }
}
```

□4.运行Demo0Application,无须修改任何配置

□5.打开浏览器并在地址栏输入,

➤ <http://localhost:8080/hello>

➤ 运行成功,页面如下



**谢谢！**