spring boot 初试，springboot入门，springboot helloworld例子

参考文章：https://www.cnblogs.com/demingblog/p/5478026.html

1.springboot是干啥的

做过Javaweb开发的程序员，肯定都知道spring框架。springboot就是一个能够快速搭起spring项目的神器。

2.为啥要用spring boot

在没有springboot之前，假如我们需要使用spring来搭建一个项目，比如说搭建一个spring mvc的项目。我们需要干啥：

假如说我们是使用Maven 那么步骤是这样的：

1.新建一个maven 项目，package 方式为war.

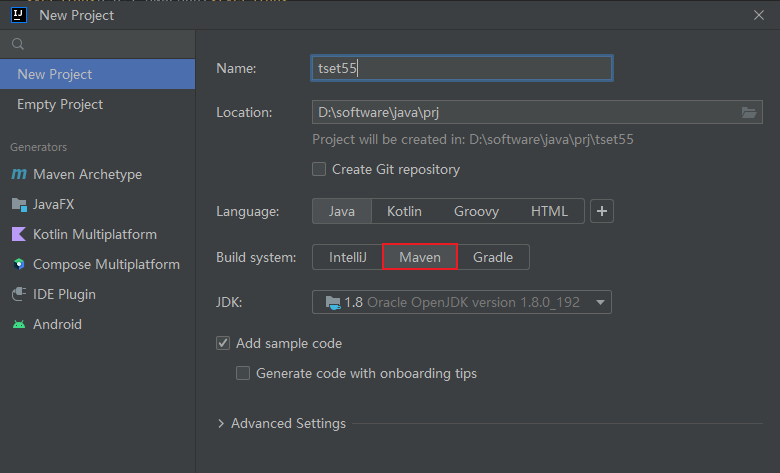
2.加入相关依赖，什么spring-bean,spring-context,spring-webmvc等等，在web.xml中配置spring listener，配置spring.xml之类的。

3.编译通过了，就可以运行项目了。

假如我们没有使用maven,就是单纯的建个web项目，然后到处copy jar包到webapp lib 下面，将会更麻烦，各种容易漏jar包。

所以，上面的方式，对于新手而言，可能一个小时也跑不起一个项目。而有了springboot之后可能只需要十几分钟。

1. 新建maven项目



2.在pom文件spring boot需要的依赖，配置文件如下：

看这个pom文件，dependencies 节点下面只配置了一个 spring-boot-starter-web 依赖，加了一个parent配置，实在是太简洁了。

vim pom.xml

<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>com.github</groupId>

<artifactId>springbootdemo</artifactId>

<version>0.0.1-SNAPSHOT</version>

<parent>

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

<artifactId>spring-boot-starter-parent</artifactId>

<version>2.1.8.RELEASE</version>

<relativePath/>

</parent>

<dependencies>

<dependency>

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

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

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

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

</build>

</project>

3.编写main方法

参考文章：https://blog.51cto.com/lavasoft/53263

参考文章：https://www.php.cn/java-article-439767.html

Java中的main方法，是一个java应用程序的入口

Vim DemoApplication.java

package org.example;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class DemoApplication {

public static void main(String[] args) throws Exception {

SpringApplication.run(DemoApplication.class, args);

//运行之后在浏览器中访问：http://localhost:8080/aello

}

}

4.编写action类，@controller注解action类，@Requestmapping映射请求地址。

package org.example;

Sping注解

参考文章：https://blog.csdn.net/weixin\_36039452/article/details/114185618

参考文章：https://blog.csdn.net/u011001843/article/details/53406661

参考文章：https://blog.csdn.net/MinggeQingchun/article/details/123586046

参考文章：https://blog.csdn.net/pan\_junbiao/article/details/103727318

参考文章：https://blog.csdn.net/huweiliyi/article/details/107758876

spring中的注解可以分为两大类：

1、springmvc的相关注解

2、spring的bean容器的相关注解；或者说bean工厂的相关注解

springmvc相关的注解有：@Controller, @RequestMapping, @RequestParam， @ResponseBody等等。

spring的bean容器相关的注解，先后有：@Required， @Autowired, @PostConstruct, @PreDestory，还有Spring3.0开始支持的JSR-330标准javax.inject.\*中的注解(@Inject, @Named, @Qualifier, @Provider, @Scope, @Singleton).

@Controller：表明这个类是一个控制器类。

@RequestMapping：

指定控制器可以处理哪些URL请求

请求路径映射，根据提供的映射信息确定请求所对应的处理方法，还可以限定请求方法、请求参数及请求头映射请求。

@controller：用于标注控制层组件，Controller控制器是通过服务接口定义的提供访问应用程序的一种行为。

@RequestMapping：请求路径映射，如果标注在某个controller的类级别上，则表明访问此类路径下的的方法都要加上其配置的路径，最常用的是标注在方法上，表明那个具体方法来接受处理某次请求。

@RequestMapping既可以作用在类级别，也可以作用在方法级别。当它定义在类级别时，标明该控制器处理所有的请求都被映射到/favsoft路径下。

@RequestMapping中可以使用method属性标记其所接受的方法类型，如果不指定方法类型的话，可以使用HTTP GET/POST方法请求数据，但是一旦指定方法类型，就只能使用该类型获取数据。

@RequestMapping可以使用@Validated与BindingResult联合验证输入的参数，在验证通过和失败的情况下，分别返回不同的视图。

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

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

@RestController

public class AelloController {

@RequestMapping("/aello")

String home() {

return "Hello ,spring boot!";

}

}