<http://www.cnblogs.com/chry/p/5876752.html>

1. 首先用eclipse创建一个maven工程, 普通maven工程即可

2. 修改pom如下：

[复制代码](javascript:void(0);)

<?xml version="1.0" encoding="UTF-8"?>

<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.chry</groupId>

<artifactId>studySpringBoot</artifactId>

<version>0.0.1-SNAPSHOT</version>

<properties>

<java.version>1.7</java.version>

</properties>

<!-- Inherit defaults from Spring Boot -->

<parent>

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

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

<version>1.4.0.RELEASE</version>

</parent>

<!-- Add typical dependencies for a web application -->

<dependencies>

<dependency>

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

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

</dependency>

</dependencies>

<!-- Package as an executable jar -->

<build>

<finalName>studySpringBoot</finalName>

</build>

</project>

[复制代码](javascript:void(0);)

3. 接着创建一个Java Class

[复制代码](javascript:void(0);)

1 package com.chry.study;

2

3 import org.springframework.boot.SpringApplication;

4 import org.springframework.boot.autoconfigure.EnableAutoConfiguration;

5 import org.springframework.stereotype.Controller;

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

7 import org.springframework.web.bind.annotation.ResponseBody;

8

9 @Controller

10 @EnableAutoConfiguration

11 public class SampleController {

12

13 @RequestMapping("/")

14 @ResponseBody

15 String home() {

16 return "Hello World!";

17 }

18

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

20 SpringApplication.run(SampleController.class, args);

21 }

22 }

[复制代码](javascript:void(0);)

4. 构件并以java application方式运行这个带main方法的类

5. 浏览器访问http://localhost:8080/，即可显示Hello World

=================================================

简单解释一下相关

**1. spring-boot-starter-parent：**

springboot官方推荐的maven管理工具，最简单的做法就是继承它。 spring-boot-starter-parent包含了以下信息：

1)、缺省使用java6编译, 如果需要改为java 1.7，则在pom中加上java.version属性即可

2)、使用utf-8编码

3)、实现了通用的测试框架 (JUnit, Hamcrest, Mockito).

4)、智能资源过滤

5)、智能的插件配置(exec plugin, surefire, Git commit ID, shade).

**2. spring-boot-starter-web**

springboot内嵌的WEB容器， 缺省会使用tomcat

**3. @Controller**

Java中的注解@Controller会启动一个Spring MVC的控制器，

**4. @EnableAutoConfiguration**

@EnableAutoConfiguration也是必须的，它用于开启自动配置

**5. @RequestMapping**

示例中该注解将”/"映射到处理Hello world 输出的home()方法

**6. @ResponseBody**

示例中该注解将home()的输出字符串“Hello world”直接作为http request的响应内容，而不是Spring MVC中常用的视图名字