# 极简Spring Boot

最近在看Spring Boot和Spring Cloud，准备把相关的技术点用代码实现起来，比如注册中心（Eureka）、网关（Zuul）、负载均衡（Ribbon）等。一步步来吧，写个系列文章记录一下。

给自己定的写作原则是注重实操，理论性的东西就不讲了，或者一带而过。

争取拿出干的不能再干的东西。

——老码农

## 需求

1. 使用Spring Boot搭建基于Restful风格的Http服务。
2. 接口1：get方法，url中包含姓名，返回“你好，{姓名}”等内容。
3. 接口2：post方法，接收返回JSON。

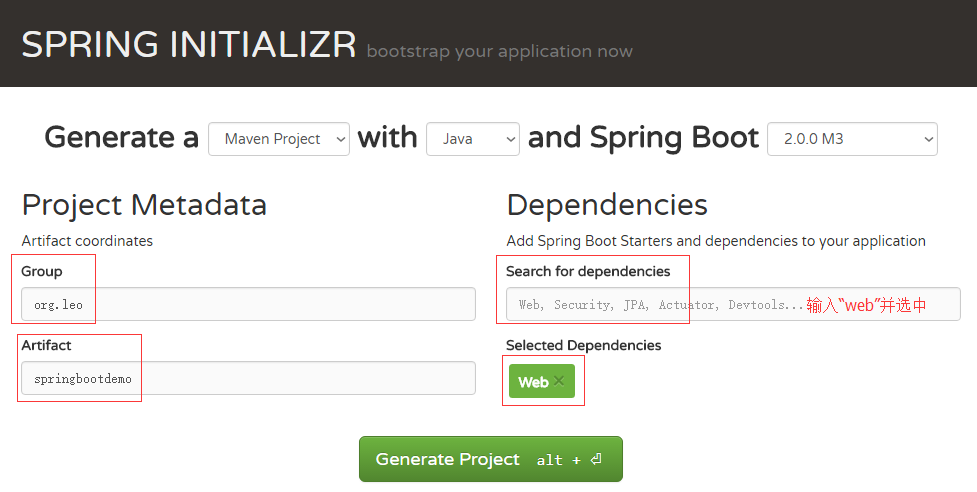
## 搭建Spring Boot工程

Spring Boot特点就是简化Spring工程的搭建、开发和部署。还记得我刚参加工作的时候，还没Maven，那一个个lib包的找啊，要命。

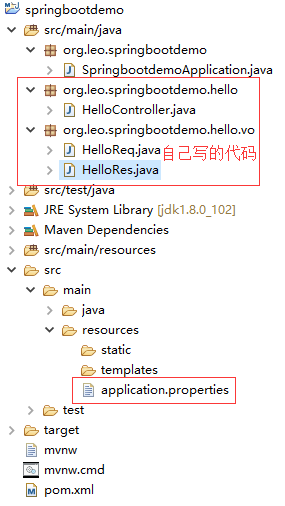
后来有Maven了，倒是省了下载，可还是要在pom里写一堆的dependency，大家可以翻翻自己工作中使用的框架里面的pom有多少行~~~

有了Spring boot就简单多了，以本Demo为例，因为我们只是简单地构建一个Http服务，所以如下操作即可：

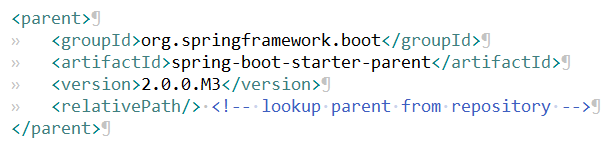
访问<http://start.spring.io>

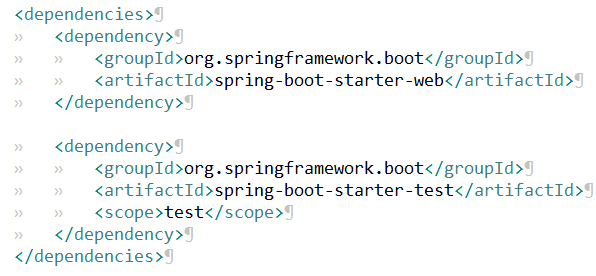


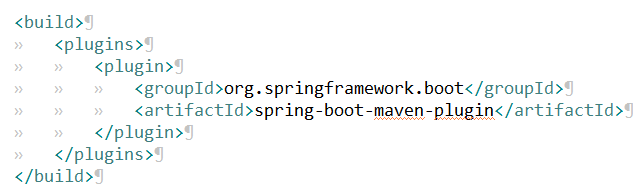
下载一个zip包，导入我们的Eclipse中，代码结构如下图：



打开pom.xml，着重的配置如下：







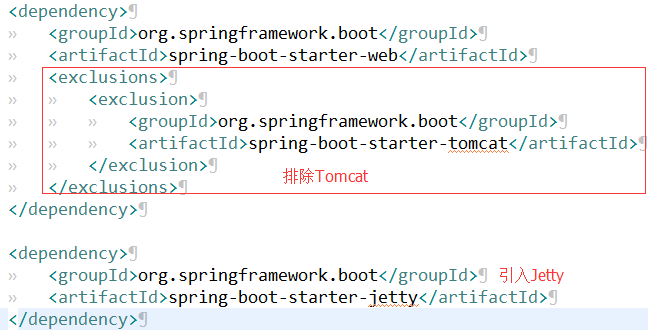
下面还有spring的镜像仓库，我们有自己的，删掉也行。

至此，搭建完毕，可以使用Maven打包发布了。

pom里面一共就配了一个parent，引入了两个dependency（其中test那个都可以不引用）。

自己手工搭建过Spring工程的都知道这有多简单多方便。

Spring Boot内嵌了Tomcat、Jetty容器，默认是Tomcat，如果想使用Jetty的话，pom做如下更改即可：



更改端口号。在resources目录下，有个application.properties文件，现在有许多教程使用的是application.yml文件，我们可以在里面修改端口号。

properties：



yml：



如图，会写properties，应该就会写yml。

用yml要注意冒号（:）后面要跟一个空格。

## 启动Spring Boot

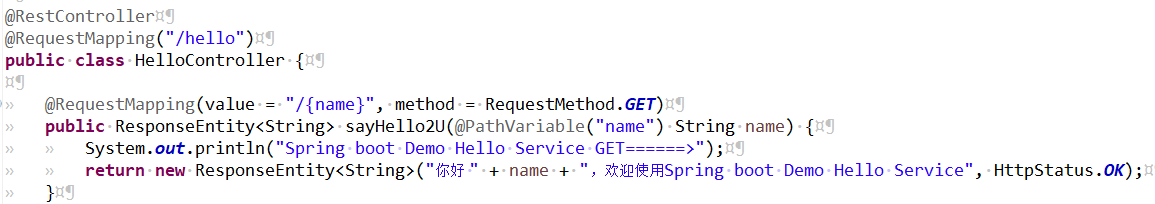
开发时，可以在Eclipse中直接启动SpringbootdemoApplication。

实际使用时，可以通过Maven打成jar、war。

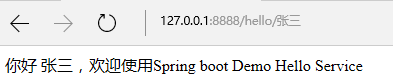
本例使用jar，在target目录中，使用“java -jar springbootdemo-0.0.1-SNAPSHOT.jar”命令。

## Get接口

创建HelloController，代码如下：



启动之后，结果如下：



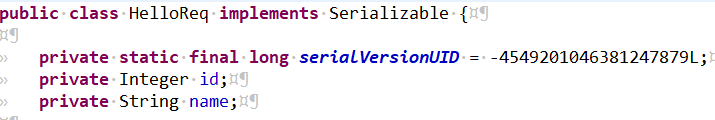
后台日志：



## Post接口

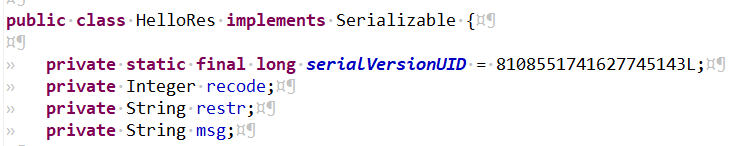
接收和返回的数据都是JSON格式，这样用起来最爽。

构建数据接收类，HelloReq：

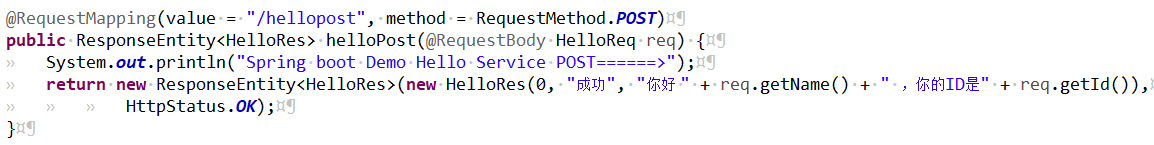


构造函数、Get/Set方法、toString方法省略。

构建数据返回类，HelloRes：

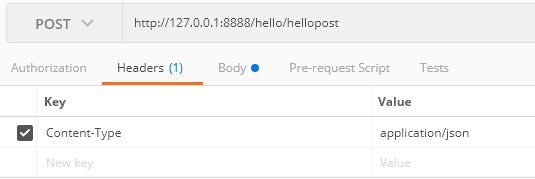


HelloController增加如下方法：

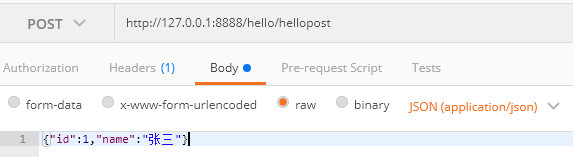


使用Postman等工具发送post请求：

设置头：



设置消息体：



结果如下：



后台日志：



至于put、delete接口就不写了，大同小异。