## 1.什么是Springboot？

SpringBoot是一个快速开发的框架，能过快速帮我们整合常用的第三方框架;

## 2.Springboot如何整合第三方框架的？

基于Maven的继承方式,完全采用注解化(使用注解的方式开启SpringMVC),简化XML配置，内置HTTP服务器（Tomcat，Jetty），最终以java应用程序进行执行；无需发布打包到服务器,直接通过Java命令即可执行(java -jar)

## 3.为什么使用Springboot？

**传统项目**：  
    a.整合SSH或SSM，考虑 配置文件，jar包冲突问题,整合起来繁琐;  
    打包方式：打成一个war包放入到tomacat的webapps目录下进行执行;  
**Springboot项目：**  
    a.Springboot 是一个快速开发框架，能够帮助我们快速整合第三方常用框架（Maven依赖关系（Maven的继承）），完全使用注解化，简化xml文件配置(在spring3.0之后采用注解方式启动Springmvc),内置嵌入了Http服务器（Tomcat,Jetty)，默认嵌入了Tomacat服务器;最终以java应用程序进行执行;使用Java创建Tomacat;Springboot项目没有web.xml;

## 4.Springboot与SpringMVC的关系？

1.Springboot web组件集成了SpringMvc框架;但是Springboot启动SpringMVc是没有传统的配置文件;

2.SpringBoot如何启动SpringMvc？

在SpringMVC3.0支持注解方式启动;

## 5. Springboot与SpringCould的关系？

1.Springboot是一个快速开发框架，可以快速帮助我们整理常用第三方框架；

完全采用注解化，简化xml文件配置，最终以java应用程序进行执行;

2.SpringCould是一套目前完整微服务解决框架;

注册中心，客户端调用工具，服务治理（负载均衡，断路器，分布式配置中心，网关，服务链路等），

3.关系：

Springboot是 微服务框架？

SpringBoot+SpringCould 实现微服务开发;

SpringBoot实现快速开发;

微服务通讯技术http+json(restful)轻量级Springboot Web组件默认集成了SpringMVC，SpringCould依赖于Springboot实现微服务, 使用SpringMVC编写微服务接口;

6. SpringCould和Springboot的区别?    
     a.微服务开发---》RPC远程通讯技术、服务治理;    
     核心区别：SpringCould微服务框架--》rpc远程调用  
             单纯的只是集成Springboot实现快速开发