**1. Spring Boot是什么，解决哪些问题？**

1) Spring Boot使编码变简单

2) Spring Boot使配置变简单

3) Spring Boot使部署变简单

4) Spring Boot使监控变简单

5) Spring Boot的不足

**2. Spring Boot在平台中的定位，相关技术如何融合？**

1) SpringBoot与SEDA +MicroService + RESTful

2) SpringBoot与Mock

**3. 采用了SpringBoot之后，技术管理应该如何进行？**

首先，我们来看一下spring boot是什么，它帮助我们解决了哪些问题：



**一、SpringBoot是啥？**

**SpringBoot是伴随着Spring4.0诞生的；**

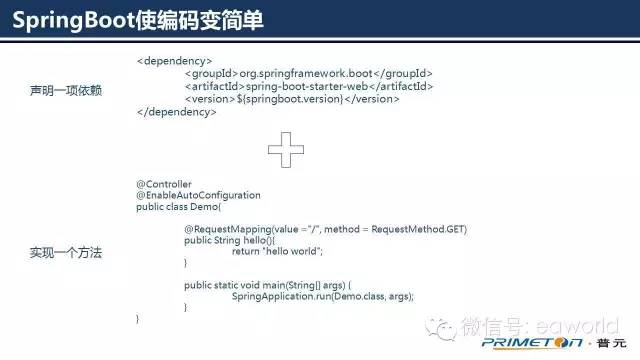
**从字面理解，Boot是引导的意思，因此SpringBoot帮助开发者快速搭建Spring框架；**

**SpringBoot帮助开发者快速启动一个Web容器；**

**SpringBoot继承了原有Spring框架的优秀基因；**

**SpringBoot简化了使用Spring的过程。**

**1) Spring Boot使编码变简单**

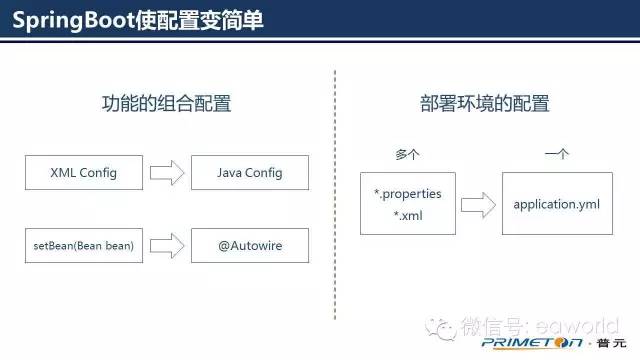


**1.声明一个你想要的依赖，就可以直接注解使用了。**

**Spring由于其繁琐的配置，一度被人认为“配置地狱”，各种XML、Annotation配置，让人眼花缭乱，而且如果出错了也很难找出原因。**

**2) Spring Boot使配置变简单**

Spring Boot更多的是采用Java Config的方式，对Spring进行配置。



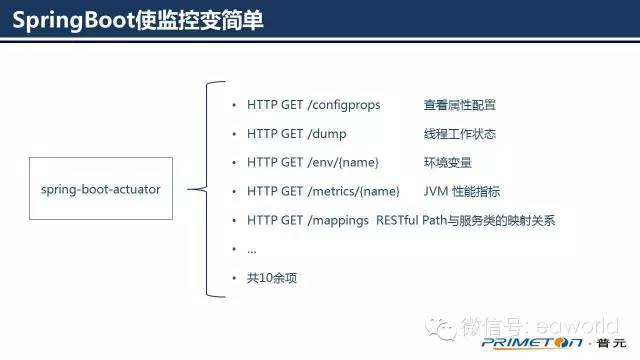
**2.配置（如redis阿，然后还有mysql阿），只需要一个yml直接搞定。**

**3) Spring Boot使部署变简单**



**3.比如tomcat之类的都不用自己配置了。**

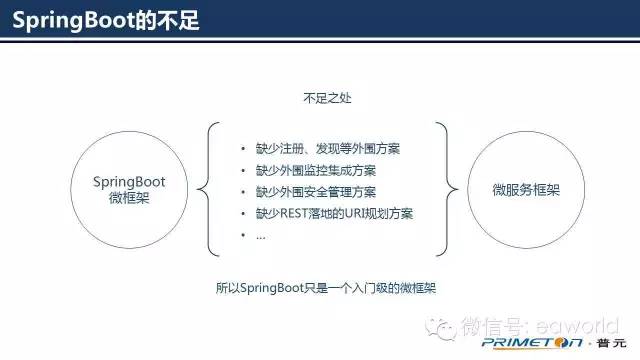
**4) Spring Boot使监控变简单**



可以看到，采用了spring-boot-start-actuator之后，直接以REST的方式，获取进程的运行期性能参数。

当然这些metrics有些是有敏感数据的，spring-boot-start-actuator为此提供了一些Basic Authentication认证的方案，这些方案在实际应用过程中也是不足的。

**5) Spring Boot的不足**



Spring Boot作为一个微框架，离**微服务的实现还是有距离的。**

**没有提供相应的服务发现和注册等配套功能。**

安全性：

**SpringBoot自身对于前面提到的配置文件：“application.yml”提供了多个「Profile」，可以便于开发者描述不同环境的配置，这些配置例如数据库的连接地址、用户名和密码。**



1. 丰富的工具链为SpringBoot的推广带来了利好。
2. SpringBoot的工具链主要来自于两个方面：

1) 原有Spring积累的工具链；

2) SpringMVC或者其他REST框架使用HTTP协议，使得HTTP丰富的工具成为SpringBoot天然的资源。

以上是我分享的主要内容，下面我们总结一下：

