# =====springboot=====

## springboot启动过程中做了哪些事情？

<https://www.jianshu.com/p/dc12081b3598>

## 注解 @spring boot Application说明？

1. Springboot如何判断当前应用是否时web应用？
2. <https://www.jianshu.com/p/dc12081b3598>

4、Spring boot整合jsp的流程，需要注意哪些点

# =====**SpringCloud**=====

1、SpringCloud和dubbo的区别

2、项目中用到了哪些组件

3、eureka的原理，如何保证高可用性，和Zookeeper有什么区别

4、feign如何调用的

5、处理生产环境上配置生效问题

6、hystrix的降级策略有哪些

7、Springcloud eureka是如何注册服务、如何监测心跳的，它注册的流程是怎么样的

8、在分布式环境中如何快速发现某一台服务有问题

9、分布式集群系统对外提供接口的时候如何验证对方的身份

## 10、Eureka和zookeeper作为注册中心有什么区别

<https://blog.csdn.net/weixin_33725126/article/details/94308610?utm_medium=distribute.pc_relevant.none-task-blog-BlogCommendFromMachineLearnPai2-2.nonecase&depth_1-utm_source=distribute.pc_relevant.none-task-blog-BlogCommendFromMachineLearnPai2-2.nonecase>

1.ZooKeeper保证的是CP,Eureka保证的是AP

ZooKeeper在选举期间注册服务瘫痪,虽然服务最终会恢复,但是选举期间不可用的  
Eureka各个节点是平等关系,只要有一台Eureka就可以保证服务可用,而查询到的数据并不是最新的自我保护机制会导致

Eureka不再从注册列表移除因长时间没收到心跳而应该过期的服务  
Eureka仍然能够接受新服务的注册和查询请求,但是不会被同步到其他节点(高可用)  
当网络稳定时,当前实例新的注册信息会被同步到其他节点中(最终一致性)  
Eureka可以很好的应对因网络故障导致部分节点失去联系的情况,而不会像ZooKeeper一样使得整个注册系统瘫痪

2.ZooKeeper有Leader和Follower角色,Eureka各个节点平等  
3.ZooKeeper采用过半数存活原则,Eureka采用自我保护机制解决分区问题  
4.Eureka本质上是一个工程,而ZooKeeper只是一个进程