# Spring Cloud 预备知识

面向切面编程（AOP）完善spring的依赖注入（DI），面向切面编程在spring中主要表现为两个方面   
1.面向切面编程提供声明式事务管理   
2.spring支持用户自定义的切面   
  
面向切面编程（aop）是对面向对象编程（oop）的补充，   
面向对象编程将程序分解成各个层次的对象，面向切面编程将程序运行过程分解成各个切面。   
AOP从程序运行角度考虑程序的结构，提取业务处理过程的切面，oop是静态的抽象，aop是动态的抽象，   
是对应用执行过程中的步骤进行抽象，，从而获得步骤之间的逻辑划分。   
  
aop框架具有的两个特征：   
1.各个步骤之间的良好隔离性    
2.源代码无关性

## 设计模式

### 发布/订阅模式

### 事件/监听模式

## Spring事件监听

org.springframework.context.ApplicationEvent:应用事件

org.springframework.context.ApplicationListener:应用监听器

@FunctionalInterface  
public interface ApplicationListener<E extends ApplicationEvent> extends EventListener {  
/\*\*  
\* Handle an application event.  
\* @param event the event to respond to  
\*/  
void onApplicationEvent(E event);  
}

# Spring Cloud Config Client

## Spring Boot事件/监听

## Spring Cloud事件/监听

### 每个Bean都有一个代理

### 所有Bean共享一个代理基类

## ApplicationContext层次性

### READ\_UNCOMMITTED

## EnvironmentEndpoint

### PROPAGATION\_REQUIRED

# 服务注册与发现

# 客户端负载均衡

# Spring Cloud 服务熔断