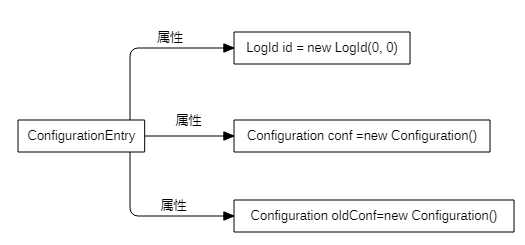
# 选举

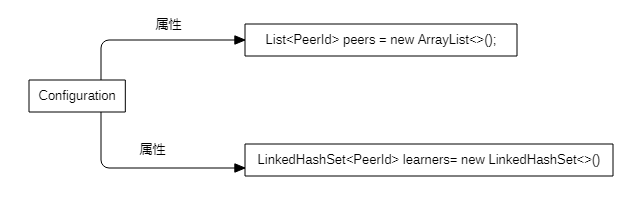
## ConfigurationEntry分析



1. boolean isStable()  
   1.1 功能：判断属性oldConf是否为空
2. boolean isEmpty()  
   2.1 功能：判断属性conf是否为空
3. Set<PeerId> listPeers()  
   3.1 功能：将属性conf和oldConf放入Set集合进行去重
4. boolean isValid()  
   4.1 功能：  
    （1）如果属性conf包含learners集合中元素，则返回false

（2）如果属性conf和属性oldConf包含learners集合中元素，则返回false

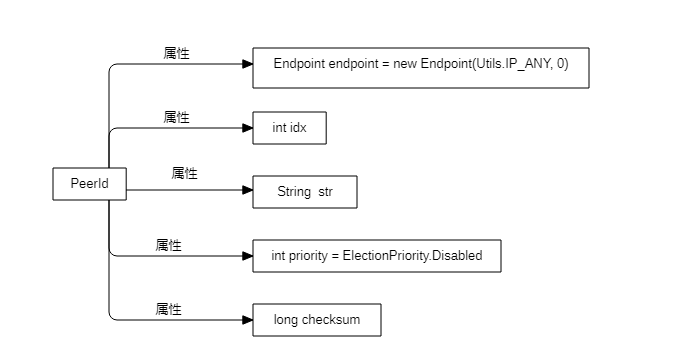
## Configuration类分析



1. boolean parse(final String conf)  
 1.1 参数conf：多个配置用逗号分割  
 1.2 功能：  
 （1）如果Learner，则配置示例为：localhost:8081/learner  
 🡪添加到learners属性中  
 （2）如果不是Learner，则添加到peers属性中  
 🡪最终解析到PeerId#parse(final String s)

2. boolean isValid()  
2.1 功能：返回peers里是否包含learners集合中元素

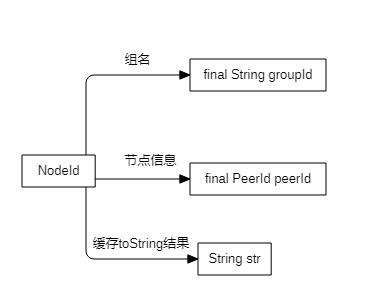
## PeerId类分析



Boolean parse(final String s)

1.1 支持的参数格式  
 （1）a:b  
 （2）a:b:c  
 （3）a:b::d  
 （4）a:b:c:d（ip:port:idx:priority）  
 1.2 功能：将参数分别解析到endpoint、idx、priority属性中

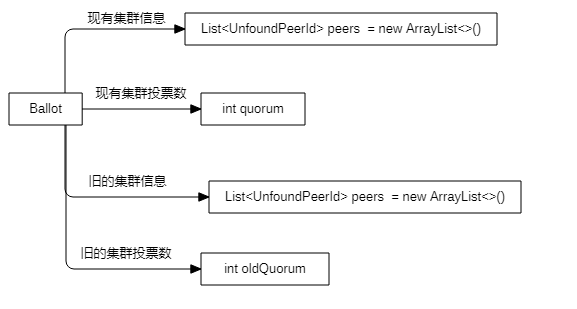
## NodeId类分析



## NodeImpl类分析

1. 构造方法：NodeImpl(final String groupId,final PeerId serverId)  
   1.1 初始化属性：groupId、serverId、state、currTerm

## Ballot类分析（投票的选票）



# RPC

## AbstractClientService(详情见mdj文件中的RPC-AbstractClientService)

### Init(final RpcOptions)初始化属性：rpcClient、rpcExecutor、rpcOptions

# SPI

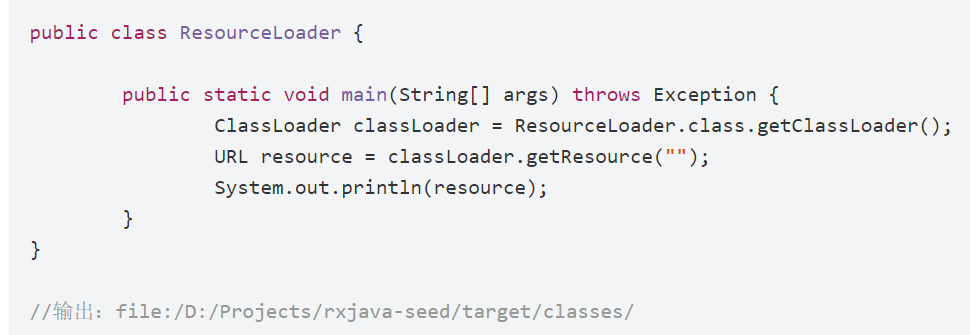
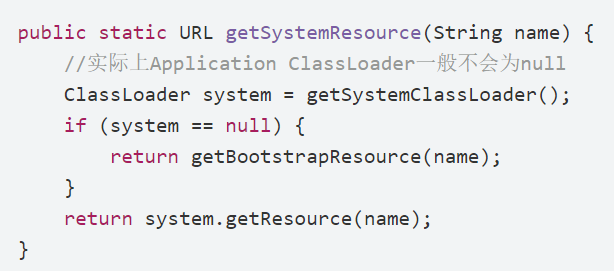
## SPI介绍

1. SPI是JDK内置的一种服务提供发现机制
2. SPI是一种动态替换发现的机制

## 类加载机制

<https://www.cnblogs.com/throwable/p/9785944.html>

### ClassLoader提供的资源加载API

1. ClassLoader#getResource(String name)方法中查找的路径  
   🡪是基于用户程序的ClassPath搜索资源
   1. 验证方法（输出结果就是当前应用的ClassPath）  
      
2. ClassLoader#getSystemResource(String name)  
   2.1 如果应用程序类加载器为null，则使用启动类加载器进行资源加载  
    🡪 如果应用程序类加载器不为null的情况下，实际上退化为ClassLoader#getResource

### 线程上下文类加载器Thread Context ClassLoader

1. 双亲委派缺陷

1.1 无法解决基础类需要回调用户的代码  
 🡪典型例子：JNDI  
 （1）JNDI的类库代码是启动类加载器加载的  
 （2）需要调用在应用的ClassPath的JNDI的SPI的代码  
 🡪 启动类加载器无法加载ClassPath下的类库

1. 这个类加载器可以通过java.lang.Thread类的setContextClassLoader()设置  
   🡪 JNDI服务可以使用线程上下文类加载器去加载所需的SPI类库