Zookeeper实现分布式锁需要满足一下条件

1、在分布式系统环境下，一个方法在同一时间只能被一个机器的一个线程执行；

2、高可用的获取锁与释放锁；

3、高性能的获取锁与释放锁；

4、具备可重入特性；

5、具备锁失效机制，防止死锁；

6、具备非阻塞锁特性，即没有获取到锁将直接返回获取锁失败。

这些条件全部满足需要考虑的问题很多，有没有现成的框架呢。

有，那就是curator，让我们一起来及时用一下吧

1、安装zookeeper

2、引入curator

引入依赖

<dependency>  
 <groupId>org.apache.zookeeper</groupId>  
 <artifactId>zookeeper</artifactId>  
 <version>3.6.1</version>  
</dependency>  
  
<dependency>  
 <groupId>org.apache.curator</groupId>  
 <artifactId>curator-client</artifactId>  
 <version>4.1.0</version>  
</dependency>  
<dependency>  
 <groupId>org.apache.curator</groupId>  
 <artifactId>curator-framework</artifactId>  
 <version>4.2.0</version>  
</dependency>  
<dependency>  
 <groupId>org.apache.curator</groupId>  
 <artifactId>curator-recipes</artifactId>  
 <version>4.2.0</version>  
</dependency>

创建锁工厂

@Configuration  
@Slf4j  
public class CuratorConfiguration {  
 @Value("${curator.retryCount}")  
 private int retryCount;  
  
 @Value("${curator.elapsedTimeMs}")  
 private int elapsedTimeMs;  
  
 @Value("${curator.connectString}")  
 private String connectString;  
  
 @Value("${curator.sessionTimeoutMs}")  
 private int sessionTimeoutMs;  
  
 @Value("${curator.connectionTimeoutMs}")  
 private int connectionTimeoutMs;  
  
 @Bean(initMethod = "start")  
 public CuratorFramework curatorFramework() {  
 CuratorFramework framework= CuratorFrameworkFactory.newClient(  
 connectString,  
 sessionTimeoutMs,  
 connectionTimeoutMs,  
 new RetryNTimes(retryCount, elapsedTimeMs));  
 return framework;  
 }  
}

在yml中添加参数