

# 问题日志

狼星:20:27:45

源码在哪下载吧

答: <https://github.com/spring-cloud/spring-cloud-netflix>

Jazq:21:07:16

Git 访问慢, 13.250.177.223 github.com 配这个 hosts 试试

张珂:21:25:08

什么时候会发生微服务下架, 如果我直接把服务停掉, 应该不会触发下架的逻辑

答: 直接停掉不会马上触发, 下架可以 `actuator` 发送下架请求, 也可以平滑上下架

ye17186:21:27:10

这是设置成 `down`, 什么时候从注册表删除呢, 还是说一直不会删除 `eureka` 页面上显示这个实例 `down`, 什么时候页面上看不到这个实例

答: 提交请求就会删除, 是服务端做的

Reythor 雷:21:38:51

服务端的分析应该从哪里入手呢

答: starter

金华:21:39:29

老师, spring cloud yml 中这么多配置信息, 记住这些 key, 有木有啥规律 or 技巧 ?

答: 多用, 看源码, 有的是有默认值的, 不用急, 记常用的就行

维浩:22:28:16

心跳不等待 register 成功就发送, 心跳都是这么用的吗?

答: 心跳是一个任务, 多线程跑的, 不会等待

ye17186:22:47:30

这里的 read 锁, 是哪个类

答: 单独的锁 lock 类

李东雪:22:52:03

不太明白先删除后插入这个操作, 这样应该会报错吧?

```

    /**
     *
     */
    public void register(InstanceInfo registrant, int leaseDuration, boolean isReplication)
    try {
        read.lock();
        Map<String, Lease<InstanceInfo>> gMap = registry.get(registrant.getAppName());
        REGISTER.increment(isReplication);
        if (gMap == null) { 删除了这里为null,不会报错
            final ConcurrentHashMap<String, Lease<InstanceInfo>> gNewMap = new Concurrent
            gMap = registry.putIfAbsent(registrant.getAppName(), gNewMap);
            if (gMap == null) {
                gMap = gNewMap;
            }
        }
    }
}

```

珠峰:22:54:15

相当于通过读锁来规避不操作同一条数据，各操作各的数据；互不干涉；