

SpringCloud Eureka 源码分析高级

1.分析了EnableEurekaServer 启动注解

- 1)、Eureka 注解他只是去加载了 Maker 对象，并没有做任何其他事情
- 2)、由于springCloud 的包中Meta-inf 下有spring-facotrys 配置，同时头上还挂着SpringBootApplication注解所以，此时就会加载spring-facotrys 的自动装配的类 EurekaServerAutoConfiguration
- 3)、初始化了 jersey的过滤器
 - a)、注意这个框架是用来处理eureka 中进行的http数据交互 jersey 的框架的 过滤器的初始化
 - b)、这个框架是及其类似于springmvc 框架 -> 也有类似于controller 的东西， ApplicationResource 类
- 4)、核心源码中 两行代码
 - a)、handleRegistration(info, resolveInstanceLeaseDuration(info), isReplication) springboot里边的监听器里边讲解过 你可以去进行监听 -> 监听对象
 - b) 这一行代码 只是去发布一个监听器对象
 - c) super.register(info, isReplication);
 - 1.首先获得心跳续约时间，如果你自己配置的话，会覆盖
 2. super.register(info, isReplication,加上心跳续约时间传递过下一个方法); 具体的注册逻辑
 3. replicateToPeers(Action.Register, info.getAppName(), info.getId(), info, null, isReplication); 进行eureka集群的续约
- 5)、具体的注册逻辑
 - a)、加了一把读锁 -> 读取数据
 - b)、从 注册资源表中拿出来 子层map，进行了子层map的初始化
 - c)、通过子map 拿到续约列表对象，取出当前操作的续约对象
 - d)、取了两个时间

已存在注册对象的 最后续约时间

当前时间

 - e)、进行比较，如果已存再的时间大于 正在注册的时间，说明在当前注册的过程中，已经有人提前把这个服务进行了注册，此时进行的操作就是用 已存再的时间去覆盖新注册时间，相当于是 使用原来那个注册信息
 - 6)、当注册完成后

a)、将数据立刻同步到readWrite 缓存中，这个缓存在定时器线程的作用下将数据刷新到我 readOnly缓存中中，我们的消费者就从readOnly缓存中欧拉取数据，又由于定时器并不是时时的，所以时有发生，拉取不到最新的信息，当一个服务不好使了之后，eureka 要对他进行删除，实际上，我们的坏掉的client，在readWrite缓存中进行全部删除，当定时器扫描到 readOnly和 readWrite不一样时，才会将数据再次加载到 readOnly缓存中，此时消费者才能拉到具体的这些信息

万一这台服务器宕机了，而order服务一直调用该服务器，或者是服务器切换了地址，那么order服务的调用逻辑代码哦也需要发生改变



