

SpringCloud 系列 --Eureka 常用配置

作者: [289306290](#)

原文链接: <https://ld246.com/article/1583838705893>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

1. 心跳检测配置

客户端的实例会向服务器发送周期性的心跳，默认30s一次，可以通过修改`eureka.instance.leaseRenewIntervalInSeconds`属性来改变这个时间。

服务端接收心跳请求，如果在一定期限内没有接收到服务实例的心跳，那么会将该实例从注册表中清除，其他的客户端无法访问这个实例。这个期限默认值是90s，可以通过修改客户端的 `eureka.instance.leaseExpirationDurationInSeconds`属性来改变这个值，需要注意清理注册表是定时器执行(默认每隔6s一次)，如果将 `leaseExpirationDurationInSeconds`设置为小于60s，虽然符合删除的条件，但是定时还没到时间执行，所以这个实例仍然存在于注册表，可以修改服务器端定时清理注册表的时间间隔 `eureka.server.eviction-interval-timer-in-ms` 单位是毫秒。

另外，如果开启了自我保护模式，则实例不会被踢除，在测试时，为避免自我保护模式的干扰，可以先关自我保护 `eureka.server.enable-self-preservation=false`

2. 注册表抓取间隔

默认情况下，客户端每隔30s去服务器端抓取注册表(可用的服务列表)，并且将服务器端的注册表保存在内存中，可以通过修改`eureka.client.registryFetch-IntervalSeconds`配置来改变注册表抓取间隔，但仍需考虑性能。

3. 配置与使用元数据

框架自带的元数据包括实例id, 主机名称, ip地址, 如果需要自定义元数据,并提供给其他客户端使用,可配置`eureka.instance.metadata-map`属性来指定,例如:

eureka :

instance:

hostname: localhost

metadata-map:

company-name: crazyit

配置了一个名为company-name的元数据，值为crazyit.使用的一方可以调用`discoveryClient`的方法取元数据，

代码如下:

```
@Autowired
private DiscoveryClient discoveryClient;

@RequestMapping(value = "/router", method = RequestMethod.GET)
public String router() {
    // 查询服务实例
    List<ServiceInstance> ins = discoveryClient.getInstances ("heart-beat-client");
    // 遍历实例并输出元数据值
    for(ServiceInstance service : ins) {
        System.out.println(service.getMetadata().get ("company-name"));
    }
    return "";
}
```

4.自我保护模式

之前文章中的红色字体提醒意味着eureka进入了自我保护模式，客户端发送给服务器端的心跳失败率过一定比例，服务会将这些实例保护起来，并不会马上剔除，此时另外的客户端可能会拿到一些无法用的实例，这种情况可能导致灾难的"蔓延"，可以使用容错机制解决，后续讲解，可以关闭自我保护式，当时关闭自我保护后出现网络或者其他问题时候，将不会保护过期的实例。