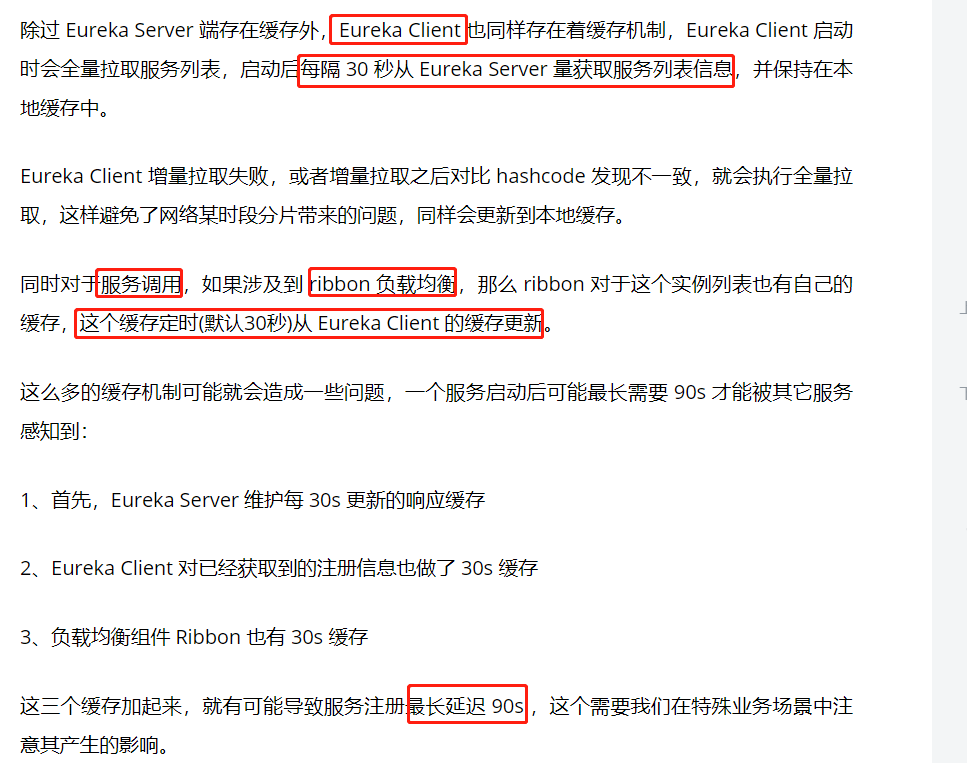
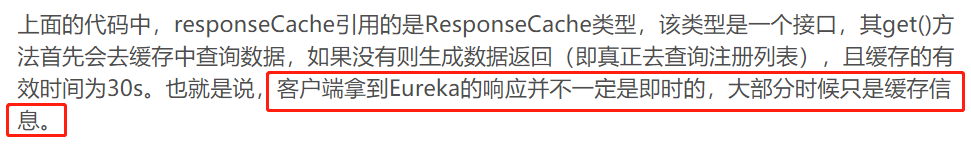


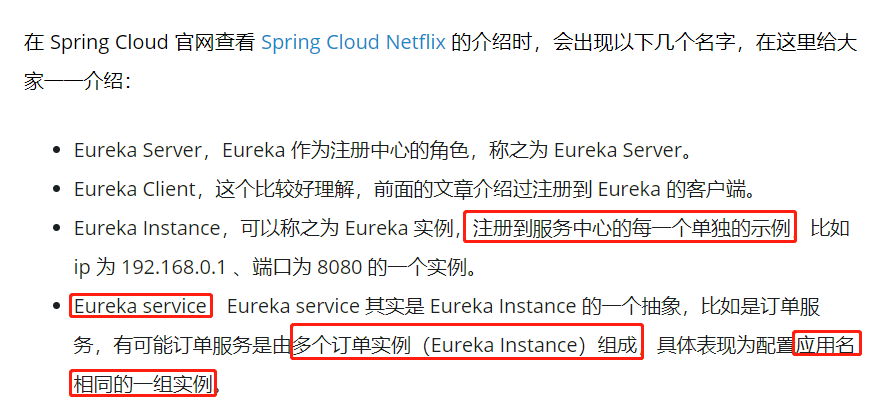
存储层是包装之后的从map扩展出来



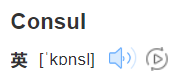
数据想到缓存层-只读缓存拿，没有缓存层-读写缓存拿，没有存储层同步到缓存层再拿

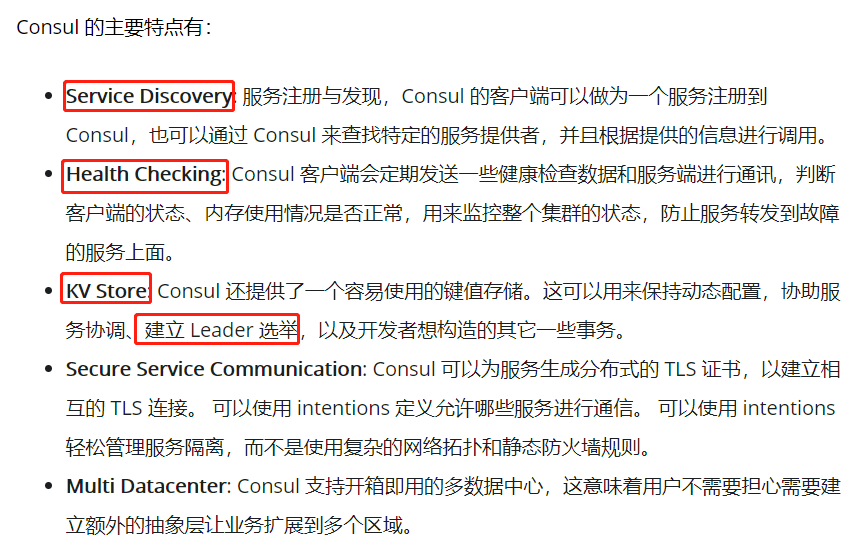






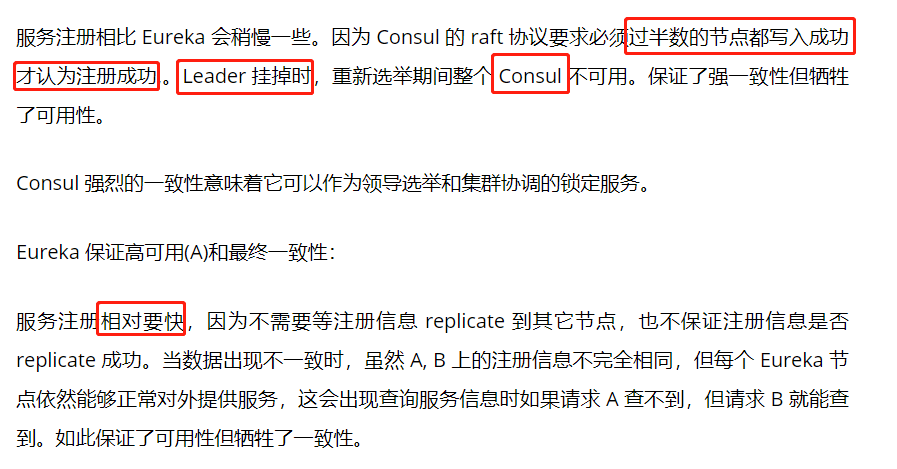
实例就是应用名相同





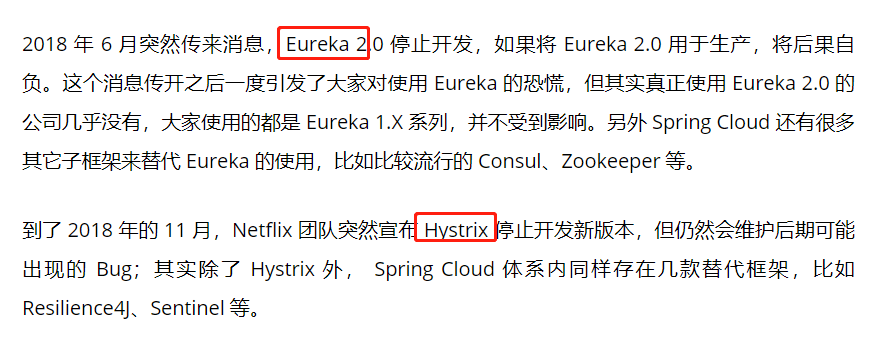




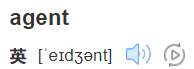


强一致性-》最终一致性









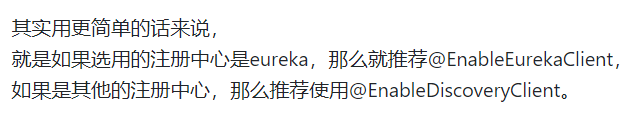


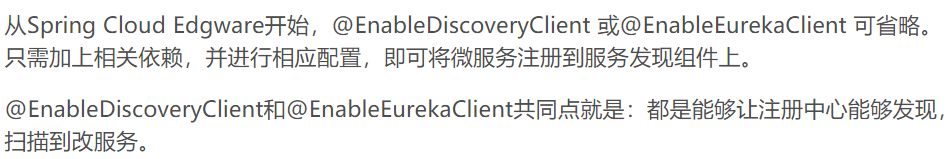
引入依赖，配置文件配置

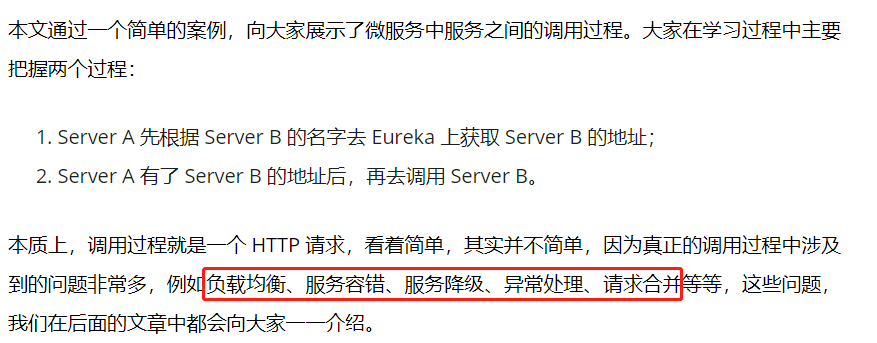


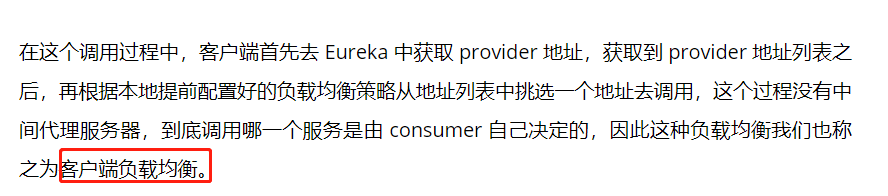
Consul

Eureka 都是EnableDiscoveryClient

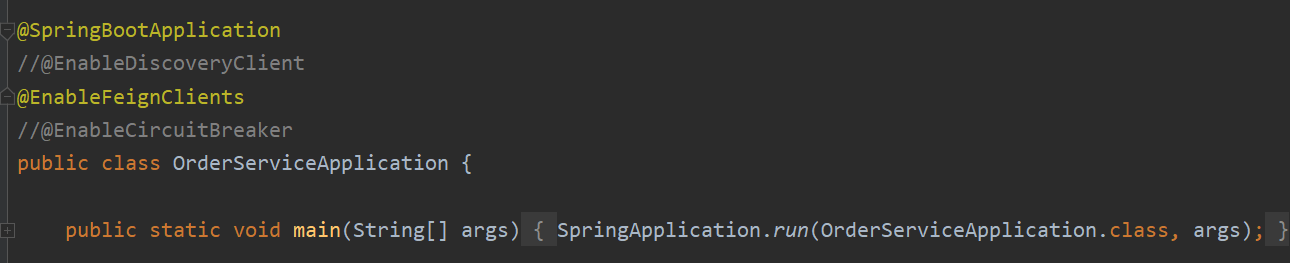






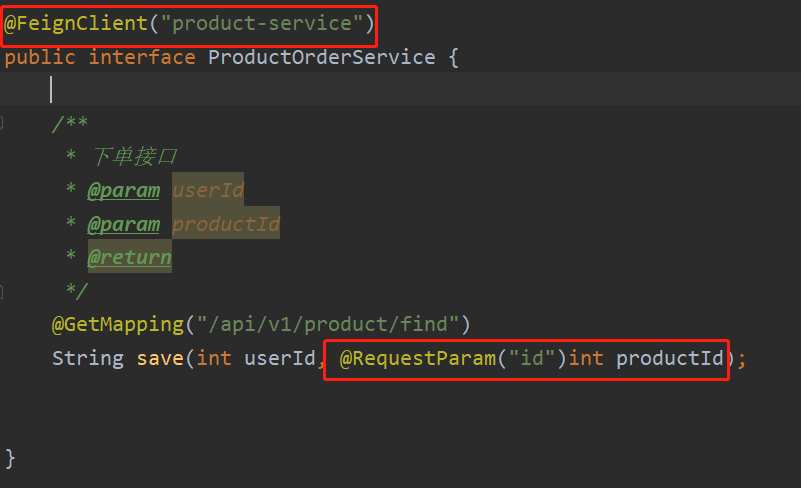


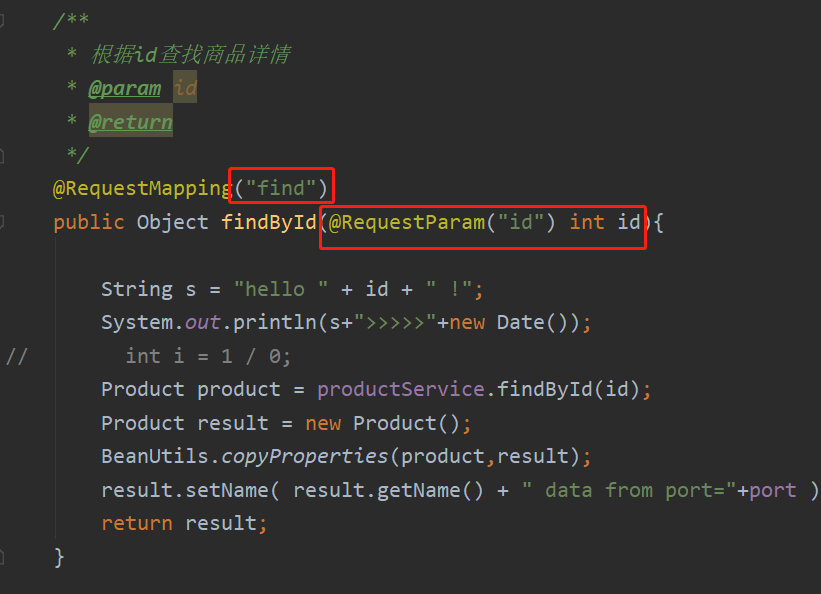






一定要加@RequestParam





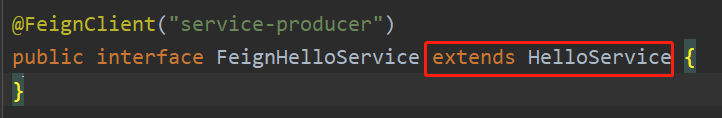




需要加上相应的注释，并且写上参数名称

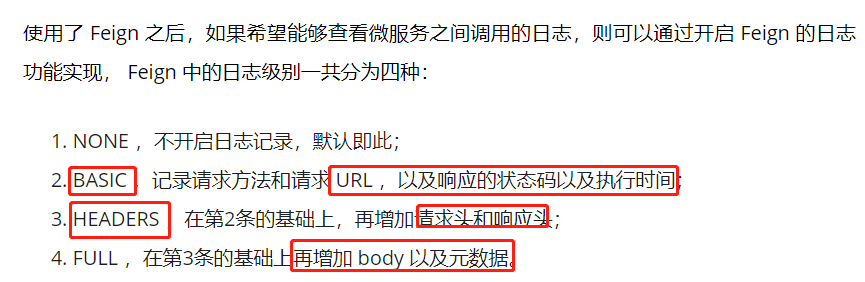






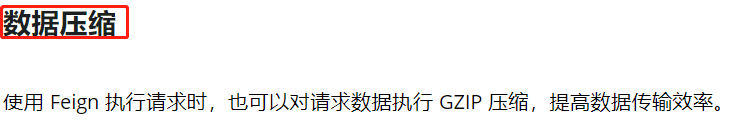
一个公共的接口。生产者实现这个接口，feign接口也实现这个接口

少去地址重复写



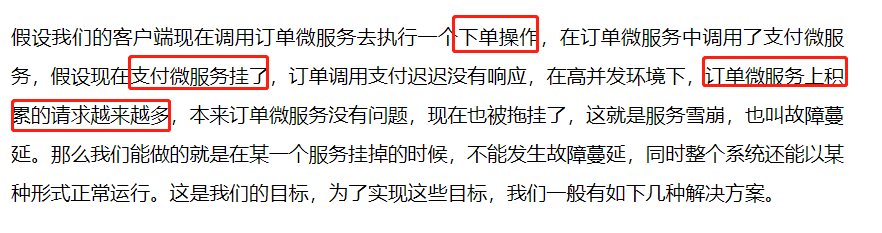


如果设置级别为INFO，则优先级高于等于INFO级别（如：INFO、WARN、ERROR）的日志信息将可以被输出,小于该级别的如DEBUG将不会被输出。



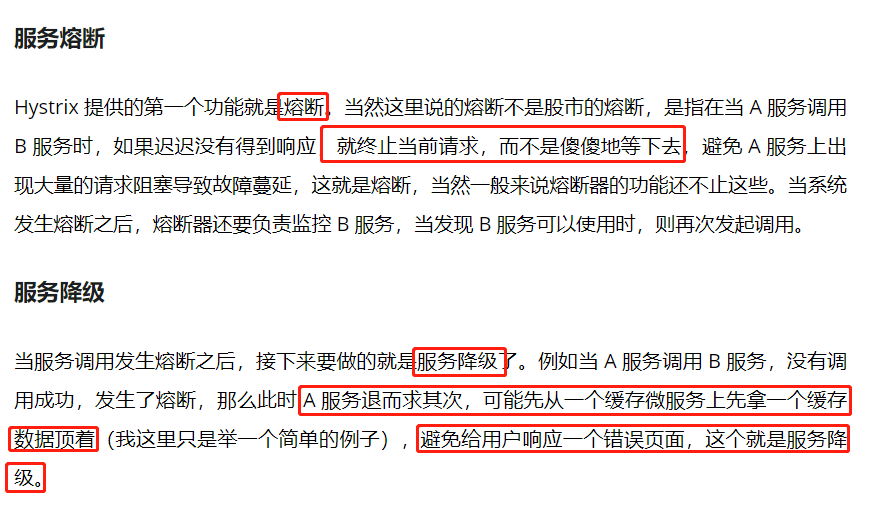


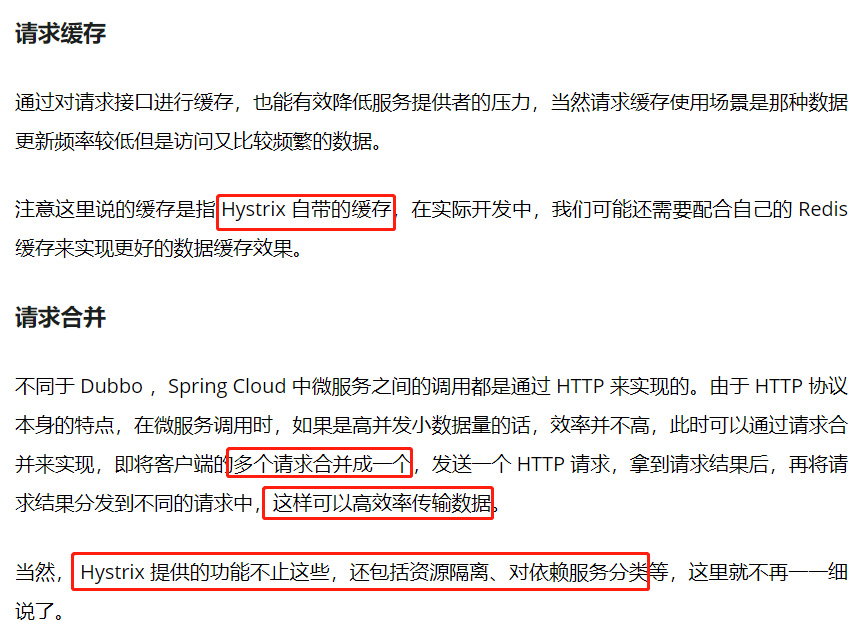
代理请求转发，数据压缩，重试，日志等功能

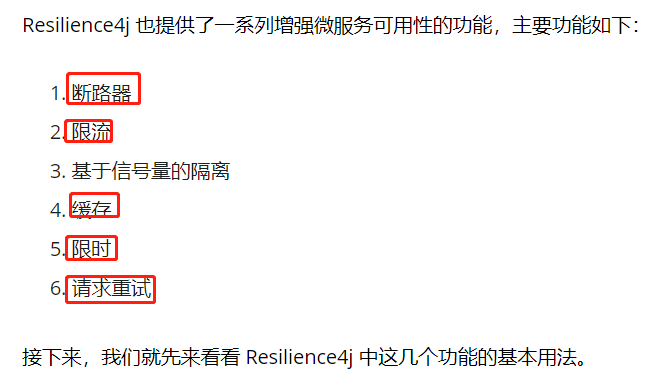


有两个服务，一个是订单服务，一个是支付服务，此时支付服务挂掉了，迟迟没有响应。订单服务越积越多，导致订单服务也别拖垮了，成为服务雪崩

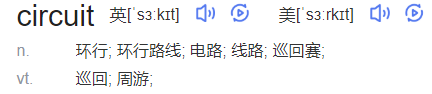


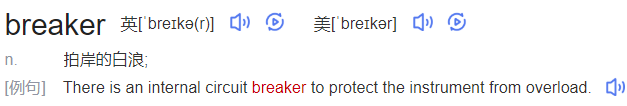






断路器，限流，限时，请求重试，缓存等功能

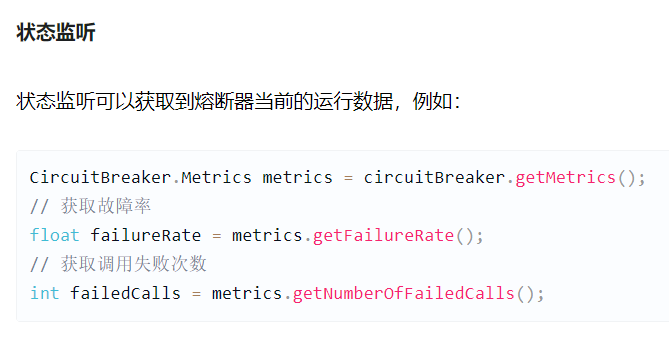












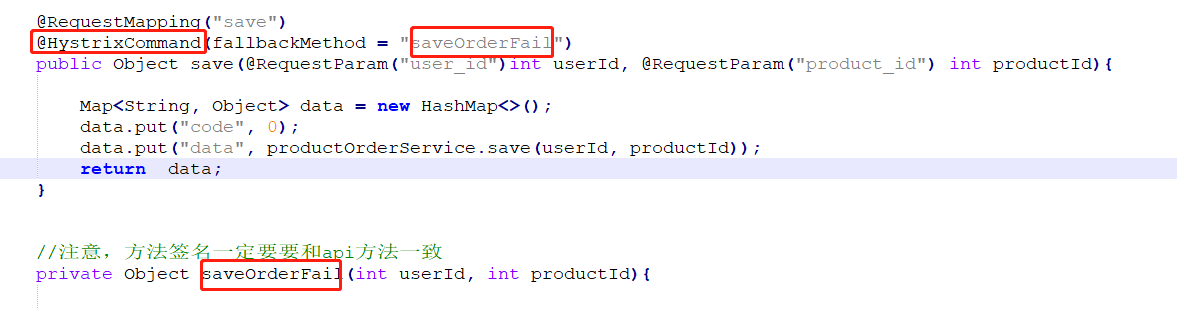
查看失败次数后，服务降级，人工介入

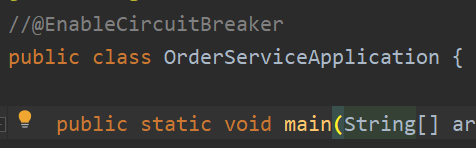
限流

想限制某个请求的频率为 2QPS（每秒处理两个请求）

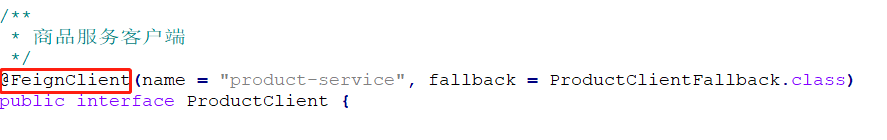


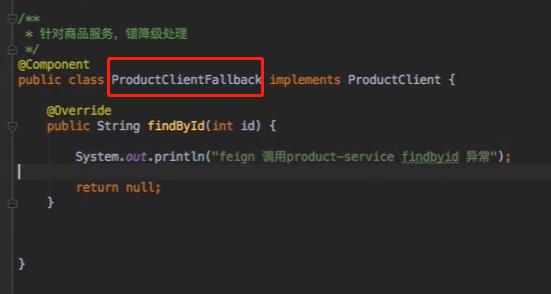
Hystrix



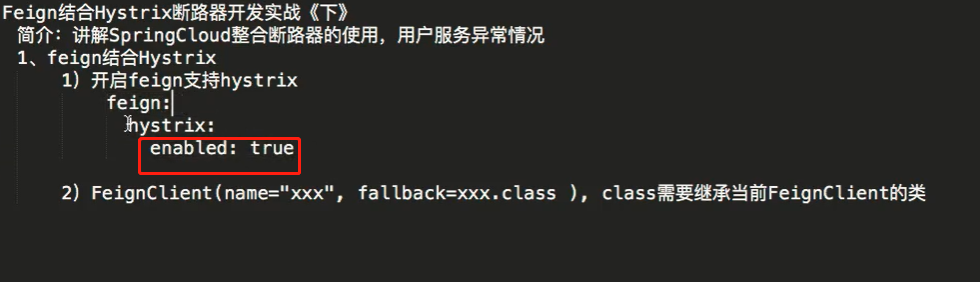


两种写在controller，service的(@hystrixcommand)，和FeignClient结合（颗粒度更小）

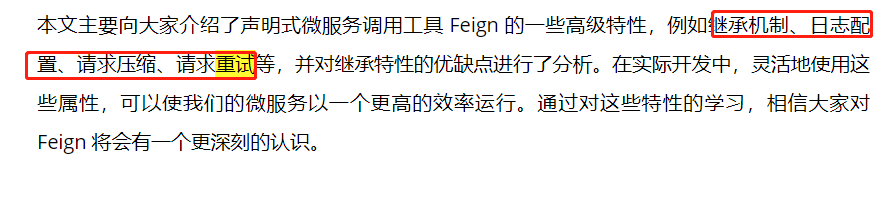


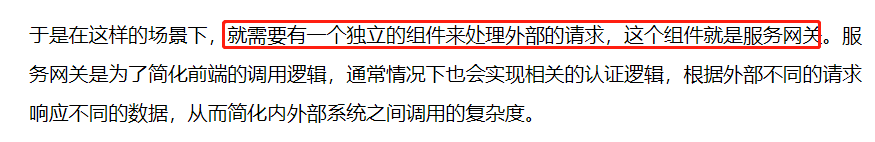


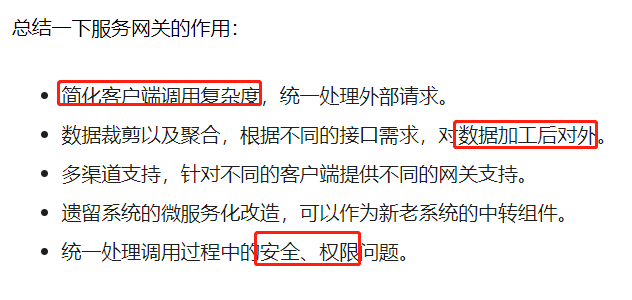
熔断器实现feign的接口

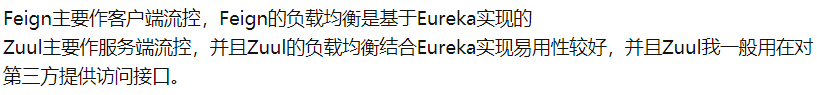


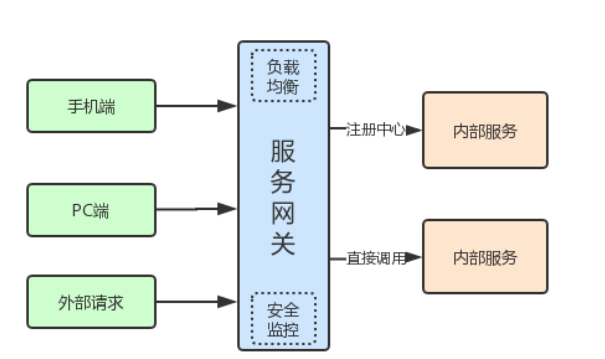
新版本feign默认关闭hystrix，在feignclient的标签加上fallback的属性，在fallback定义的方法中实现feignclient的接口并重写方法实现服务降级

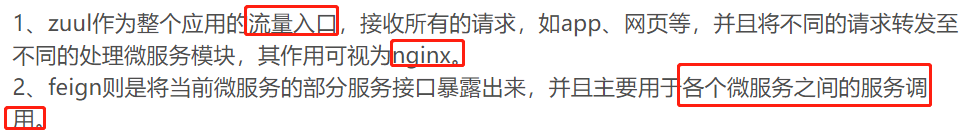




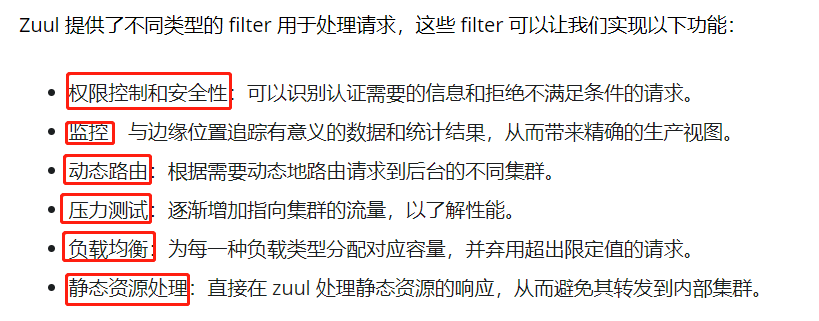








Feign主要作客户端流控， Zuul主要作服务端流控

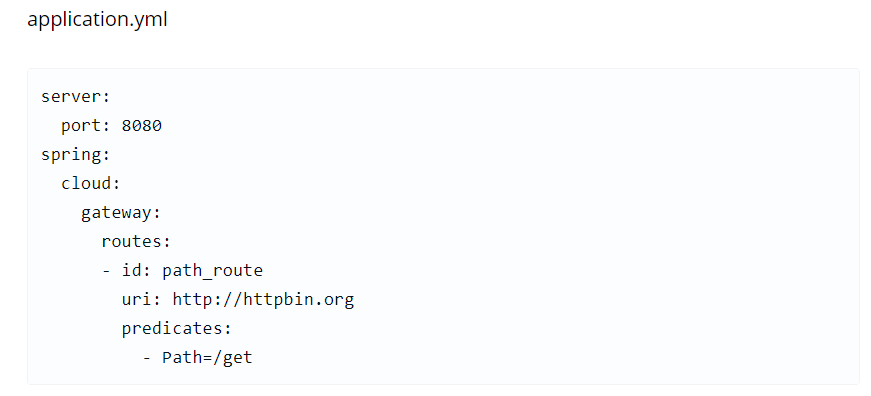


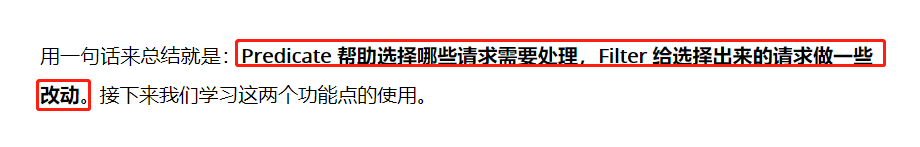


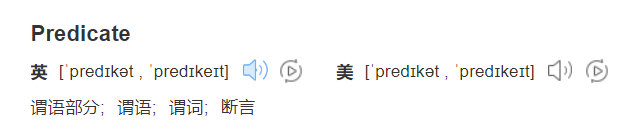




Gateway配置

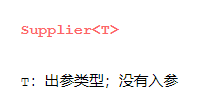


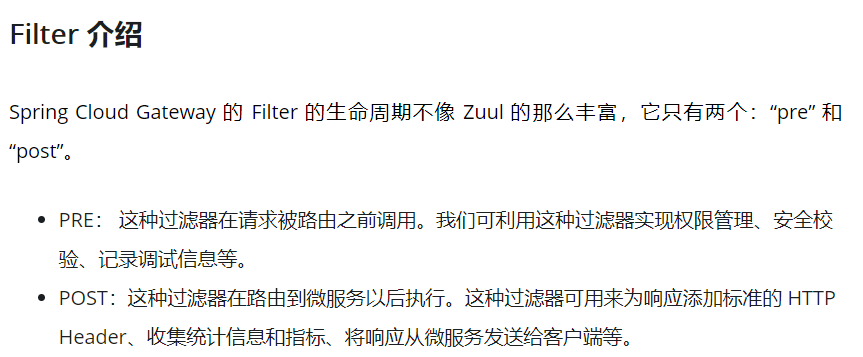




Stream filter true的留下

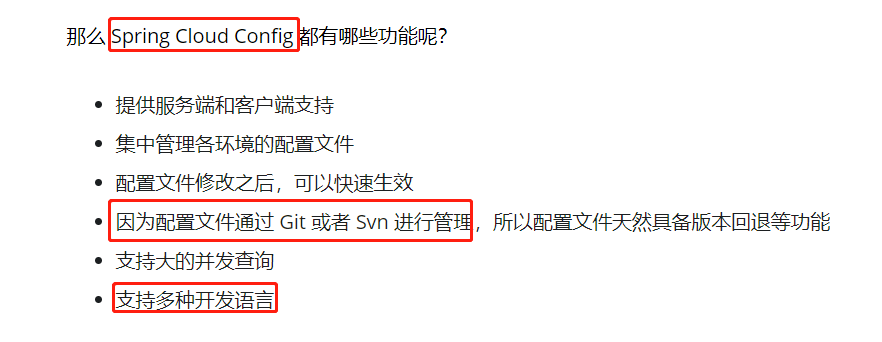


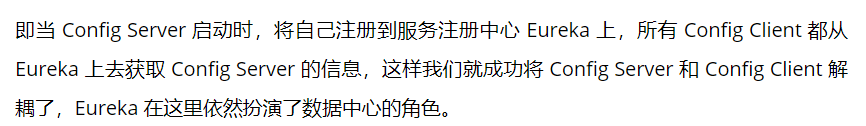


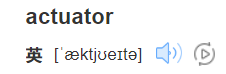


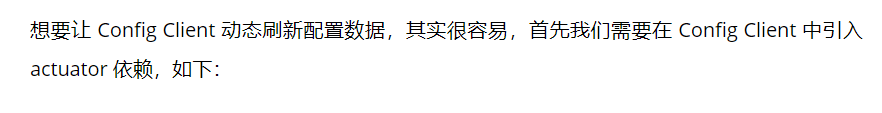
Predicate过滤，yml配置

Filter 有点像aop,之前之后处理







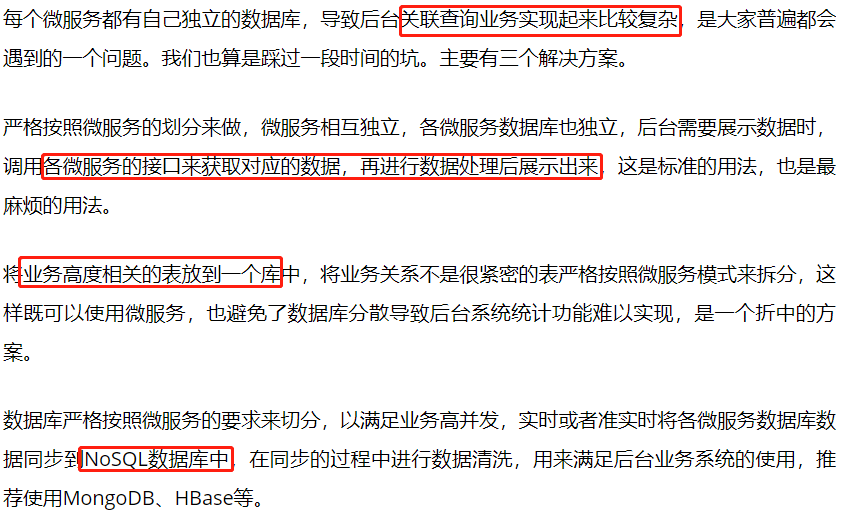


然后暴露refresh接口





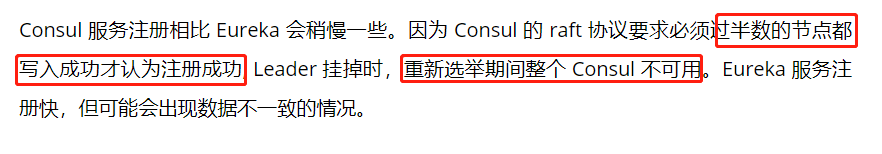




**二阶段提交 2PC，强一致性**

**消息机制，最终一致性**

**TCC 补偿模式，最终一致性**



熔断是一种保护方式，降级是熔断后的一种处理方式