

springCloud

▼ 什么springCloud

- 在springboot的基础上构建的微服务框架
- 1.引入组件的启动器
- 2.覆盖默认配置
- 3.在引导类上添加相应的注解

▼ eureka

- 注册中心，服务的注册与发现

▼ 服务端

- 1.引入服务端启动器：eureka-server
- ▼ 2.添加了配置
 - ▾ spring.application.name
 - 服务名
 - ▼ eureka.client.service-url.defaultZone
 - http://localhost:10086/eureka
 - ▼ eureka.server.eviction-interval-timer-in-ms
 - 剔除无效连接的间隔时间
 - ▼ eureka.server.enable-self-preservation
 - 关闭自我保护
- ▼ 3.@EnableEurekaServer
 - 开启eureka服务端功能

▼ 客户端

- 1.引入启动器：eureka-client
- ▼ 2.添加配置
 - spring.application.name
 - eureka.client.service-url.defaultZone
 - ▾ eureka.instance.lease-renewal-interval-in-seconds
 - 心跳时间
 - ▾ eureka.instance.lease-expiration-duration-in-seconds
 - 过期时间

- ▼ eureka.client.register-with-eureka

- 是否注册给eureka容器
- 默认true

- ▼ eureka.client.fetch-registry

- 是否拉取服务列表
- 默认true

- ▼ eureka.client.registry-fetch-interval-seconds

- 拉取服务的间隔时间

- ▼ 3. @EnableDiscoveryClient

- 启用eureka客户端

- ▼ **ribbon**

- 负载均衡组件
- eureka、feign以及zuul已集成

- ▼ 配置负载均衡策略

- ▼ <服务名>.ribbon.NFLoadBalancerRuleClassName
 - 负载均衡策略的全路径

- ▼ @LoadBalanced

- 在RestTemplate的@Bean方法上

- ▼ **hystrix**

- 容错组件

- ▼ 降级

- 1.引入hystrix的启动器
- 2.添加配置，超时时间的配置

- ▼ 3. @EnableCircuitBreaker

- 启用熔断组件

- ▼ @SpringCloudApplication

- @SpringBootApplication
- @EnableDiscoveryClient

- ▼ 代码

- ▼ 局部

- 返回值和参数列表和被熔断的方法一致

- 在被熔断的方法上@HystrixCommand(fallbackMethod="熔断方法名")

▼ 全局

- 返回值和被熔断的方法返回值一致，不能有参数列表
- 在类上@DefaultProperties(defaultFallback="全局熔断方法名")
- 在具体的被熔断方法上@HystrixCommand

▼ 熔断

▼ close

- 关闭状态
- 所有请求都正常访问

▼ open

- 打开状态
- 所有请求都无法访问
- 触发：连续失败的比例大于50%或者失败次数不少于20
- 维持5s的休眠时间

▼ half open

- 半开状态
- ▼ 释放部分请求通过

▼ 正常

- close

▼ 不正常

- open

- 触发：休眠时间之后

▼ feign

▼ 远程调用组件

- 集成ribbon和hystrix
- 使代码更加优雅

- 1.引入feign的启动器

- ▼ 2.feign.hystrix.enable=true

- 开启熔断

▼ 3.@EnableFeignClients

- 开启feign的功能

▼ 代码

▼ 定义一个接口

- @FeignClient(value="服务名", fallback=实现类.class)
- 方法上的注解使用的都是springMVC的注解

▼ zuul

▼ 网关组件

- 路由
- 过滤器

▪ 1.引入启动器

▼ 2.添加配置

- 路由的配置

▼ zuul.prefix

- 路由前缀

▼ 四种路由配置方式

- zuul.routes.<路由名>.path=/service-provider/**
zuul.routes.<路由名>.url=http://localhost:8081
- zuul.routes.<路由名>.path=/service-provider/**
zuul.routes.<路由名>.serviceId=service-provider
- zuul.routes.<路由名:服务名>=/service-provider/**

▼ 不配置

- 默认服务的入口就是以服务名为前缀

▼ 3.@EnableZuulProxy

- 开启zuul网关组件

▼ 自定义过滤器

▼ IZuulFilter

- 默认的抽象实现类: ZuulFilter

▼ filterType

- pre

post

- post
- route

- error

- ▼ filterOrder

- 执行顺序，返回值越小优先级越高

- ▼ shouldFilter

- 是否执行run方法

- ▼ true

- 执行

- ▼ run

- 过滤器的具体逻辑

- ▼ 四种执行顺序

- pre-->route-->post

- ▼ 异常

- ▼ pre或者route出现异常

- 直接执行error-->post

- ▼ post出现异常

- error-->响应

- ▼ error出现异常

- error-->post

- **自由主题**