

alpsCloud 智能网关使用说明(简化版)

版本号:V1.0.0

@Date:2019/12/14

技术: Springcloud ZUUL + shiro + Redis

Springcloud gateway + Security + oauth2 + Redis

微服务 Provider:

1. **alps-zuul --- zuul 网关服务 port:8300**
2. **alps-gateway --- gateway 网关服 port:9200**

**简化版:只提供开发者使用,暂不提供操作的 UI

一: zuul 网关使用方法

- 1.配置链路追踪 zipkin(开发可暂不配置)

```
#zipKin 指向
zipkin:
  base-url: http://localhost:9411
  enabled: true
sleuth:
  web:
    client:
      enabled: true
  sampler:
    probability: 1.0 #zipkin采集率 0.1表示10%采集
```

- 2.配置消息管理 rabbitmq. 用户我和密码是你本地设置的账/密.具体可以参考

alps 搭建日志

```
rabbitmq:
  host: localhost
  port: 5672
  password: guest
  username: guest
  listener:
    direct:
      retry:
        enabled: true
    simple:
      retry:
        enabled: true
  queue: zipkin
```

3.熔断设置

```
feign.hystrix.enabled: true
hystrix:
  shareSecurityContext: true
  command:
    default:
      circuitBreaker:
        sleepWindowInMilliseconds: 100000
        forceClosed: true
      execution:
        isolation:
          thread:
            timeoutInMilliseconds: 600000
```

3. 网关路由设定:

可以设定 url 或者服务.具体配置可以参考以下:

设定完成以后启动 8300 端口.外部访问统一由 8300 端口进入.

```
#zuul:
#关闭统一前缀 prefix: /alps
# routes:
#   alps-web:
#     path: /web/**
#     serviceId: alpsWeb
#   docs:
#     url: http://10.202.86.138:8001/

#   alps-app:
#     path: /app/**
#     serviceId: webPro
#   alps-file:
#     path: /file/**
#     serviceId: alpsfile
# ignored-services: "*"
# add-host-header: true
# sensitive-headers: "*"
# retryable: true

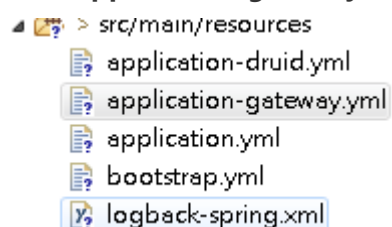
zuul:
  routes:
    system-service:
      path: /system/** #配置请求URL的请求规则
      serviceId: sysPro
    alps-web:
      path: /web/**
      serviceId: webPro
    file-service:
      path: /file/**
      serviceId: alpsFile
  ignored-services: "*"
  add-host-header: true
  sensitive-headers: "*"
  retryable: true
```

二: gateway 网关使用方法

1. 进入到 alpscloud\alps-gateway\src\main\resources 目录下:

I. Application-druid 下配置 gate 所需的数据源.

II. Application -gateway 为网关配置文件



2. Gateway 配置详解

#gateway config 越在前面优先级越高,id用来区分不同的路由规则,不可重复。uri指需要跳转的地址。Predicates可以配置多个,使用正则匹配。

```
spring:
  cloud:
    gateway:
      discovery:
        locator:
          lowerCaseServiceId: true
          enabled: true
      routes:
        - id: alps-eureka
          uri: lb://eureka-server
          order: 100
          predicates:
            - Path=/eureka/**
          filters:
            - StripPrefix=1
            - name: Hystrix
              args:
                name : default
                fallbackUri: 'forward:/defaultfallback'
```

1)#gateway config 越在前面优先级越高,id 用来区分不同的路由规则,不可重复。uri 指需要跳转的地址。Predicates 可以配置多个,使用正则匹配。

2)gateway 可以通过开启以下配置来打开根据服务的 serviceId 来匹配路由,默认是大写,可以通过 spring.cloud.gateway.locator.lowerCaseServiceId:true 来开启小写

3) URL 与服务配置

uri: <http://example.org>

uri: lb://eureka-server

4.Filter

路由过滤器.

- StripPrefix=1 #去掉 Path 前缀,参数为 1 代表去掉/ribbon

5.限流:

支持 计数器,令牌桶,漏桶三种限流方法

```
- name: Hystrix
  args:
    name : default
    fallbackUri: 'forward:/defaultfallback'
- name: RequestRateLimiter
  args:
    redis-rate-limiter.replenishRate: 1 # 令牌桶的容积
    redis-rate-limiter.burstCapacity: 3 # 流速 每秒
    key-resolver: "#{@IpAddressKeyResolver}" #SPEL表达式去的对应的bean
...
```