

算法服务添加权限验证

- 张聪聪 华润智慧能源有限公司
- 2020-06-10

maven 依赖

```
<dependency>
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-starter-openfeign</artifactId>
  <version>2.1.5.RELEASE</version>
</dependency>
<dependency>
  <groupId>cn.hutool</groupId>
  <artifactId>hutool-all</artifactId>
  <version>5.3.5</version>
  <scope>compile</scope>
</dependency>
```

yml 配置文件

添加算法服务加密所需信息

```
algorithm:
  service:
    appId: 79686876 # 算法引用id
    appKey: GJhLJLHIU@0^%$0&^I&^%$KJJFKGTU*^(0&)Tjggjjgj # 算法key
    hmacKey: GJhLJLHIU@0^%$0&^I&^%$KJJFKGTU*^(0&)TjggjjgjGJhLJLHIU@0^%$0&^I&^%$KJJFKGTU*^(0&)Tjggjjgj # 签名
```

Feign 配置类

```

package com.crie.photovoltaic.config;

import feign.Logger;
import feign.codec.Encoder;
import feign.form.spring.SpringFormEncoder;
import org.springframework.beans.factory.ObjectFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.autoconfigure.http.HttpMessageConverters;
import org.springframework.cloud.openfeign.support.SpringEncoder;
import org.springframework.context.annotation.Bean;

/**
 * @author: 张聪聪
 * @date: 2020/6/8 15:17
 * @description:
 */
public class FeignConfigure {

    @Bean
    Logger.Level feignLoggerLevel() {
        return Logger.Level.FULL;
    }

    @Autowired
    private ObjectFactory<HttpMessageConverters> messageConverters;

    @Bean
    public Encoder feignFormEncoder() {
        return new SpringFormEncoder(new SpringEncoder(messageConverters));
    }
}

```

Feign 拦截器类

该类用于在 feign 调用算法服务是构造请求 Header,header 中包含请求时间、traceId、签名信息等

```

package com.crie.photovoltaic.interceptor;

import cn.hutool.core.lang.UUID;
import com.crie.photovoltaic.util.HMACUtil;
import com.google.common.base.Strings;
import feign.RequestInterceptor;
import feign.RequestTemplate;
import lombok.Setter;
import org.slf4j.MDC;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.boot.autoconfigure.condition.ConditionalOnClass;
import org.springframework.stereotype.Component;

/**
 * @author: 张聪聪
 * @date: 2020/6/8 19:01
 * @description:
 */
@Component
@ConditionalOnClass(RequestInterceptor.class)
@SuppressWarnings("unused")
@Setter
public class FeignInterceptor implements RequestInterceptor {
    @Value("${algorithm.service.appId}")
    private String appId;
    @Value("${algorithm.service.appKey}")
    private String appKey;
    @Override
    public void apply(RequestTemplate requestTemplate) {
        String traceId = UUID.randomUUID().toString().replaceAll("-", "");
        requestTemplate.header("TRACE_ID", traceId);
        if (!Strings.isNullOrEmpty(appId) && !Strings.isNullOrEmpty(appKey)) {
            long timestamp = System.currentTimeMillis();
            String hmac = HMACUtil.encode(appId + appKey + traceId + timestamp);
            requestTemplate.header("appId", appId);
            requestTemplate.header("timestamp", Long.toString(timestamp));
            requestTemplate.header("sign", hmac);
        }
    }
}

```

Hmac 签名工具类

```
package com.crie.photovoltaic.util;

import cn.hutool.crypto.digest.HMac;
import cn.hutool.crypto.digest.HmacAlgorithm;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.stereotype.Component;

/**
 * @author: 张聪聪
 * @date: 2020/6/9 09:12
 * @description:
 */
@Component
public class HMACUtil {
    @Value("${algorithm.service.hmacKey}")
    public void setKey(String key) {
        HMACUtil.key = key;
    }
    private static String key;
    public static String encode(String message) {
        HMac mac = new HMac(HmacAlgorithm.HmacSHA256, key.getBytes());
        String macHex = mac.digestHex(message);
        return macHex;
    }
}
```

feign 客户端类

```

package com.crie.photovoltaic.client;

import com.alibaba.fastjson.JSONObject;
import com.crie.photovoltaic.config.FeignConfigure;
import org.springframework.cloud.openfeign.FeignClient;
import org.springframework.http.MediaType;
import org.springframework.web.bind.annotation.*;

/**
 * @author: 张聪聪
 * @date: 2020/6/8 15:16
 * @description:
 */
@FeignClient(name="sidecar-server", url = "http://127.0.0.1:5000/users/",configuration = FeignConfigure.class)
public interface SidecarAPIClient {
    /**
     *
     * @param jsonValue json字符串
     * @return
     */
    @PostMapping(value = "/photovoltaic/capacity",consumes = MediaType.APPLICATION_JSON_VALUE)
    JSONObject capacity(@RequestBody String jsonValue);
}

```

@FeignClient 中的 url 参数和@PostMapping 中的 value 参数拼接在一起就是完整的算法服务 url 地址，需根据实际地址修改。