

spring cloud 自定义ribbon客户端

一、自定义Ribbon客户端-【方式一】配置类

1.1、自定义负载规则

增加RibbonConfiguration.java配置类

```
public class RibbonConfiguration {

//  @Autowired
//  IClientConfig config;

    @Bean
    public IRule ribbonRule() {
        return new RandomRule();
    }
}
```

其中：RibbonClient中name 微服务名称，configuration配置类

注意：configuration等于的TestConfiguration必须是@Configuration， 但要注意它不在主应用程序上下文的@ComponentScan中， 否则它将被所有@RibbonClients共享。 如果使用@ComponentScan（或@SpringBootApplication）， 则需要采取措施以避免包含它（例如， 将其放在单独的， 不重叠的包中， 或者指定要在@ComponentScan中显式扫描的包）。

方式1、TestConfiguration不放在spring boot启动类的当前包或子包中即可

方式2、如果TestConfiguration确实需要放在当前包， 需要设置如下

增加注解

```
public @interface ExcludeFromComponentScan {

}
```

将注解增加至RibbonConfiguration

```
package com.pupei yuan.config;

import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;

import com.netflix.loadbalancer.IRule;
import com.netflix.loadbalancer.RandomRule;

@Configuration
@ExcludeFromComponentScan
public class RibbonConfiguration {

//  @Autowired
//  IClientConfig config;

    @Bean
    public IRule ribbonRule() {
        return new RandomRule();
    }
}
```

将排除注解增加至启动类， 将自定义客户端增加至启动类

```
package com.pupei yuan.config;

import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.boot.autoconfigure.jdbc.DataSourceAutoConfiguration;
import org.springframework.boot.builder.SpringApplicationBuilder;
import org.springframework.boot.web.support.SpringBootServletInitializer;
import org.springframework.cloud.client.discovery.EnableDiscoveryClient;
import org.springframework.cloud.client.loadbalancer.LoadBalanced;
import org.springframework.cloud.context.config.annotation.RefreshScope;
import org.springframework.cloud.netflix.ribbon.RibbonClient;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.ComponentScan;
import org.springframework.context.annotation.Configuration;
import org.springframework.context.annotation.FilterType;
import org.springframework.web.client.RestTemplate;

@Configuration
//扫描bean
@ComponentScan(basePackages = "com.pupei yuan.*")
//不用自动配置数据源
```

```
@EnableDiscoveryClient
@SpringBootApplication(exclude=DataSourceAutoConfiguration.class)
@RibbonClient(name = "multiple", configuration = RibbonConfiguration.class)
@ComponentScan(excludeFilters = { @ComponentScan.Filter(type = FilterType.ANNOTATION, value = ExcludeFromComponentScan.class) })
public class MainApplication extends SpringBootServletInitializer {

    //相当于xml中的bean标签 用于调用当前方法获取到指定的对象
    @Bean(name="remoteRestTemplate")
    @LoadBalanced
    public RestTemplate getRestTemplate(){
        return new RestTemplate();
    }
    @Override
    protected SpringApplicationBuilder configure(SpringApplicationBuilder builder) {
        return builder.sources(MainApplication.class);
    }
}
```

多个自定义客户端可以依次添加@RibbonClient(name = "multiple", configuration = RibbonConfiguration.class)

1.2、源码查看

查看：@RibbonClient注解发现主要属性为 RibbonClientConfiguration，查看其实现

Spring Cloud Netflix默认为Ribbon提供以下Bean（BeanType beanName: ClassName）：

- IClientConfig ribbonClientConfig: DefaultClientConfigImpl
- IRule ribbonRule: ZoneAvoidanceRule
- IPing ribbonPing: DummyPing
- ServerList<Server> ribbonServerList: ConfigurationBasedServerList
- ServerListFilter<Server> ribbonServerListFilter: ZonePreferenceServerListFilter
- ILoadBalancer ribbonLoadBalancer: ZoneAwareLoadBalancer
- ServerListUpdater ribbonServerListUpdater: PollingServerListUpdater

二、自定义Ribbon客户端-【方式二】配置文件

从1.2.0版开始，Spring Cloud Netflix现在支持使用属性自定义Ribbon客户端以与Ribbon文档兼容。

配置属性

复制代码

```
<clientName>.ribbon.:
NFLoadBalancerClassName: should implement ILoadBalancer
NFLoadBalancerRuleClassName: should implement IRule
NFLoadBalancerPingClassName: should implement IPing
NIWSServerListClassName: should implement ServerList
NIWSServerListFilterClassName should implement ServerListFilter
```

复制代码

注意：配置优先级高于代码自定义和默认配置，并且不会有代码方式的干扰

示例

```
users: #微服务名称
  ribbon:
    NIWSServerListClassName: com.netflix.loadbalancer.ConfigurationBasedServerList
    NFLoadBalancerRuleClassName: com.netflix.loadbalancer.WeightedResponseTimeRule
```

- NFLoadBalancerRuleClassName，可以通过这个配置项定制需要的负载均衡规则，可以是ribbon提供的原生的几种规则类，也可以是自己实现的规则类，这些类都实现了IRule接口。指定负载均衡器的实现类。当然，可以设置自己实现的负载均衡器。
- NFLoadBalancerPingClassName 用于配置查看服务器是否存活。
- NIWSServerListClassName 是服务器列表的处理类，用来维护服务器列表的。Ribbon已经实现了动态服务器列表。
- NIWSServerListFilterClassName 是服务器的拦截类。

四、Ribbon内置负载均衡规则

Ribbon框架按照不同需求，已经为我们实现了许多实现了IRule接口的实现类，适用于常用的负载均衡规则。以下规则能够实现大部分负载均衡需求的应用场景，如果有更复杂的需求，可以自己实现IRule。

内置负载均衡规则类	规则描述
RoundRobinRule	简单轮询服务列表来选择服务器。它是Ribbon默认的负载均衡规则。
AvailabilityFilteringRule	对以下两种服务器进行忽略： （1）在默认情况下，这台服务器如果3次连接失败，这台服务器就会被设置为“短路”状态。短路状态将持续30秒，如果再次连接失败，短路的持续时间就会几何级地增加。 注意：可以通过修改配置loadbalancer.<clientName>.connectionFailureCountThreshold来修改连接失败多少次之后被设置为短路状态。默认是3次。 （2）并发数过高的服务器。如果一个服务器的并发连接数过高，配置了AvailabilityFilteringRule规则的客户端也会将其忽略。并发连接数的上线，可以由客户端的<clientName>.<clientConfigNameSpace>.ActiveConnectionsLimit属性进行配置。

WeightedResponseTimeRule	为每一个服务器赋予一个权重值。服务器响应时间越长，这个服务器的权重就越小。这个规则会随机选择服务器，这个权重值会影响服务器的选择。
ZoneAvoidanceRule	以区域可用的服务器为基础进行服务器的选择。使用Zone对服务器进行分类，这个Zone可以理解为一个机房、一个机架等。
BestAvailableRule	忽略哪些短路的服务器，并选择并发数较低的服务器。
RandomRule	随机选择一个可用的服务器。
Retry	重试机制的选择逻辑

附录：AvailabilityFilteringRule的三个默认配置

```
# successive connection failures threshold to put the server in circuit tripped state, default 3
niws.loadbalancer.<clientName>.connectionFailureCountThreshold

# Maximal period that an instance can remain in "unusable" state regardless of the exponential increase, default 30
niws.loadbalancer.<clientName>.circuitTripMaxTimeoutSeconds

# threshold of concurrent connections count to skip the server, default is Integer.MAX_INT
<clientName>.<clientConfigNameSpace>.ActiveConnectionsLimit
```

相关阅读:

- [Oracle 表空间查询](#)
- [FlaskAppBuilder 中文文档 markdown格式](#)
- [Springboot + Stomp + React 实现通过前后端通信](#)
- [Jmeter自动生成UUID](#)
- [Jmeter响应断言中，中文匹配失败问题解决](#)
- [IOS使用纯代码布局替换掉默认的storyboard](#)
- [MySQL在Windows上创建服务和删除服务](#)
- [IOS15上纯代码布局之导航控制器的导航条为透明的问题](#)
- [PHP编译FTP扩展](#)
- [ansible处理一些逻辑请求思路](#)

原文地址：<https://www.cnblogs.com/pypua/p/10132767.html>

最新文章

- [python学习](#)
- [Memcached](#)
- [layui镜像文档](#)
- [fastadmin 增删数组类型数据](#)
- [css img引入的图片宽高固定 不想让图片变形](#)
- [fastadmin之Workerman在线聊天](#)
- [Skipped 'runtime' Node remains in conflict](#)
- [linux搭建svn,配置自动更新钩子](#)
- [微信开发者工具能调试小程序,手机端加载不出来](#)
- [fastadmin常用一键创建curd](#)

热门文章

- [selenium 添加cookie和token](#)
- [Python调试器pdb](#)
- [selenium 多窗口切换](#)
- [selenium 弹出框处理](#)
- [selenium 鼠标操作（ActionChains）](#)
- [年终总结](#)
- [接口的安全性测试](#)
- [pytest简单介绍](#)
- [selenium 复选框处理](#)
- [selenium 下拉框（select）处理](#)