

@LoadBalanced用作标记注释，指示被注释的对象RestTemplate应使用RibbonLoadBalancerClient与您的服务进行交互。这允许您对传递给的网址使用“逻辑标识符”RestTemplate。这些逻辑标识符通常是服务的名称。例如：restTemplate.getForObject("http://some-service-name/user/{id}", String.class, 1);

@RibbonClient用于配置功能区客户端。如果您正在使用服务发现，并且对所有默认的配置设置都没问题，则不需要使用@RibbonClient注释。至少有两种情况需要使用@RibbonClient：1 您需要为特定的微服务有特定的配置，如调用A服务用随机ribbon，调用B服务用轮询等。2 您没有使用任何服务发现

定义一个@RibbonClient：@RibbonClient(name = "some-service", configuration = SomeServiceConfig.class)。name - 微服务名。configuration-将其设置为@Configuration所有定义的类@Beans，如下图。

ribbon有7种负载均衡策略可供选择：

策略类	命名	描述
RandomRule	随机策略	随机选择server
RoundRobinRule	轮询策略	按照顺序选择server（ribbon默认策略）
RetryRule	重试策略	在一个配置时间段内，当选择server不成功，则一直尝试选择一个可用的server
BestAvailableRule	最低并发策略	逐个考察server，如果server断路器打开，则忽略，再选择其中并发链接最低的server
AvailabilityFilteringRule	可用过滤策略	过滤掉一直失败并被标记为circuit tripped的server，过滤掉那些高并发链接的server（active connections超过配置的阈值）
ResponseTimeWeightedRule	响应时间加权重策略	根据server的响应时间分配权重，响应时间越长，权重越低，被选择到的概率也就越低。响应时间越短，权重越高，被选中的概率越高，这个策略很贴切，综合了各种因素，比如：网络，磁盘，io等，都直接影响响应时间
ZoneAvoidanceRule	区域权重策略	综合判断server所在区域的性能，和server的可用性，轮询选择server并且判断一个AWS Zone的运行性能是否可用，剔除不可用的Zone中的所有server

如果想要创建一个全局的负载策略，只需添加一个配置类，也可自己扩展，添加逻辑，如下：

```
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;

import com.netflix.client.config.IClientConfig;
import com.netflix.loadbalancer.IRule;
import com.netflix.loadbalancer.RandomRule;

@Configuration
public class RibbonConfiguration {

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

如果想要对某个服务源设置特有的策略，可以在工程启动类上添加@RibbonClient注解，当然，对应配置代码也需要调整：

```
@RibbonClient(name = "client-a", configuration = RibbonConfiguration.class)//表示针对client-a服务使用的负责策略是经过RibbonConfiguration配置类的。
//@RibbonClients(value = {
//    @RibbonClient(name = "client-a", configuration = RibbonConfiguration.class),
//    @RibbonClient(name = "client-b", configuration = RibbonConfiguration.class)
//})//这种方式跟@RibbonClient类似，不过这个是针对多个服务进行策略指定。
@ComponentScan(excludeFilters = {@ComponentScan.Filter(type = FilterType.ANNOTATION, value = {AvoidScan.class}})//表示让工程在启动的时候，不让Spring扫描被@AvoidScan注解标记的类，
```

如果想使用配置文件的方式，进行配置负责策略，语法是 **client name.ribbon.***，client name是我们自己给服务取的名字，即：spring.application.name设置的值。如下：

```
client-a:
  ribbon:
    NFLoadBalancerRuleClassName: com.netflix.loadbalancer.RandomRule #针对client-a服务使用随机策略
```

ribbon的重试机制，默认是开启的，需要添加超时与重试的策略配置，如下：

```
client-a:
  ribbon:
    ConnectTimeout: 30000
    ReadTimeout: 60000
    MaxAutoRetries: 1 #对第一次请求的服务的重试次数
    MaxAutoRetriesNextServer: 1 #要重试的下一个服务的最大数量（不包括第一个服务）
    OkToRetryOnAllOperations: true

#说明：这里配置的ConnectTimeout和ReadTimeout是当HTTP客户端使用的是HttpClient才生效，这个时间最终会被设置到HttpClient中。
#在设置的时候需要结合hystrix的超时时间来综合考虑，针对使用的场景，设置太小会导致很多请求失败，设置太大会导致熔断控制变差。

提供了7个核心接口：
```

接口	简述	默认实现
IClientConfig	定义ribbon中管理配置的接口	DefaultClientConfigImpl
IRule	定义ribbon中负载均衡策略的接口	ZoneAvoidanceRule
IPing	定义定期ping服务，检查可用性的接口	DummyPing
ServerList<Server>	定义获取服务列表方法的接口	ConfigurationBasedServerList
ServerListFilter<Server>	定义特定场景下，获取服务列表的方法接口	ZonePreferenceServerListFilter
ILoadBalancer	定义负载均衡选择服务的核心方法接口	ZoneAwareLoadBalancer
ServerListUpdater	为DynamicServerListLoadBalancer定义动态更新服务列表的接口	PollingServerListUpdater