

5.3 使用Java代码自定义Ribbon配置

很多场景下，我们可能需要自定义Ribbon的配置。在Spring Cloud Camden中，可以使用Java代码或配置文件自定义Ribbon的配置，两种方式是等价的。

在Spring Cloud中，Ribbon默认的配置类是 `RibbonClientConfiguration`，它也允许使用一个POJO自定义配置，自定义配置会覆盖默认配置，并且这种配置是细粒度的，不同的RibbonClient可以使用不同的配置。

我们以修改Ribbon的负载均衡规则为例，讲解如何自定义Ribbon的配置。找到

`org.springframework.cloud.netflix.ribbon.RibbonClientConfiguration`，可以看到以下内容：

```
@Bean
@ConditionalOnMissingBean
public IRule ribbonRule(IClientConfig config) {
    ZoneAvoidanceRule rule = new ZoneAvoidanceRule();
    rule.initWithNiwsConfig(config);
    return rule;
}
```

说明在Spring Cloud中，默认的规则是 `ZoneAvoidanceRule`，这是一种复合判断服务提供者所在区域的性能和可用性选择服务提供者节点的负载均衡规则。

可使用`@RibbonClient`注解，为指定名称的Ribbon Client自定义配置。下面我们来为名称为 `microservice-provider-user` 的Ribbon Client自定义配置。

(1) 复制项目 `microservice-consumer-movie-ribbon`，将ArtifactId修改为 `microservice-consumer-movie-ribbon-customizing`。

(2) 创建Ribbon的配置类。

```
/**
 * 该类为Ribbon的配置类
 * 注意：该不应该在主应用程序上下文的@ComponentScan中。
 * @author 周立
 */
@Configuration
public class RibbonConfiguration {
    @Bean
    public IRule ribbonRule() {
        // 负载均衡规则，改为随机
        return new RandomRule();
    }
}
```

(3) 创建一个空类，并在其上添加`@Configuration`注解和`@RibbonClient`注解。

```

/**
 * 使用RibbonClient，为特定name的Ribbon Client自定义配置。
 * 使用@RibbonClient的configuration属性，指定Ribbon的配置类。
 * 可参考的示例：
 * http://spring.io/guides/gs/client-side-load-balancing/
 * @author 周立
 */
@Configuration
@RibbonClient(name = "microservice-provider-user", configuration = RibbonConfiguration.class)
public class TestConfiguration {
}

```

必须注意的是，`RibbonConfiguration` 类不应该在主应用程序上下文的 `@ComponentScan` 中，否则它将会被所有的 `@RibbonClient` 共享。因此，如果只想配置某一个RibbonClient，必须防止 `@Configuration` 注解的类所在的包与 `@ComponentScan` 扫描的包重叠，或者显式指定 `@ComponentScan` 不扫描 `@Configuration` 类所在的包。

测试

- (1) 启动Eureka Server (`microservice-discovery-eureka`)
- (2) 启动2个或更多用户微服务实例 (`microservice-provider-user`)
- (3) 启动电影微服务实例 (`microservice-consumer-movie-ribbon-customizing`)
- (4) 访问<http://localhost:8761>，结果如图5-3所示。
- (5) 多次访问<http://localhost:8010/log-user-instance>，日志如下：

```

2016-11-01 10:56:44.717 INFO 5624 --- [nio-8010-exec-2] c.i.c.study.user.service.MovieService
: microservice-provider-user:192.168.0.59:8000
2016-11-01 10:56:46.657 INFO 5624 --- [nio-8010-exec-6] c.i.c.study.user.service.MovieService
: microservice-provider-user:192.168.0.59:8000
2016-11-01 10:56:46.849 INFO 5624 --- [io-8010-exec-10] c.i.c.study.user.service.MovieService
: microservice-provider-user:192.168.0.59:8001
2016-11-01 10:56:46.996 INFO 5624 --- [nio-8010-exec-9] c.i.c.study.user.service.MovieService
: microservice-provider-user:192.168.0.59:8000
2016-11-01 10:56:47.148 INFO 5624 --- [nio-8010-exec-7] c.i.c.study.user.service.MovieService
: microservice-provider-user:192.168.0.59:8001
2016-11-01 10:56:47.312 INFO 5624 --- [nio-8010-exec-2] c.i.c.study.user.service.MovieService
: microservice-provider-user:192.168.0.59:8001

```

此时请求会随机分布到两个用户微服务节点上，说明已经实现了Ribbon的自定义配置。