主讲老师：Fox

1. 什么是Spring Cloud LoadBalancer

Spring Cloud LoadBalancer是Spring Cloud官方自己提供的客户端负载均衡器, 用来替代Ribbon。

Spring官方提供了两种负载均衡的客户端：

RestTemplate

RestTemplate是Spring提供的用于访问Rest服务的客户端，RestTemplate提供了多种便捷访问远程Http服务的方法，能够大大提高客户端的编写效率。默认情况下，RestTemplate默认依赖jdk的HTTP连接工具。

WebClient

WebClient是从Spring WebFlux 5.0版本开始提供的一个非阻塞的基于响应式编程的进行Http请求的客户端工具。它的响应式编程的基于Reactor的。WebClient中提供了标准Http请求方式对应的get、post、put、delete等方法，可以用来发起相应的请求。

2. RestTemplate整合LoadBalancer

1）引入依赖

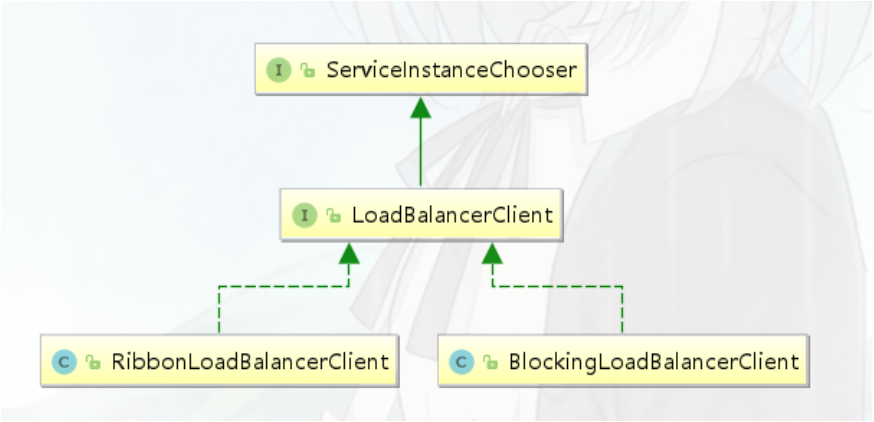
```
<!-- LoadBalancer -->
<dependency>
 <groupId>org.springframework.cloud</groupId>
 <artifactId>spring-cloud-starter-loadbalancer</artifactId>
</dependency>

<!-- 提供了RestTemplate支持 -->
<dependency>
 <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-web</artifactId>
</dependency>

<!-- nacos服务注册与发现 移除ribbon支持-->
<dependency>
 <groupId>com.alibaba.cloud</groupId>
 <artifactId>spring-cloud-starter-alibaba-nacos-discovery</artifactId>
 <exclusions>
 <exclusion>
 <groupId>org.springframework.cloud</groupId>
 <artifactId>spring-cloud-starter-netflix-ribbon</artifactId>
 </exclusion>
 </exclusions>
</dependency>
```

注意： nacos-discovery中引入了ribbon，需要移除ribbon的包

如果不移除，也可以在yml中配置不使用ribbon

```yaml
spring:
 application:
 name: mall-user-loadbalancer-demo
 cloud:
 nacos:
 discovery:
 server-addr: 127.0.0.1:8848
 # 不使用ribbon
 loadbalancer:
  ribbon:
  enabled: false
```

原理：默认情况下，如果同时拥有RibbonLoadBalancerClient和BlockingLoadBalancerClient，为了保持向后兼容性，将使用RibbonLoadBalancerClient。要覆盖它，可以设置spring.cloud.loadbalancer.ribbon.enabled属性为false。



2）使用@LoadBalanced注解配置RestTemplate

```java
@Configuration
public class RestConfig {
 @Bean
 @LoadBalanced
 public RestTemplate restTemplate() {
 return new RestTemplate();
 }
}

```

3) 使用

```java
@RestController
@RequestMapping("/user")
public class UserController {
```

```
4
5  @Autowired
6  private RestTemplate restTemplate;
7
8  @RequestMapping(value = "/findOrderByUserId/{id}")
9  public R findOrderByUserId(@PathVariable("id") Integer id) {
10  String url = "http://mall-order/order/findOrderByUserId/"+id;
11  R result = restTemplate.getForObject(url,R.class);
12  return result;
13  }
14 }
```

3. WebClient整合LoadBalancer

1）引入依赖

```
1  <!-- LoadBalancer -->
2  <dependency>
3   <groupId>org.springframework.cloud</groupId>
4   <artifactId>spring-cloud-starter-loadbalancer</artifactId>
5  </dependency>
6
7  <!-- webflux -->
8  <dependency>
9   <groupId>org.springframework.boot</groupId>
10   <artifactId>spring-boot-starter-webflux</artifactId>
11  </dependency>
12
13  <!-- nacos服务注册与发现 -->
14  <dependency>
15   <groupId>com.alibaba.cloud</groupId>
16   <artifactId>spring-cloud-starter-alibaba-nacos-discovery</artifactId>
17   <exclusions>
18   <exclusion>
19   <groupId>org.springframework.cloud</groupId>
20   <artifactId>spring-cloud-starter-netflix-ribbon</artifactId>
21   </exclusion>
22   </exclusions>
23  </dependency>
```

2) 配置WebClient作为负载均衡器的client

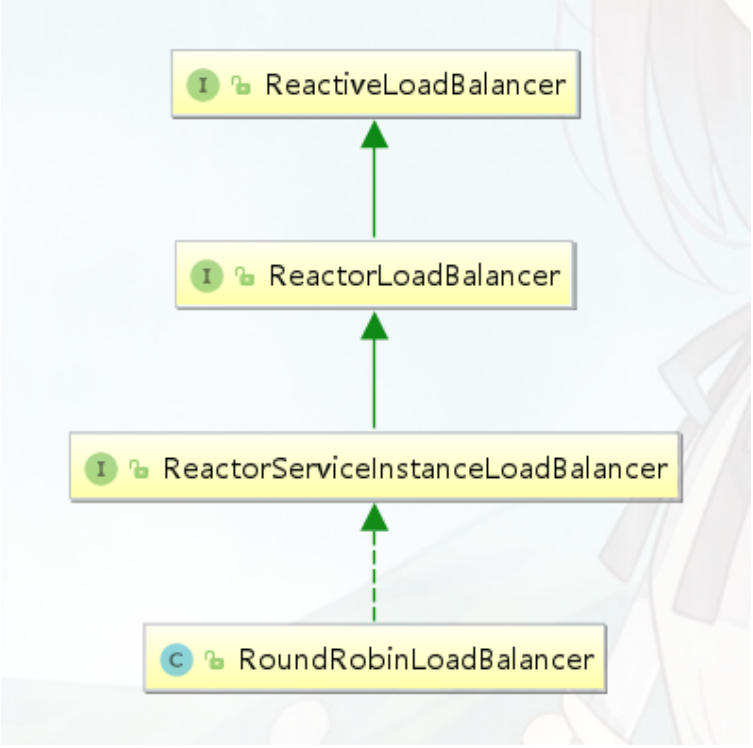
```
1  @Configuration
2  public class WebClientConfig {
3
4   @LoadBalanced
```

```
@Bean
WebClient.Builder webClientBuilder() {
return WebClient.builder();
}

 @Bean
 WebClient webClient() {
 return webClientBuilder().build();
 }
}
```

3) 使用

```
@Autowired
private WebClient webClient;

@RequestMapping(value = "/findOrderByUserId2/{id}")
public Mono<R> findOrderByUserIdWithWebClient(@PathVariable("id") Integer id)
{

 String url = "http://mall-order/order/findOrderByUserId/"+id;
 //基于WebClient
 Mono<R> result = webClient.get().uri(url)
  .retrieve().bodyToMono(R.class);
 return result;
}
```

原理： 底层会使用ReactiveLoadBalancer



引入webFlux

```java
@Autowired
private ReactorLoadBalancerExchangeFilterFunction lbFunction;

@RequestMapping(value = "/findOrderByUserId3/{id}")
public Mono<R> findOrderByUserIdWithWebFlux(@PathVariable("id") Integer id) {

 String url = "http://mall-order/order/findOrderByUserId/"+id;
 //基于WebClient+webFlux
 Mono<R> result = WebClient.builder()
  .filter(lbFunction)
  .build()
  .get()
  .uri(url)
  .retrieve()
  .bodyToMono(R.class);
 return result;
}
```

文档：02-1 微服务负载均衡器LoadBalancer实?..

链接：http://note.youdao.com/noteshare?id=36adba6814ddc01d62363c2a54595a00&sub=E37EABF2A6A946ABBF4F676793FCD780