

主讲老师: Fox

1. Spring Cloud整合Dubbo

1. provider端配置

引入依赖

```
1
2 <dependency>
3   <groupId>com.alibaba.cloud</groupId>
4   <artifactId>spring-cloud-starter-dubbo</artifactId>
5 </dependency>
6
7 <dependency>
8   <groupId>com.alibaba.cloud</groupId>
9   <artifactId>spring-cloud-starter-alibaba-nacos-discovery</artifactId>
10 </dependency>
```

application.yml

```
1 dubbo:
2   scan:
3     # 指定 Dubbo 服务实现类的扫描基准包
4     base-packages: com.tuling.mall.user.service
5   # application:
6   # name: ${spring.application.name}
7   protocol:
8     # dubbo 协议
9     name: dubbo
10    # dubbo 协议端口 ( -1 表示自增端口, 从 20880 开始)
11    port: -1
12  # registry:
13  # #挂载到 Spring Cloud 注册中心 高版本可选
14  # address: spring-cloud://127.0.0.1:8848
15
16 spring:
17   application:
18     name: spring-cloud-dubbo-provider-user
19     main:
20     # Spring Boot2.1及更高的版本需要设定
21     allow-bean-definition-overriding: true
22   cloud:
```

```
23 nacos:
24 # Nacos 服务发现与注册配置
25 discovery:
26 server-addr: 127.0.0.1:8848
```

服务实现类上配置@DubboService暴露服务

```
1 @DubboService
2 public class UserServiceImpl implements UserService {
3
4     @Autowired
5     private UserMapper userMapper;
6
7     @Override
8     public List<User> list() {
9         return userMapper.list();
10    }
11
12    @Override
13    public User getById(Integer id) {
14        return userMapper.getById(id);
15    }
16 }
```

2. consumer端配置

引入依赖

```
1 <dependency>
2   <groupId>org.springframework.boot</groupId>
3   <artifactId>spring-boot-starter-web</artifactId>
4 </dependency>
5
6 <dependency>
7   <groupId>com.alibaba.cloud</groupId>
8   <artifactId>spring-cloud-starter-dubbo</artifactId>
9 </dependency>
10
11 <dependency>
12   <groupId>com.alibaba.cloud</groupId>
13   <artifactId>spring-cloud-starter-alibaba-nacos-discovery</artifactId>
14 </dependency>
```

application.yml

```

1 dubbo:
2   cloud:
3   # 指定需要订阅的服务提供方，默认值*，会订阅所有服务，不建议使用
4   subscribed-services: spring-cloud-dubbo-provider-user
5   # application:
6   # name: ${spring.application.name}
7   protocol:
8   # dubbo 协议
9   name: dubbo
10  # dubbo 协议端口（-1 表示自增端口，从 20880 开始）
11  port: -1
12  # registry:
13  # #挂载到 Spring Cloud 注册中心 高版本可选
14  # address: spring-cloud://127.0.0.1:8848
15
16  spring:
17  application:
18  name: spring-cloud-dubbo-consumer-user
19  main:
20  # Spring Boot2.1及更高的版本需要设定
21  allow-bean-definition-overriding: true
22  cloud:
23  nacos:
24  # Nacos 服务发现与注册配置
25  discovery:
26  server-addr: 127.0.0.1:8848

```

当应用使用属性 `dubbo.cloud.subscribed-services` 为默认值时，日志中将会输出警告：

```

end you using the externalized property 'dubbo.cloud.subscribed-services' to specify the services
end you using the externalized property 'dubbo.cloud.subscribed-services' to specify the services
end you using the externalized property 'dubbo.cloud.subscribed-services' to specify the services
end you using the externalized property 'dubbo.cloud.subscribed-services' to specify the services
end you using the externalized property 'dubbo.cloud.subscribed-services' to specify the services

```

服务消费方通过 `@DubboReference` 引入服务

```

1 @RestController
2 @RequestMapping("/user")
3 public class UserController {
4
5     @DubboReference
6     private UserService userService;
7
8     @RequestMapping("/info/{id}")

```

```

9 public User info(@PathVariable("id") Integer id){
10
11     return userService.getById(id);
12 }
13
14 @RequestMapping("/list")
15 public List<User> list(){
16
17     return userService.list();
18 }
19 }

```

4. 从Open Feign迁移到Dubbo

Dubbo Spring Cloud 提供了方案，即 `@DubboTransported` 注解，支持在类，方法，属性上使用。能够帮助服务消费端的 Spring Cloud Open Feign 接口以及 `@LoadBalanced RestTemplate Bean` 底层走 Dubbo 调用（可切换 Dubbo 支持的协议），而服务提供方则只需在原有 `@RestController` 类上追加 Dubbo `@Service` 注解（需要抽取接口）即可，换言之，在不调整 Feign 接口以及 RestTemplate URL 的前提下，实现无缝迁移。

修改服务提供者

```

1 @DubboService
2 @Slf4j
3 @RestController
4 @RequestMapping("/user")
5 public class UserServiceImpl implements UserService {
6
7     @Autowired
8     private UserMapper userMapper;
9
10    @Override
11    @RequestMapping("/list")
12    public List<User> list() {
13        log.info("查询user列表");
14        return userMapper.list();
15    }
16
17    @Override
18    @RequestMapping("/getById/{id}")
19    public User getById(@PathVariable("id") Integer id) {
20        return userMapper.getById(id);

```

```
21 }  
22 }
```

服务消费端引入依赖

```
1 <dependency>  
2 <groupId>org.springframework.boot</groupId>  
3 <artifactId>spring-boot-starter-web</artifactId>  
4 </dependency>  
5  
6 <dependency>  
7 <groupId>com.alibaba.cloud</groupId>  
8 <artifactId>spring-cloud-starter-dubbo</artifactId>  
9 </dependency>  
10  
11 <dependency>  
12 <groupId>com.alibaba.cloud</groupId>  
13 <artifactId>spring-cloud-starter-alibaba-nacos-discovery</artifactId>  
14 </dependency>  
15  
16 <dependency>  
17 <groupId>org.springframework.cloud</groupId>  
18 <artifactId>spring-cloud-starter-openfeign</artifactId>  
19 </dependency>
```

feign的实现, 启动类上添加@EnableFeignClients

```
1 @SpringBootApplication  
2 @EnableFeignClients  
3 public class SpringCloudDubboConsumerUserFeignApplication {  
4  
5     public static void main(String[] args) {  
6         SpringApplication.run(SpringCloudDubboConsumerUserFeignApplication.class, args);  
7     }  
8  
9 }
```

feign接口添加 @DubboTransported 注解

```
1 @FeignClient(value = "spring-cloud-dubbo-provider-user-feign", path =  
2     "/user")  
3 @DubboTransported(protocol = "dubbo")  
4 public interface UserDubboFeignService {  
5  
6     @RequestMapping("/list")
```

```

6 public List<User> list();
7
8 @RequestMapping("/getById/{id}")
9 public User getById(@PathVariable("id") Integer id);
10 }
11
12 @FeignClient(value = "spring-cloud-dubbo-provider-user-feign",path = "/user")
13 public interface UserFeignService {
14
15     @RequestMapping("/list")
16     public List<User> list();
17
18     @RequestMapping("/getById/{id}")
19     public User getById(@PathVariable("id") Integer id);
20 }

```

调用对象添加 @DubboTransported 注解

```

1 @RestController
2 @RequestMapping("/user")
3 public class UserController {
4
5     @DubboReference
6     private UserService userService;
7
8     @RequestMapping("/info/{id}")
9     public User info(@PathVariable("id") Integer id){
10     return userService.getById(id);
11     }
12
13     @Autowired
14     @DubboTransported
15     private UserFeignService userFeignService;
16
17     @RequestMapping("/list")
18     public List<User> list(){
19     return userFeignService.list();
20     }
21
22     @Autowired
23     private UserDubboFeignService userDubboFeignService;
24

```

```
25 @RequestMapping("/list2")
26 public List<User> list2(){
27
28     return userDubboFeignService.list();
29 }
30
31 @Autowired
32 private RestTemplate restTemplate;
33
34 @Bean
35 @LoadBalanced
36 @DubboTransported
37 public RestTemplate restTemplate() {
38     return new RestTemplate();
39 }
40
41 @RequestMapping("/list3")
42 public List<User> list3(){
43     String url = "http://spring-cloud-dubbo-provider-user-feign/user/list";
44     return restTemplate.getForObject(url, List.class);
45 }
46
47 }
```

文档: 03-1 微服务调用组件Dubbo实战.note

链接: [http://note.youdao.com/noteshare?](http://note.youdao.com/noteshare?id=d04900ac7cd7a7fc0d9031cf52b2a46f&sub=9A9403B46FEE4B1485BEC1D71D3BEE68)

id=d04900ac7cd7a7fc0d9031cf52b2a46f&sub=9A9403B46FEE4B1485BEC1D71D3BEE68