**[学习微服务-第6天] 负载均衡之ServiceComb + SpringCloud Ribbon**

在微服务架构中，客户端负载均衡是指负载均衡器作为客户端软件的一部分,客户端得到可用的服务实例列表然后按照特定的负载均衡策略,分发请求到不同的服务。

ServiceComb内置了客户端负载均衡组件，开发者可以非常简单的使用。具体可参考:

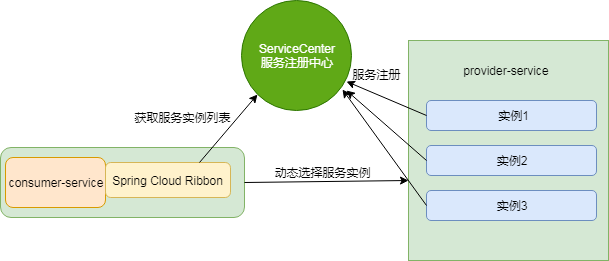
https://docs.servicecomb.io/java-chassis/zh\_CN/references-handlers/loadbalance.html

本文将介绍ServiceComb与SpringCloud的Ribbon负载均衡组件协同工作，以构建微服务应用。ServiceComb已适配对应的接口和配置，用户用极简单的方法配置后即可使微服务应用具备负载均衡的能力。

**示例**

以下通过一个服务提供者provider-service和消费者consumer-service作为demo演示。

provider-service会启动3个微服务实例，消费者端consumer-service使用Ribbon负载均衡调用proveder-service服务的接口。其中consumer-service在调用provider-service提供的接口时会打印出真实调用的URL ↓↓↓



**完整示例地址：**

https://github.com/lisenwork/servicecomb-demo/tree/master/servicecomb-ribbon

**预置条件:**示例应先安装启动服务与注册中心ServiceCenter，详细步骤请参考官网↓↓↓

http://servicecomb.apache.org/cn/users/setup-environment/#%E8%BF%90%E8%A1%8Cservice-center

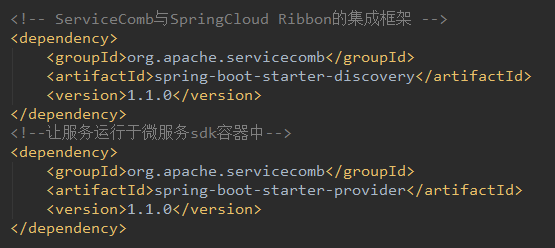
|  |  |
| --- | --- |
| **一** | **开发服务消费者comsumer-service** |

只需三步即可开发拥有负载均衡能力的微服务

步骤如下：

**01.添加依赖**

新建pom文件，引入如下依赖。



完整pom文件内容请参考

https://github.com/lisenwork/servicecomb-demo/blob/master/servicecomb-ribbon/consumer-service/pom.xml

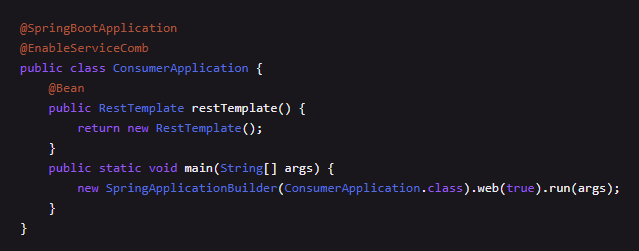
**02．配置**

在resources目录下新建ServiceComb配置文件microservice.yaml。配置微服务信息

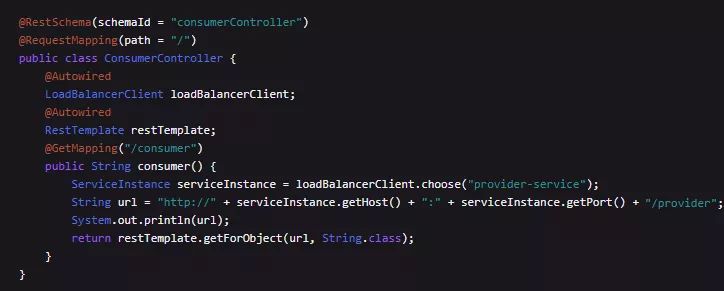


**03．项目入口**

新建启动类ConsumerApplication.java。如下图，启动类里同时实例化一个RestTemplate对象。该对象用于后面的服务间接口调用。



新建ConsumerController.java。在该类的consumer方法里使用Ribbon的API动态获取服务实例，并打印出被选中的实例的真实IP地址和端口。最后调用服务实例的接口，获取结果并返回。



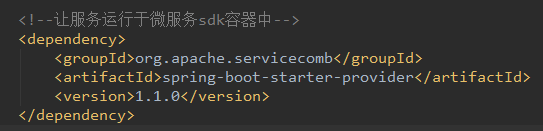
**04．启动**

项目根目录下执行命令 mvn spring-boot:run

|  |  |
| --- | --- |
| **二** | **开发服务提供者provider-service** |

**01．添加依赖**

新建pom文件，引入如下依赖。



完整pom文件内容请参考

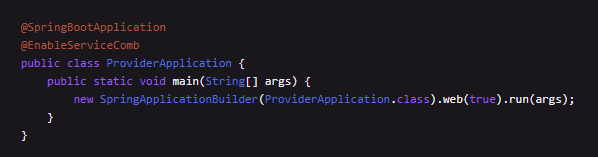
https://github.com/lisenwork/servicecomb-demo/blob/master/servicecomb-ribbon/provider-service/pom.xml

**02．配置**

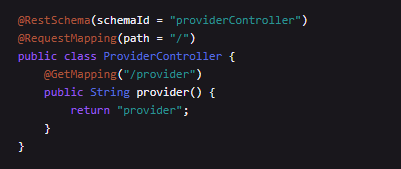
在src/main/resources目录下新建microservice.yaml

**03.项目入口**

新建启动类ProviderApplication.java



新建ProviderController.java。只向外提供/provider接口



**04.启动**

服务提供者要启动3个微服务实例。

打开microservice.yaml文件，分别修改微服务监听端口为8888，8889，8890，在项目根目录下执行3次命令 mvn spring-boot:run

**演示**

浏览器访问http://localhost:7777/consumer，重复刷新一定次数，观察控制台，会发现服务消费者会轮询调用服务提供者的三个实例。

