# 微服务

微服务的特点:

单一职责:微服务中每一个服务都对应唯一的业务能力,做到单一职责微:微服务的服务拆分粒度很小

面向服务:面向服务是说每个服务都要对外暴露Rest风格服务接口API.并不关心服务的技术实现,做到与平台和语 言无关.也不限定用什么技术实现,只要提供Rest的接口即可.

自治:自治是说服务间互相独立,互不干扰.

# 服务调用方式

## RPC和HTTP

常见的远程调用方式有以下2种:

RPC:(Remote Produce Call)远程过程调用,类似的还有RMI(Remote Method Invoke).自定义数据格式,基于原生TCP通信,速度快,效率高.早期的webservice,现在热门的dubbo.都是RPC的典型代表

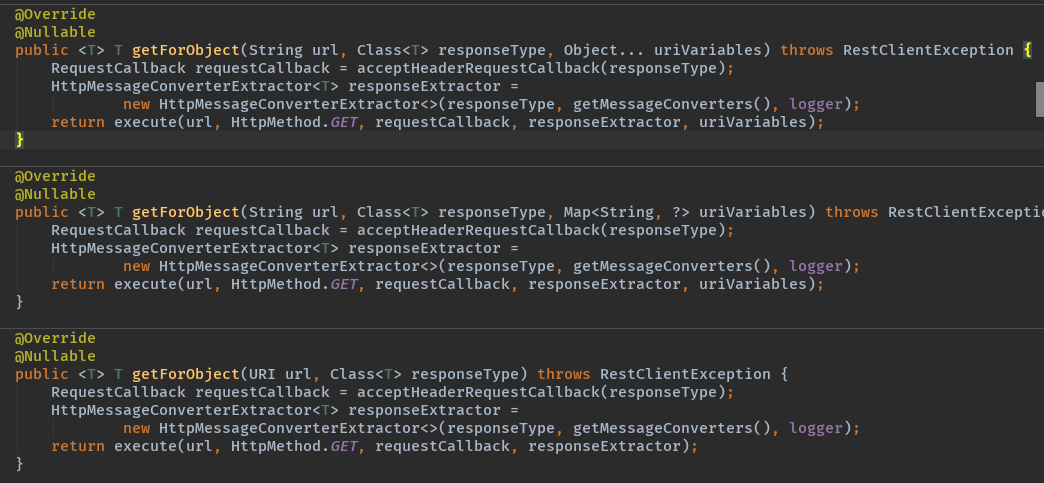
Http:Http是一种网络传输协议,基于TCP,规定了数据传输的格式.现在客户端浏览器与服务端通信的方式基本都是 采用http协议.也可以用来进行远程服务调用.缺点就是消息封装臃肿.优势是对服务的提供和调用方没有任何技术 限定,自由灵活,更符合微服务理念

## HTTP客户端工具

Spring提供了一个**RestTemplate**模板工具类,对基于Http的客户端进行了封装,并且实现了对象与json的序列化和反 序列化.非常方便.RestTemplate并没有限定Http的客户端类型,而是进行了抽象.

首先在项目中注入一个RestTemplate对象,可以在启动类位置注册,在服务的消费者端直接 @Autowired 并使用即可.通过RestTemplate的 getForObject() 方法,传递url地址及实体类的字节码,RestTemplate会自动发起请求,接收响应,并且帮我们对响应结果进行反序列化.

目前RestTemplate提供了三种重载方法.



# Eureka

## CAP原则

CAP原则又称CAP定理,指的是在一个分布式系统中,Consistency(一致性),Availability(可用性),Partition tolerance(分区容错性),三者不可兼得.

Eureka遵循的的就是AP原则.

## Eureka对比Zookeeper

作为服务注册中心,Eureka比Zookeeper好在哪里.

注明的CAP理论指出,一个分布式系统不可能同时满足C(一致性),A(可用性)和P(分区容错性).由于分区容错性P在是在分布式系统中必须要保证的,因此我们只能在A和C之间进行权衡.

因此:**Zookeeper保证的是CP,Eureka是AP**

### Zookeeper保证CP

在Zookeeper中会出现这样一种情况,当master节点因为网络问题故障与其他节点失去联系时候,剩余节点会重新进行leader选举.问题在于,选举leader的时间太长,30-120s,且选举期间整个zookeeper集群都是不可用的.这就导致在选择 具期间注册服务瘫痪.在云部署的环境下,因网络问题使得zookeeper集群失去master节点是较大概率会发生的事.虽然 服务能够最终恢复,但是漫长的选举时间导致的注册长期不可用是不能容忍的.

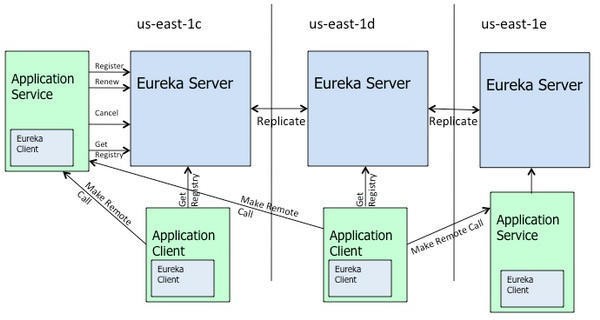
### Eureka保证AP

Eureka各个节点都是平等的,几个节点挂掉不会影响正常节点的工作,剩余节点依然可以提供注册和查询服务.而Eureka的客户端在向某个Eureka注册或如果发现连接失败,则会自动切换至其他节点,只要一台Eureka还在,就能保证 注册服务可用(保证可用性),只不过查到的信息可能不是最新的(不保证强一致性).除此之外,**Eureka还有一种自我保护 机制.如果在15分钟内超过85%的节点都没有正常的心跳,那么Eureka就认为客户端与注册中心出现了网络故障**,此时 会出现以下几种情况.

* + - 1. Eureka不再从注册列表中移除因为长时间没有收到心跳而应该过期的服务
      2. Eureka仍然能够接受新服务的注册和查询请求,但是不会被同步到其他节点上(即保证当前节点依然可用).
      3. 当网络稳定时,当前实例新的注册信息会被同步到其它节点中.

因此,Eureka可以很好的应对因网络故障导致部分节点失去联系的情况,而不会像Zookeeper那样使整个注册服务 瘫痪.

## Eureka原理



处于不同节点的eureka通过replicate(复制)进行数据同步. Application Sservice为服务提供者

Application Client为服务消费者Make Remote Call完成一次服务调用

当服务启动后向Eureka注册,Eureka Server会将注册信息向其他Eureka Server进行同步,当服务消费者要调用服务提供者,则向服务注册中心获取服务提供者地址,然后会将服务提供者地址缓存在本地,下次再调用时,则直接从本地缓存中 取,完成一次调用.

当服务注册中新Eureka Server检测到服务者因为宕机,网络原因不可用时,则在服务注册中心将服务置为DOWN状态,

并把当前服务提供者向订阅者发布,订阅过的服务消费者更新本地缓存.

服务提供者在启动后,周期性(默认30s)向Eureka Server发送心跳,以证明当前服务是可用状态.Eureka Server在一定时间(默认90s)未收到客户端的心跳,则认为服务宕机,注销该实例.

## Eureka注册中心程序

父pom文件(确定SpringCloud及SpringBoot的版本)

<parent>