微服务

简述：

对整体的一个项目进行服务拆分，

对外暴露接口，rest接口风格，http调用方式（跨语言），

自治

每个服务都是独立的，互不干扰

技术分离，前后端分离，

数据库分离，每个服务使用自己的数据源

部署独立，服务重启不影响其他的服务可复用可替换，降低耦合易维护。

服务间的调用：

远程调用种类

RPC（webservice、dubbo）基于Java，TCP通信速度快效率高

HTTP（spring cloud）跨语言

接口风格Restful

例子流程简述：

用户访问百度

——网络运营商（电信移动联通等）

——分配最近最优的一台或者一群Nginx服务器

——转发至以下网关

——网关到注册中心eureka拉取服务列表

——返回网关进行分配后直接转发到各对应的功能首个访问的服务

——在服务间的调用处理请求完成后返回用户端

注册中心直接分配给集群服务

采用Ribbon负载均衡分配

服务间调用

**SpringCloud**

简述：微服务框架

**调用restTemplate**

简述：

使用Spring中的 restTemplate调用其他服务

直接使用RestTemplate对象的getForObject 或者其他的方法 传入url参数和调用获取结果类型.class 参数

User user = this.restTemplate.getForObject("http://localhost:8088/user/1", User.class);

或者直接在Controller上加@RestController

**Eureka注册中心**

简述：

服务工程和调用服务的工程 中间通过注册中心eureka分配访问地址，eureka本身也设置成多个的集群模式，防止单个挂机。

由于一个服务可能是有多个的集群模式，Eureka在存储服务工程信息时，采用双Map形式Map<String,Map<String,ServiceInstance>>储存

配置流程：

1.配置pom.xml或者导入对应jar

2.创建eureka项目（集群）

3.配置文件，做一些对应的配置心跳，拉取服务列表周期，其他eureka服务的url等

4.给每个对应服务，配置上所有eureka服务的地址

**Ribbon负载均衡器**

简述：

负责从1到 + 的均衡分配。

解决负载均衡问题，比如一个服务调用另一个服务集群，采用轮询或者随机等方式均衡分配的访问集群接收的请求

使用Ribbon的拦截器，拦截resttemplate发出的请求链接，进行均衡的分配后转至目标集群服务中的一个。

配置流程：

引入相关jar包后

在restTemplate上加@LoadBalanced 自动在restTemplate加负载均衡

配置中可以配置均衡的方式 @RestController

后期直接集成进Hystrix中

**Hystrix熔断器**

简述：

在某个服务断连或者超市时进行切断操作，以节省资源，定时回访查看是否恢复。

调用一系列服务中其中一个宕机了，线程阻塞导致一个个请求卡在此处，占用总服务的线程资源导致雪崩。进行线程隔离，服务降级。每个服务分配一定的线程，占满后进行服务降级返回失败信息

配置流程：

在调用方做Hystrix处理与配置，

1.启动器加上@EnableHystrix 或者@EnableCircuitBreaker，或者使用@SpringCloudApplication整合了注册中心和雪崩处理

2.在控制器上添加

@DefaultProperties(defaultFallback = "queryByIdFallback" 定义雪崩返回的方法，

或在方法上添加

@HystrixCommand(fallbackMethod = "queryByIdFallback") //失败熔断处理

或

@HystrixCommand(commandProperties = {

//类上使用@DefaultProperties,则不定义fallbackMethod,只作启动用

@HystrixProperty(name="execution.isolation.thread.timeoutInMilliseconds",value="5000")

//设置超时时间

})

3.定义雪崩失败返回方法

氛围打开关闭休眠三个状态

当请求某个服务时，阈值比例50%的请求超时时，则开启熔断器接下来的请求全部失败返回

打开后休眠5秒后，放一部分请求进来测试是否能正常访问，

假如可以则关闭，假如不行则循环5秒

修改阈值比例为50%

修改休眠时间

//熔断属性测试

@HystrixCommand(commandProperties = {

@HystrixProperty(name="circuitBreaker.requestVolumeThreshold",value="10"),

//10次统计一轮，超过5次则打开熔断

@HystrixProperty(name="circuitBreaker.sleepWindowInMilliseconds",value="10000"),

//休眠时间窗 10s

@HystrixProperty(name="circuitBreaker.errorThresholdPercentage",value="60"),

//错误百分比

})

**Feign服务调用**

简述：

解决调用路径固定的问题

配置流程：

1.引用pom依赖

2.启动器添加@EnableFeignClients

3.编写接口 加@FeignClient(“服务名”) 给Feign

定义

@GetMapping("user/{id}")

User querybyid(@PathVariable("id") Long id);

4.加配置fegin.hystrix.enabled: = true 开启Fegin的Hystrix熔断功能

Feign请求压缩

文件上传下载可以使用，有效减少请求数据量，加强上传下载的效率，平时可以不做

Feign 日志级别

打印Feign日志信息

**Zuul服务网关**

简述：

1.网关：作为Ngix分配的入口

功能：限流，权限

流程步骤：

1. 引入pom依赖

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-starter-netflix-zuul</artifactId>

</dependency>

2. 启动器类加@EnableZuulProxy，@EnableZuulService中有大量过滤器不使用

3. 进行配置

**zuul:  
 routes:  
 user-service:** /user/\*\* *#直接简化服务后直接默认配置  
 #path: /user-service/\*\* #匹配拦截user为前缀的所有路径  
 #url: http://127.0.0.1:8081 #http://localhost:10010/user-service/user/1  
 #serviceId: user-service #不使用固定url的写法，用eureka拉取服务列表，会一并实现负载均衡等  
 #strip-prefix: true #默认true 去除前缀名 由 #http://localhost:10010/user-service/user/1 变为#http://localhost:10010/user/1* **ignored-services:** *#配置忽略的服务，主要禁用默认一套，防止访问权限问题* - consumer-demo  
 - user-service  
 **prefix:** /api *#在 /user/user/1 改为 /api/user/user/1 此前缀在服务名前*

过滤器、权限控制

ZuulFilter过滤器顶级父类，其中有四个方法

abstract public String filterType(); //过滤器类型

pre ：请求在被路由执行之前

routing：在路由请求时调用

post：在routing和error过滤器之后调用

error：处理请求时发生错误调用

abstract public int filterOrder(); // 过滤器顺序 来自实现的接口IZuulFilter 越大优先级越低

boolean shouldFilter(); //要不要过滤 来自实现的接口IZuulFilter

Object run() throws ZuulException; // 过滤逻辑 来自实现的接口IZuulFilter

**Nginx**

简述：

用户直接访问Nginx，由Nginx进行分配至网关

访问百度

——网络运营商（电信移动联通等）

——分配最近最优的一台或者一群Nginx服务器

——转发至以下网关

——

**SpringCloud-Config配置中心**

简述：

与Git的Webhook钩子连用，从Git上获取一个总的配置管理，在服务请求的时候给予对应的配置，

或者通过spring-cloud-bus消息总线在有更新时进行主动推送给对应微服务。

**Spring-cloud-bus 消息总线**

**Spring-cloud-stream 消息通信**

**Spring-cloud-hystrix-dashboard 容错统计，图形化界面**

**Spring-cloud-sleuth 链路追踪**

简述：

链路追踪，结合Zipkon

**其他知识**

Ctrl + shift + > ( <) 加减字号