**Spring Cloud与Docker微服务架构实战笔记**

**151150072 姚政**

基本概念： 2

项目Step: 2

Chapter1-5 2

Chapter6 3

Chapter7 4

Chapter8 4

GitHub：<https://github.com/yaozheng1998/MovieTicket>

# 基本概念：

1. 微服务：敏捷响应平台；组件化，松耦合，自治，去中心化；每个服务自己的进程，API交互，可全自动部署；不同的服务可用 不同语言开发，使用不同的数据存储技术。sc轻轻松松几行代码就完成了熔断、均衡负责、服务中心的各种平台功能
2. 微服务设计原则：单一指责原则，服务自治原则，轻量级通信机制，微服务粒度

Rest：<https://www.cnblogs.com/imyalost/p/7923230.html>

1. Eureka: 实现微服务的注册与发现；集群；
2. Ribbon：控制HTTP和TCP客户端的行为;请求均匀分布到用户微服务节点上实现负载均衡；
3. Feign: 便捷优雅调用HTTP API；仅需创建接口，添加注解;实际业务中会根据不同的角色，执行不同的业务操作，可以自定义完成复杂的服务鉴权；
4. Hystrix：实现微服务的容错（高负载场景下，服务提供者的缓慢响应会使得消费者强制等待，直到提供者响应或超时）；

* 为网络请求设置超时，让资源尽快释放
* 使用断路器模式，统计一段时间内调用失败的次数，决定正常请求依赖的服务还是直接返回；还能够恢复正常状态；

Hystrix实现方式：

包裹请求，命令模式；跳闸机制；资源隔离；监控；回退机制；自我修复（半开）

1. Zuul：微服务网关是介于客户端和服务端之间的中间层，所有的外部请求都先经过微服务网关；客户端只需与网关交互；
2. RxJava:依赖于4个基础接口，Publisher，Subscriber，Subscription，Processor

Publisher发出一系列的时间，Subscriber负责和处理这些事件；

1. Spring Cloud Config：为分布式系统外部化配置提供了服务器和客户端的支持；

Config Server：可横向扩展、集中式的配置服务器，用于集中管理应用程序各个环境下的配置；

Config Client：用于操作存储在Config Server中的配置属性；

1. SpringCloud有一个“引导上下文”的概念，是主应用程序的父上下文，负责从配置服务器加载配置属性，以及解密外部配置文件中的属性。
2. Spring Cloud Bus: 使用轻量级的消息代理连接分布式系统的节点，可以广播传播状态的更改（配置更新）或其他管理指令；
3. 微服务之间通过网络进行通信，能够跟踪每个请求，了解请求的详细信息，可以更好地分析系统瓶颈、解决系统问题；

Spring Cloud Sleuth为Spring Cloud提供了分布式跟踪的解决方案；

# 项目Step:

## Chapter1-5

1. WARNING: Your ApplicationContext **is** unlikely **to** start due **to** a @ComponentScan **of** the **default** **package**.

解决：[main启动类需要放在一个包里]

1. schema.sql文件的路径问题；

解决：classpath

1. Spring Boot Actuator
2. EurekaServer在pom.xml中的配置
3. <http://localhost:8019/log-instance> 页面空白问题
4. Controller调用Feign接口报错：Could not Autowire.

解决：此错误提示不会产生影响，1.ij本身问题；2.导入包的问题；只需设置降低Autowired检测的级别即可；

1. SpringBoot启动多个实例？

解决：<https://blog.csdn.net/forezp/article/details/76408139>

## Chapter6

1. Feign报错 Hystrix熔断器会判断服务是否能通，超时则错误返回；

Nested exception is com.netflix.hystrix.exception.HystrixRuntimeException:

UserFeignClient#findById(Long) timed-out and no fallback available.] with root cause

解决：<https://www.cnblogs.com/EasonJim/p/7722372.html>

采用了F3: feign.hystrix.enabled: false

1. Collection<? Extends GrantedAuthority>

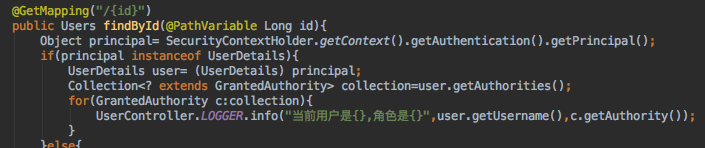
声明该Collection模版类的类型变量可以是所有GrantedAuthority的子类；

1. Logger VS System.out logger打印出详细信息，可以配置，还能保存到日志文件；sys在正式项目基本被禁用，不雅观，还影响效率；【指slj4的logger】

再看6.7 Feign的日志，Feign的logger只对DEBUG级别作出响应；

1. 自定义feign实现服务鉴权，登录无效问题

解决：在WebSecurityConfig配置文件中，isEnabled等参数默认为false，修改为true；而且配置文件需要放在com.yz文件夹下。



user.getAuthorties是Collection集合类；user一个，authority多个？

解决：这里我们只添加了一个auth！

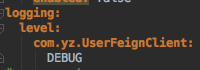


1. 6.7报错：

feign.FeignException: status 401 reading UserFeignClient#findById(Long); content:

{"timestamp":1522402291489,"status":401,"error":"Unauthorized","message":"Full authentication is required to access this resource","path":"/1"}

解决：因为我运行了with-auth，需要验证安全性



需要根据自己的项目结构配置。

## Chapter7

1. FiegnClient 注解的fallback属性，未成功回退；而是显示服务器500

解决：yml配置文件中未启用hystrix

../../屏幕快照%202018-03-31%20下午2.15.22.png

1. 使用消息中间件RabbitMQ实现Turbine监控多个微服务还未实现

## Chapter8

1. ZuulApplication报错java.lang.IllegalStateException: Failed to load property source from location 'classpath:/application.yml'

解决：jar

../../屏幕快照%202018-03-31%20下午4.04.31.png

1. pom.xml不加parent时，记得写<version></version>版本号;fileUpload也要注意版本号的问题
2. 上传文件报错：

{"timestamp":1522639187550,"status":400,"error":"Bad Request","exception":"org.springframework.web.multipart.support.MissingServletRequestPartException","message":"Required request part 'file' is not present","path":"/upload"}

解决：file=@filename 路径问题

1. zuul aggregation运行时报错：java.lang.reflect.UndeclaredThrowableException: Failed to invoke event listener method

## Chapter9

1. 访问远程仓库时出错：org.eclipse.jgit.errors.NoRemoteRepositoryException:
2. Mac下java安装路径 /Library/JAVA/…

## Chapter10