**Zuul**

**特点：**

**1 路由+过滤器 = zuul**

**2 核心是一系列的过滤器**

**我的理解就是分布式过滤器对所有请求进行统一管理的方式。**

**Zuul的四种过滤器API**

**前置：Pre (可以用作限流，鉴定权限，参数校验，请求转发)**

**后置：Post （统计 日志）**

**路由：Route**

**错误：Error**

**高可用 ：注册多个节点**

**从实用性角度出发可以nginx和zuul的混搭，达到取长补短的目的。**

**实战：**

在启动类上添加路由注解

@EnableZuulProxy

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*坑\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

在springcloud Finchley.BUILD-SNAPSHOT

Springboot 2.1.0.BUILD-SNAPSHOT

不能运行，有bug。

但是在视频中的版本就可以运行 springcloud Finchley.M2

Springboot 2.0.0.M3

路由(访问)规则为：端口号/服务名称/接口

若是想自定义路由，按如下配置

**zuul:  
 routes:  
 myProuct:(此处为自定义，可以瞎写)  
 path:** /myProduct/\*\*  
 **serviceId:** product

#简洁写法 Producct: /myProduct/\*\*

**ignored-patterns: #排除某些路由**

查看路由规则时的权限问题，加上如下注解

**management:  
 security:  
 enabled:** false

不配置的情况下是拿不到cookie的，如果想要拿到cookie的话需要排除敏感头的过滤。加上如下配置

sensitiveHead：

动态路由的实现：

1 动态路由就是指将配置放在配置中心。

2 也可以写在代码里。

@ConfigurationProperties("zuul")  
@RefreshScope  
public ZuulProperties zuulProperties(){  
 return new ZuulProperties();  
}

常量类叫做FilterConstants，官方推荐用常量来写

Filterorder，顺序越小，优先级越高

限流的解决方案:

令牌桶算法。固定速率往令牌桶中放入令牌，客户端请求只有领取到令牌才能继续通行。

详细请看图

可以用谷歌的开源组件

private static final RateLimiter *RATE\_LIMITER* = RateLimiter.*create*(100); //每秒放多少个令牌

使用zuul的过滤器的时候 继承ZuulFilter，run方法是要执行的过程，自己封装了request和response对象的获取方法。

**Zuul的跨域问题**

解决方案： 1在被调用的类或者方法上加@CrossOrigin注解

2 在Zuul里增加CorsFilter过滤器