1. 创建一个普通的Spring Boot工程名为api-gateway，然后添加相关依赖，主要添加两个依赖spring-cloud-starter-zuul和spring-cloud-starter-eureka，spring-cloud-starter-zuul依赖中则包含了ribbon、hystrix、actuator等
2. 在入口类上添加@EnableZuulProxy注解表示开启Zuul的API网关服务功能
3. 配置路由规则

application.properties文件中的配置可以分为两部分，一部分是Zuul应用的基础信息，还有一部分则是路由规则

# 基础信息配置

spring.application.name=api-gateway

server.port=2006

# 路由规则配置

zuul.routes.api-a.path=/api-a/\*\*

zuul.routes.api-a.serviceId=feign-consumer

# API网关也将作为一个服务注册到eureka-server上

eureka.client.service-url.defaultZone=http://localhost:1111/eureka/

这里配置了路由规则所有符合/api-a/\*\*的请求都将被转发到feign-consumer服务上，至于feign-consumer服务的地址到底是什么则由eureka-server去分析

**请求过滤**

1. 定义一个过滤器继承自ZuulFilter



1.filterType方法的返回值为过滤器的类型，过滤器的类型决定了过滤器在哪个生命周期执行，pre表示在路由之前执行过滤器，其他可选值还有post、error、route和static，当然也可以自定义。  
2.filterOrder方法表示过滤器的执行顺序，当过滤器很多时，这个方法会有意义。  
3.shouldFilter方法用来判断过滤器是否执行，true表示执行，false表示不执行，在实际开发中，我们可以根据当前请求地址来决定要不要对该地址进行过滤，这里我直接返回true。  
4.run方法则表示过滤的具体逻辑，假设请求地址中携带了login参数的话，则认为是合法请求，否则就是非法请求，如果是非法请求的话，首先设置ctx.setSendZuulResponse(false);表示不对该请求进行路由，然后设置响应码和响应值。这个run方法的返回值在当前版本(Dalston.SR3)中暂时没有任何意义，可以返回任意值。

1. 在入口类中配置相关的Bean

@Bean  
PermisFilter permisFilter() {  
 return new PermisFilter();  
}