# 网关的概念

网关相当于一个网络服务架构的入口，所有网络请求必须通过网关转发到具体的服务。

# 网关的作用

统一管理微服务请求，权限控制、负载均衡、路由转发、监控、安全控制、黑名单和白名单等。

# 基本概念

1、Zuul是对SpringCloud提供的成熟的路由方案，他会根据请求的路径不同，网关会定位到指定的微服务，并代理请求到不同的微服务接口，他对外隐蔽了微服务的真正接口地址。 三个重要概念：动态路由表，路由定位，反向代理。

动态路由表：Zuul支持Eureka路由，手动配置路由，这两种都支持自动更新。

路由定位：根据请求路径，Zuul有自己的一套定位服务规则以及路由表达式匹配。

反向代理：客户端请求到路由网关，网关受理之后，在对目标发送请求，拿到响应之后在给客户端。

2、它可以和Eureka、Ribbon、Hystrix等组件配合使用。

# 应用场景

对外暴露，权限校验，服务聚合，日志审计等。

# 功能

Zuul的主要功能是路由转发和过滤器。路由功能是微服务的一部分，比如／api/user转发到到user服务，/api/shop转发到到shop服务。zuul默认和Ribbon结合实现了负载均衡的功能。

# 代码示例

## 1、创建service-zuul项目

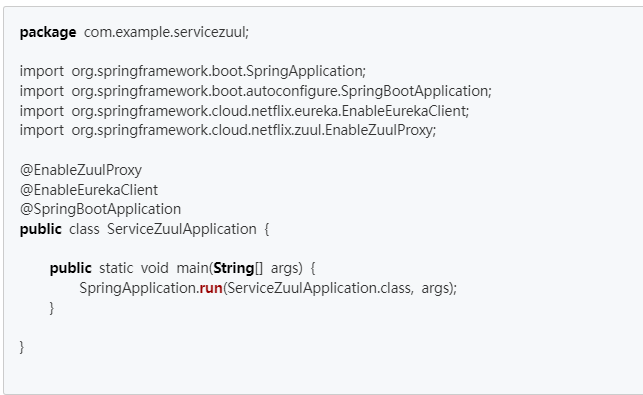
### pom.xml文件





### 编写入口类

在其入口applicaton类加上注解@EnableZuulProxy，开启zuul的功能。



### 编写配置文件

配置文件application.yml加上以下的配置代码：



首先指定服务注册中心的地址为http://localhost:8761/eureka/，服务的端口为10086，服务名为service-zuul；以/api-customer/ 开头的请求都转发给eureka-ribbon服务；以/api-admin/开头的请求都转发给service-feign服务；

依次eureka-service、eureka-client、eureka-ribbon、service-feign、service-zuul运行

打开浏览器访问：

http://localhost:10086/api-customer/helloRibbon?name=helloRibbonZuul ;

浏览器显示：



打开浏览器访问：

http://localhost:10086/api-admin/helloRibbon?name=feigeZuul

浏览器显示：



这说明zuul起到了路由的作用

# 服务过滤

zuul不仅只是路由，并且还能过滤，做一些安全验证。继续改造工程；



说明：

1、filterType：返回一个字符串代表过滤器的类型，在zuul中定义了四种不同生命周期的过滤器类型，具体如下：

pre：路由之前

routing：路由之时

post： 路由之后

error：发送错误调用

2、filterOrder：过滤的顺序

3、shouldFilter：这里可以写逻辑判断，是否要过滤，本文true,永远过滤。

4、run：过滤器的具体逻辑。可用很复杂，包括查sql，nosql去判断该请求到底有没有权限访问。

这时访问：

http://localhost:10086/api-admin/helloFeign?name=feigeZuul ；

网页显示：



访问

http://localhost:10086/api-customer/helloRibbon?name=helloRibbonZuul&token=3

网页显示：



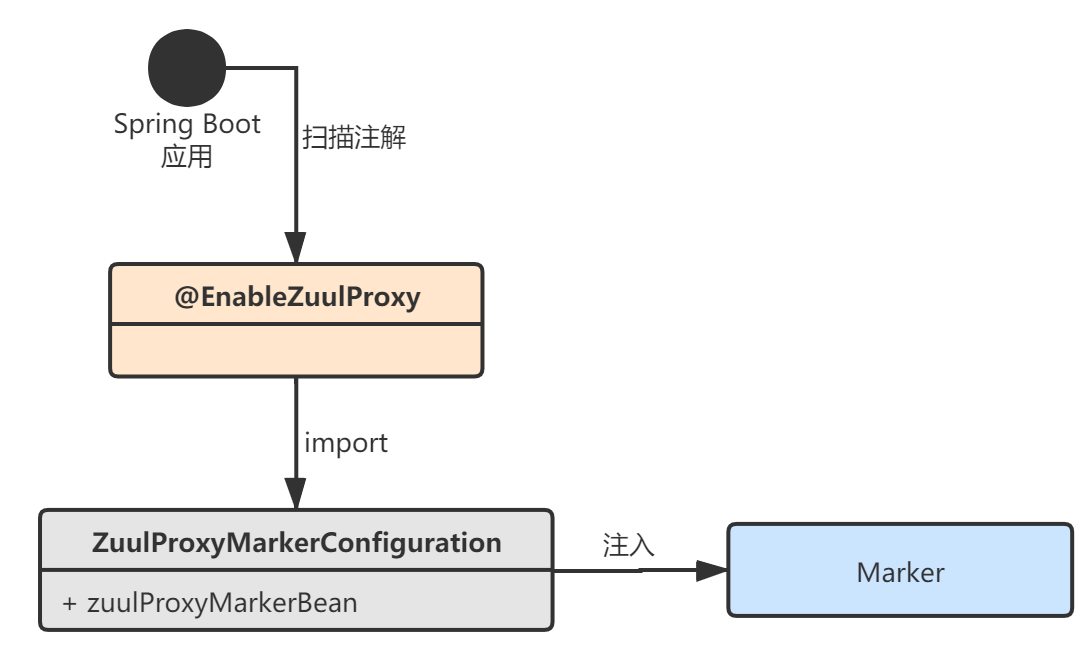
# 初始化流程

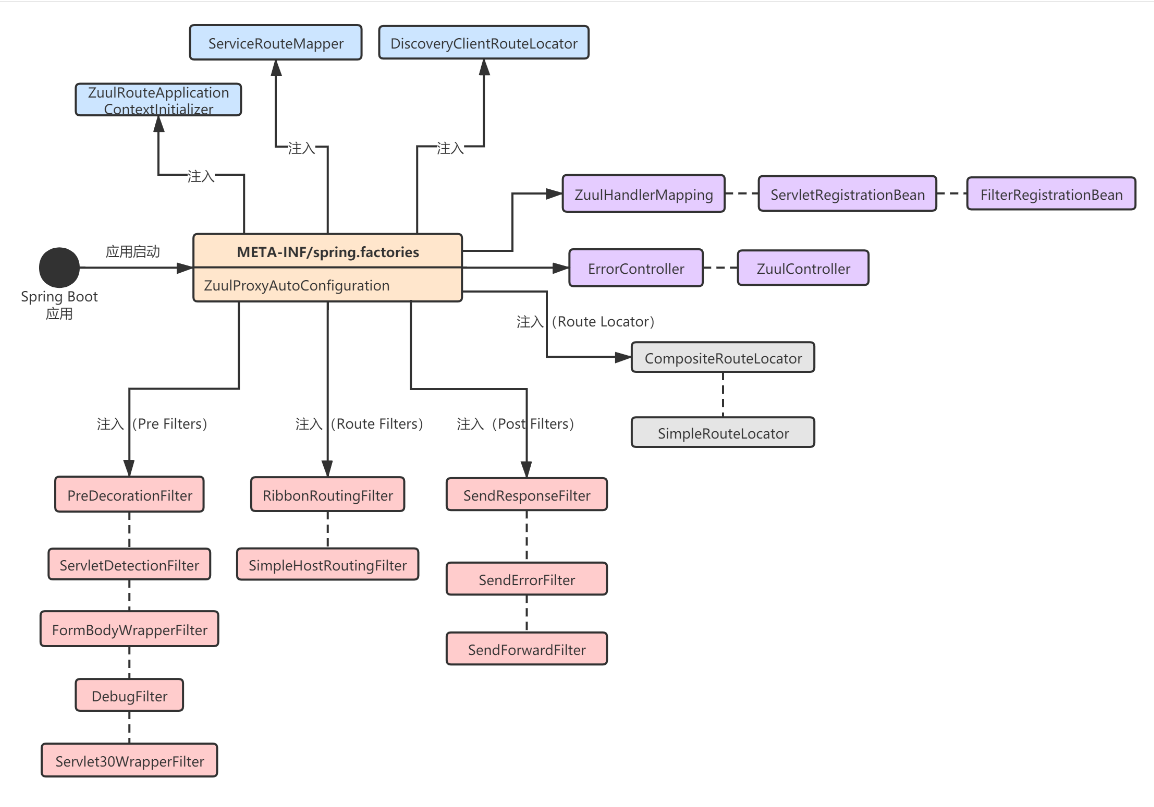
Spring Boot应用启动后，会去两个地方加载配置：

1、扫描到@EnableZuulProxy注解，执行注解中的@Import操作。

2、加载spring-cloud-netflix-zuul源码包META-INF目录下的spring.factories配置。

我用下面这两张图表述Spring Cloud Netflix Zuul的初始化流程。第一张图很简单，就是注入了一个标记对象，后续自动装配时会根据是否存在该对象选择并注入合理的Bean；第二张图主要描述了Spring Cloud向容器中注入的核心Zuul组件：





# 如何实现动态Zuul网关路由转发

通过path配置拦截请求，通过ServiceId到配置中心获取转发的服务列表，Zuul内部使用Ribbon实现本地负载均衡和转发。

# Zuul网关如何搭建集群

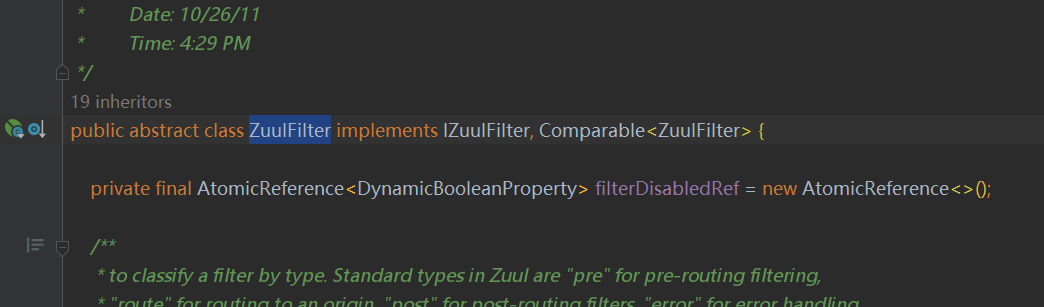
使用Nginx的upstream设置Zuul服务集群，通过location拦截请求并转发到upstream，默认使用轮询机制对Zuul集群发送请求。

# 常用的类

## ZuulFilter

### 基本概念

### 源码



### 方法

#### run()

过滤器的具体业务逻辑

#### shouldFilter()

判断过滤器是否有效

#### filterOrder()

过滤器执行顺序

#### filterType()

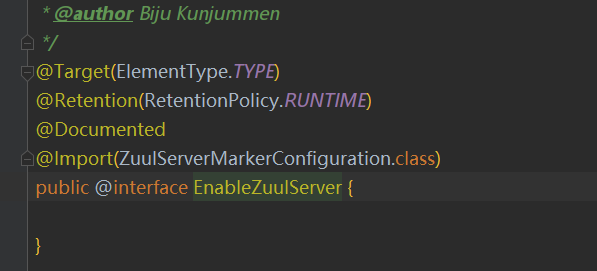
过滤器拦截位置

# 常用的注解

## @EnableZuulServer

### 基本概念

### 源码

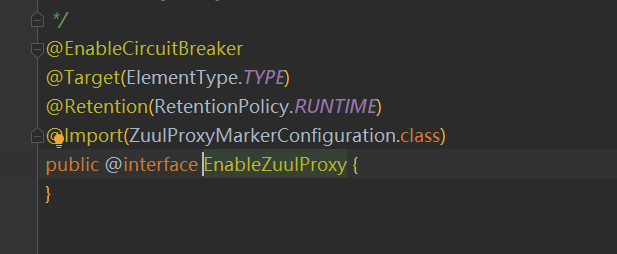


## @EnableZuulProxy

### 基本概念

@EnableZuulProxy 是 Spring Cloud Netflix Zuul 的核心注解，用于启用 API 网关功能。

### 源码



### 作用

#### 1、启用 Zuul 代理服务器

将当前应用转换为 API 网关

#### 2、自动配置路由规则

根据服务发现自动创建路由映射

#### 3、集成服务发现

与 Eureka/Consul 等注册中心配合工作

### 核心功能

#### 1、请求路由

将外部请求转发到后端微服务

#### 2、负载均衡

集成 Ribbon 实现客户端负载均衡

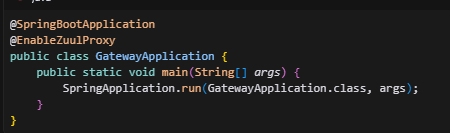
#### 3、过滤器链

支持 pre、route、post、error 四种过滤器

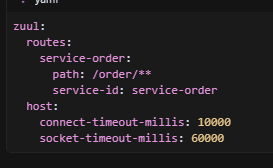
#### 4、熔断保护

集成 Hystrix 提供容错机制

### 基本用法



### 配置示例



### 注意事项