

# 乐字节教育高级架构课程

正所谓“授人以鱼不如授人以渔”，你们想要的 **Java 学习资料** 来啦！

不管你是学生，还是已经步入职场的同行，希望你们都要珍惜眼前的学习机会，奋斗没有终点，知识永不过时。

扫描下方二维码即可领取



乐字节晓啡

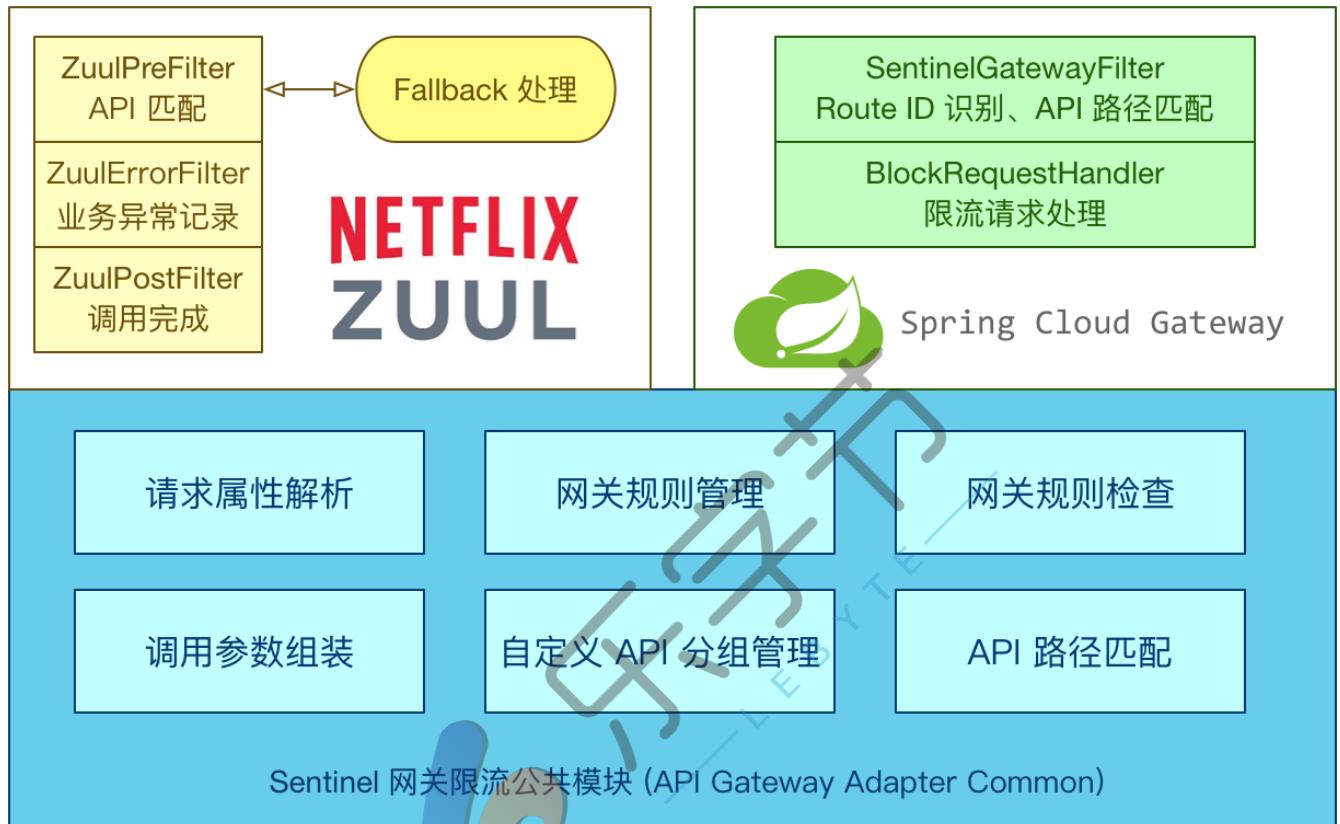


乐字节官方交流群



## Zuul 和 Sentinel 整合

Sentinel 支持对 Spring Cloud Gateway、Netflix Zuul 等主流的 API Gateway 进行限流。



官网文档:

- <https://github.com/alibaba/spring-cloud-alibaba/wiki/Sentinel>
- <https://github.com/alibaba/Sentinel/wiki/网关限流#zuul-1x>
- 、

## 网关限流和容错

### 创建项目

创建 `zuul-server-sentinel` 项目。

### 添加依赖

单独使用添加 `sentinel-zuul-adapter` 依赖即可。

若想跟 Sentinel Starter 配合使用，需要加上 `spring-cloud-alibaba-sentinel-gateway` 依赖，同时需要添加 `spring-cloud-starter-netflix-zuul` 依赖来让 `spring-cloud-alibaba-sentinel-gateway` 模块里的 Zuul 自动化配置类生效。

同时请将 `spring.cloud.sentinel.filter.enabled` 配置项置为 `false`（若在网关流控控制台上看到了 URL 资源，就是此配置项没有置为 `false`）。

```
<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>com.example</groupId>
  <artifactId>zuul-server-sentinel</artifactId>
  <version>1.0-SNAPSHOT</version>

  <!-- 继承父依赖 -->
  <parent>
    <groupId>com.example</groupId>
    <artifactId>zuul-demo</artifactId>
    <version>1.0-SNAPSHOT</version>
  </parent>

  <!-- 项目依赖 -->
  <dependencies>
    <!-- spring cloud netflix zuul 依赖 -->
    <dependency>
      <groupId>org.springframework.cloud</groupId>
      <artifactId>spring-cloud-starter-netflix-zuul</artifactId>
    </dependency>
    <!-- netflix eureka client 依赖 -->
    <dependency>
      <groupId>org.springframework.cloud</groupId>
      <artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>
    </dependency>
    <!-- 单独使用 -->
    <!-- sentinel zuul adapter 依赖 -->
    <dependency>
      <groupId>com.alibaba.csp</groupId>
      <artifactId>sentinel-zuul-adapter</artifactId>
    </dependency>
    <!-- 和 Sentinel Starter 配合使用 -->
    <!--
    <dependency>
      <groupId>com.alibaba.cloud</groupId>
      <artifactId>spring-cloud-starter-alibaba-sentinel</artifactId>
    </dependency>
    <dependency>
      <groupId>com.alibaba.cloud</groupId>
      <artifactId>spring-cloud-alibaba-sentinel-gateway</artifactId>
    </dependency>
  </dependencies>
</project>
```

```
        </dependency>
    -->
</dependencies>

</project>
```

## 配置文件

```
server:
  port: 9001 # 端口

spring:
  application:
    name: zuul-server-sentinel # 应用名称
  cloud:
    sentinel:
      filter:
        enabled: false

# 配置 Eureka Server 注册中心
eureka:
  instance:
    prefer-ip-address: true # 是否使用 ip 地址注册
    instance-id: ${spring.cloud.client.ip-address}:${server.port} # ip:port
  client:
    service-url: # 设置服务注册中心地址
    defaultZone: http://localhost:8761/eureka/,http://localhost:8762/eureka/
```

## 网关服务配置类

配置网关服务过滤器和网关限流规则。

```
package com.example.config;

import com.alibaba.csp.sentinel.adapter.gateway.common.rule.GatewayFlowRule;
import com.alibaba.csp.sentinel.adapter.gateway.common.rule.GatewayRuleManager;
import com.alibaba.csp.sentinel.adapter.gateway.zuul.fallback.ZuulBlockFallbackManager;
import com.alibaba.csp.sentinel.adapter.gateway.zuul.filters.SentinelZuulErrorFilter;
import com.alibaba.csp.sentinel.adapter.gateway.zuul.filters.SentinelZuulPostFilter;
import com.alibaba.csp.sentinel.adapter.gateway.zuul.filters.SentinelZuulPreFilter;
import com.netflix.zuul.ZuulFilter;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;

import javax.annotation.PostConstruct;
```

```
import java.util.HashSet;
import java.util.Set;

/**
 * 网关服务配置类
 */
@Configuration
public class ZuulConfig {

    // 底层继承了 ZuulFilter
    @Bean
    public ZuulFilter sentinelZuulPreFilter() {
        // We can also provider the filter order in the constructor.
        return new SentinelZuulPreFilter();
    }

    // 底层继承了 ZuulFilter
    @Bean
    public ZuulFilter sentinelZuulPostFilter() {
        return new SentinelZuulPostFilter();
    }

    // 底层继承了 ZuulFilter
    @Bean
    public ZuulFilter sentinelZuulErrorFilter() {
        return new SentinelZuulErrorFilter();
    }

    /**
     * Spring 容器初始化的时候执行该方法
     */
    @PostConstruct
    public void doInit() {
        // 加载网关限流规则
        initGatewayRules();
    }

    /**
     * 网关限流规则
     */
    private void initGatewayRules() {
        Set<GatewayFlowRule> rules = new HashSet<>();
        /**
         * resource: 资源名称, 可以是网关中的 route 名称或者用户自定义的 API 分组名称
         * count: 限流阈值
         * intervalSec: 统计时间窗口, 单位是秒, 默认是 1 秒
         */
        rules.add(new GatewayFlowRule("order-service")
            .setCount(3) // 限流阈值
            .setIntervalSec(60)); // 统计时间窗口, 单位是秒, 默认是 1 秒
        // 加载网关限流规则
        GatewayRuleManager.loadRules(rules);
    }
}
```

```
}
```

## 启动类

ZuulServerSentinelApplication.java

```
package com.example;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.cloud.netflix.zuul.EnableZuulProxy;

@SpringBootApplication
// 开启 zuul 注解
@EnableZuulProxy
// 开启 EurekaClient 注解, 目前版本如果配置了 Eureka 注册中心, 默认会开启该注解
// @EnableEurekaClient
public class ZuulServerSentinelApplication {

    public static void main(String[] args) {
        SpringApplication.run(ZuulServerSentinelApplication.class, args);
    }

}
```

## 访问

多次访问: <http://localhost:9001/order-service/order/1> 触发限流后会返回固定的提示:

