# Sentinel入门及安装

## 什么是Sentinel

Sentinel (分布式系统的流量防卫兵) 是阿里开源的一套用于**服务容错**的综合性解决方案。它以流量为切入点, 从**流量控制、熔断降级、系统负载保护**等多个维度来保护服务的稳定性。

### Sentinel 具有以下特征:

**丰富的应用场景**：Sentinel 承接了阿里巴巴近 10 年的双十一大促流量的核心场景, 例如秒杀（即突发流量控制在系统容量可以承受的范围）、消息削峰填谷、集群流量控制、实时熔断下游不可用 应用等。

**完备的实时监控**：Sentinel 提供了实时的监控功能。通过控制台可以看到接入应用的单台机器秒级数据, 甚至 500 台以下规模的集群的汇总运行情况。

**广泛的开源生态：**Sentinel 提供开箱即用的与其它开源框架/库的整合模块, 例如与 Spring Cloud、Dubbo、gRPC 的整合。只需要引入相应的依赖并进行简单的配置即可快速地接入Sentinel。

**完善的 SPI 扩展点**：Sentinel 提供简单易用、完善的 SPI 扩展接口。您可以通过实现扩展接口来快速地定制逻辑。例如定制规则管理、适配动态数据源等。

### Sentinel 分为两个部分:

核心库（Java 客户端）不依赖任何框架/库,能够运行于所有 Java 运行时环境，同时对 Dubbo / Spring Cloud 等框架也有较好的支持。

控制台（Dashboard）基于 Spring Boot 开发，打包后可以直接运行，不需要额外的 Tomcat 等应用容器。

## 微服务集成Sentinel

为微服务集成Sentinel非常简单, 只需要加入Sentinel的依赖即可

1. 在pom.xml中加入下面依赖

<dependency>

<groupId>com.alibaba.cloud</groupId>

<artifactId>spring-cloud-starter-alibaba-sentinel</artifactId>

</dependency>

1. 编写一个Controller测试使用

## 安装Sentinel控制台

@RestController @Slf4j

public class OrderController3 { @RequestMapping("/order/message1") public String message1() {

return "message1";

}

@RequestMapping("/order/message2") public String message2() {

return "message2";

}

}



### 安装Sentinel控制台

Sentinel 提供一个轻量级的控制台, 它提供机器发现、单机资源实时监控以及规则管理等功能。

1 下载jar包,解压到文件夹https://github.com/alibaba/Sentinel/releases 2 启动控制台

# 直接使用jar命令启动项目(控制台本身是一个SpringBoot项目)

java -Dserver.port=8080 -Dcsp.sentinel.dashboard.server=localhost:8080 -Dproject.name=sentinel-dashboard -jar sentinel-dashboard-1.7.0.jar

3 修改shop-order ,在里面加入有关控制台的配置

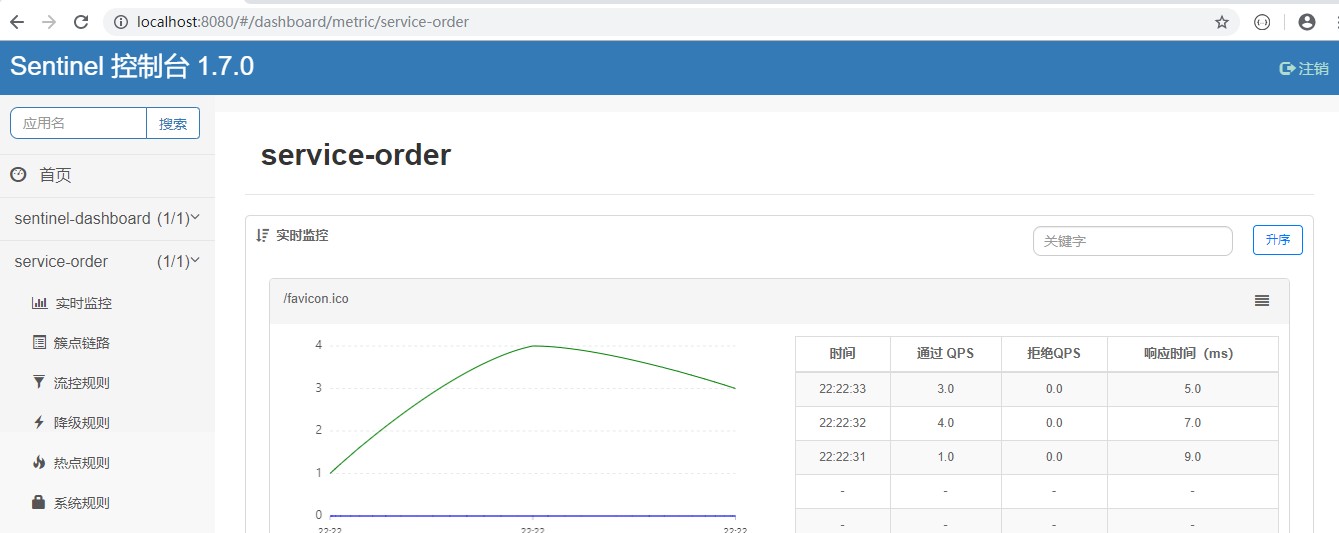
spring: cloud:

sentinel: transport:

port: 9999 #跟控制台交流的端口,随意指定一个未使用的端口即可

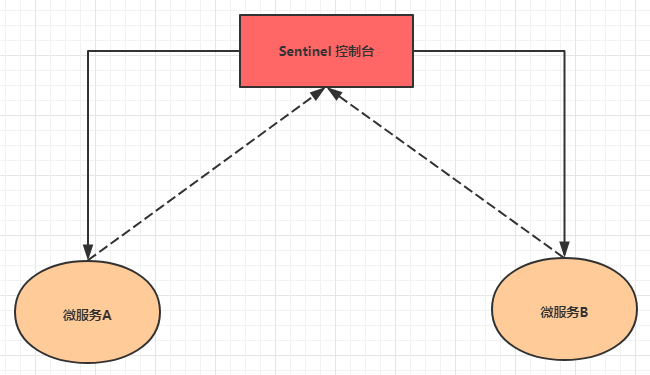
dashboard: localhost:8080 # 指定控制台服务的地址

第4步: 通过浏览器访问localhost:8080 进入控制台 ( 默认用户名密码是 sentinel/sentinel )



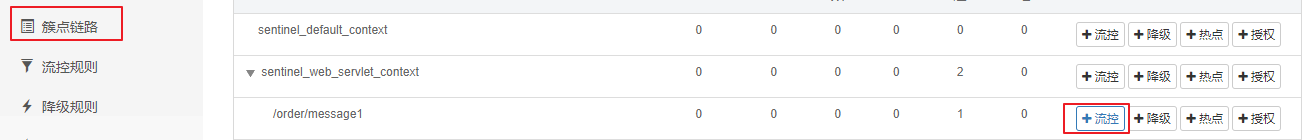
### 补充：了解控制台的使用原理

Sentinel的控制台其实就是一个SpringBoot编写的程序。我们需要将我们的微服务程序注册到控制台上, 即在微服务中指定控制台的地址, 并且还要开启一个跟控制台传递数据的端口, 控制台也可以通过此端口调用微服务中的监控程序获取微服务的各种信息。



## 实现一个接口的限流

1. 通过控制台为message1添加一个流控规则





1. 通过控制台快速频繁访问, 观察效果

