# 【Spring Cloud Sleuth】

**【主要内容】**

1. Sleuth简介
2. Zipkin搭建
3. Sleuth快速入门

**【学习目标】**

|  |  |
| --- | --- |
| 知识点 | 要求 |
| Sleuth简介 | 了解 |
| Zipkin搭建 | 掌握 |
| Sleuth快速入门 | 掌握 |

# 什么是链路追踪

官网：<https://spring.io/projects/spring-cloud-sleuth>

链路追踪就是：**追踪微服务的调用路径**

# 链路追踪的由来

在微服务框架中，一个由客户端发起的请求在后端系统中会经过多个不同的服务节点调用来协同产生最后的请求结果，每一个请求都会开成一条复杂的分布式服务调用链路，链路中的任何一环出现高延时或错误都会引导起整个请求最后的失败。

# 分布式链路调用的监控

sleuth+zipkin（zipkin就是一个可视化的监控控制台）

Zipkin 是 Twitter 的一个开源项目，允许开发者收集 Twitter 各个服务上的监控数据，并提供查询接口。

该系统让开发者可通过一个 Web 前端轻松的收集和分析数据，例如用户每次请求服务的处理时间等，可方便的监测系统中存在的瓶颈。

# zipkin

## 下载zipkin

SpringCloud从F版起已不需要自己构建Zipkin server了，只需要调用jar包即可

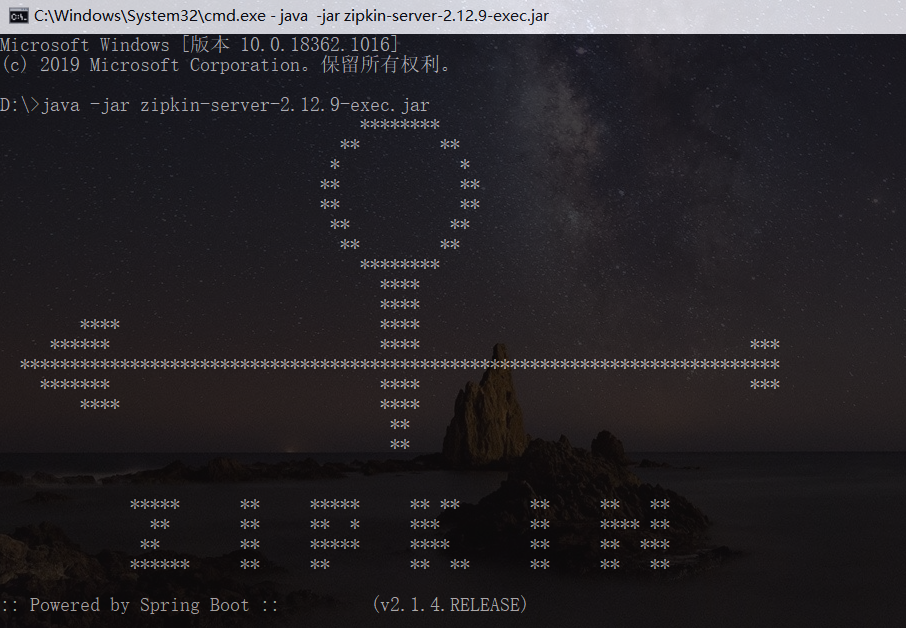
<https://dl.bintray.com/openzipkin/maven/io/zipkin/java/zipkin-server/>

<https://dl.bintray.com/openzipkin/maven/io/zipkin/java/zipkin-server/2.12.9/>



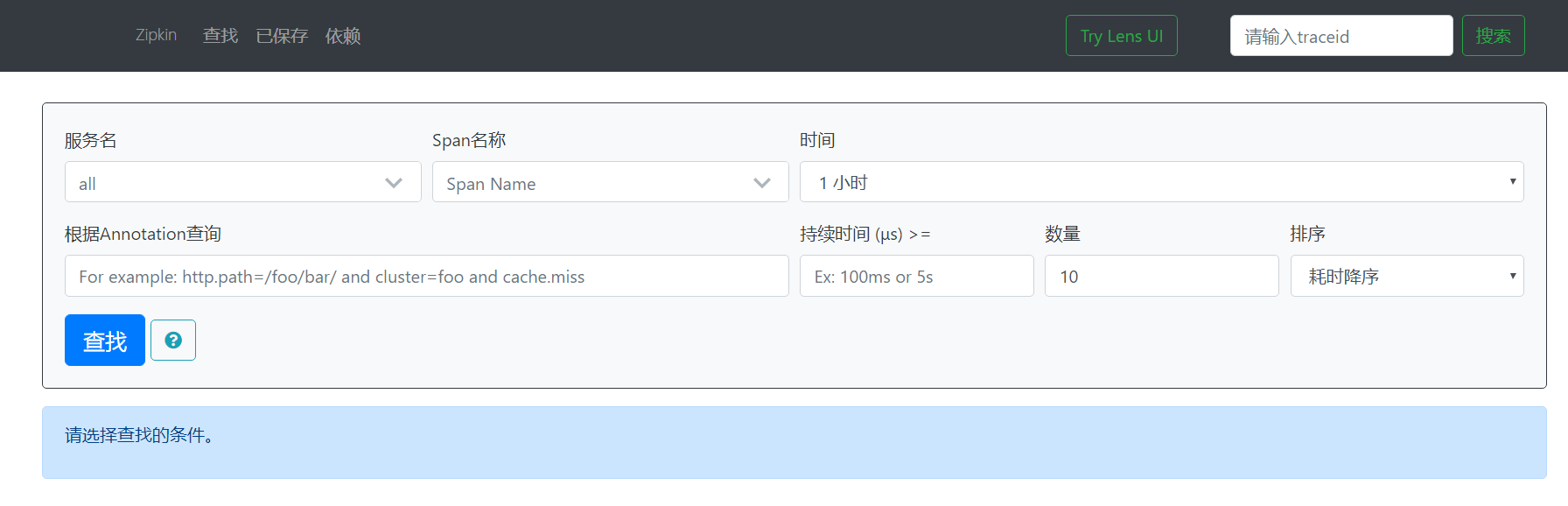
## 运行zipkin

java -jar zipkin-server-2.12.9-exec.jar



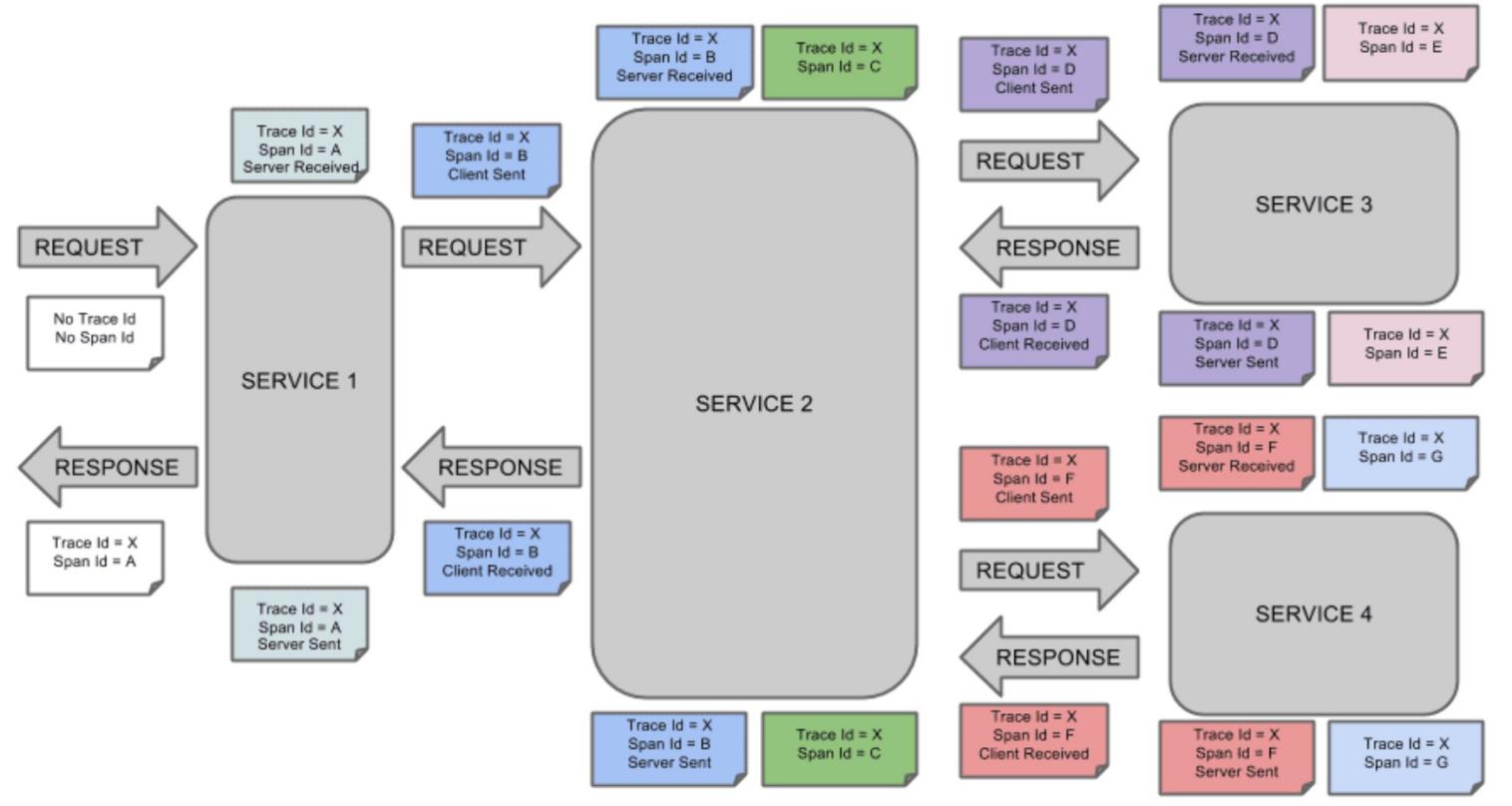
## 查看zipkin的控制台

<http://localhost:9411>

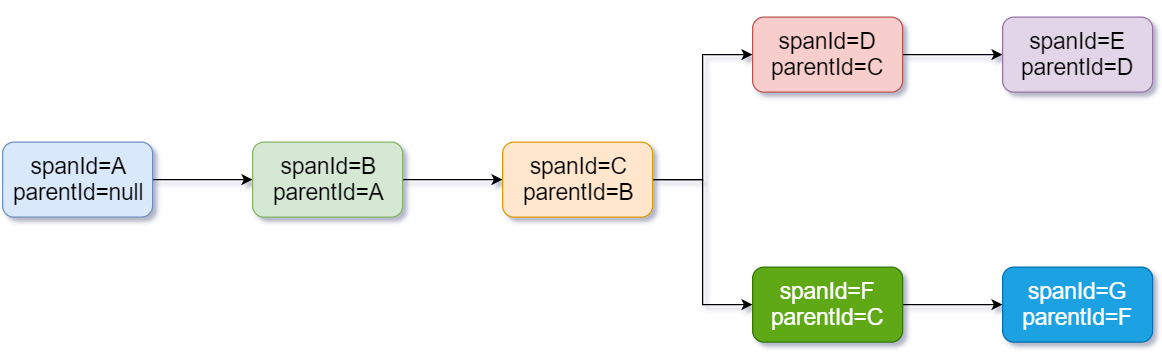


# 相关术语和名词解释

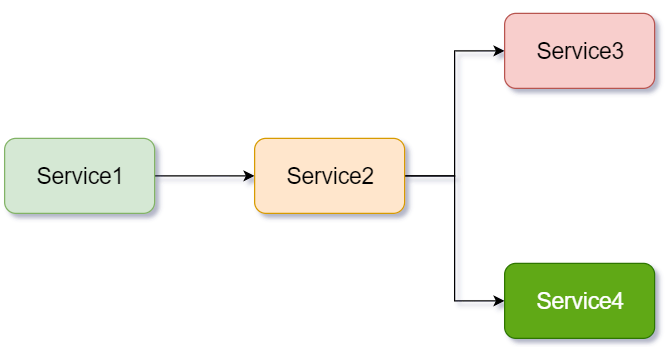
## 完整调用链路



## 简化后



调用链路的依赖关系



## 名词解释

Trace:类似于树结构的Span集合，表示一条调用链路，存在唯一标识

span:表示调用链路来源，通俗的理解span就是一次请求信息

# Sleuth快速入门

## 案例说明

这里我们就不搭建项目了，直接用之前的案例

eureka-server

consumer-user-service

provider-order-service

## 添加依赖

**consumer-user-service和provider-order-service 都要加**

|  |
| --- |
| <dependency>  <groupId>org.springframework.cloud</groupId>  <artifactId>spring-cloud-starter-zipkin</artifactId> </dependency> |

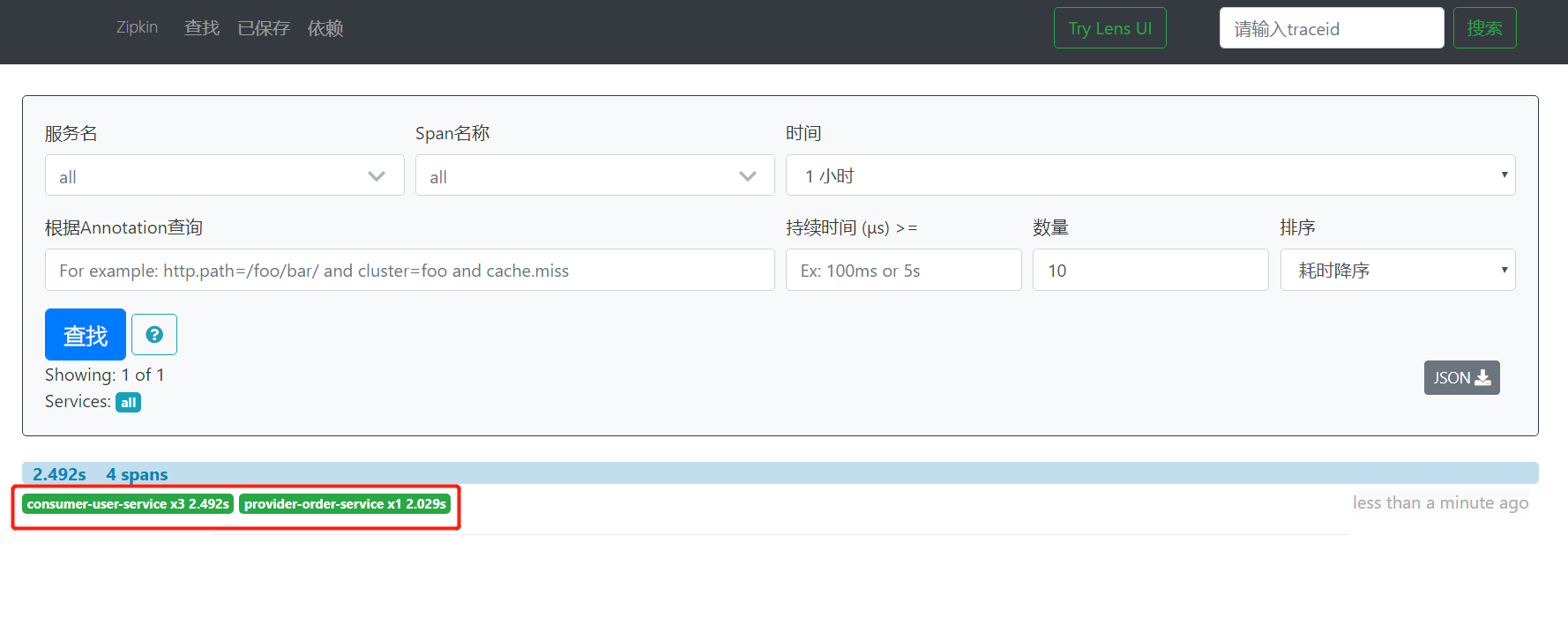
## 修改配置文件

**consumer-user-service和provider-order-service 都要该配置文件**

|  |
| --- |
| spring:  zipkin:  base-url: http://localhost:9411  sleuth:  sampler:  probability: 1 #配置采样率 默认的采样比例为: 0.1，即 10%，所设置的值介于 0 到 1 之间，1 则表示全部采集  rate: 10 #为了使用速率限制采样器,选择每秒间隔接受的trace量，最小数字为0，最大值为2,147,483,647（最大int） 默认为10。 |

## 启动访问，远程调用一下

查看zipkin



点一下依赖

