

# 链路追踪 Zipkin

---

## 学习目标

- 1.什么是链路追踪
- 2.链路追踪能做什么
- 3.使用链路追踪需要注意的点
- 4.如何使用链路追踪
- 5.链路追踪同类产品对比

## 1 什么是链路追踪

微服务架构是一个分布式架构，它按业务划分服务单元，一个分布式系统往往有很多个服务单元。

由于服务单元数量众多，业务的复杂性，如果出现了错误和异常，很难去定位。主要体现在，一个请求可能需要调用很多个服务，而内部服务的调用复杂性，决定了问题难以定位。所以微服务架构中，必须实现分布式链路追踪，去跟进一个请求到底有哪些服务参与，参与的顺序又是怎样的，从而达到每个请求的步骤清晰可见，出了问题，很快定位。

在微服务系统中，一个来自用户的请求，请求先达到前端A（如前端界面），然后通过远程调用，达到系统的中间件B、C（如负载均衡、网关等），最后达到后端服务D、E，后端经过一系列的业务逻辑计算最后将数据返回给用户。对于这样一个请求，经历了这么多个服务，怎么样将它的请求过程的数据记录下来呢？这就需要用到服务链路追踪。

## 2.链路追踪能做什么

### 2.1.1 链路追踪能做什么

简单来讲就是，项目在生产的环境下如果出了错，我们没有办法使用debug，而且传统的日志打印，可能随时会过期，这样定位错误就会很难，所以我们使用链路追踪这个技术来对我们认为可能发生异常的地方进行日志的打印与追踪。

最重要的一点就是，虽然可以打印出来日志但是我们不知道哪个请求发生了异常，所以使用链路的方式通过前缀id的方式生成一条链路，追踪问题的所在

### 2.1.2 链路追踪的埋点

所谓埋点就是在应用中特定的流程收集一些信息，用来跟踪应用使用的状况，后续用来进一步优化产品或是提供运营的数据支撑，包括访问数（Visits），访客数（Visitor），停留时长（Time On Site），页面浏览数（Page Views）和跳出率（Bounce Rate）。这样的信息收集可以大致分为两种：页面统计（track this virtual page view），统计操作行为（track this button by an event）。

## 3 使用链路追踪需要注意的点

### 3.1.1 链路日志的格式

日志格式:

[order-server,c323c72e7009c077,fba72d9c65745e60,false]

- 1、第一个值, spring.application.name的值
- 2、第二个值, c323c72e7009c077, sleuth生成的一个ID, 叫Trace ID, 用来标识一条请求链路, 一条请求链路中包含一个Trace ID, 多个Span ID
- 3、第三个值, fba72d9c65745e60、spanID 基本的工作单元, 获取元数据, 如发送一个http
- 4、第四个值: false, 是否要将该信息输出到zipkin服务中来收集和展示。

## 4 如何使用我们的链路追踪

### 4.1.1 使用 zipkin + sleuth

通常我们的链路追踪都是搭配 sleuth来使用的

- 1.首先需要在消费者上面配置依赖

```
1 <!--sleuth-->
2 <dependency>
3   <groupId>org.springframework.cloud</groupId>
4   <artifactId>spring-cloud-starter-sleuth</artifactId>
5 </dependency>
```

- 2.在需要写日志的类上贴 @Slf4j,然后再消费者中打印日志。

```
@Service
@Slf4j
public class OrderServiceImpl implements OrderService {
    @Autowired
    private ProductFeignApi productFeignApi;
    @Override
    public Order findById(Long id) {
        Product product = null;
        Order order = null;
        try {
            product = productFeignApi.find(id);
            log.info("OrderServiceImpl : " + product.toString());
            log.info("OrderServiceImpl : " + productFeignApi.getClass().toString());
            order = new Order();
            order.setOrderId(1L);
            order.setProductName(product.getName());
        } catch (Exception e) {
            e.printStackTrace();
            log.error(e.getMessage());
        }
    }
}
```

### 3.在配置文件中加入zipkin的地址

```
1  spring:
2    zipkin:
3      base-url: http://localhost:9411
4    sleuth:
5      sampler:
6        probability: 1
```

### 4.引入zipkin的依赖

```
1  <dependency>
2    <groupId>org.springframework.cloud</groupId>
3    <artifactId>spring-cloud-starter-zipkin</artifactId>
4  </dependency>
```

## 5.链路追踪同类产品对比

#### 1.CAT

由大众点评开源，基于Java开发的实时应用监控平台，包括实时应用监控，业务监控。

#### 2.Pinpoint

由韩国团队naver团队开源，针对大规模分布式系统用链路监控，使用Java写的工具。

#### 3.SkyWalking

2015年由个人吴晟（华为开发者）开源，2017年加入Apache孵化器。

针对分布式系统的应用性能监控系统，特别针对微服务、cloud native和容器化(Docker, Kubernetes, Mesos)架构，其核心是个分布式追踪系统

## 补充

[相关阅读](#)