


课时8

分布式链路跟踪

1. 分布式链路跟踪使用场景
2. 分布式链路跟踪简介
3. 分布式链路跟踪框架
4. 分布式链路数据存储
5. 手动埋点检测性能



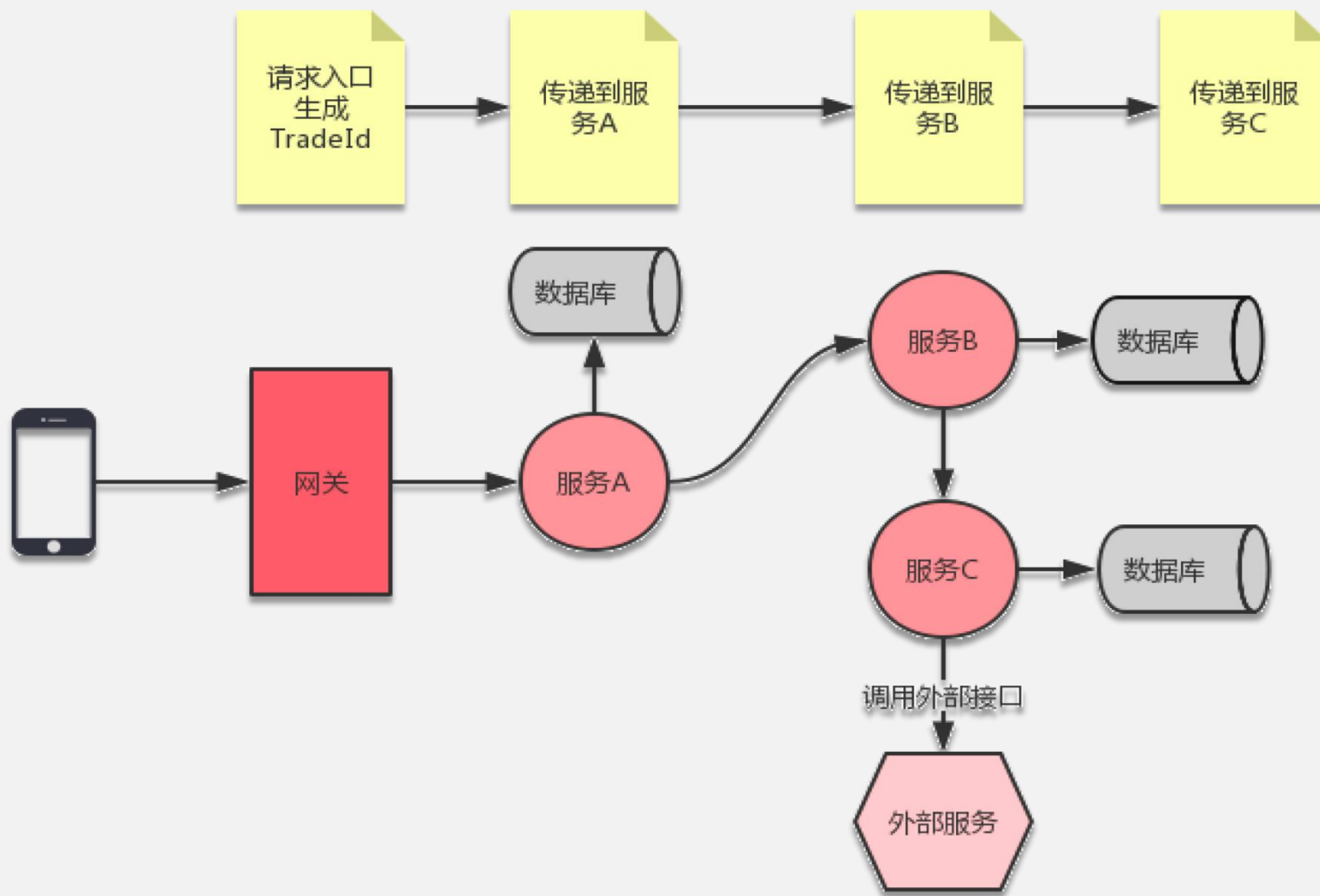
单体架构升级到微服务架构
带来的一系列问题

01. 接口响应慢, 怎么排查?

02. 服务间的依赖关系如何查看?

03. 请求贯穿多个服务, 如何将日志串起来?

分布式链路跟踪介绍



链路跟踪核心概念

Span

基本工作单元

例如：发送RPC请求是一个新的span，发送HTTP请求是一个新的span，内部方法调用也可以是一个新的span

Trace

一次分布式调用的链路信息，将所有踪迹信息串连起来



Annotation

用于记录事件的信息

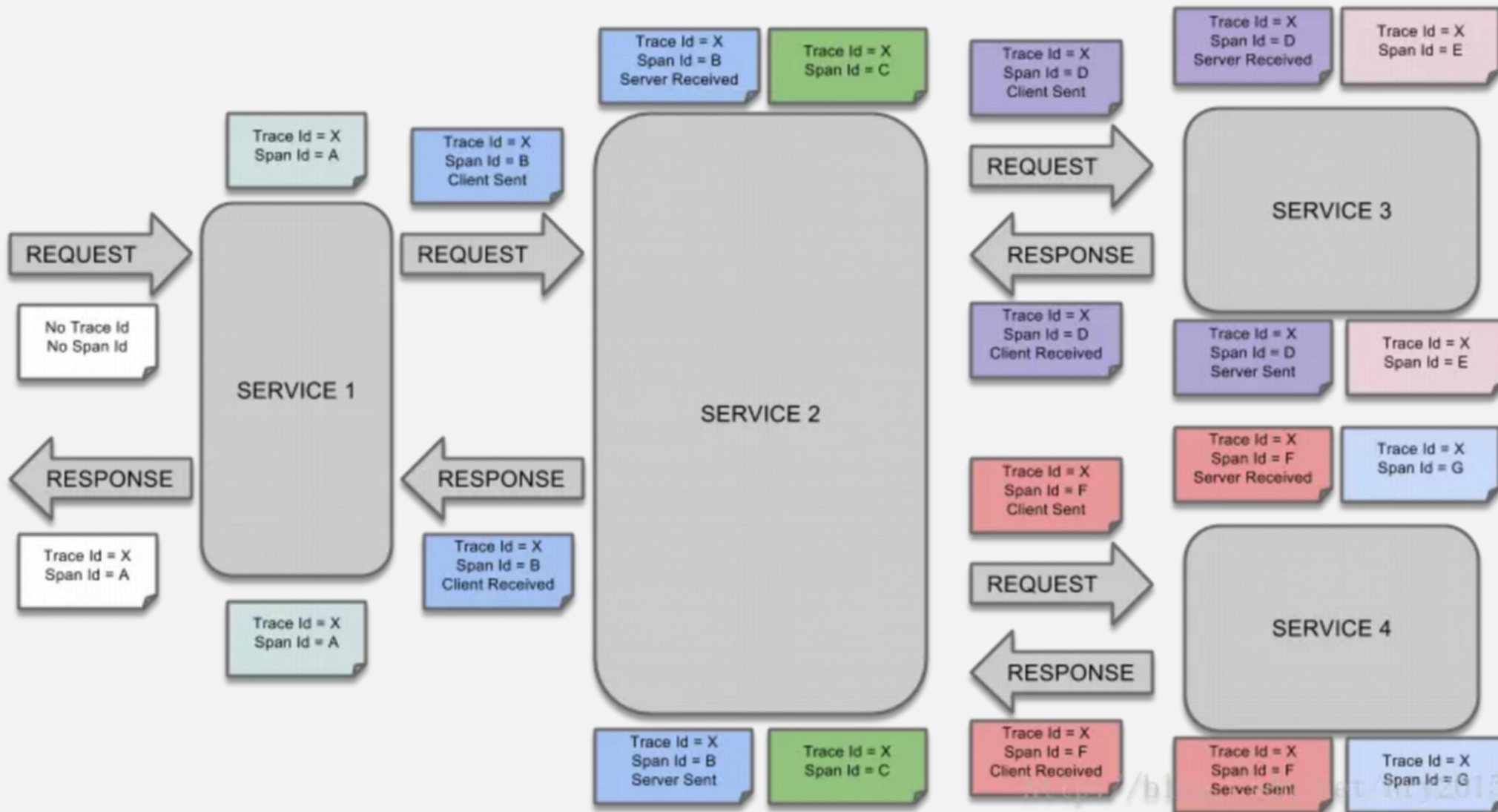
cs : Client Sent, 客户端发起了一个请求, 这个annotation表示span的开始

sr : Server Received, 服务器端获得了请求并开始处理它, 从此时间戳中减去 cs 时间戳会显示网络延迟

ss : Server Sent, 在请求处理完成时将响应被发送回客户端时, 从此时间戳中减去 sr 时间戳会显示服务器端处理请求所需的时间

cr : Client Received, 表示span的结束, 客户端已成功从服务器端收到响应, 从此时间戳中减去 cs 时间戳会显示客户端从服务器接收响应所需的全部时间

请求追踪过程分解



Spring Cloud Sleuth介绍

Spring Cloud Sleuth

是一种分布式的服务链路跟踪解决方案

通过使用Spring Cloud Sleuth可以让我们快速定位某个服务的问题以及服务间的依赖关系

功能点

1. 增加链路信息到SLF4J
2. 数据可直接上报给Zipkin
3. 内置很多框架的埋点

如: Feign, Hystrix,

RestTemplate等

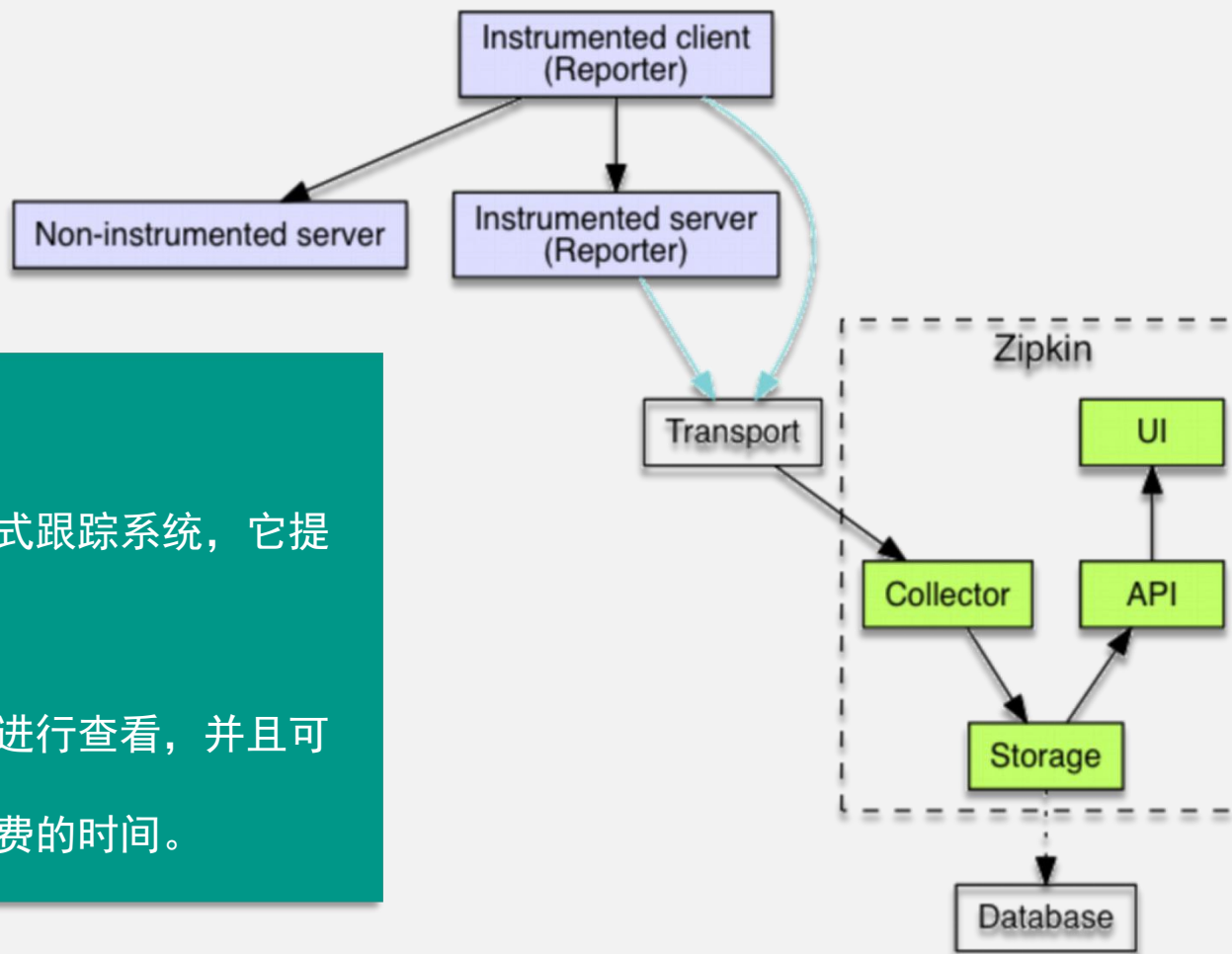


Zipkin

是 Twitter 的一个开源项目

是一个致力于收集所有服务的监控数据的分布式跟踪系统，它提供了收集数据和查询数据两大接口服务

有了 Zipkin ，我们就可以很直观地对调用链进行查看，并且可能很方便看出服务之间的调用关系以及调用耗费的时间。



日志的格式为: [application name, traceId, spanId, export]

application name : 应用的名称, 也就是application.properties中的spring.application.name参数配置的属性

traceId : 为一个请求分配的唯一请求号, 用来标识一条请求链路

spanId : 表示一个基本的工作单元, 一个请求可以包含多个步骤, 每个步骤都拥有自己的spanId。一个请求包含一个TraceId, 多个SpanId

export : 布尔类型。表示是否要将该信息输出到Zipkin进行收集和展示

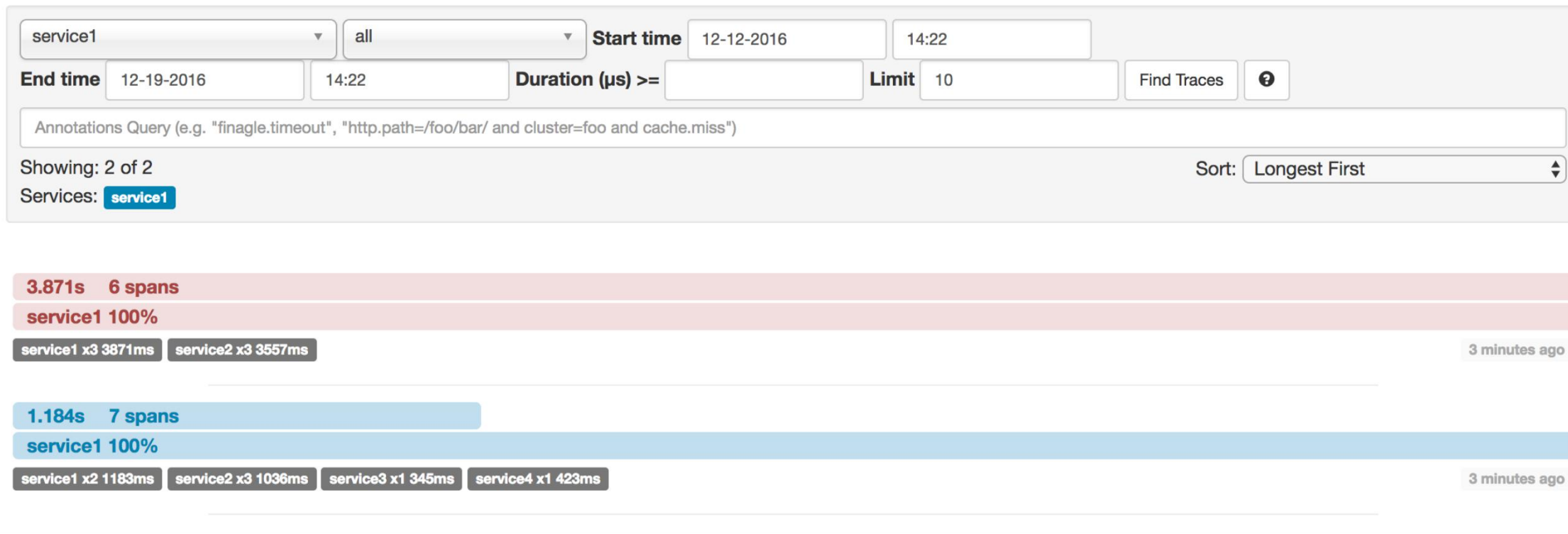
日志:

```
2019-10-26 16:53:41.191 INFO [sleuth-zuul,97143cc4c54f6559,97143cc4c54f6559,false] 26404 --- [nio-8081-exec-1] com.example.zuul.MyFilter : MyFilter run ...
```

```
2019-10-26 16:53:41.631 INFO [sleuth-article,97143cc4c54f6559,235e03bf22f6f435,false] 26405 --- [nio-8084-exec-1] c.e.a.controller.ArticleController : article/get
```

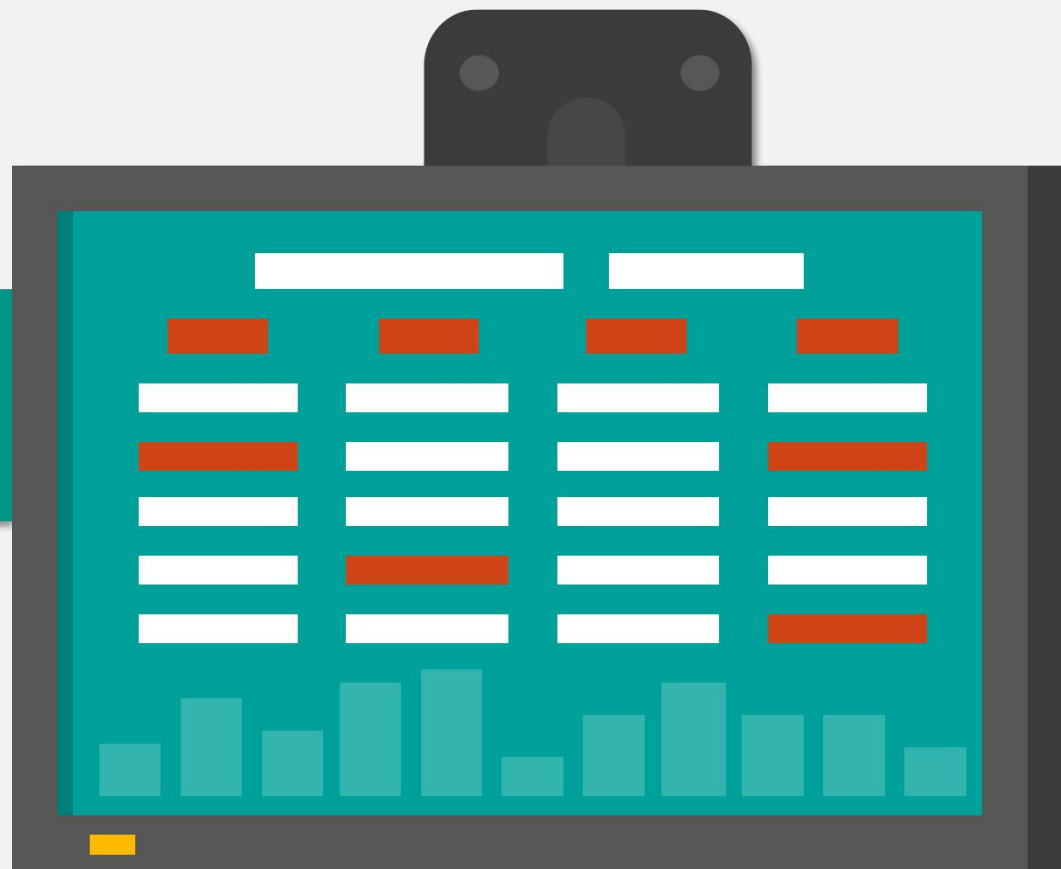
```
2019-10-26 16:53:41.928 INFO [sleuth-user,97143cc4c54f6559,b95dc1de7e689a60,false] 26406 --- [nio-8085-exec-1] c.e.user.controller.UserController : user/get
```

使用Zipkin展示链路跟踪数据



抽样比例：1为100%

```
spring.sleuth.sampler.probability=1.0
```



RabbitMq 代替 Http 发送调用链数据

Maven依赖

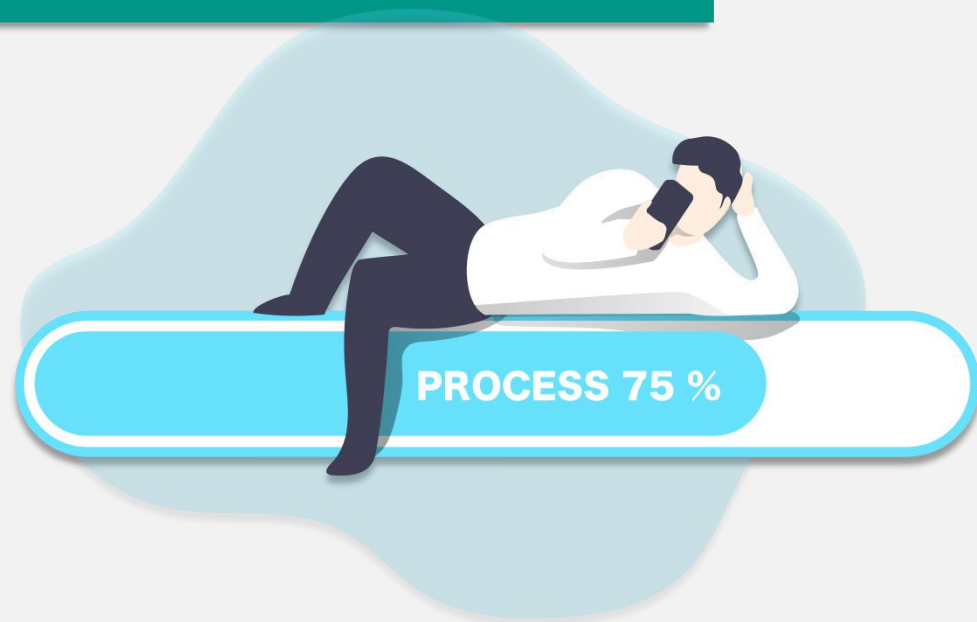
```
<dependency>  
  
    <groupId>org.springframework.amqp</groupId>  
  
    <artifactId>spring-rabbit</artifactId>  
  
</dependency>
```

配置

```
spring.zipkin.sender.type=RABBIT  
  
spring.rabbitmq.addresses=amqp://192.168.10.124:5672  
  
spring.rabbitmq.username=yinjihuan  
  
spring.rabbitmq.password=123456
```

启动Zipkin Server指定ElasticSearch信息

```
java -DSTORAGE_TYPE=elasticsearch -DES_HOSTS=http://localhost:9200 -jar zipkin.jar
```



```
@Autowired
Tracer tracer;

@Override
public void saveLog(String log) {
    ScopedSpan span = tracer.startScopedSpan("saveLog");
    try {
        Thread.sleep(200);
    } catch (Exception | Error e) {
        span.error(e);
    } finally {
        span.finish();
    }
}
```

Hystrix埋点

服务之间的信息传递

Next: 课时9 《微服务安全认证》