前言

虽然最终tmallpush没有使用eagleeye的方案，但是由于eagleeye的业务日志API链路追踪方式比较实用，在此分享一下之前学习到的一部分内容。

原始目标

能通过eagleeye查看每条push完整链路，从读取push的一个task任务开始，到最后发送到agoo或者apns为止。可以根据给定device、userId、statisticsId（活动Id）查看该条push的状态，具体是被被疲劳还是被发送未达，甚至是丢失等。

步骤

1. 可使用eagleeye API的日志包含：业务明细日志（记录每一个业务的变更明细）、统计日志（固定时间统计）和调用链日志（梳理调用关系）  
  
2. 明确本目标为第一种类型——业务明细日志，确定调用类为TraceLogger

3. 使用接口打印日志  
TraceLogger.trace(domain, eventType).escapeAndLogline(logContent)  
4. 若eagleeye-core的版本在1.4.2版本以上，则可使用以下接口，记录更加详细的业务Id，关联出业务相关的一系列日志。

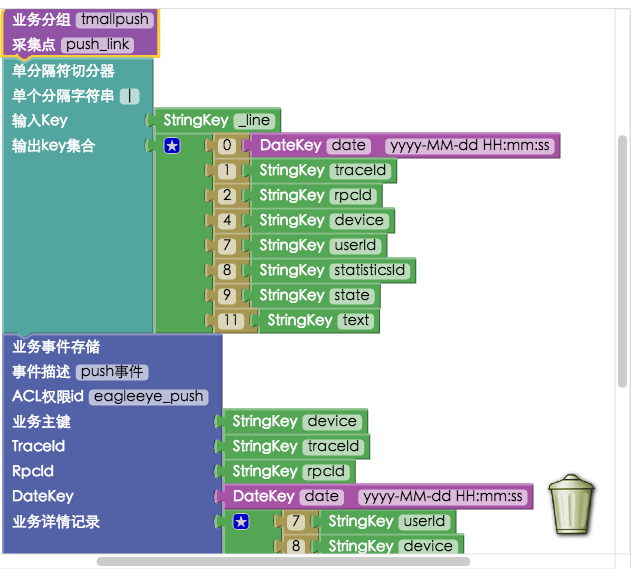
TraceLogger.trace(domain, eventType)

.traceBiz(bizKey, bizValue)

.escapeAndLogline(logContent)  
由于eagleeye-core的版本主要依赖HSF的版本，eagleeye-core 1.4.2需要HSF 2\_5\_150826，pandora 2.0.5.5.8版本及以上，升级HSF版本涉及修改太多，因此暂时没有使用详细版本的接口

5.  在/home/admin/logs/eagleeye/目录下生成trace-\*.log日志，日志格式为

timestamp|traceId|rpcId|domain|eventType|logType|traceBizs|logContent

6.  在服务端生成日志后，需要使用TLog解析日志，自定义格式解析输出  
  
7. 在TLog系统中创建分组、采集点、采集规则后，需要配置业务事件模块，通过该模块控制日志解析和输出  
  
8. 这里日志的分隔符使用默认的“|”分隔符，在ACL权限控制平台创建相应的查看权限。流程模块的输入是采集源，包含业务分组和采集点。第二步，对每行数据今夕解析，使用切分器（单分隔符切分器、多分隔符切分器、顺序分隔符切分器等），修改分隔符和输出key集合。若输出的keys个数超出当前，点击五角星，可以添加或删除keys的个数。第三步，添加业务事件，编辑事件名、刚创建的acl权限id，业务主键、traceId、rpcId、DateKey、业务详情，这个自定义。最后若要方便测试，可添加debug节点。最后的结构如下图所示：  
  
  
9. 点击右侧的保存，然后启动，流程启动，开始解析采集的日志。  
  
10. 在采集点处，点击激活，激活采集点，然后点击“根据规则生成任务”，创建一条采集任务。  
  
11. 完成采集任务，过一段时间，在eagleeye的输入框，输入业务主键或者traceId的业务轨迹均可看到日志解析输出结果

失败原因

若不绑定上下文，则traceId是根据threadLocal变量变动，若是多线程中用到了线程池，则会出现两个不相干的线程因为共用线程导致traceId相同，无法区分。要绑定上下文和traceId，则需要传递context，对于当前的push链路改动较大，不适合。

参考文档

eagleeye的业务日志API的官方文档：<http://tbdocs.alibaba-inc.com/pages/viewpage.action?pageId=192046759>业务轨迹接入eagleeye的步骤：<http://gitlab.alibaba-inc.com/middleware/tlog2-docs/wikis/TLog2BizEvent>tlog交流群：965281479（密码tlog1234）  
  
eagleeye交流群：928687313（密码hello1234）