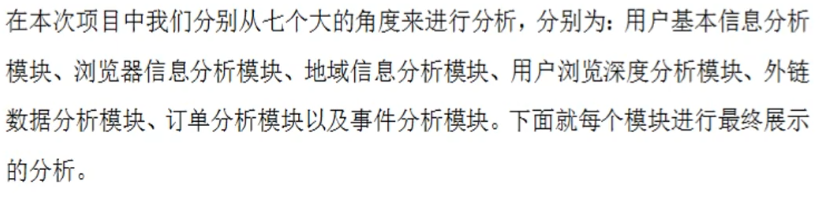
目录

[需求分析 1](#_Toc47475477)

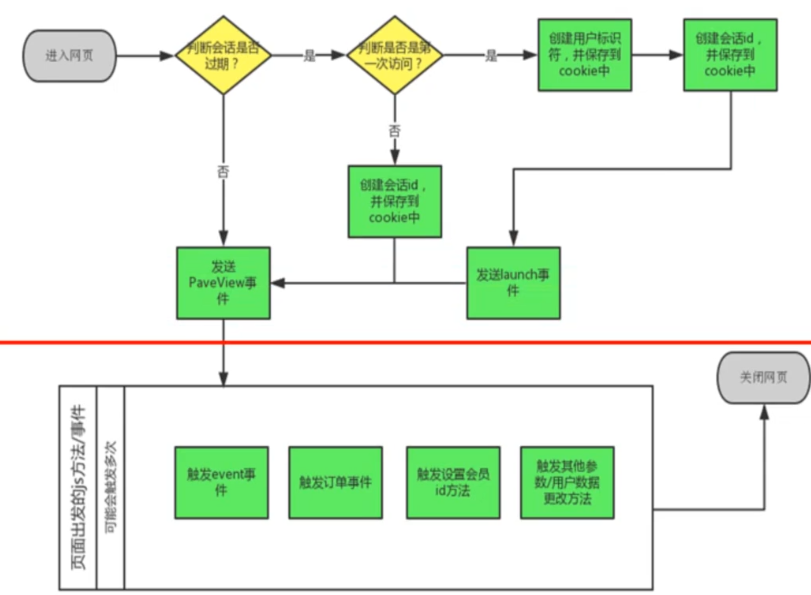
[Flume 4](#_Toc47475478)

# 需求分析



Jsp文件写了页面，运行在本地的tomcat

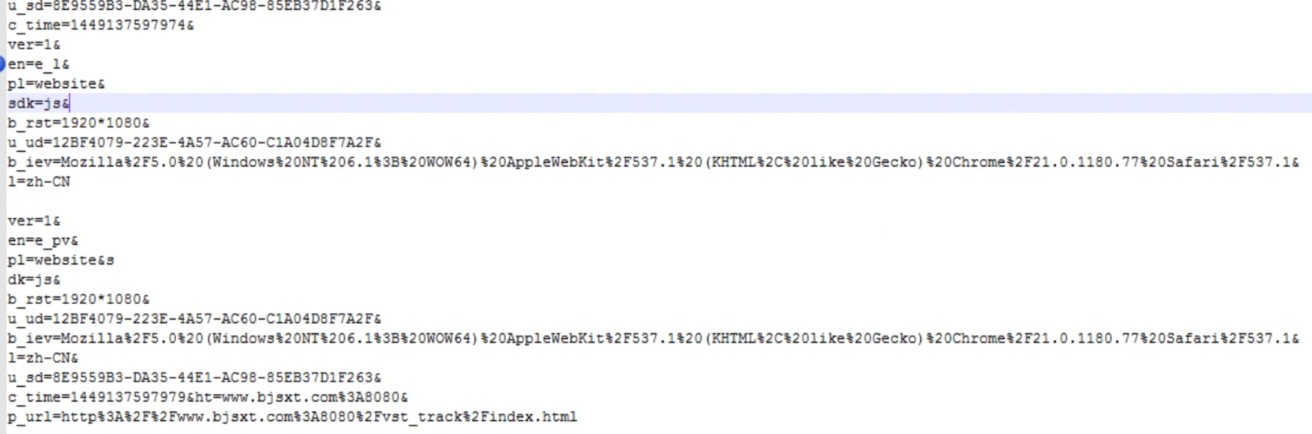
Js代码用于链接，用的key-value的方式链接，js完成下面的逻辑，



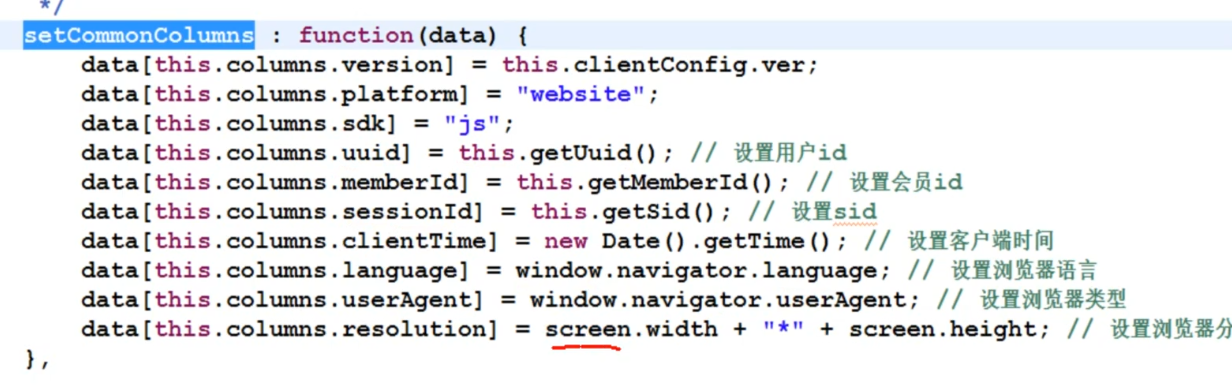
用startSession开始运行



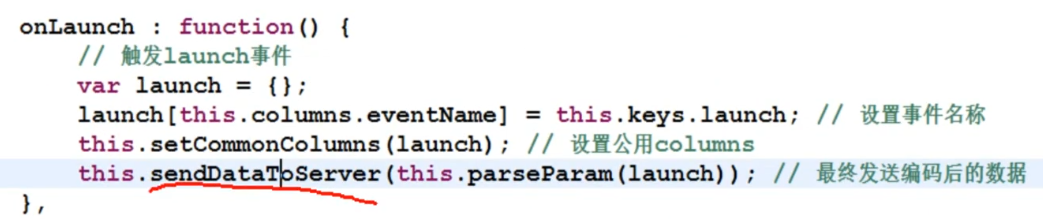
源数据



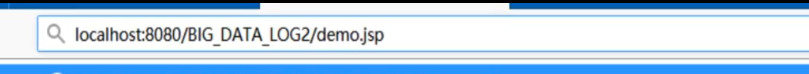
获取数据方法



发送数据方法

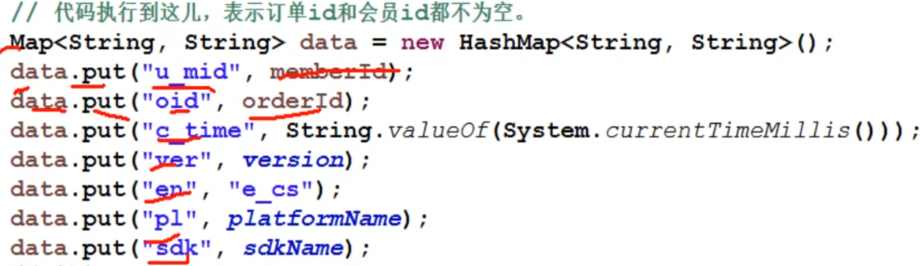


访问方式

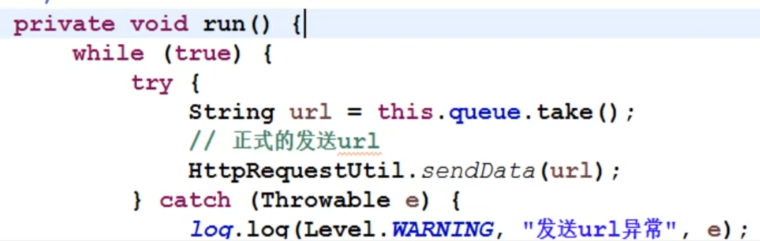


Js用于实现前端采集

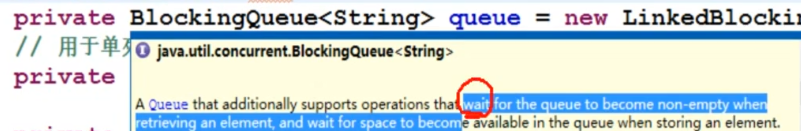
Java-sdk用于收集数据写后端

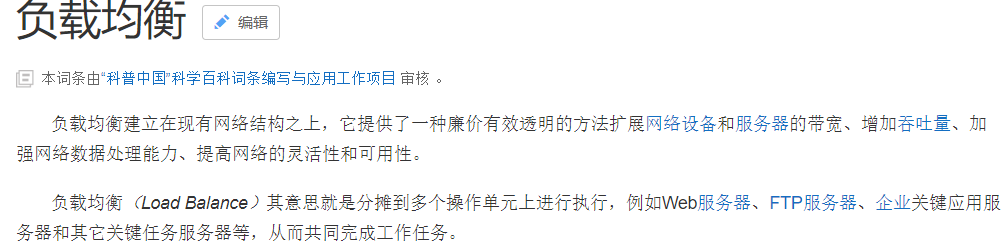


运用单列模式，保证取数据顺序，创建了一个队列用线程跑起来保证能收到数据

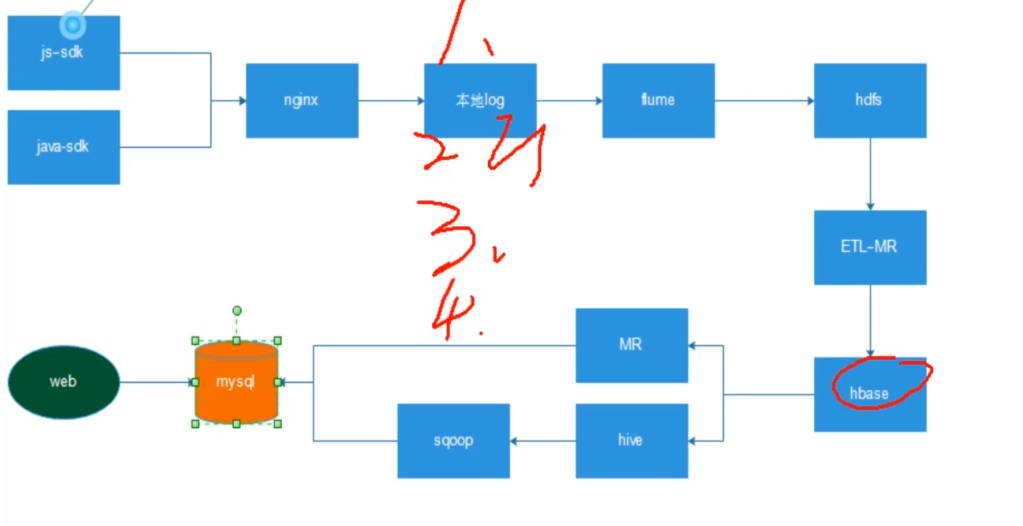


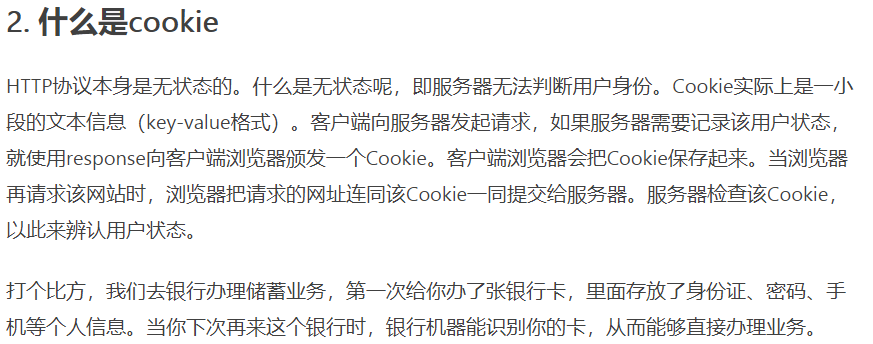
没数据了会等待



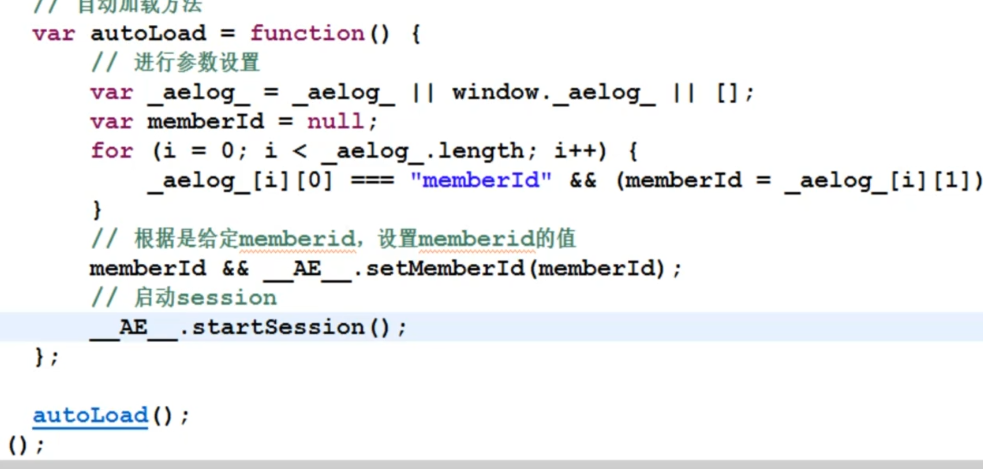


数据流图





autoLoad()不是方法是执行程序



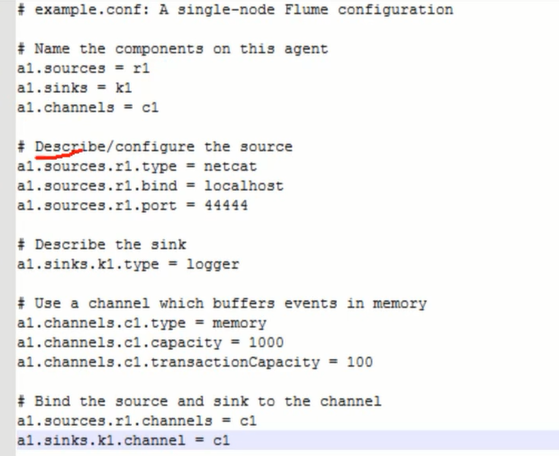
# Flume

运行demo





Flume核心配置 先起名称 再配置链接



channels部分是通道容量和每次转换的数据

执行



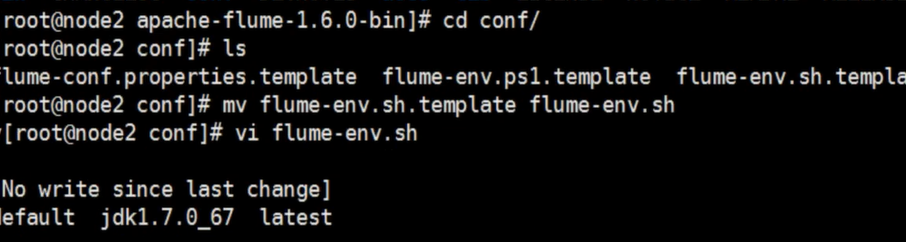
被划掉的部分是flume自己的配置文件可以不选

--conf-file 后面跟我们自己配置的配置文件（上面的代码放的地方）

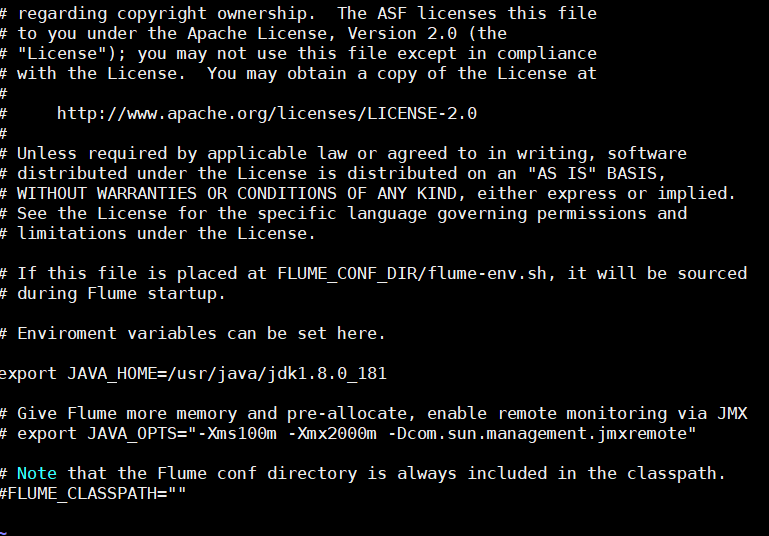
--name 是agent的名称

## 设置flume内存

修改解压包中得flume-env.sh.template文件



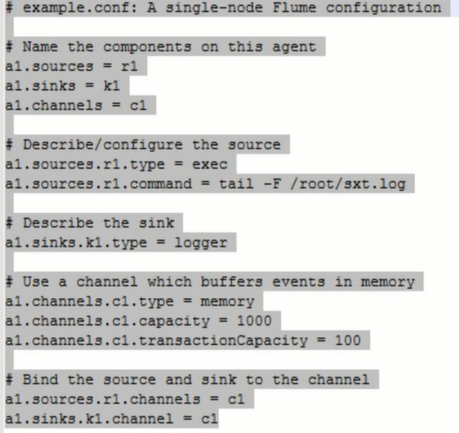
修改java路径



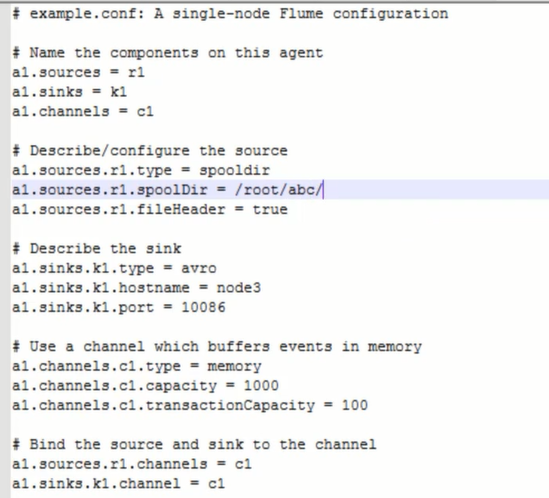
## Flume主要功能以及其配置实现

主要是配置sources的type属性来对应不同场景的应用

1. 监控文件

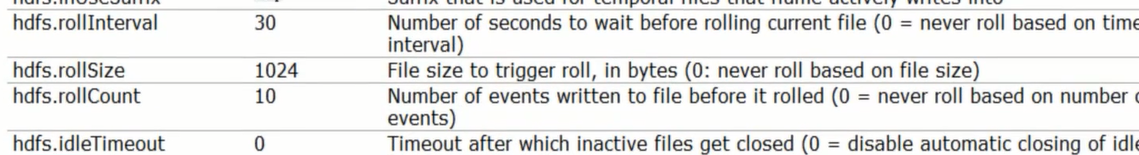


1. 监控文件夹

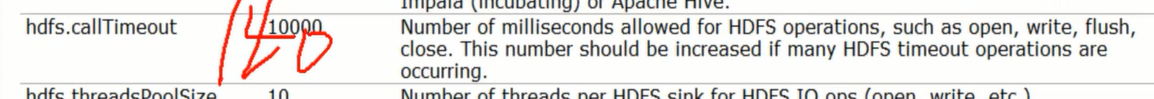


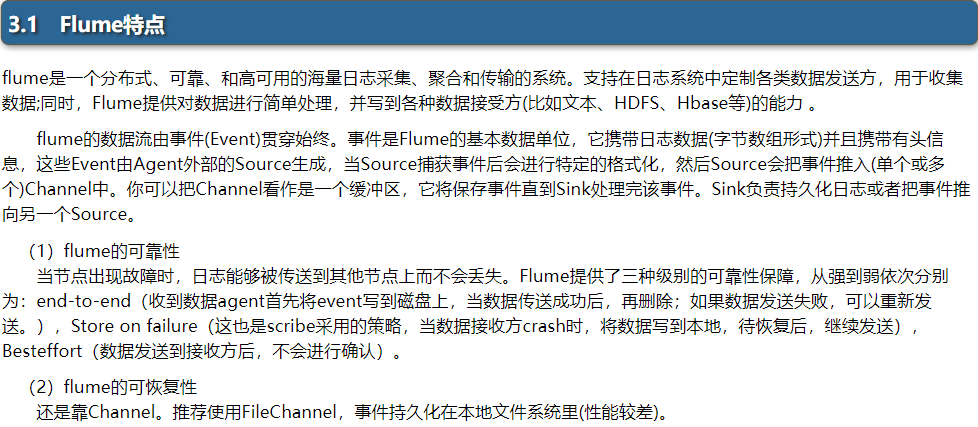
1. 配置sink HDFS

滚动的参数



多少秒写入一次数据





K1往hdfs里面写东西



Tail命令再Linux中是监控的意思

ETL在本项目的作用（IPseek）

1. 脏数据过滤
2. 地址解析
3. 浏览器先验信息解析

ETL过后的数据输出

1. 过滤的多少数据
2. 留下了多少数据
3. 输入了多少数据

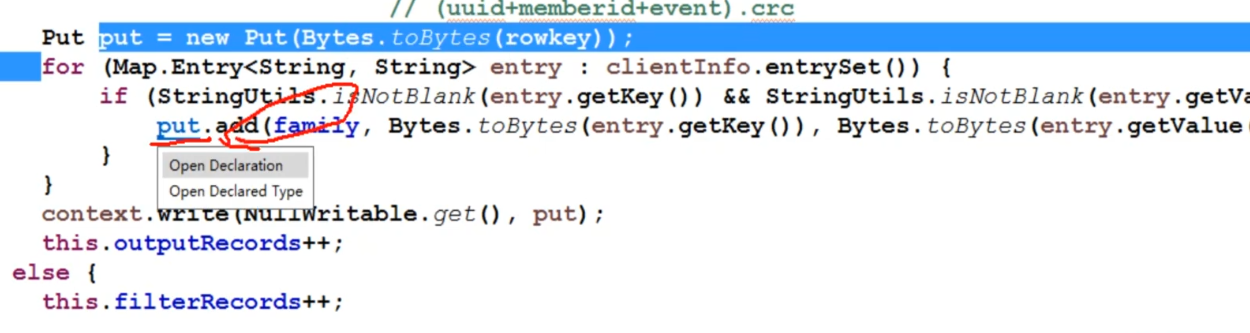
让etl可以从hdfs里面拿数据，能写道hbase里面的demo



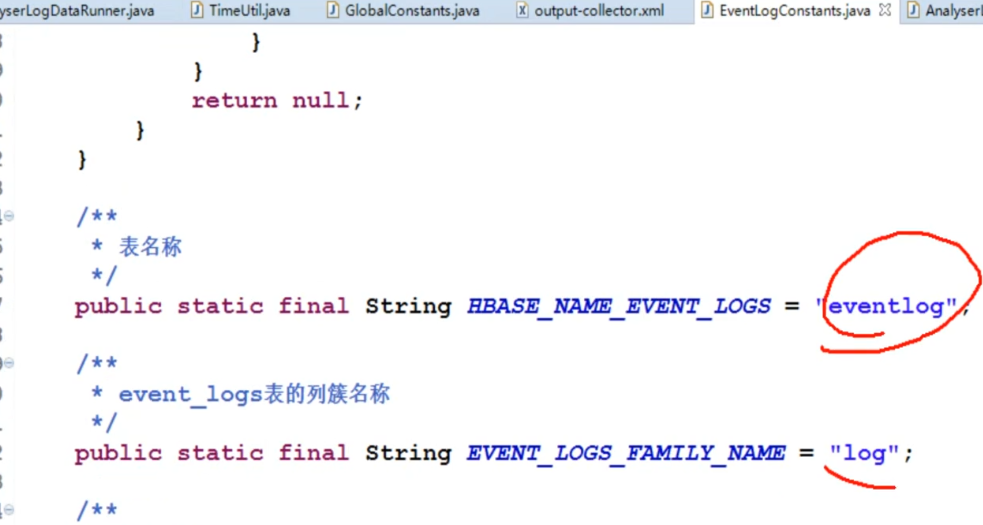
Etl解析log split方法进行切分



切割统计后的数据保存在rowkey里面，创建put对象



运行前先创建hbase表，名字是已经写好的才可以传入





在win的eclipse上找到run.java右键run as config

http://flume.apache.org/FlumeUserGuide.html#a-simple-example