**日志抽取程序**

**使用说明**



修 改 记 录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本号 | 日 期 | 地点 | 修订 | 描 述 |
| V1.0 | 2014/05/27 | 杭州 | 黄建丽 | 文档修订与维护 |
|  |  |  |  |  |

1.对于新增特定日志需求可以通过修改配置实现格式配置

2.实现对于特定字段的合并,拼接,加减乘除的功能

3.配置文件修改,详见下文

**3.domain.properties:**

channel与sp名称的映射表,sp名称必须和customer-ref.xml中的customer.name一致,channel字段无法匹配的将使用default格式输出

示例如下:

img01.taobaocdn.com=default  
img02.taobaocdn.com=default  
########################################   
www.cloudvsp.com=cloudvsp   
www.cloudvsp.com.watone.net=cloudvsp   
consulting.cloudvsp.com=cloudvsp   
consulting.cloudvsp.com.watone.net=cloudvsp   
solution.cloudvsp.com=cloudvsp   
solution.cloudvsp.com.watone.net=cloudvsp   
dc.cloudvsp.com=cloudvsp   
dc.cloudvsp.com.watone.net=cloudvsp

**4.customer-ref.xml:**

sp用户定义与策略,模板的关系定义配置

示例如下:

<?xml version="1.0" encoding="utf-8"?>   
<customers>   
<!-- SP 默认-->   
<customer>

<!-- SP 名称,默认的日志使用default-->   
<name>default</name>

<!-- 日志输出目录-->   
<outputdir>outputdir</outputdir>

<!-- 输出日志的日志文件前缀-->   
<filePrefix>default\_htcdn\_</filePrefix>

**<!-- 策略名称,默认的策略 template策略,可以统一使用该策略,在snappy.properties文件中配置不同的策略模板细节 -->   
<strategy>template</strategy>**

**<!-- 策略模板名称,默认的策略模板名称是default,对应snappy.properties文件配置模板的配置项前缀-->**

**<template>default</template>**   
</customer>   
<!-- SP 淘宝-->   
<customer>   
<name>taobao</name>   
<outputdir>outputdir</outputdir>   
<filePrefix>tb\_htcdn\_</filePrefix>   
<strategy>template</strategy>   
<template>test</template>   
</customer>   
<!-- SP 天地祥云-->   
<customer>   
<name>cloudvsp</name>   
<outputdir>outputdir</outputdir>   
<filePrefix>cloudvsp\_cache\_</filePrefix>

**<!-- 使用template策略-->   
<strategy>template</strategy>**

**<!-- 使用cloudvsp的策略模板 -->   
<template>cloudvsp</template>**   
</customer>   
</customers>

5.snappy.properties

策略模板配置以及全局配置细节,可以通过配置支持多种处理函数

示例如下:

####### PUBLIC CONFIGURATION ########   
logHome=hdfs\://hdfs1/flume/pic/   
partitions=001,006   
  
####### TEMPLATE CONFIGURATION ########   
  
#############   
# \* 0 = string.const   
# \* -1 = double.quote   
# \* -2 = string.concat   
# \* -3 = int.plus   
# \* -4 = int.minus   
# \* -5 = int.multiply   
# \* -6 = int.divide   
# \* -7 = int.power   
# \* -8 = int.modulo   
# \* -9 = double.plus   
# \* -10 = double.minus   
# \* -11 = double.multiply   
# \* -12 = double.divide   
# \* -13 = double.power   
# \* -14 = single.quote   
#   
## template is named 'default'   
# The index of fields from source into target line,these index is source's.   
default.fields=3,4,5,6,7,17,8,9,10,12,22,15,23,18,24   
default.time.format=yyyy-MM-dd HH:mm:ss  
# default fields' separator, this is a space blank,it's should be '\ ' or '\u0020' in UTF-8   
# The separator may be a space symbol,a tab symbol('\t') or a symbol such as '|'   
default.separator=\u0020   
default.double.quote.fields=5,12,15,23,18   
default.resp.fields=   
  
################   
#### 3,19,12,concat("TCP\_HIT/",6),9,4,concat(24,"://",25,5),const("-"),const("NONE/-"),const("-"),quote(15),quote(18)####   
cloudvsp.separator=\t   
cloudvsp.fields=3,19,12,-2,9,4,-2,0,0,0,-1,-1   
cloudvsp.time.format=ms   
## target line fields index in quotes ##   
cloudvsp.double.quote.fields=11,12   
cloudvsp.double.quote.params=$15;$18   
  
cloudvsp.string.const.fields=8,9,10   
cloudvsp.string.const.params=-;NONE/-;-   
  
cloudvsp.string.concat.fields=4,7   
cloudvsp.string.concat.params=TCP\_HIT/,$6;$24,://,$25,$5   
  
#################   
#### test ####   
#### 3,3,12,concat("TCP\_HIT/",6),9,4,concat(24,"://",25,5),const("-"),const("NONE/-"),const("-"),quote(15),quote(18)####   
test.separator=\t   
test.fields=3,3,12,-2,9,4,-2,0,0,0,-1,-1,-3,-4,-5,-6   
test.time.format=ms   
## target line fields index in quotes ##   
test.double.quote.fields=11,12   
test.double.quote.params=$15;$18   
  
test.string.const.fields=8,9,10   
test.string.const.params=-;NONE/-;-   
  
test.string.concat.fields=4,7   
test.string.concat.params=TCP\_HIT/,$6;$24,://,$25,$5   
  
test.int.plus.fields=13   
test.int.plus.params=6,$8   
  
test.int.minus.fields=14   
test.int.minus.params=$6,8   
  
test.int.multiply.fields=15   
test.int.multiply.params=$6,$8   
  
test.int.divide.fields=16   
test.int.divide.params=6,8   
  
test.params=$1,$2   
################   
workers=10   
############

######################################

snappy.properties中模板配置:

配置项格式:

**[templateName].{option}**

**说明:**

**1.字段序列**

[templateName].fields = 3,3,12,-2,9,4,-2,0,0,0,-1,-1,-3,-4,-5,-6

key:需要配置的日志字段

value:数字是日志在源日志中字段的序号(自然顺序,从1开始),需要进行函数功能处理的字段用各个负数定义,常量用0定义,各个函数定义与数值约定如下:

  0 = string.const ---> 字符串常量   
 -1 = double.quote ---> 双引号列   
 -2 = string.concat ---> 字符串拼接   
 -3 = int.plus ---> Integer求和   
 -4 = int.minus ---> Integer求差   
 -5 = int.multiply ---> Integer求积   
 -6 = int.divide ---> Integer求商   
 -7 = int.power ---> Integer求幂   
 -8 = int.modulo ---> Integer求模   
 -9 = double.plus ---> Double求和   
 -10 = double.minus ---> Double求差   
 -11 = double.multiply ---> Double求积   
 -12 = double.divide ---> Double求商   
 -13 = double.power ---> Double求幂   
 -14 = single.quote ---> 单引号列

**2.时间格式**

[templateName].time.format=ms

日志中请求时间格式,不设置该选项或者置为空时则使用原日志中的时间格式,值域:

ms  ---> 以ms为单位的时间戳

yyyy-MM-dd HH:mm:ss  ---> yyyy-MM-dd HH:mm:ss格式的时间戳

yyyyMMddHHmmss       ---> yyyyMMddHHmmss格式的时间戳

其他java支持的时间格式

3.**日志分隔符**

[templateName].separator=\t

定义解析出来的日志格式使用的分隔符,可以是tab('\t'),空格('\ ','\u0020'(UFT-8的空格字符)),|,以及其他自定义的字符,默认使用'|'

4.双引号引用字段

[template].double.quote.fields=11,12

在目标日志中双引号区域的位置,自然序号(从1开始)   
[template].double.quote.params=$15;$18

在源日志中需要加双引号的字段位置,用'$'符号引用,表示引用params配置项中使用源日志的字段($15,表示原日志中第15个字段,自然序号),params中对应fields中值,以';'分隔数组(即fields中count(',')=params中count(';'),以下皆同)

5.常量字段

[template].string.const.fields=8,9,10

目标日志中常量字段区域,自然序号 (','分隔)  
[template].string.const.params=-;NONE/-;-

对应fields中常量字段区域的常量字符串的值(';'分隔)

6.字符串拼接

[template].string.concat.fields=4,7

目标日志中的输出需要进行日志拼接的字段,自然序号(','分隔)   
[template].string.concat.params=TCP\_HIT/,$6;$24,://,$25,$5

对应fields中拼接字段区域的拼接参数(各个字段的参数间用';'分隔,字段内参数用','分隔,按拼接顺序进行排列参数,原日志中字段用'$'引用字段的自然序号)

7.整型求和

[template].int.plus.fields=13

目标日志以求和为结果的字段(','分隔),字段序号  
[template].int.plus.params=6,$8

求和参数,对应fields以';'分隔,参数数组内部以','分隔,无'$'引用的数字是整型数值,'$'引用的数字表示原日志中字段的自然序号

8.整型求差以及其他数值计算,释义同7