# SpoutRule维护说明 V1.0

## 说明

通常情况下，我们维护SpoutRule的原则，是一个数据源维护一个（或者多条）过滤规则。这样在预估数据量的时候，能够尽量确保各个过滤规则之间没有重叠的数据。（当一个数据源维护了多条过滤规则的话，不可避免的会有重叠数据的情况）

如果多个数据源的数据过滤规则，基本一样，比如都是blog类型，关键词，时间也完全相同，那么这几个数据源的过滤规则可以维护在一条中。

按照不同的需求，SpoutRule主要分为以下几种：

一种spoutrule维护一个数据源

* Keywords (ElasticKeywords) => 从es中查询

关键词查询。

通过查询索引获取数据，大部分情况下都是基于“分析关键词”进行索引查询。

* Comments(HBaseComments)

comments查询。

直接查询hbase中的comments，通常需要搭配父数据的idstr

* Answer(HBaseAnswer)

answer查询。

直接查询hbase中的answer，通常需要搭配父数据的idstr

* KeywordsWithUserFilter(ElasticUserFilter)

关键词查询+用户过滤。

方式基本等同keywords，只是在keywords查出数据以后，再查询数据的user信息，校验是否匹配user filter条件。

* KeywordsProductComments(elasticProductHBaseComments)

comments类型子数据查询。

分两步，第一步等同于keywords，基于关键词从索引中查询数据，第二步，将第一步得到的数据作为父数据，从hbase中查询父数据下的comments数据。

* KeywordsQuestionAnswer(elasticQuestionHBaseAnswer)

answer类型子数据查询。

同5，只是数据类型是answer，父数据类型是qa

* ElasticProductElasticComments(elasticProductElasticComments)

comments类型子数据查询。

分两步，第一步等同于keywords，基于关键词从索引中查询数据，第二步，将第一步得到的数据作为父数据，依旧从索引中查询父数据下的comments数据。

* KeywordsQuestionAnswer(elasticQuestionHBaseAnswer)

answer类型子数据查询。

同5，只是数据类型是answer，父数据类型是qa

## 维护规范

### Keywords：(ElasticKeywords)

从索引中查询数据，可以告诉索引的信息项包括：你的关键词是什么，数据类型是什么，数据源是什么，时间范围是什么，extra是什么，uid是什么。于是，在keywords这个类型的过滤规则中，需要维护以下内容：

***数据类型（type）***

：blog，forum，tweet等，可以有多个数据类型，多个数据类型中间 以英文逗号分隔

***数据源（source）***

：weibo，baidu，twitter等，可以有多个数据源，多个数据源中间以英文逗号分隔

***关键词过滤字段***

：是指关键词在数据哪个列上进行过滤，通常是不填，表示对title和content进行过滤，如果需求要求仅对title字段进行过滤，那么可以只填写title。对于普通数据来说，基本就是title和content。

***过滤关键词***

：关键词，用于索引查询时的过滤规则。完整的单词，需要用英文双引号包含，确保匹配的是完整的词语。多个关键词之间，可以有AND、OR和NOT的关系组合，组合关系之间通过英文括号进行分隔和二次组合，请注意AND，NOT和OR必须大写，如果不需要关键词过滤，可以填\*。例如：("word1" AND "word2") OR ("word3" AND "word4") NOT ("word5" OR "word6")

***开始时间*** 按照数据发布时间（真实网路数据的发布时间），数据爬取时间

：限制过滤数据的发布时间的起始时间。按需求填写即可。

***结束时间***

：限制过滤数据的发布时间的结束时间。此处注意，由于过滤规则是前闭后开的设置，所以，如果项目要求的数据是截止到某天的话，通常结束时间需要维护为第二天的0点。例如需求要求结束时间为2018-08-31，意思是包含8月31号这一天的数据，那么结束时间需要维护为2018-09-01.

***开始Indices*** 爬取时间

：我们的索引时按照月份来进行分开存储的，所以查询的时候，也是按照月份来分别查询，此处是为了限制最早的查询索引Indices。如果不填，则等同于开始时间，最多不超过24个月。

***Extra*** extra表示不同的属性列，特殊爬取的时候，填上去的信息，原始的数据中包括这个列

：有些数据过滤需要制定extra，那么可以在此维护，可以填多个extra，多个extra之间以英文逗号分隔。extra不需要加英文双引号。

***指定用户ID***

：有些时候，需要只查询指定用户发布的数据，那么可以在此处维护用户的id。但是此处尽量不要超过30个，超过30个，尽量按照30为一组，一组一个过滤规则。

***SpamWords***

：等同于过滤关键词中的NOT。不同之处是，NOT是在查询索引的时候就去掉了NOT词匹配的数据。SpamWords是先将数据查询出来，然后对数据的title和content进行查找，如果匹配到SpamWords，那么这个数据就不要了。使用情况是，如果NOT词太多了，放在查询中会导致查询变慢，那么可以维护在这里。

***正则匹配指定列***

：搭配【正则匹配表达式】一起使用，是指对索引查询出来的数据，进行再一次的过滤，必须满足维护的正则表达式，否则数据会被扔掉。使用情况是：例如要查询某个航空公司的数据，有大量的航班号，是以航空公司编码+航班号组成，如CA969，查询索引时，我们用关键词“CA”，但是进项目的数据，必须是CA后面带有几位数字的，此时维护正则表达式为【.\*CA[0-9]{2,4}.\*】.

***正则匹配表达式***

：请参考【正则匹配指定列 】。具体的正则表达式可以找开发人员咨询。

### Comments(HBaseComments)

直接从hbase中查询comments，但是需要知道comments所属父数据的idstr，例如商品数据的idstr，forum的idstr等等。

***数据源（source）***

：weibo，baidu，twitter等，此时一个过滤规则只能维护一个source。

***父数据IDSTR***

：

父数据的IDSTR。

***列过滤\_Family***

: 从HBase中查询到comments数据后，如果针对指定的列有过滤规则，可在此处指定column\_family。例如：dcf。

**列过滤\_Column**

:

从HBase中查询到comments数据后，如果针对指定的列有过滤规则，可在此处指定列名。例如：content。

**列过滤\_Compare :**

从HBase中查询到comments数据后，如果针对指定的列有过滤规则，可在此处指定过滤比较符，例如：like；支持>,>=,<,<=,==,like

***列过滤\_Value***

：

从HBase中查询到comments数据后，如果针对指定的列有过滤规则，可在此处指定过滤比较的值，例如：iphonex.

***列过滤\_type***

：

从HBase中查询到comments数据后，如果针对指定的列有过滤规则，可在此处指定过滤比较的值的类型，例如：String

开始时间

结束时间

***正则匹配指定列***

：搭配【正则匹配表达式】一起使用，是指对索引查询出来的数据，进行再一次的过滤，必须满足维护的正则表达式，否则数据会被扔掉。使用情况是：例如要查询某个航空公司的数据，有大量的航班号，是以航空公司编码+航班号组成，如CA969，查询索引时，我们用关键词“CA”，但是进项目的数据，必须是CA后面带有几位数字的，此时维护正则表达式为【.\*CA[0-9]{2,4}.\*】.

***正则匹配表达式***

：请参考【正则匹配指定列 】。具体的正则表达式可以找开发人员咨询。

### Answer(HBaseAnswer)

查询原理同comments一样，不同在于，此处是从answer表里查询answer数据。

***数据源（source）***

：基本只能是qa，此时一个过滤规则只能维护一个source。

***父数据IDSTR***

：父数据的IDSTR。

***列过滤\_Family***

: 从HBase中查询到comments数据后，如果针对指定的列有过滤规则，可在此处指定column\_family。例如：dcf。

**列过滤\_Column**

:

从HBase中查询到comments数据后，如果针对指定的列有过滤规则，可在此处指定列名。例如：content。

**列过滤\_Compare :**

从HBase中查询到comments数据后，如果针对指定的列有过滤规则，可在此处指定过滤比较符，例如：like；支持>,>=,<,<=,==,like

***列过滤\_Value***

：

从HBase中查询到comments数据后，如果针对指定的列有过滤规则，可在此处指定过滤比较的值，例如：iphonex.

***列过滤\_type***

：

从HBase中查询到comments数据后，如果针对指定的列有过滤规则，可在此处指定过滤比较的值的类型，例如：String

***正则匹配指定列***

：搭配【正则匹配表达式】一起使用，是指对索引查询出来的数据，进行再一次的过滤，必须满足维护的正则表达式，否则数据会被扔掉。使用情况是：例如要查询某个航空公司的数据，有大量的航班号，是以航空公司编码+航班号组成，如CA969，查询索引时，我们用关键词“CA”，但是进项目的数据，必须是CA后面带有几位数字的，此时维护正则表达式为【.\*CA[0-9]{2,4}.\*】.

***正则匹配表达式***

：请参考【正则匹配指定列 】。具体的正则表达式可以找开发人员咨询。

### KeywordsWithUserFilter(ElasticUserFilter)

### 第一次查完数据后，然后对用户进行一次过滤

本过滤规则，和keywords过滤规则差不多，是在keywords过滤的基础上，增加一层user过滤规则。

***数据类型（type）***

：请参考【keywords】中的说明

***数据源（source）***

：请参考【keywords】中的说明

***关键词过滤字段***

：请参考【keywords】中的说明

***过滤关键词***

：请参考【keywords】中的说明

***开始时间***

：请参考【keywords】中的说明

***结束时间***

：请参考【keywords】中的说明

***开始Indices***

：请参考【keywords】中的说明

***Extra***

：请参考【keywords】中的说明

***指定用户ID***

：请参考【keywords】中的说明

***用户过滤规则***

：针对keywords过滤后的数据，查询user数据，如果user满足用户过滤规则，那么该数据保留，否则扔掉。维护用户过滤规则的规范如下：一个规则按照【列名 | 判断符 | 判断值 | 判断值类型】进行维护，举例：【verify\_type|<|1|int】，多个规则之间，&&表示and关系，||表示or的关系，常用例子如下：

去蓝V数据 ： verify\_type|=|null|string||verify\_type|<|1|int||verify\_type|>|7|int

去黄V数据 ： verify\_type|=|null|string||verify\_type|<|0|int||verify\_type|>|7|int

只要蓝V数据：verify\_type|>|0|int&&verify\_type|<|8|int

只要黄V数据：verify\_type|=|0|int

### KeywordsProductComments(elasticProductHBaseComments)

此种数据过滤规则表示，先用keywords的方式过滤出父数据类型，然后查找父数据下的comments子数据。 comments时间和父数据时间必须在时间范围内

***数据类型（type）***

：blog，forum，product等，虽然最终查询出来的事comment数据，但是此处要求填写父数据的数据类型，且，一个过滤规则，只维护一个数据类型。

***数据源（source）***

：请参考【keywords】中的说明

***关键词过滤字段***

：请参考【keywords】中的说明

***过滤关键词***

：请参考【keywords】中的说明

***开始时间***

：请参考【keywords】中的说明

***结束时间***

：请参考【keywords】中的说明

***开始Indices***

：请参考【keywords】中的说明

***Extra***

：请参考【keywords】中的说明

***指定用户ID***

：请参考【keywords】中的说明

### KeywordsQuestionAnswer(elasticQuestionHBaseAnswer)

此数据和KeywordsProductComments查询很类似，不同之处在于，父数据是qa，子数据是answer。

***数据类型（type）***

：qa，虽然最终查询出来的事answer数据，但是此处要求填写父数据的数据类型，且，一个过滤规则，只维护一个数据类型。

***数据源（source）***

：请参考【keywords】中的说明

***关键词过滤字段***

：请参考【keywords】中的说明

***过滤关键词***

：请参考【keywords】中的说明

***开始时间***

：请参考【keywords】中的说明

***结束时间***

：请参考【keywords】中的说明

***开始Indices***

：请参考【keywords】中的说明

***Extra***

：请参考【keywords】中的说明

***指定用户ID***

：请参考【keywords】中的说明

### ElasticProductElasticComments(ElasticProductElasticComments)

此种数据过滤规则表示，先用keywords的方式过滤出父数据类型，然后在通过索引查找父数据下的comments子数据id，最后从hbase中查找comments的详细信息。

***数据类型（type）***

：blog，forum，product等，虽然最终查询出来的事comment数据，但是此处要求填写父数据的数据类型，且，一个过滤规则，只维护一个数据类型。

***数据源（source）***

：请参考【keywords】中的说明

***关键词过滤字段***

：请参考【keywords】中的说明

***过滤关键词***

：请参考【keywords】中的说明

***开始时间***

：请参考【keywords】中的说明

***结束时间***

：请参考【keywords】中的说明

***开始Indices***

：请参考【keywords】中的说明

***Extra***

：请参考【keywords】中的说明

***指定用户ID***

：请参考【keywords】中的说明

### ElasticQuestionElasticAnswer(ElasticQuestionElasticAnswer)

此数据和ElasticProductElasticComments查询很类似，不同之处在于，父数据是qa，子数据是answer。

***数据类型（type）***

：qa，虽然最终查询出来的事answer数据，但是此处要求填写父数据的数据类型，且，一个过滤规则，只维护一个数据类型。

***数据源（source）***

：请参考【keywords】中的说明

***关键词过滤字段***

：请参考【keywords】中的说明

***过滤关键词***

：请参考【keywords】中的说明

***开始时间***

：请参考【keywords】中的说明

***结束时间***

：请参考【keywords】中的说明

***开始Indices***

：请参考【keywords】中的说明

***Extra***

：请参考【keywords】中的说明

***指定用户ID***

：请参考【keywords】中的说明