SearchCondition

**一：简介**

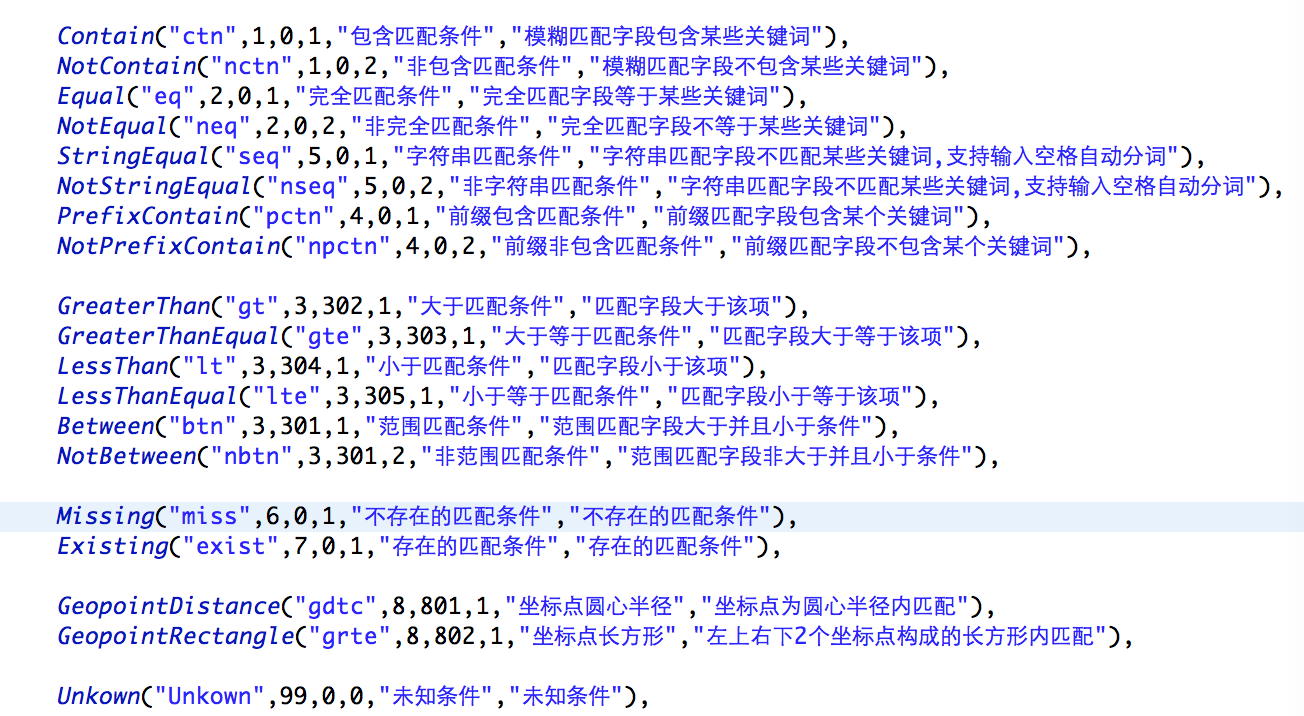
SearchConditionMessage是满足动态搜索条件组合的一种通俗简化的json数据格式

SearchCondition是把简单的SearchConditionMessage转换成复杂的搜索引擎匹配elasticsearch builder，从而进行检索。

**二：数据格式**

1. **搜索条件方式 pattern**

搜索引擎支持丰富的条件匹配方式，为了通俗简化，封装定义了一套适用于不熟悉elasticsearch的人群使用的全新条件方式。

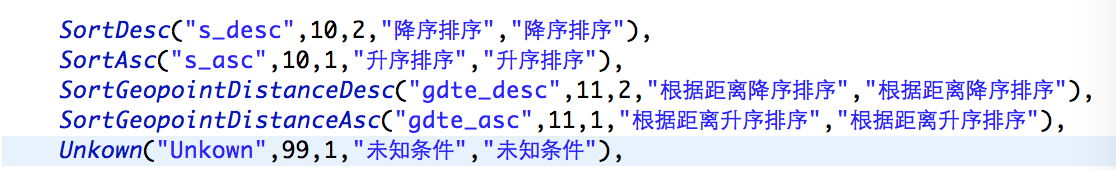


举例来说，其中Contain这个元素，对应的**pattern**是ctn，描述是模糊匹配字段包含某些关键词。类似于数据库中的like ‘%xx%’。

客户端只需要知道所有的**pattern** 对应的效果，并且灵活使用。

1. **排序条件方式pattern**

排序条件也进行了全新定义

****

举例来说，其中SortDesc这个元素，对应的**pattern**是s\_desc，描述是降序排序。类似于数据库中的order by xx desc。

1. **一个简单的独立搜索条件数据格式**

比如搜索引擎有个用户库，包含一个用户昵称的字段叫nick，我们需要查询和”lawliet”字符串完全一样的用户数据。

**{"key":"nick","pattern":"eq","payload":"lawliet"}**

key: 表示对应的需要检索的业务字段名称

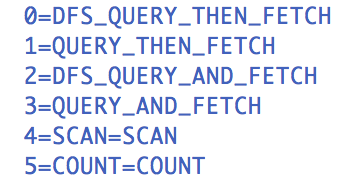
pattern: eq 代表 Equal完全匹配字段等于某些关键词 类似数据库中的 nick = xx

payload: 表示此条件对应的附加信息 对于我们这个条件，附加信息就是用户的昵称 lawliet

1. **完整的搜索多组合条件数据格式**

一个完整的搜索条件数据格式需要2个要素，一些条件和搜索方式

搜索方式如下图：



使用人群并没必要完全了解搜索方式的含义，只记住1代表搜索数据

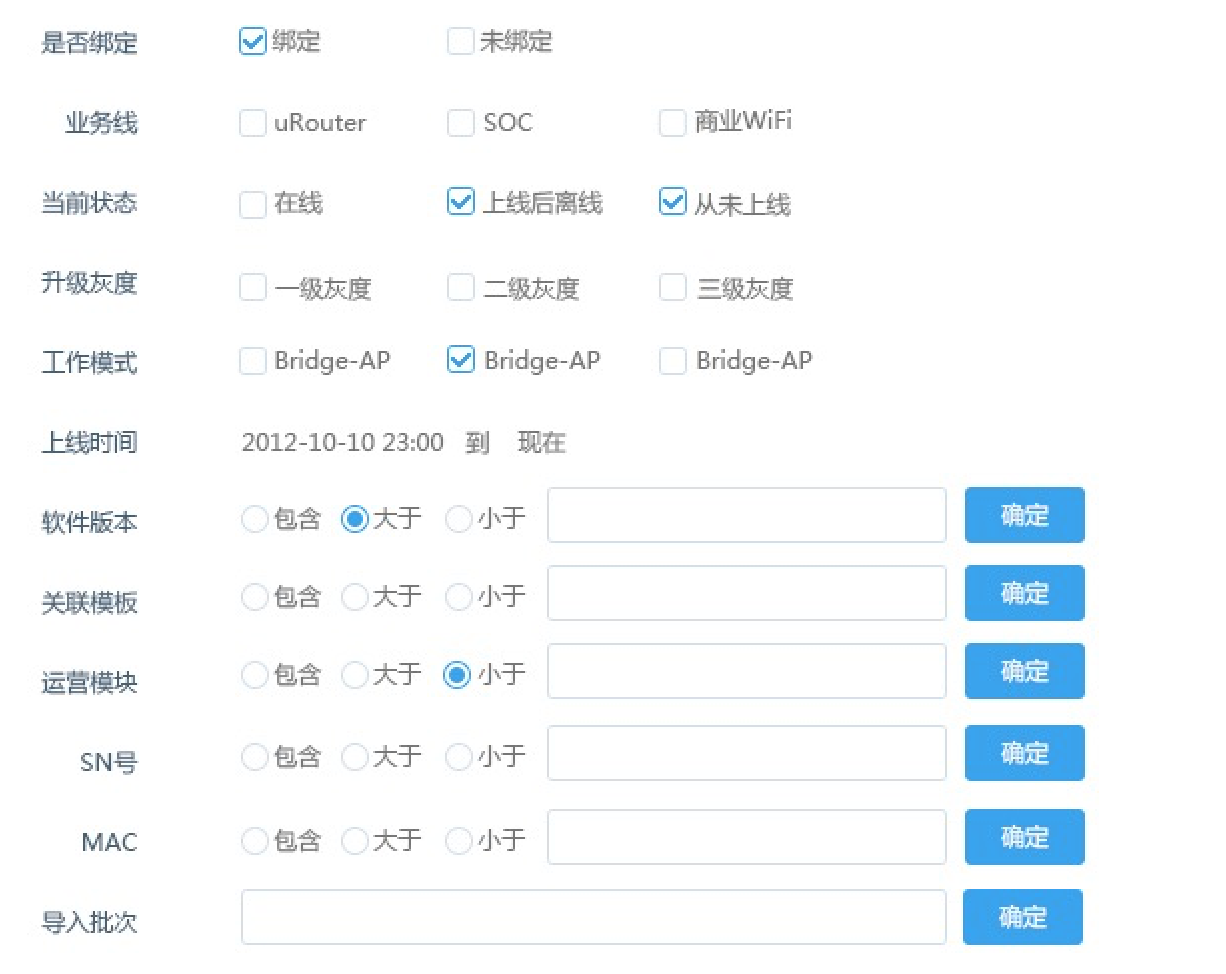
5代表搜索数量

**{"search\_t":1,"search\_cs":[{"key":"nick","pattern":"eq","payload":"lawliet"}]}**

search\_t：搜索方式 我们选1 代表搜索数据

search\_cs: 搜索多组合条件数组 包含了一组在节点3中提到的条件数据

**三：针对现在业务需求实现的条件数据格式**

****

**首选我们针对设备的定义的索引字段如下**

****

1. **是否绑定：对应的索引字段是u\_id，如果是绑定状态的, u\_id字段就会存在用户**id**数据。所以只需要找到u\_id字段存在值的数据。对应条件如下：**

**{"key":"u\_id","pattern":"exist","payload":null}**

1. **业务线：对应的索引字段是d\_dut，分析所见，此条件有多选情况，所以条件方式我们选择StringEqual，因为支持输入自动空格分词。比如我们选择了urouter和soc，这两个值业务线对应的payload值分别为TU和TS，对应条件如下：**

**{"key":"d\_dut","pattern":"** **seq","payload":"TU TS"}**

1. **当前状态：对应的索引字段是d\_online，因为也是支持多选，同2号条件一样，选用StringEqual条件方式。比如我们选择了从未上线和上线后离线，这两个值对应的payload分别为0和2，对应条件如下：**

**{"key":"d\_online ","pattern":"** **seq","payload":"0 2"}**

1. **升级灰度：对应的索引字段是o\_graylevel，因为也是支持多选，同2号条件一样，选用StringEqual条件方式。比如我们选择了一级灰度和二级灰度，这两个值对应的payload分别为gl1和gl2，对应条件如下：**

**{"key":"o\_graylevel ","pattern":"** **seq","payload":"gl1 gl2"}**

1. **工作模式：对应的索引字段是d\_workmodel，因为也是支持多选，同2号条件一样，选用StringEqual条件方式。比如我们选择了router-ap和bridge-ap，对应条件如下：**

**{"key":"d\_workmodel","pattern":"seq","payload":"router-ap bridge-ap"}**

1. **软件版本：对应的索引字段是d\_origswver，比如包含选项，则选用Contain条件方式。对应条件如下：**

**{"key":"d\_origswver","pattern":"ctn","payload":"1.2.18"}**

1. **关联模板: 对应的索引字段是o\_template，比如大于选项，则选用GreaterThan条件方式。此条件方式对应的payload是一个json字符串，对应条件如下:**

**{"key":"o\_template","pattern":"gt","payload":"{\"gtv\":\"style01 \",\"ltv\":null}"}**

**gtv 表示大于或大于等于的值，ltv 表示小于或小于等于的值**

1. **运营模块：对应的索引字段是d\_origvapmodule，比如小于选项，则选用LessThan条件方式。此条件方式对应的payload是一个json字符串，对应条件如下:**

**{"key":"d\_origvapmodule","pattern":"lt","payload":"{\"ltv\":\"H106V1.3.2M8888\",\"gtv\":null}"}**

1. **上线时间: 对应的索引字段是 d\_lastregedat，我们需要范围匹配，则选用Between条件方式。此条件方式对应的payload是一个json字符串，对应条件如下:**

**{"key":"d\_lastregedat","pattern":"btn","payload":"{\"gtv\":\"1446312000000\",\"ltv\":\"1446312600000\"}"}**

**四：SearchCondition分析引擎实现**

**在此不在赘述 见java类**

1. **AbstractDataSearchConditionService.java 实现条件格式转换成elasticsearch builder**
2. **WifiDeviceDataSearchService1Test.java 十多个用例的测试**