

elasticsearch-client封装-接口确认工作基于vap的api-alarmdeal当中关于查询elasticsearch预警相关方法进行对应的整理工作。下面是详细接口整理

1.ElasticsearchConfiguration

作用：连接es数据源的作用

```
1 //es链接配置，在Spring容器当中注入ES的客户端
2 //入参： 1.es集群名称； 2.es node信息
3 //说明： 入参采用yaml配置完成
4 public Client getEsClient(String clustername,String nodes)
```

2.ElasticSearchManage

作用：封装es基础方法

```
1 //判断集群索引是否存在
2 //入参 索引名称: String index
3 public Boolean isExistEsIndex(String index);
4
5 /**
6  * 判断指定索引的类型是否存在
7  * @param indexName
8  * @param type
9  * @return
10  */
11 public Boolean isExistEsTypeOfIndex(String indexName,String type);
12
13 //创建索引-直接创建
14 //入参: 索引名称 String indexName
15 public Boolean createEsIndex(String indexName)
16
17 //创建索引-属性创建索引（包含index, shardCount, replceCount）
18 //入参: String index: 索引名称, int shardCount: 分片数, int replceCount: 副本数
19 public Boolean createEsIndex(String indexName,int shardCount,int replceCount)
20
21 // 创建索引-属性创建索引（包含index, mapping, shardCount, replceCount, fileds）
22 public Boolean createEsIndex(String indexName,String mapping,int
    shardCount,int replceCount,Field[] fileds)
23
24 //刷新所有索引
25 //入参: 无
26 public Boolean refreshAllIndex()
27
28 //刷新指定索引-通过索引名称
```

```
29 //入参: 索引名称: String indexName
30 public void refreshIndexByName(String indexName)
31
32 //获取mapping
33 //入参: 索引名称: String indexName:索引名称,类型: String type
34 public Map<String,Object> getMapping(String indexName,String type)
35
36 //重新设置index的副本数。
37 //入参: String indexName:索引名称,int replicateCount:新副本数
38 public void setReplicateCountNum(String indexName,String replicateCount)
39
40 //索引文档库总数
41 //入参: String indexName:索引名称
42 public int getDocCountByName(String indexName)
43
44 //获得索引分片数
45 //入参: String indexName:索引名称
46 public int getShardCountByName(String indexName)
47
48 //获得索引成功启动的分片总数
49 //入参: String indexName:索引名称
50 public int getSuccessShardCount(String indexName)
51
52 //获得索引库副本总数
53 //入参: String indexName:索引名称
54 public int getAllReplicateCount(String indexName)
55
56 //根据索引获得对应的setting信息
57 //入参: String indexName:索引名称
58 public Map<String,Object> getSetting(String indexName)
59
60 //根据index,type,id,field字段创建对应doc
61 //入参: String indexName:索引名称 type:类型, id:主键id, field: 字段
62 public String createDoc(String indexName,String type,String
    id,Map<String,Object> field)
63
64 //根据id删除对应的doc
65 //入参: String indexName:索引名称,String type: 类型,String id: doc主键
66 public boolean delDocByName(String indexName,String type,String id)
67
68 //删除索引
69 //入参: String indexName:索引名称
70 public boolean delIndexByName(String indexName)
71
72 //获得所有的索引名称
73 //入参: 无
```

```
74 public String[] getAllIndex();
75
76 //获得index的数量
77 //入参: 无
78 public int getIndexCount()
79
80 //获得某一个index下面所有的type
81 //入参: String indexName:索引名称
82 public List<String> getTypesByIndexName(String indexName)
83
84 //获得index所有字段的名称
85 //入参: String indexName:索引名称
86 public Set<String> getAllFieldsByIndexName(String indexName)
87
88 //获得index某一个type所有字段的名称
89 //入参: String indexName:索引名称, String type
90 public Set<String> getAllFieldsByIndexNameAndType(String indexName,String
    type)
91
92 //根据index,type,id获得某一条记录
93 //入参: String indexName:索引名称, String type, String id
94 public GetResponse getDoc(String indexName,String type,String id)
95
96 //根据index, type, searchType, input进行查询
97 //入参String index, String type, String searchType, String input, int start,
    int size
98 public SearchHits getAllDocs(String index, String type, String searchType,
    String input, int start, int size)
99
100 //获得集群名称
101 //入参:无
102 public String getClusterName()
103
104 //获得集群健康度
105 //入参: 无
106 public String getEsClusterHealthStatus()
107
108 //返回ES集群node的个数
109 //入参: 无
110 public int getDataNodeCount()
111
112 //返回集群节点数
113 //入参无
114 public int getClusterNodeCount()
115
116 //判断别名是否存在
```

```

117 //入参: String aliasName: 别名名称
118 public boolean isExist(String aliasName)
119
120 //新增别名
121 //入参: String indexName:索引名称, String aliasName: 别名名称
122 public void addAlias(String index,String aliasName)
123
124 //删除别名
125 //入参: String indexName:索引名称, String aliasName: 别名名称
126 public void delAlias(String index,String aliasName)
127
128 //检查索引状态
129 //入参: String indexName:索引名称
130 public boolean checkEsIndexState(String indexName)
131
132 //更新索引设置
133 //入参: String index:索引名称, Settings settings:索引设置
134 public void updateSettingsByIndex(String index, Settings settings)
135
136 //ES筛选查询
137 //入参: List<QueryCondition> conditions: 筛选条件
138 public QueryBuilder toQueryBuilder(List<QueryCondition> conditions)

```

3.ElasticSearchService<T>

作用：基于ElasticSearchManage的业务封装

```

1 //获得IndexName
2 //入参: 无
3 public abstract String getIndexName()
4
5 //获得TypeName
6 //入参:无
7 public abstract String getTypeName()
8
9 //检查索引是否存在
10 public Boolean isEsIndexExist()
11
12 //检查索引状态(OPEN OR CLOSE)
13 //入参: 无
14 public boolean getEsIndexState()
15
16 //刷新索引
17 public void refreshIndex()
18 //创建索引
19 public boolean createIndex(int shardCount,int replceCount)

```

```
20
21 //通过id获得doc
22 //入参: 主键 Serializable id
23 public T getOne(Serializable id)
24
25 //查询所有的数据
26 //入参: 无
27 public List<T> findAll()
28
29 //查询所有的个数
30 //入参: 无
31 public long count()
32
33 //根据筛选条件查询数据
34 //入参: List<QueryCondition> conditions: 筛选条件
35 public List<T> findAll(List<QueryCondition> conditions)
36
37 //根据筛选条件查询个数
38 //入参: List<QueryCondition> conditions: 筛选条件
39 public long count(List<QueryCondition> conditions)
40
41 //带排序字段以及排序方式查询
42 //入参: List<QueryCondition> conditions: 筛选条件, String value:排序字段,String
    sort:排序方式
43 public List<T> findAll(List<QueryCondition> conditions, String value,String
    sort)
44
45 //根据筛选条件分页查询
46 //入参: PageReq pageReq:分页属性, List<QueryCondition> conditions: 筛选条件
47 public PageRes<T> findByPage(PageReq pageReq,List<QueryCondition> conditons)
48
49 //根据筛选条件进行复杂查询
50 public <Map<String,Object>> queryStatistics(List<QueryCondition> conditions,
    SearchField field)
51
52 //保存数据
53 //T entity:实体数据
54 public Serializable save(T entity)
55
56 //批量保存
57 //List<T> entities: 批量保存数据
58 public void addList(List<T> entities)
59
60 //更新数据
61 //T entity:更新数据
```

```
62 //public void update(T entity)
63
64 //保存或者更新数据
65 //T entity: 更新某条doc
66 //public void saveOrUpdate(T entity)
67
68 //根据筛选条件批量删除doc
69 //入参: List<T> list (筛选出来需要删除的doc集合)
70 public void deleteAll(List<T> list)
71
72 //根据doc的id删除对应的doc
73 //入参: Serializable id: 根据id进行删除
74 public void deleteById(Serializable id)
75
76 //根据实体属性删除对应的doc
77 //T entity:需要删除的实体
78 public void delete(T entity)
79
```