复制从 索引查询接口文档

一、添加maven依赖sdk

<dependency>

<groupId>com.peter.search</groupId>

<artifactId>search-api</artifactId>

<version>参见版本号说明页</version>

</dependency>

二、查询参数示例

```
{
    "serviceTag":"product",
    "queryParam":{
        "iRowSize":10,
        "tenantCode":80001,
        "sFLParam":"title,subTitle",
        "sAggParam":{
            "field":"tenantCode,type",
            "function":"max,min,sum,avg",
            "functionField":"listPrice"
        }
    }
}
```

完整版的查询示例，请参考附件： 查询全集.json

三、常用API使用介绍

单请求查询方法：SearchServiceApi::search(SearchRequestDTO<JSONObject> requestParam)

批量查询请求：SearchServiceApi::mSearch(SearchRequestDTO<JSONOArray> requestParam)

SearchRequestDTO

字段名称	类型	是否必填	描述

queryParam	json / json array	是	查询条件，具体查询语法见下方
searchConfig	map	否	配合查询条件，高级查询使用，具体使用，见本页第六项
serviceTag	String	是	业务标识
useCache	boolean	否	是否使用搜索服务来缓存查询数据

四、查询语法

queryParam中的key默认全部使用下表中的FQ解析器，key就是索引定义的字段名称。

条件之间默认使用and关系。

一些特殊场景需要配合 searchConfig参数使用，详见 下方 高级使用部分 的介绍

解析器名称	功能描述	默认参数名称	值类型	值示例	备注
FQ	等于	字段原始名称	String	注： 1、value中如果有冒号，需要使用 `&colon;` 参见下方FQ值示例	更多规则见下方FQ规则介绍
PAGE_START	起始行	iStart	int	0	默认是从第0行开始，最大值5000
PAGE_SIZE	返回条数	iRowSize	int	10	最大值200
ROUTING	查询路由	routing	String	80001	当数据写入时设置了路由信息时，查询路由可以让查询速度更快
POLYGON	图形区域查询	sPolygonParam	json array	["-10|30", "-40|40", "-10|20", "40|20", "0|0"]	坐标点集合
SORT	排序	sSort	string	column1 desc,column2 asc	多个排序字段以逗号隔开 通过如下字段设置默认排序 [serviceTag]_DF_SORT
SORT_MODE	指定排序模式	sSortMode	string	random	目前仅支持random模式

FL	返回字段	sFLParam	string	title,subTitle	设置返回字段 可以设置默认值，key如下 `searviceTag_DF_FL_FIELD`
EXCLUDE	需要排除掉的字段	sExcludeParam	string	title,subTitle	设置要排除的字段 可以设置默认值，key如下 `searviceTag_DF_EXCLUDE_FIELD`
HL	高亮字段	sHLParam	string	title,subTitle	设置高亮字段

AGGREGATION	聚合查询	sAggParam	json	{
				"field":"tenantCode,type",
				"function":"max,min,sum,avg",
				"functionField":"listPrice",
				"filterJsonStr":{
				"price":10
				},
				"order":"desc",
				"fetchSize":1,
				"source":"roomId,comName,roomStatus"
				}
					字段说明：
					field:聚合字段，多字段以逗号隔开
					function：子聚合函数
					functionField:子聚合函数字段
					fetchSize:返回条数
					source:返回字段
					filterJsonStr：过滤聚合数据条件
					order:排序，desc或asc

RANGE_AGGREGATION	范围聚合查询	sRangeAgg	json	{
				"field":"esUpdateTime",
				"rangeType":"general",
				"ranges": [
				{"from":10, "to":20, "key": "low"},
				{"from":20, "to":30, "key": "medium"},
				{"from":30, "to":40, "key": "high"}
				],
				“format”：“yyyy-MM”
				}
					rangeType可选值：
					- general : 数值范围聚合
					- date_range : 日期范围聚合
					- ip_range : IP 范围聚合
					ranges :范围
					format: 当聚合类型为日期时，可以指定该参数日期格式
					function、functionField、filterJsonStr、order、fetchSize
					source 参数同样支持，同上方的聚合查询

GEO	圆形区域查询	sGeoParam	json	{
				"lat":31.245,
				"lng":121.509,
				"distanceMeters":200,
				"iGeoSort":"desc"
				}
					lat:纬度
					lng:经度
					distanceMeters:查询范围半径
					iGeoSort： desc:由近到远，asc:由远到近

Q	分词查询	sKw	string	我爱中国	可以设置查询字段的查询权重 设置格式如下： sKw_[serviceTag]_ES_NAME = title^0.5,description^0.7,brand^0.3

MULTIPLE_OR	多条件或	sRawORParam	JsonObject或JsonArray	{
				"sRawORParam":{
				"roomId":123,
				"sourceType":1,
				"sRawANDParam":{
				"price":123,
				"size":124
				}
				}
				}
				或者
				{
				"sRawORParam":[
				{
				"roomId":123,
				"sourceType":1
				},
				{
				"roomId":1223,
				"sourceType":2
				},
				{
				"roomId":1233,
				"sourceType":13
				}
				]
				}
					sRawORParam中的所有条件为或的关系
					可以设置参数
					【paramName】_【searviceTag】_MINIMUM_SHOULD_MATCH
					的值为0或1来控制最少要匹配的条件个数，默认是1

MULTIPLE_AND	多条件与	sRawANDParam	JsonObject或JsonArray	{
				"sRawANDParam":{
				"roomId":123,
				"sourceType":1,
				"sRawANDParam":{
				"price":123,
				"size":124
				}
				}
				}
				或者
				{
				"sRawANDParam":[
				{
				"roomId":123,
				"sourceType":1
				},
				{
				"roomId":1223,
				"sourceType":2
				},
				{
				"roomId":1233,
				"sourceType":13
				}
				]
				}
					sRawANDParam中的所有条件为与的关系

WILDCARD	模糊查询	sWildcardWord	string	code00123	等同于sql里的通配符查询*code00123* 需要配合[paramName]_[serviceTag]_ES_NAME使用
RANGE_START	范围开始值	iRangeStart	int	100	需要配合[paramName]_[serviceTag]_ES_NAME使用
RANGE_END	范围结束值	iRangeEnd	int	200	需要配合[paramName]_[serviceTag]_ES_NAME使用
ID	doc id查询	`docIds`	int []	[1,2,3,4]	根据ES文档的doc id查询

RESCORE_ID	二次排序模型	reScoreId	string	"scriptId"	通过设置二次排序的id,对查询结果进行二次排序
RESCORE	二次排序模型	reScore	json	{ "windowSize":200, "reScoreId":"scriptId", "params":{ "p1":"sss", "p2":"sdf" } }	通过设置二次排序的id及入参,对查询结果进行二次排序

FQ值示例

规则名称	值示例	解释	
gt	gt:5	大于5	
lt	lt:5	小于5	
gte	gte:5	大于等于5	
lte	lte:5	小于等于5	
or	or:1,2,3	等于1或等于2或等于3	
and	and:1,2,3	等于1且等于2且等于3	
ra	ra:1,2	大于等于1小于等于2	
ral	ral1,2	大于等于1小于2	
rar	rar:1,2	大于1小于等于2	
prefix	prefix:code0	前缀匹配查询like 'code0%'	
!	!:1	不等于1 该规则可与以上规则共同使用 如：!and:2,3 意思为不等于2且不等于3 !gt:5 意思为小于5	

五、返回参数示例

```
{
    "status":1,
    "totalNum":510,
    "returnNum":3,
    "data":[
        {
            "roomId":21,
            "roomTitle":"1",
            "layout":[
                1,
                2
            ]
        },
        {
            "roomId":22,
            "roomTitle":"2",
            "layout":[
                1,
                2
            ]
        },
        {
            "roomId":23,
            "roomTitle":"3",
            "layout":[
                1,
                2
            ]
        }
    ],
   "aggregationDataMap": {
    "code": [
      {
        "key": "10895252",
        "count": 1,
        "max": null,
        "min": null,
        "sum": null,
        "avg": null,
        "sourceData": {
          "skuList": [
            {
              "code": "23697306",
              "salePrice": "478.0000"
            },
            {
              "code": "23611401",
              "salePrice": "478.0000"
            },
            {
              "code": "23548962",
              "salePrice": "478.0000"
            },
            {
              "code": "23697307",
              "salePrice": "478.0000"
            },
            {
              "code": "23611402",
              "salePrice": "478.0000"
            },
            {
              "code": "23548963",
              "salePrice": "478.0000"
            }
          ]
        }
      },
      {
        "key": "11006002",
        "count": 1,
```

```
        "max": null,
        "min": null,
        "sum": null,
        "avg": null,
        "sourceData": {
          "skuList": [
            {
              "code": "21984529",
              "salePrice": "98.0000"
            },
            {
              "code": "21984549",
              "salePrice": "98.0000"
            },
            {
              "code": "21984554",
              "salePrice": "98.0000"
            },
            {
              "code": "21984530",
              "salePrice": "98.0000"
            },
            {
              "code": "21984550",
              "salePrice": "98.0000"
            },
            {
              "code": "21984555",
              "salePrice": "98.0000"
            },
            {
              "code": "21984531",
              "salePrice": "98.0000"
            },
            {
              "code": "21984551",
              "salePrice": "98.0000"
            },
            {
              "code": "21984556",
              "salePrice": "98.0000"
            },
            {
              "code": "21984532",
              "salePrice": "98.0000"
            },
            {
              "code": "21984552",
              "salePrice": "98.0000"
            },
            {
              "code": "21984557",
              "salePrice": "98.0000"
            }
          ]
        }
      }
    ]
  }
}
```


六、高级使用

1、当自己使用的参数名称与索引中使用的字段名称不一致时，需要指定索引中的字段名称，可以通过设置searchConfig来实现

示例：searchConfig.put("minShowPrice_product_ES_NAME", "showPrice" ）其中minShowPrice是客户端使用的参数名称，showPrice是索引中使用的名称，product是业务标识

2、当多个参数需要使用同一解析器是，需要为参数指定解析器类型，可以通过设置searchConfig来实现

示例：searchConfig.put("minShowPrice_product_QUERY_BUILDER", "RANGE_START" ）其中minShowPrice是客户端使用的参数名称，RANGE_START是解析器类型，product是业务标识

3、当你再创建索引时，指定了某个字段的数据类型为NESTED，则，在查询时，也必须指定该字段为NESTED类型，通过如下代码来进行设置

示例：searchConfig.put("minShowPrice_product_NESTED", "true")， 默认为false

七、附录：SQL转换查询示例