方法列表：

1、new LJSearch();

方法初始化，默认参数为jar包内部system.properties配置

1. new LJSearch(int start,int pageSize,String hostIP,int hostPort)

配置start/pageSize/hostIP/hostPort参数，起始位置/页容量/索引的IP地址/索引的端口号

3、getStart()

getPageSize()

getHostIP()

getHostPort()

setStart()

setPageSize()

setHostIP()

setHostPort()

/\*\*

对start/pageSize/hostIP/hostPort参数的get和set方法

\*/

1. **void** search\_all()

/\*\*

\* 语句： 列出所有数据

\*/

**5、void** search\_and(String attr,String... words)  
/\*\*

\* 语句：且，包含

\* **@param** attr 判断的字段

\* **@param** words 依据的关键词，不定长

\*/

1. **void** search\_combine(LJSearch cmd1,LJSearch cmd2)

/\*\*

\* 语句：两个语句联合搜索

\* **@param** cmd1 语句1

\* **@param** cmd2 语句2

\*/

7、String search\_compare(String attr,LJSearch cmd1,LJSearch cmd2)

/\*\*

\* 语句：比较搜索

\* **@param** attr 字段

\* **@param** cmd1 搜索语句1

\* **@param** cmd2 搜索语句2

\* **@return** 搜索结果

\*/

8、**void** search\_complex(String attr1,Collection word1,Collection word2)

/\*\*

\* 语句：复杂搜索

\* **@param** attr1 搜索的字段

\* **@param** word1 关键词列表1，等同于keyword1|| keyword2||…||keywordN

\* **@param** word2 关键词列表2，等同于word1|| word2||…||wordM

\* 两个列表之间逻辑且，列表元素之间逻辑或

\*/

1. **void** search\_complex(String attr1,Collection word1,Collection word2,**int** dis)

/\*\*

\* 语句：复杂搜索

\* **@param** attr1 搜索的字段

\* **@param** word1 关键词列表1，等同于keyword1|| keyword2||…||keywordN

\* **@param** word2 关键词列表2，等同于word1|| word2||…||wordM

\* **@param** dis 两个关键词之间的距离

\* 两个列表之间逻辑且，列表元素之间逻辑或

\*/

10、**void** search\_fuzzy(String attr,String... words)

/\*\*

\* 语句：模糊搜索

\* **@param** attr 搜索的字段

\* **@param** words 依据的关键词

\*/

11、**void** search\_jzor(LJSearch... cmds)

/\*\*

\* 语句：搜索结果逻辑或

\* **@param** cmds 多个搜索语句，对它们的结果进行逻辑或运算

\*/

12、**void** search\_max(String attr,String v1)/\*\*

\* 语句：最大值搜索

\* **@param** attr 判断的字段，必须为数值型或datetime型

\* **@param** v1 值

\*/

13、**void** search\_near(String attr,String value1,String value2,**int** dis)

/\*\*

\* 语句：临近搜索

\* **@param** attr 搜索的字段

\* **@param** v1 依据的第一个关键词

\* **@param** v2 依据的第二个关键词

\* **@param** dis 两个关键词之间的最大距离

\*/

14、**void** search\_not(String attr,String... words)

/\*\*

\* 语句：非，不包含

\* 必须和其他语句联合使用，否则无结果

\* **@param** attr 判断的字段

\* **@param** words 依据的的关键词

\*/

1. **void** search\_or(String attr,String... words)

/\*\*

\* 语句：或

\* **@param** attr 判断的字段

\* **@param** words 依据的关键词

\*/

1. **void** search\_precision(String attr,String... words)

/\*\*

\* 语句：精准匹配，（对应模糊匹配）

\* 其他搜索中，将关键词加双引号，有同样效果

\* **@param** attr 搜索的字段

\* **@param** words 依据的关键词，不定长

\*/

1. **void** search\_prefix(String attr,String v1)

/\*\*

\* 语句：前缀匹配搜索

\* **@param** attr 搜索的字段，必须为字符型

\* **@param** v1 依据的关键词

\*/

18、**void** search\_range(String attr,String v1,String v2)

/\*\*

\* 语句：范围搜索

\* **@param** attr 搜索的字段，字段类型要求为数值型或日期datetime型，其表示形式为：2012/06/20\_00:57:17，需要在日期与时间之间加上“\_”

\* **@param** v1 范围的起始

\* **@param** v2 范围的终结

\*/

19、set\_sentiment(**boolean** isOpt)

/\*\*

\* 设置：筛选正负面新闻

\* **@param** isOpt 选择显示正面信息？

\*/

1. **void** set\_sort(String attr,**boolean** notReverse)

/\*\*

\* 设置：排序方式

\* **@param** attr 排序依据的字段名称

\* **@param** notReverse 是否按从小到大的正序排列

\*/

1. **void** set\_showDeleted(**boolean** isShow)

/\*\*

\* 设置：是否显示被删除的数据

\* **@param** isShow 是否显示

\*/

1. set\_cluster(**int** num,String...fieldnames)

/\*\*

\* 设置：聚类

\* **@param** num 类别数量

\* **@param** fieldnames 聚类依据的字段名称，为自动生成的字段

\*/

1. **void** set\_groupby(String... fieldnames)

/\*\*

\* 设置：分组

\* **@param** fieldnames 分组依据的字段，字段类型必须是数值型

\*/

1. **void** set\_unique(String key)

/\*\*

\* 设置：去重

\* **@param** key 去重依据的字段

\*/

1. search\_not(String attr,String... words)

/\*\*

\* 语句：非，不包含

\* 必须和其他语句联合使用，否则无结果

\* **@param** attr 判断的字段

\* **@param** words 依据的的关键词

\*/

1. List<Map<String,String>> run()

/\*\*

\* 运行，对query进行搜索

\* **@return** 结果，List<Map<String,String>>类型

\*/

调用步骤：

引入jar包

新建LJSearch对象

调用对象的成员函数，运行set\_\*与search\_\*方法，进行功能设置

运行run()方法，执行搜索，返回值类型为List<Map<String,String>>

示例：

LJSearch a = new LJSearch();

a.setHostIP("127.0.0.1");a.setHostPort(98);

// a = new LJSearch(0,20,"127.0.0.1",98);

a.search\_all();

a.set\_unique("id");

List<Map<String,String>> res = a.run();

System.out.println(res);