# 介绍

## 官方主页

<https://www.elastic.co/guide/cn/index.html>

ElasticSearch以下简称ES

## 为什么选择ES

Elasticsearch 是一个开源的搜索引擎，建立在一个全文搜索引擎库 Apache Lucene™ 基础之上。 Lucene 可以说是当下最先进、高性能、全功能的搜索引擎库—​无论是开源还是私有。

Elasticsearch 不仅仅是 Lucene，并且也不仅仅只是一个全文搜索引擎。 它可以被下面这样准确的形容：

一个分布式的实时文档存储，每个字段 可以被索引与搜索

一个分布式实时分析搜索引擎

能胜任上百个服务节点的扩展，并支持 PB 级别的结构化或者非结构化数据

## 版本

本文档使用的是新版 7.8.1

# 安装

## Jdk环境

ES需要JDK环境最低要求是jdk8，也可以下载带有JDK的安装包。

## ES下载安装

<https://www.elastic.co/cn/downloads/elasticsearch>

下载后解压即可

## 安装中文分词插件

下载对应版本：<https://github.com/medcl/elasticsearch-analysis-ik>

解压到：elasticsearch-7.8.1\plugins

## 安装拼音分词插件

下载对应版本：<https://github.com/medcl/elasticsearch-analysis-pinyin>

解压到：elasticsearch-7.8.1\plugins

# 配置

## 配置文件

elasticsearch-7.8.1\config\ elasticsearch.yml

## 修改配置

可修改索引数据目录与日志地址，端口，集群等

## 更多配置说明



此文件可打开查看

## 启动ES

Windows: elasticsearch-7.8.1\bin\ elasticsearch.bat

Linux: elasticsearch-7.8.1\bin\ elasticsearch

# 新建索引

## 测试工具

Curl、postman、浏览器ElasticSearch Head插件

推荐使用ElasticSearch Head插件，可在google插件商城安装

## 新建索引

提交PUT: http://127.0.0.1:9200/test\_index

创建一个索引名称为：test\_index的索引，提交参数如下json

索引设置及字段mappings

{

"settings": {

"number\_of\_shards": 3,

"number\_of\_replicas": 2,

"analysis": {

"analyzer": {

"pinyin\_analyzer": {

"tokenizer": "my\_pinyin"

}

},

"tokenizer": {

"my\_pinyin": {

"type": "pinyin",

"keep\_separate\_first\_letter": false,

"keep\_full\_pinyin": true,

"keep\_original": true,

"limit\_first\_letter\_length": 16,

"lowercase": true,

"remove\_duplicated\_term": true

}

}

}

},

"mappings": {

"properties": {

"content": {

"type": "text",

"analyzer": "ik\_max\_word",

"search\_analyzer": "ik\_max\_word",

"store": true,

"fields": {

"py": {

"type": "text",

"store": false,

"term\_vector": "with\_offsets",

"analyzer": "pinyin\_analyzer"

}

}

}

}

}

}

## 添加文档

POST: <http://localhost:9200/test_index/_create/1/>

{"content": "美国留给伊拉克的是个烂摊子吗"}

{"content": "中国邮政储蓄银行湖南省株洲市茶陵县交通街支行"}

{"content": "公安部：各地校车将享最高路权"}

{"content": "中韩渔警冲突调查：韩警平均每天扣1艘中国渔船"}

{"content": "中国驻洛杉矶领事馆遭亚裔男子枪击 嫌犯已自首"}

逐条执行添加文档数据到索引，方便后面测试（每条\_create后的id要更换）

# RestApi简单查询

## 测试分词插件

POST: <http://localhost:9200/test_index/_analyze/>

{

"analyzer": "ik\_max\_word",

"text": "中国邮政银行"

}

提交结果如下，表示分词成功：

{

"tokens": [

{

"token": "中国邮政",

"start\_offset": 0,

"end\_offset": 4,

"type": "CN\_WORD",

"position": 0

},

{

"token": "中国",

"start\_offset": 0,

"end\_offset": 2,

"type": "CN\_WORD",

"position": 1

},

{

"token": "邮政",

"start\_offset": 2,

"end\_offset": 4,

"type": "CN\_WORD",

"position": 2

},

{

"token": "银行",

"start\_offset": 4,

"end\_offset": 6,

"type": "CN\_WORD",

"position": 3

}

]

}

## 高亮查询

POST: <http://localhost:9200/test_index/_search/>

{

"query": {

"match": {

"content": "中国调查"

}

},

"highlight": {

"pre\_tags": [

"<tag>"

],

"post\_tags": [

"</tag>"

],

"fields": {

"content": {}

}

}

}

查询结果如下：

截取一部分：

{

"\_index": "test\_index",

"\_type": "\_doc",

"\_id": "3",

"\_score": 1.3909361,

"\_source": {

"content": "中韩渔警冲突调查：韩警平均每天扣1艘中国渔船"

},

"highlight": {

"content": [

"中韩渔警冲突<tag>调查</tag>：韩警平均每天扣1艘<tag>中国</tag>渔船"

]

}

}

## Term查询

term是代表完全匹配，即不进行分词器分析，文档中必须包含整个搜索的词汇

POST: <http://localhost:9200/test_index/_search/>

{

"query": {

"term": {

"content": "中国邮政"

}

}

}

查出的所有文档都包含"中国邮政"这个词组的词汇

## match\_phrase查询

match\_phrase：必须全部匹配，还要索引位置相邻(顺序一致)

POST: <http://localhost:9200/test_index/_search/>

{

"query": {

"match\_phrase": {

"content": {

"query": "中国邮政",

"slop": 1

}

}

}

}

如果换成查询邮政中国，则没有任何结果

我们会希望有个可调节因子，相邻夸度可调节放宽，那就需要使用到slop

## bool查询

如果我们想要请求"content中带中国，但是content中不带洛杉矶"这样类似的需求，就可以用到bool联合查询

POST: <http://localhost:9200/test_index/_search/>

{

"query": {

"bool": {

"must": {

"term": {

"content": "中国"

}

},

"must\_not": {

"term": {

"content": "洛杉矶"

}

}

}

}

}

# Java客户端使用

## 添加maven依赖

需要和ElasticSearch服务端版本一致

<dependency>

<groupId>**org.elasticsearch.client**</groupId>

<artifactId>**elasticsearch-rest-high-level-client**</artifactId>

<version>**7.8.1**</version>

</dependency>

## 单元测试

RestHighLevelClient client;

### 创建索引

String indexName **=** "test\_doc"**;**

XContentBuilder builder **=** XContentFactory**.**jsonBuilder**()**

**.**startObject**()**

**.**field**(**"properties"**)**

**.**startObject**()**

**.**field**(**"id"**).**startObject**()**

**.**field**(**"store"**,** **true)**

**.**field**(**"index"**,** **true)**

**.**field**(**"type"**,** "long"**)**

**.**endObject**()**

**.**field**(**"title"**).**startObject**()**

**.**field**(**"analyzer"**,** "ik\_max\_word"**)**

**.**field**(**"search\_analyzer"**,** "ik\_max\_word"**)**

**.**field**(**"store"**,** **true)**

**.**field**(**"index"**,** **true)**

**.**field**(**"type"**,** "text"**)**

**.**endObject**()**

**.**endObject**()**

**.**endObject**();**

CreateIndexRequest createIndexRequest **=** **new** CreateIndexRequest**(**indexName**);**

createIndexRequest**.**mapping**(**builder**);**

CreateIndexResponse createIndexResponse **=** client**.**indices**().**create**(**createIndexRequest**,** RequestOptions**.**DEFAULT**);**

boolean acknowledged **=** createIndexResponse**.**isAcknowledged**();**

System**.**out**.**println**(**"isAcknowledged::" **+** acknowledged**);**

### 添加文档

@Test

public void testAdd**()** **throws** IOException **{**

IndexRequest indexRequest **=** **new** IndexRequest**(**"test\_doc"**);**

Map**<**String**,** Object**>** source **=** **new** HashMap**<>();**

source**.**put**(**"id"**,** 1000**);**

source**.**put**(**"title"**,** "中国农业银行东莞南城分行"**);**

source**.**put**(**"test\_not"**,** "这是一个不存在的字段"**);**

indexRequest**.**source**(**source**);**

indexRequest**.**id**(**source**.**get**(**"id"**).**toString**());**

IndexResponse res **=** client**.**index**(**indexRequest**,** RequestOptions**.**DEFAULT**);**

log**.**info**(**res**.**toString**());**

**}**

### 文档查询

@Test

public void testScrollQuery**()** **throws** Exception **{**

String indexName **=** "test\_doc"**;**

SearchSourceBuilder searchBuilder **=** **new** SearchSourceBuilder**();**

searchBuilder**.**size**(**3**)**

**.**highlighter**(new** HighlightBuilder**().**field**(**"title"**).**preTags**(**"<em>"**).**postTags**(**"</em>"**))**

**.**query**(**QueryBuilders**.**multiMatchQuery**(**"农业银行"**,** "title"**)**

**.**type**(**MultiMatchQueryBuilder**.**Type**.**MOST\_FIELDS**));**

SearchResponse resp **=** client**.**searchScroll**(**indexName**,** **null,** TimeValue**.**timeValueMinutes**(**5**),** searchBuilder**);**

List**<**Map**<**String**,** Object**>>** list **=** ElasticSearchUtils**.**getHitSourceAsMap**(**resp**,** **true);**

log**.**info**(**"scrollId:{}"**,** resp**.**getScrollId**());**

log**.**info**(**JSON**.**toJSONString**(**list**,** **true));**

**}**

### 更多查询

1．请参考官网

2．个人测试示例：<https://github.com/phh2008/elastic-search-client>