ElasticSearch

源码分析：

<http://www.jianshu.com/u/1dbe35392705>

权威指南：

<https://es.xiaoleilu.com/040_Distributed_CRUD/15_Create_index_delete.html>

翻译:

<http://cwiki.apachecn.org/pages/viewpage.action?pageId=4260558>

1 安装head插件

<http://www.jianshu.com/p/dc1eee5a82ef>

2 elastic-5 依赖jdk8

1 选用哪种api？

<https://www.elastic.co/blog/benchmarking-rest-client-transport-client>

2 客户端选择

<http://qkxue.net/info/200166/elasticsearch>

5 查询

区分match和term的：

1 term是把查询的内容作为一个整体。如：“ hello world”

2 match是需要对查询的内容进行分词：“Hello word” -> “

# 一、查询

## 1 term查询

## 2 terms查询

1 简单用法是：类似sql中的in

2 复杂用法: in中的数组，可以通过再另外的type中查找的结果。

## 3 range查询

1 for string fields, the TermRangeQuery，怎么对字符串使用范围？

## 4 Exists查询

某个字段下面是有值或没值。

## 5 Prefix Query

匹配term中是以指定字符串为前缀的。 输入的查询内容（前缀）并不会分词。

## 6 Wildcard Query

使用通配符匹配term。

支持的通配符为：\*和？， 为了保证效率，不能使用通配符开头。

## 7 Regexp Query

对term使用正则表达式匹配。

## 8 模糊查询

与我们理解的使用正则，模糊查询时不同的。在6.0版本就被废弃，使用match替代。

## 9 Ids Query

根据mapping中的\_id来查询

# 二、聚合

指定列相同的值汇总在一起。

<http://xiaorui.cc/2015/11/01/%E8%81%8A%E8%81%8Apython-elasticsearch%E8%81%9A%E5%90%88%E8%AE%A1%E7%AE%97%E7%9A%84%E7%94%A8%E6%B3%95/>

## 1 Metric Aggregations

指标聚合，生成一系列指标。

需要给出1 指定列名 2 对这一列生成什么指标统计：是最大值、最小值、还是计数。

1 值计数聚合

2 最大值聚合

3 最小值聚合

4 平均值聚合

## 2 Bucket Aggregations

### 1 Children Aggregations

父子文档类型

### 2 Date Histogram Aggregation

时间直方图聚合：根据date类型列聚合，设置间隔时长。

### 3 Date Range Aggregation

日期范围聚合。

### 4 Filter Aggregation

过滤器聚合。

aggs同filter一起使用。

先根据字段聚合后，然后找出过滤器中指定内容的bucket.

还是先过滤，再聚合呢？？？？

### 5 Filters Aggregation

aggs中有多个filters。

过滤字段就是聚合字段。

### 6 Term Aggregation

参考：<http://qindongliang.iteye.com/blog/2374138>

# 三、Mapping

## 1 Field datatypes

<https://www.elastic.co/guide/en/elasticsearch/reference/current/mapping-types.html>

1 date类型

2 ip类型

## 2 mapping parameters

<https://www.elastic.co/guide/en/elasticsearch/reference/current/mapping-params.html#mapping-params>

### 1 norms

1存储了计算得分的factors。 如果不需要某个字段计算得分，可以设置为false。

2 使用磁盘

### 2 fields

如果某个字段有两种用途

1 分词后查询

2 排序和聚合

## 3 Meta-Fields

<https://www.elastic.co/guide/en/elasticsearch/reference/current/mapping-fields.html>

#### 1 \_routing

1 根据\_routing字段的哈希值，来定位分片id。

2 写入和查找都可以指定\_routing

# 四、Analysis

## Analyzers

## Tokenizers

#### 1 NGram Tokenizer

# Compound queries

组合查询：类似sql中的子查询。利用上次查询的结果。

## 1 bool query

1 must

2 filter

must和filter的区别：

must需要计算score

filter：在filter上下文中执行，不计算score，并会缓存结果。

3 should

1最少匹配should中的一个条件，可以通过minimun\_should\_match来设置。

2 根据在query context和filter context中不同的表现形式。

在query context中，有must和should， 则should可以一个都不匹配。

在filter context中，

# How to

## 1 一般建议

1 不要返回大量的数据集

2 与db是不同的使用场景: 不要像db一样对某一个特定条件进行全部查询。

## 2 避免大文件（单条Doc）

1 大文件，es还会有一个source字段，存储双倍。

2 索引这个文档的时候会占用大量的内存。

3 邻近搜索（例如短语搜索）和高亮显示更为昂贵。

## 3 避免稀疏

# Meta-Fields

<https://www.elastic.co/guide/en/elasticsearch/reference/current/mapping-fields.html>

## 1 \_source

## 2 \_id

所有的查询，都会获取\_id这个字段。

## 3 \_all

1 默认情况下，除了给你所有的字段分别创建索引，elasticsearch还会把他们一起放进一个叫\_all的新字段里做索引。好处是你可以在\_all里搜索那些你不在乎在哪个字段找到的东西。另一面是在创建索引和增大索引大小的时候会使用额外更多的CPU。所以如果你不用这个特性的话，关掉它。即使你用，最好也考虑一下定义清楚限定哪些字段包含进\_all里。详见[链接](http://www.elasticsearch.org/guide/reference/mapping/all-field.html)。

2 所有的内容不管什么类型（无论是日期还是数值等）拼接成字符串，即为\_all, 然后对\_all进行分词，默认但是不会被store（可以设置）。

3 用途：不关心在哪个字段被检索到。如果不需要这个用途，则关闭。

4 可以自定义\_all字段，使用哪些字段进行拼接。

# Mapping属性

## 1 store

原始数据是否会被存储。如果不被存储，则不能被retrieved。

# Blog

## 1 Text vs keyword

<https://www.elastic.co/blog/strings-are-dead-long-live-strings>

1 string类型被废弃了，使用Text(分词，全问检索)和keyword（不分词，关键字检索）

2 动态mapping，默认类型

## 2 多index还是多个type

原理：

1 父子文档，必须使用一个index下的多个type

2 同时查询多个index，则最好使用多个type。

因为每个index都是独立查询。

## 3 es的ttl

<https://www.elastic.co/blog/ttl-documents-shield-and-found>

## 4 深度分页：scroll

<http://xiaorui.cc/2016/03/28/%E4%BD%BF%E7%94%A8elasticsearch%E7%9A%84scroll%E5%92%8Cscan%E8%A7%A3%E5%86%B3%E6%B7%B1%E5%BA%A6%E5%88%86%E9%A1%B5%E9%97%AE%E9%A2%98/>

## 5 选择哪种客户端

<https://www.elastic.co/blog/found-interfacing-elasticsearch-picking-client>

<http://qkxue.net/info/200166/elasticsearch>

## 6 两种客户端基准测试

<https://www.elastic.co/blog/benchmarking-rest-client-transport-client>

# Java REST Client

<https://www.elastic.co/guide/en/elasticsearch/client/java-rest/current/index.html>

<http://www.cnblogs.com/shihuc/p/6000725.html>

# 项目中的问题

## 1 精准匹配

a 如果基于标准的分词，搜索内容是多个完整的单词，可使用phrase\_match,把它们看做一个整体进行匹配。

b 如果不是基于标准的分词，类似数据库的模糊查询。

<http://jingyan.baidu.com/article/380abd0a1aa8631d90192cf6.html>