<https://www.cnblogs.com/dreamroute/p/8484457.html>

查询-详细介绍

<https://blog.csdn.net/u014646662/article/details/89010759>

中文文档

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

<https://es.xiaoleilu.com/010_Intro/15_API.html>

基本操作

<https://www.cnblogs.com/shizhijie/p/10332970.html>

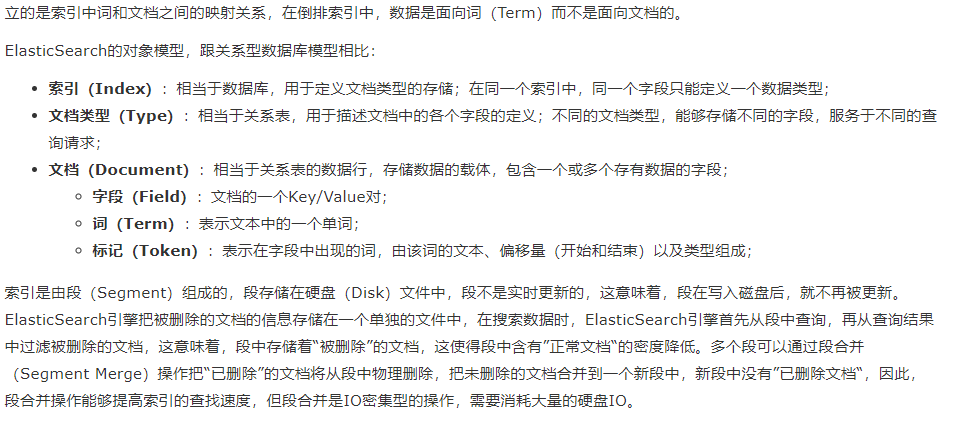
<https://www.cnblogs.com/wangshouchang/p/8049492.html>

<https://blog.csdn.net/mj86534210/article/details/79910909>

<https://blog.csdn.net/chujuntu1211/article/details/100818675>

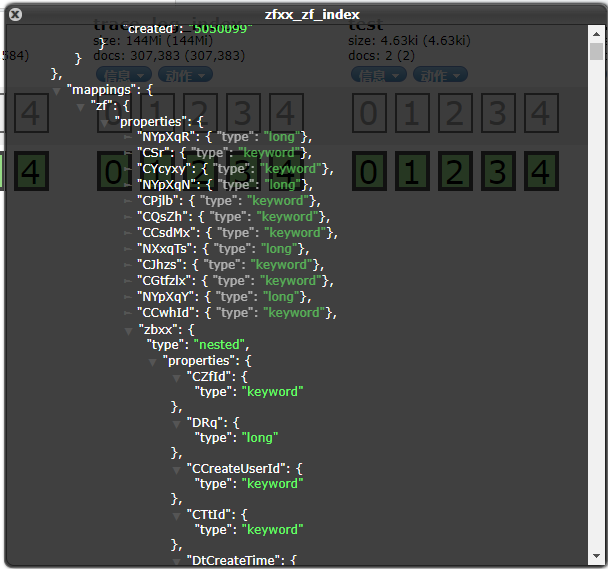
<https://www.cnblogs.com/soinve/p/9397261.html>

# 基本介绍





# Mappering映射-数据类型

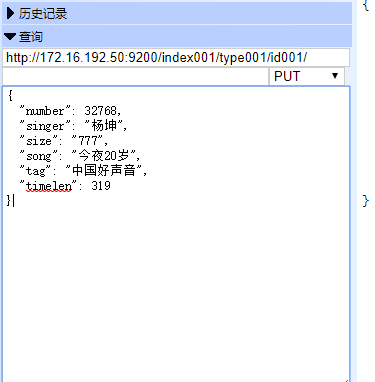


这里涉及到的是数据类型。Long类型和keyword类型就是简单的类型，像nested就是复杂的类型。Zf是一个大类型，在properties中配置各个字段的类型。但是这个zbxx就是一个复杂的nested类型，这里就是嵌套。

# 创建索引

创建索引和类型，并且给id值。并插入一条记录。

只需要改变记录,再次执行又会添加一条记录.但是,如果这个地方id已经存在,就会将之前的值覆盖.



# Elasticsearch版本变更

6.0.0移除了一个索引允许映射多个类型，虽然还支持同索引多类型查询，但是Elasticsearch 7.0.0的版本将完全放弃。

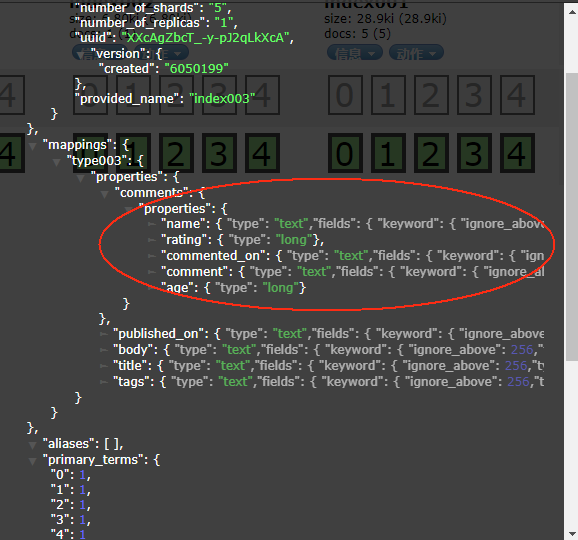
# Es保存数据

向es中添加这条数据，

|  |
| --- |
| {  "title": "Invest Money",  "body": "Please start investing money as soon...",  "tags": ["money", "invest"],  "published\_on": "18 Oct 2017",  "comments": [  {  "name": "William",  "age": 34,  "rating": 8,  "comment": "Nice article..",  "commented\_on": "30 Nov 2017"  },  {  "name": "John",  "age": 38,  "rating": 9,  "comment": "I started investing after reading this.",  "commented\_on": "25 Nov 2017"  },  {  "name": "Smith",  "age": 33,  "rating": 7,  "comment": "Very good post",  "commented\_on": "20 Nov 2017"  }  ]  } |

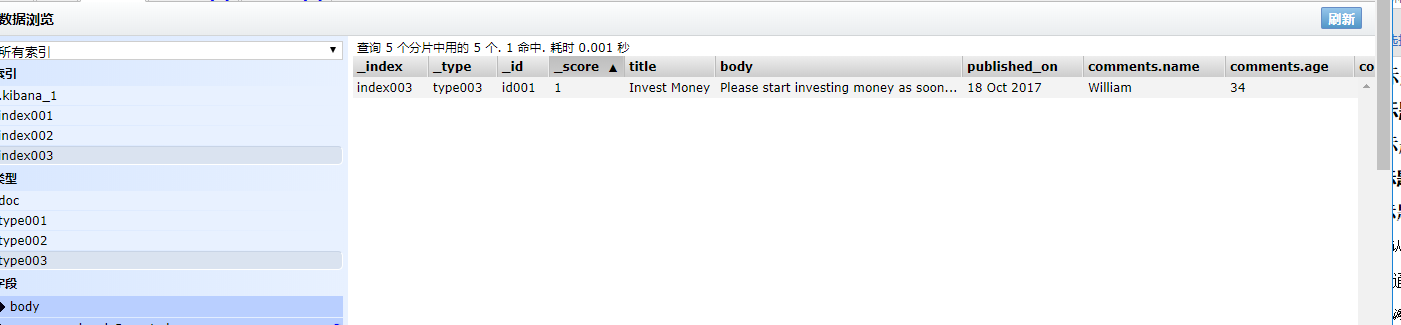
Mappering映射如下：





因为comments这个是一个复杂字段，所以就将这个字段做进一步的划分。

## 但是有个问题就是es是怎么保存的。



可以看出，es中保存了一条记录，但是comments这个是如何保存的？

# Es写数据

1. 客户端选择一个node发送请求过去，这个node就是coordinate node(协调节点)。
2. Coordinate node对document进行路由，将请求转发给对应的node（有primary shard）
3. 实际node上的primary shard处理请求，然后将数据同步到replica node
4. Coordinate node发现primary node和所有的replica node都搞定之后，就返回响应结果给客户端。

# 读数据

1. 客户端发送请求到一个coordinate node
2. 协调节点将搜索请求转发到所有的shard对应的primary shard或replica shard都可以
3. Query phase：每个shard将自己的搜索结果(其实就是一些doc id)返回给协调节点，有协调节点进行数据的合并、排序、分页、等操作产生最终结果
4. Fetch phase：接着由协调节点去各个节点上拉取实际的document数据，最终返回给客户端。

# elasticsearchTemplate

term是代表完全匹配，也就是精确查询，搜索前不会再对搜索词进行分词，所以我们的搜索词必须是文档分词集合中的一个

TermsBuilder:构造聚合函数

AggregationBuilders:创建聚合函数工具类

BoolQueryBuilder:拼装连接(查询)条件

QueryBuilders:简单的静态工厂”导入静态”使用。主要作用是查询条件(关系),如区间\精确\多值等条件

NativeSearchQueryBuilder:将连接条件和聚合函数等组合

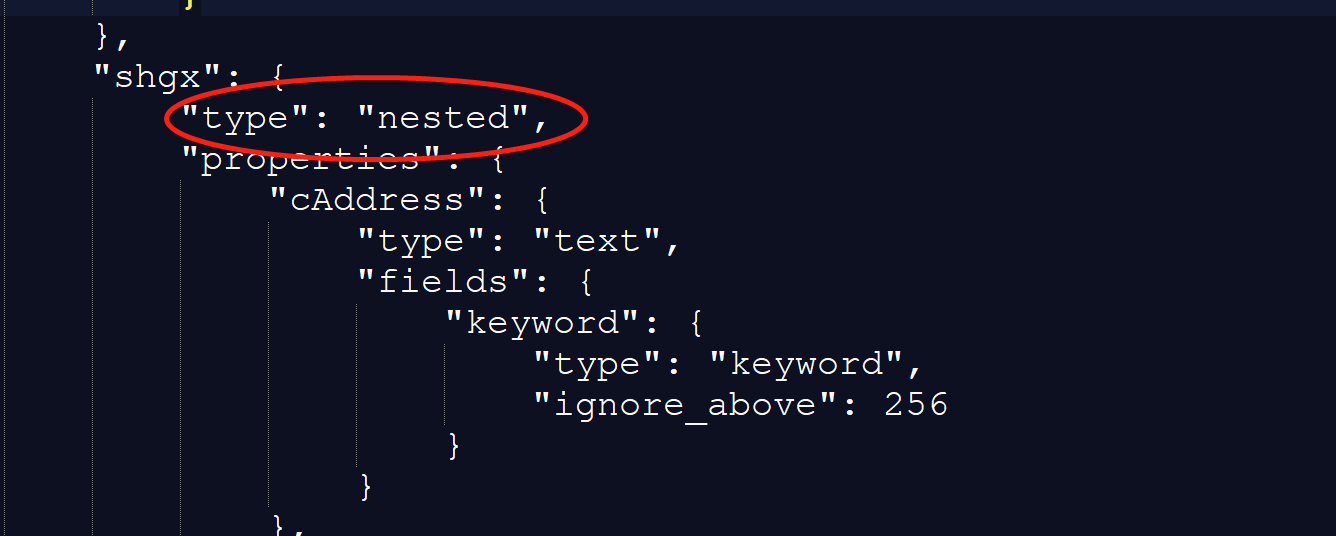
SearchQuery:生成查询

elasticsearchTemplate.query:进行查询

Aggregations:Represents a set of computed addAggregation.代表一组添加聚合函数统计后的数据

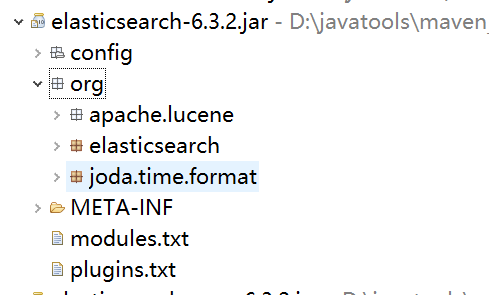
Bucket:满足某个条件(聚合)的文档集合

# 同步数据时嵌套对象如果为空同步失败问题

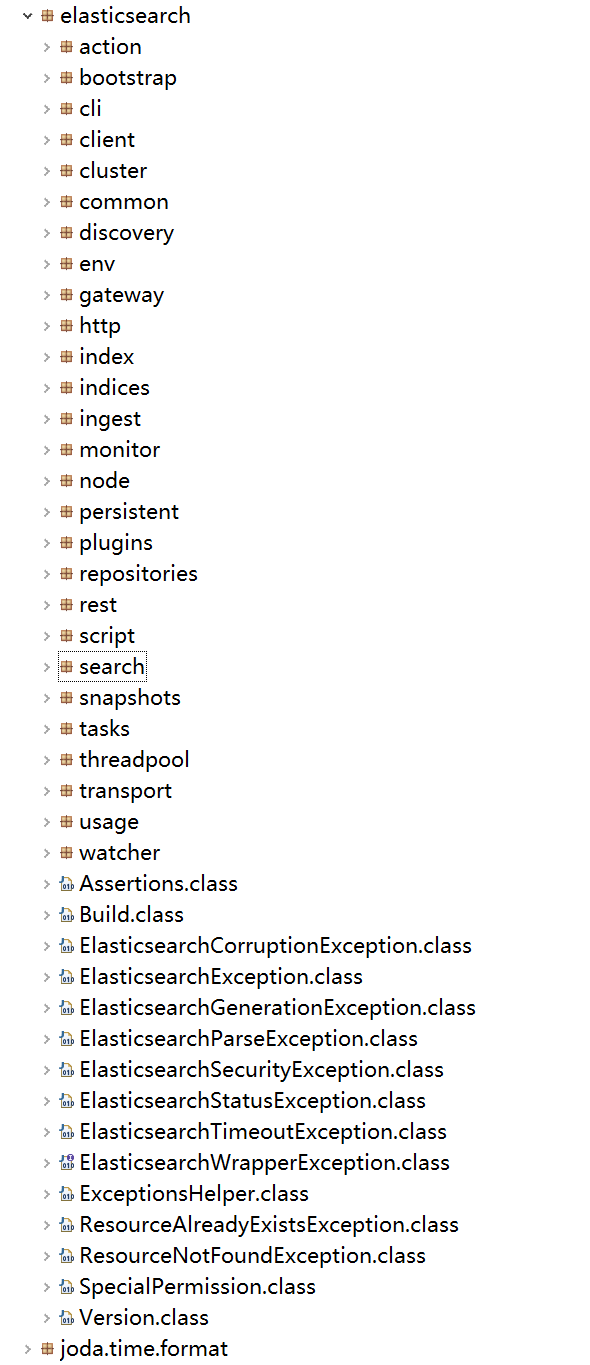


社会关系是用户的嵌套字段，如果映射文件中不指定类型，如果社会关系为空，则该用户不能进行同步到elasticsearch中。

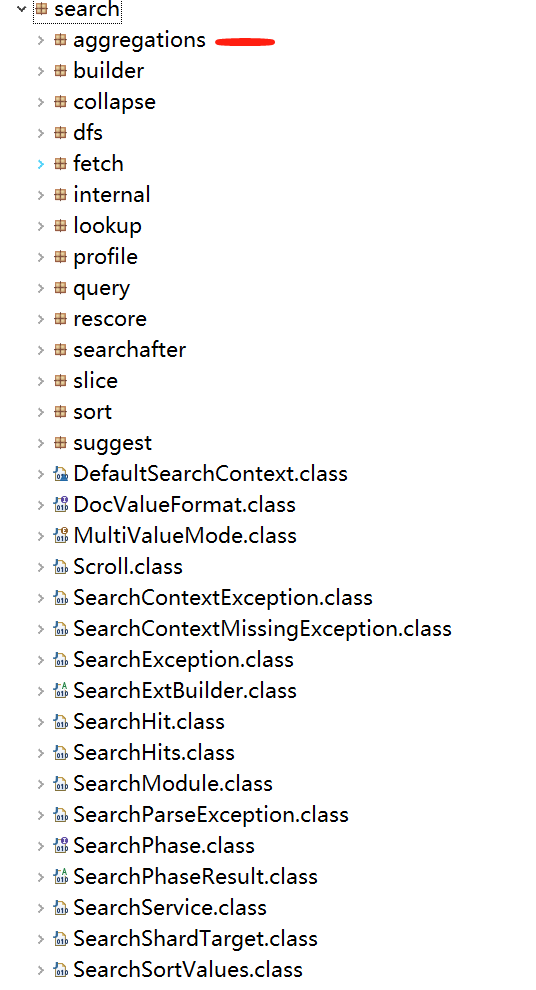
# 包结构



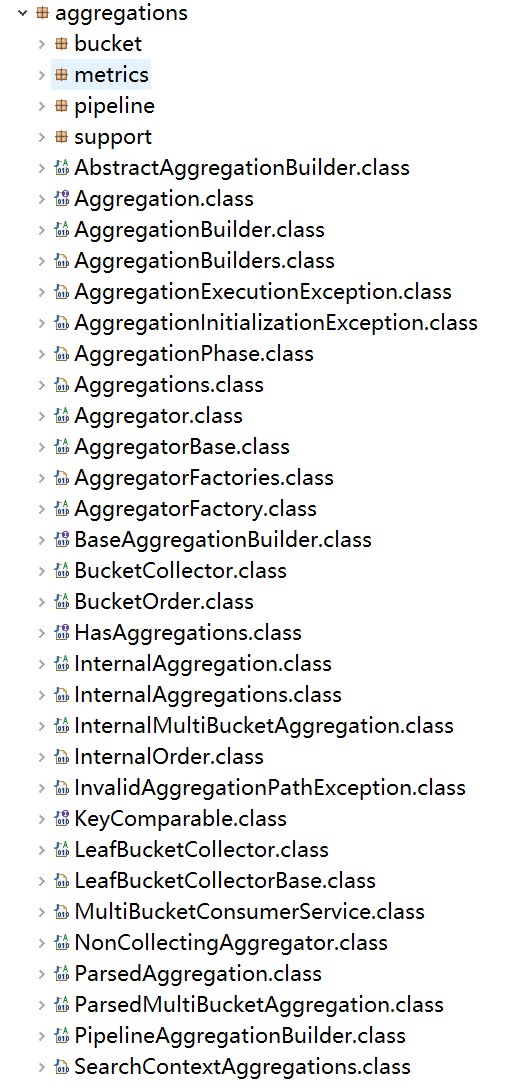
## Elasticsearch

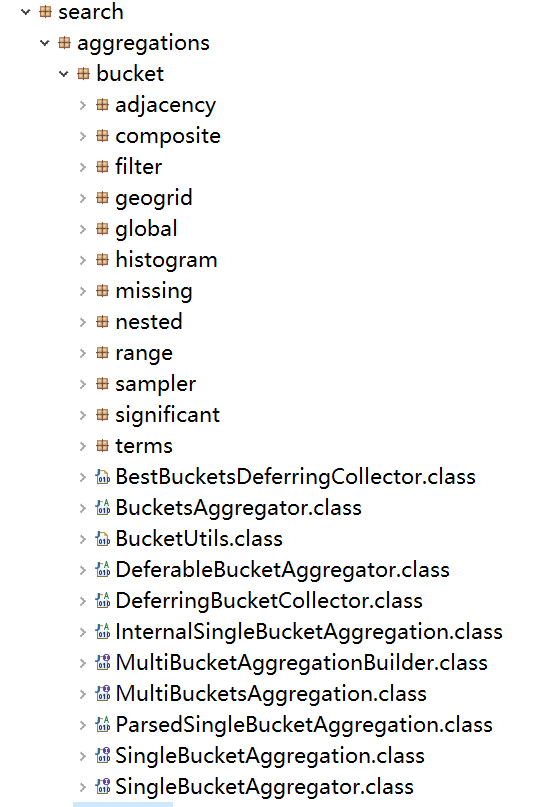


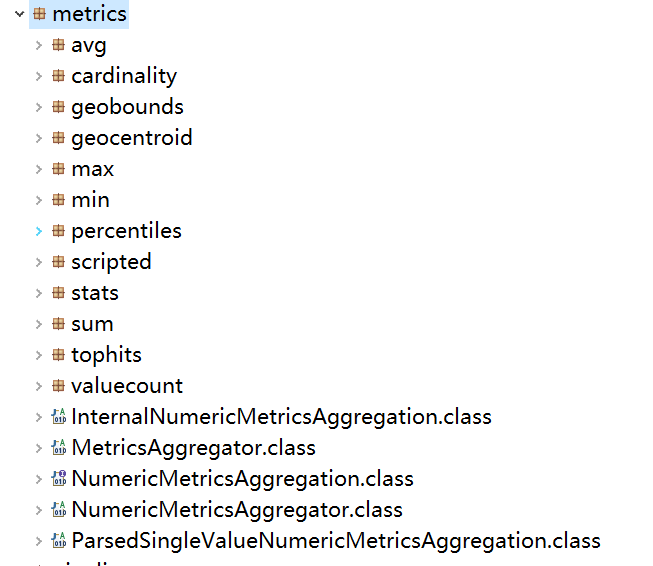
### Search

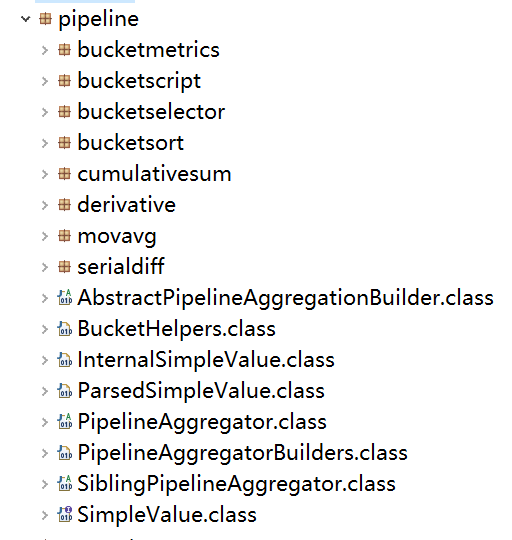


#### Aggregations









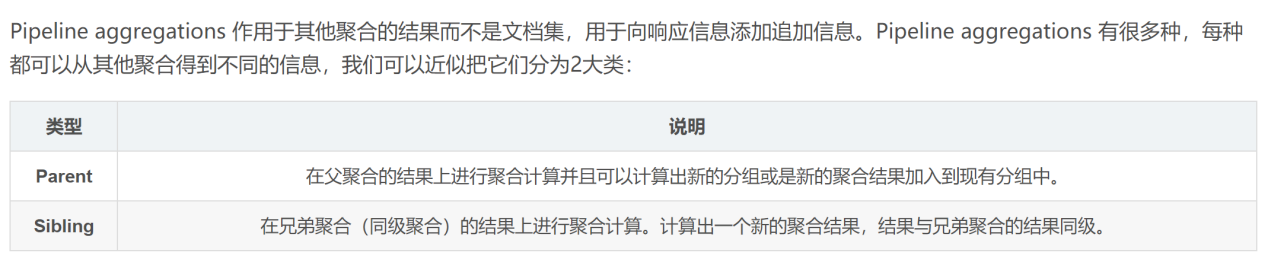
#### Builder

# Bucket metric pipeline

Bucket 一个数据分组

Metrics 对一个数据分组进行的分析

Pipeline



## Bucket