

1、重建索引

一个 field 的设置时不能被修改的，如果要修改一个 Field，那么应该重新按照新的 mapping，建立一个 index，然后将数据批量查询出来，重新用 bulk [api](#) 写入 index 中批量查询的时候，建议采用 scroll api，并且采用多线程并发的方式来reindex 数据，每次 scroll 就查询指定如期的一段数据，交给一个线程即可

```
1 GET /old_index/_search?scroll = 1m
2 {
3     "query":{
4         "range":{
5             "date":{
6                 "gte": "2017-01-01",
7                 "lt":"2017-02-01"
8             }
9         }
10    },
11    "sort":["_doc"],
12    "size": 1000
13 }
```

2、基于alias 对 client 透明切换 index

```
1 PUT my_index2/_alias/my_index_v1 //
   old_index_name/_alias/new_index_name 老index名 --》 新 index 名
```

client 可以使用 my_index_v1 对 index 进行操作，等 reindex 操作完成之后，再通过 alias 方法，切换到新的 index 即可

```
1 POST /_aliases
2 {
3     "actions":
4     [
5         {"remove":{"index":"my_index2","alias":"my_index_v1"}},
6         {"add":{"index":"ecommerce","alias":"my_index_v1"}}
7     ]
8 }
```

```
7   ]  
8 }
```

3、零停机重建索引 详细过程

1. 一开始，依靠 dynamic mapping，插入，但是不小心有些数据是 2017-01-01这种日期格式的，所以 title 这种 field 被自动映射为了 date 类型
2. 当后期想索引中加入 string 类型的 title值得时候，就会报错
3. 再不修改 index的情况下，此时想修改 title 类型，是不可能的
4. 此时，唯一的办法，就是进行 reindex，也就是重新建立一个索引，将旧索引的数据查询出来，在导入新索引
5. 如果说旧索引的名字，是 old_index，新索引的名字是new_index, 终端 java 应用，已经在使用 old_index在操作了，难道还要停止java 应用，修改使用的 index 为 new_index, 才重新启动java 应用？在这个过程中，就会导致 java 应用停机，可用性降低
6. 所以说，给 java 应用一个别名，这个别名是指向就索引的，java 应用先用着

-》起别名
7. 新建一个 index，调整其title的类型为 string

-》建立新索引，修正错误
8. 使用 scroll api将数据批量查询出来
9. 采用 bulk api 将 scroll查出来的一批数据，批量写入新索引
10. 循环 8~9 步骤，查询一批又一批的数据出来，采用 bulk api 将每一批数据批量写入新索引
-----》通过 scroll 和 bulk 批量转移数据
11. 将 work_index alias 切换到 new_index 上去，java 应用会直接通过 index 别名使用新的索引中的数据，java应用程序不需要停机，零停机，高可用
----》切换别名 到修正的新的 index 上
12. 然后直接 通过 work_index alias 别名来查询，是否 ok