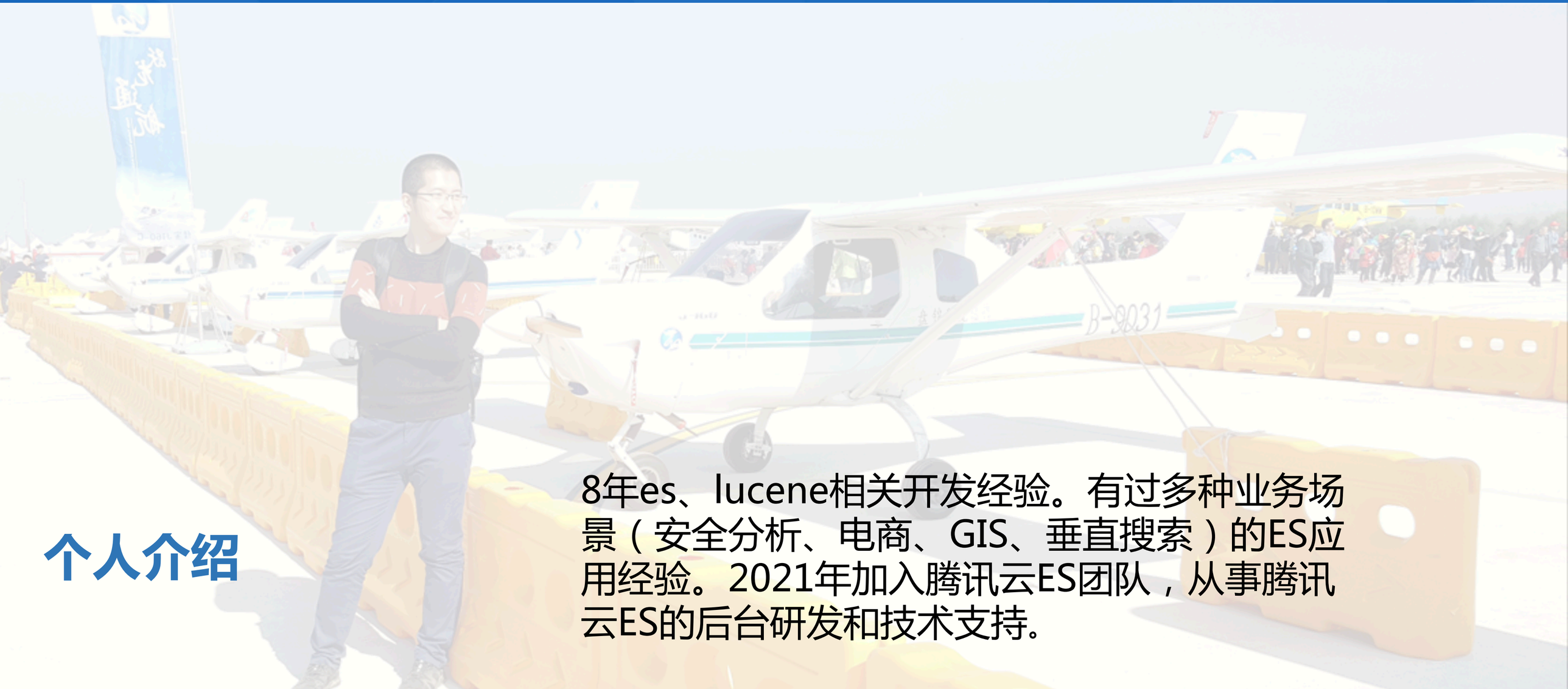
多场景下ES索引管理的最佳实践

刘忠奇
腾讯云ElasticSearch高级开发工程师



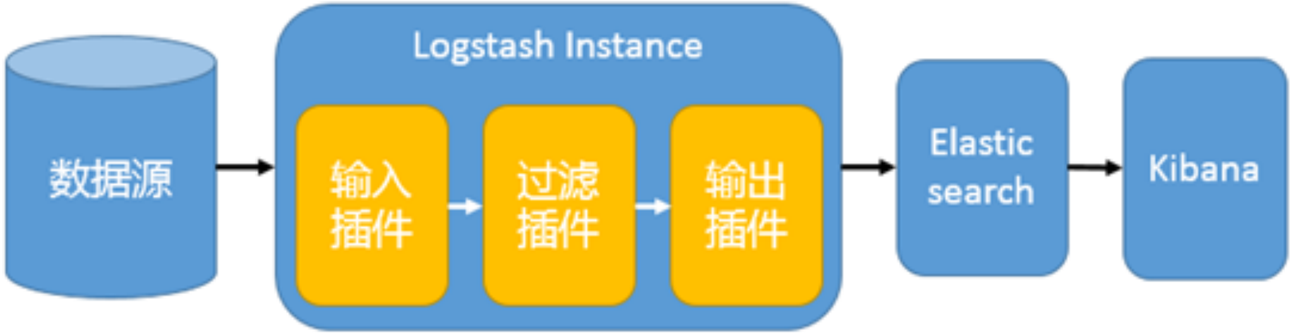
个人介绍

8年es、lucene相关开发经验。有过多种业务场景（安全分析、电商、GIS、垂直搜索）的ES应用经验。2021年加入腾讯云ES团队，从事腾讯云ES的后台研发和技术支持。

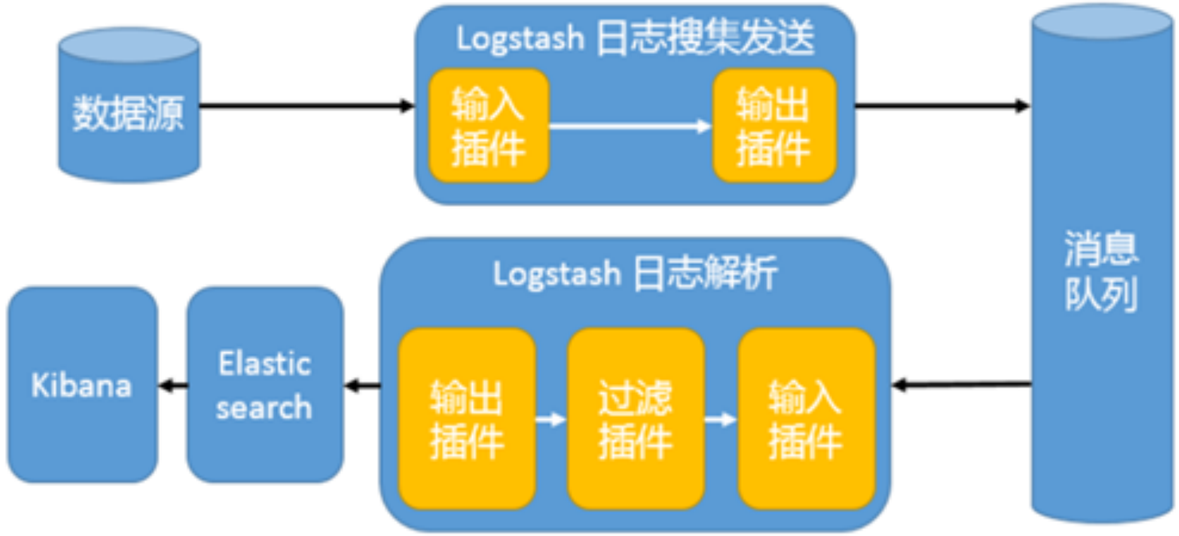
目录概要

日志、监控、时序数据类应用场景

广为人知的ELK架构



引入消息队列的ELK架构



日志、监控、时序数据类应用场景

更多采集类型

FILEBEAT
日志文件

HEARTBEAT
服务可用性监控

METRICBEAT
指标数据

AUDITBEAT
Linux审核框架事件

FUNCTIONBEAT
云服务器监控

PACKETBEAT
网络数据

更多解析、添加、修改、过滤手段

通过processor（ingest node/Logstash）

- Append
- Bytes
- Circle
- Community ID
- Convert
- CSV
- Date
- Date index name
- Dissect
- Dot expander
- Drop
- Enrich
- Fail
- Fingerprint
- Foreach
- GeoIP
- Grok
- Gsub
- HTML strip
- Inference
- Join
- JSON
- KV
- Lowercase
- Network direction
- Pipeline
- Remove
- Rename
- Script
- Set
- Set security user
- Sort
- Split
- Trim
- Uppercase
- URL decode
- URI parts
- User agent

日志、监控、时序数据类应用场景

ILM (Index Lifecycle Management) 索引生命周期管理

ILM Action: Rollover

能够实现根据**索引的大小、文档数和创建时间**自动切换到新的索引。当Rollover触发时，新的索引将被创建，Alias 别名自动指向新索引，且新索引被设置为is_write_index=true。

Rollover必须结合Alias一起使用。

Rollover 优势：

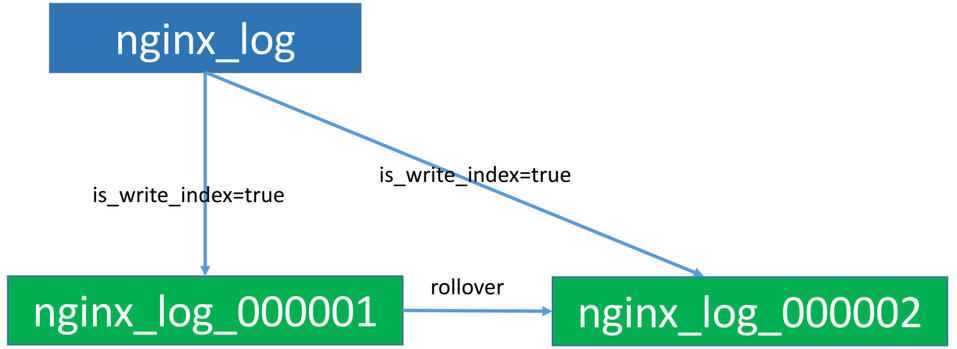
- 避免单个索引过大
- 提升索引写入性能
- 方便管理索引数据

Rollover 避坑指南：

合理设置触发 Rollover的Condition，否则容易导致集群中大量的小索引，使得总分片数过多，影响查询性能和集群稳定性。

设置索引 Rollover conditions：

```
POST /nginx-log/_rollover
{
 "conditions": {
  "max_age":  ”1d",
  "max_docs":  100000000,
  "max_size": ”30gb"
 }
}
```



ILM Action: Shrink

Shrink降索引分片的利器，Shrink API可以实现将存量的索引主分片数量收缩到一个很小的值，新索引的主分片数量必须是原索引主分片数量的因子。例如原索引主分片数量为20，则Shrink后的索引主分片数量可设置为 1、2、4、5、10。

前提条件：

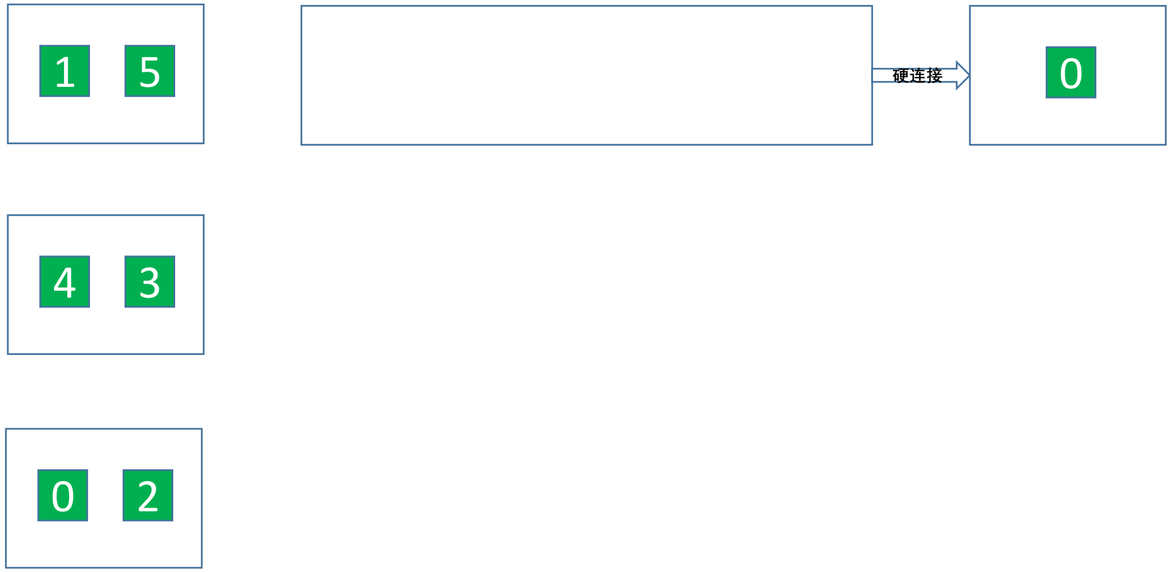
- 索引必须是只读的
- 索引的健康状态必须是Green
- 所有的主分片或副本分片都必须集中到一个节点上

将源索引设置为只读，且将分片汇集到某一个特定节点：

```
PUT nginx-log-000001
{
    "settings": {
        "index.routing.allocation.require._name": "160372000288532",
        "index.blocks.write": true
    }
}
```

对源索引执行Shrink操作，将源索引分片缩小到1个：

```
POST /nginx-log-000001/_shrink/nginx-log-000001_shrink
  {
      "settings": {
       "index.number_of_shards": 1
  }
}
```



ILM Action: Frozen

ES 索引的三种状态

- Open
- Frozen
- Close

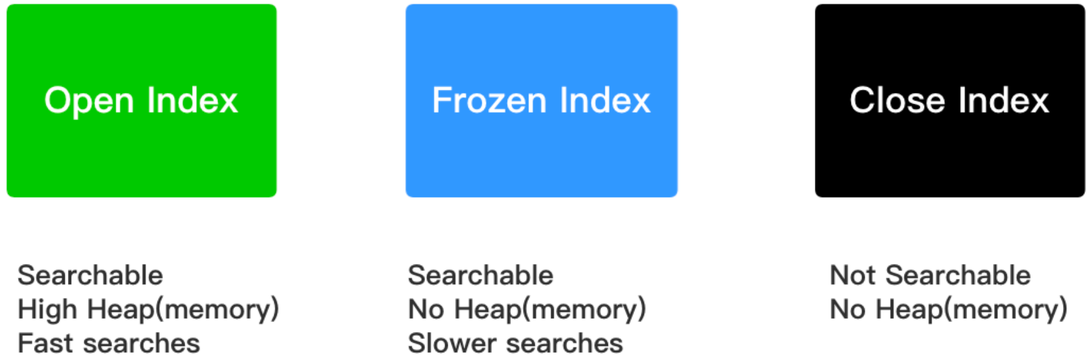
适用场景：时序数据+冷热分离架构

冻结索引能够释放JVM堆内存空间，又能够被搜索到，在需要搜索的时候，会将冻结索引的数据加载进内存，查询完成后再释放。

冻结索引API：
POST /nginx_log-000001/_freeze

解冻索引API：
POST /nginx_log-000001/_unfreeze

将冻结索引纳入搜索范围：
GET /nginx_log/_search?ignore_throttled=false



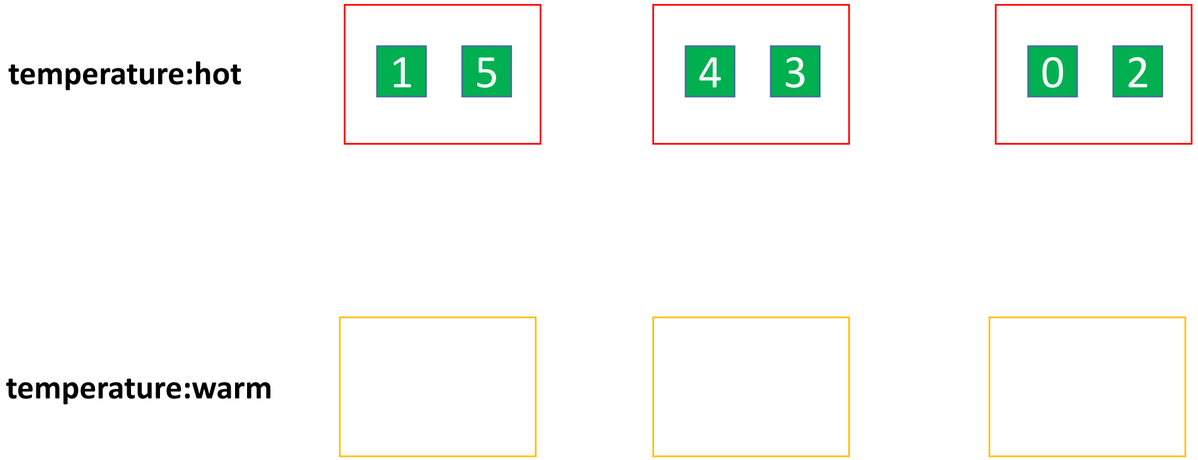
ILM Action: Allocate

Allocate 可作用的Phase为warm和cold。
必须指定include、exclude、require中的一个选项。

include: 将索引分片分配到至少满足其中一个属性的节点上；

require: 将索引分片分配到满足所有属性的节点上；

exclude: 将索引分片分配到不包含其中任何一个属性的节点上。



修改索引的温度属性，将索引分片迁移到温节点：

```
PUT nginx-log-000001
{
  "settings": {
    "index.routing.allocation.require.temperature":
"warm"
  }
}
```

日志、监控、时序数据类应用场景

以日志场景为例，结合腾讯云ES集群冷热分离架构，介绍如何使用ILM：

1、将新索引实时写入到ES集群中的热节点上，当索引达到特定条件后，数据滚动 (Rollover) 写入到新索引；

2、索引在热节点上滚动完成后在hot阶段停留三天后，迁移 (Allocate) 到温节点，即进入warm阶段；

3、warm阶段将索引设置为只读 (Read-only)，将副本设置为0，将主分片个数缩小 (Shrink) 到1个；

4、索引在温节点上停留7天后（从滚动更新时算起），进入delete阶段；

5、索引阶段delete阶段后执行删除 (Delete) 操作。

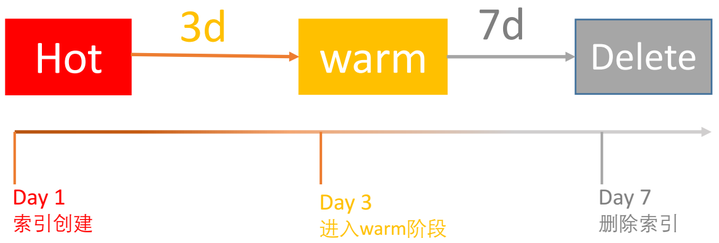
操作步骤：

第一步：创建Policy

第二步：创建索引模版

第三步：创建初始索引

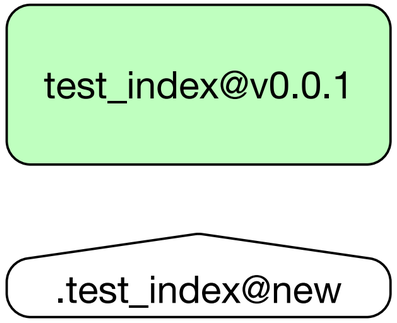
第四步：通过别名写入数据



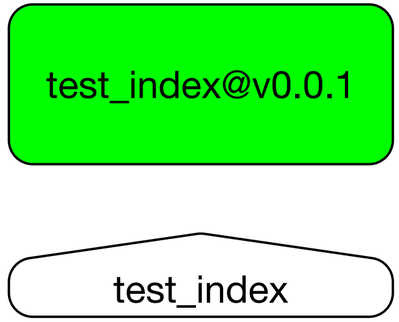
目录概要

数据库加速、搜索、推荐类应用场景

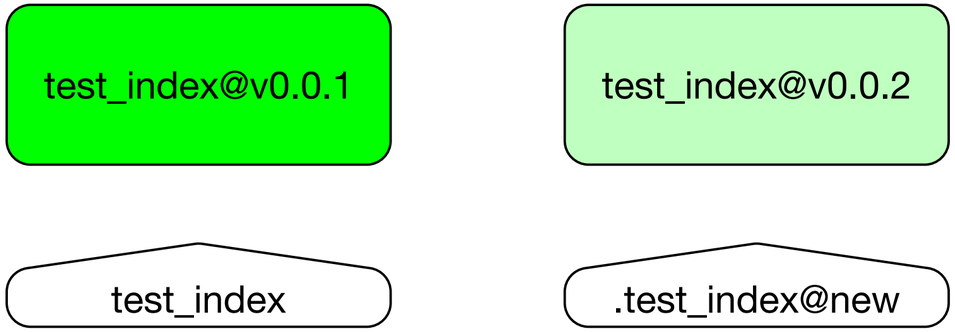
首次写入索引



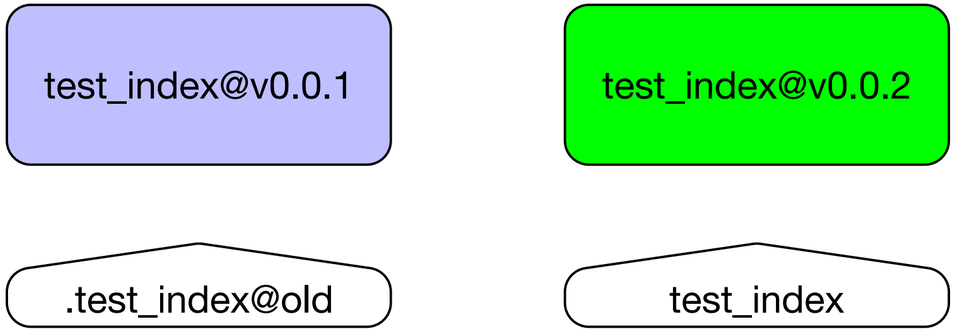
写入完成



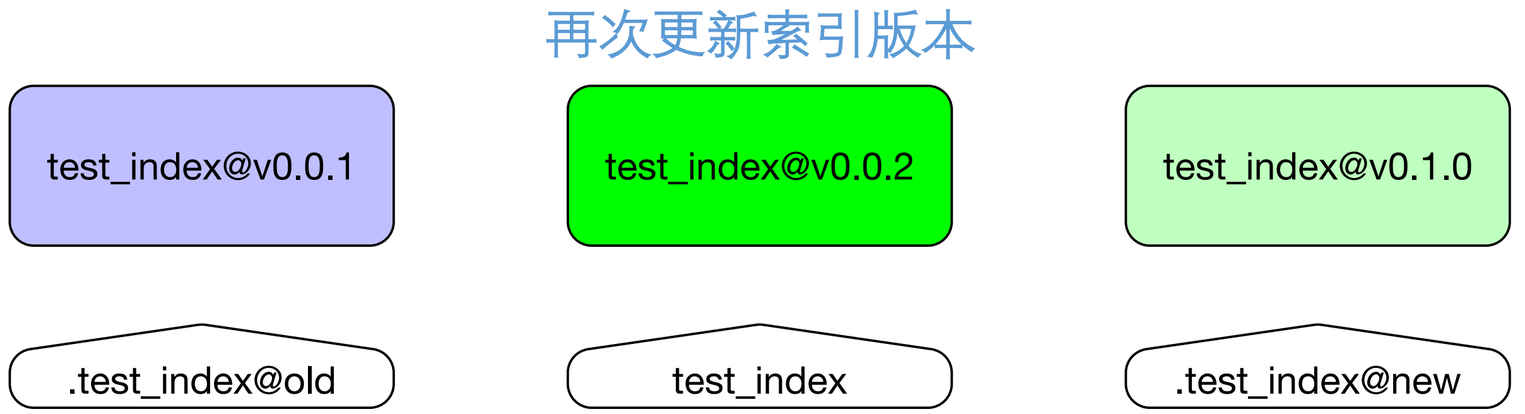
更新索引版本

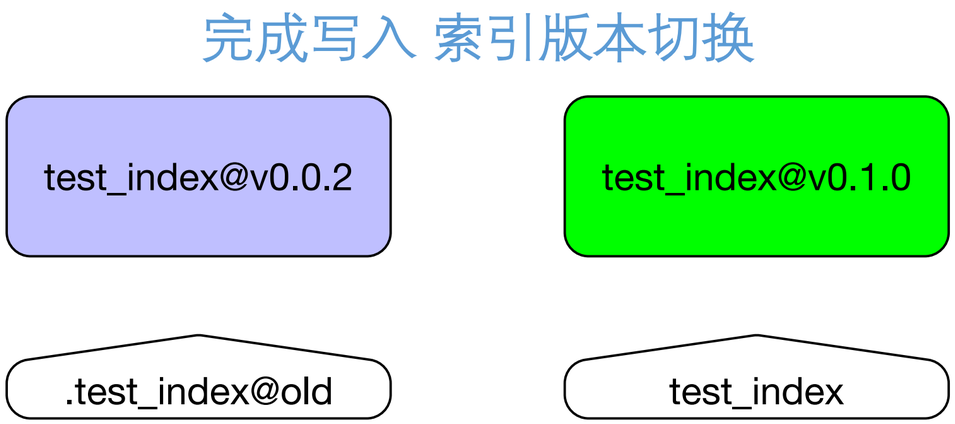


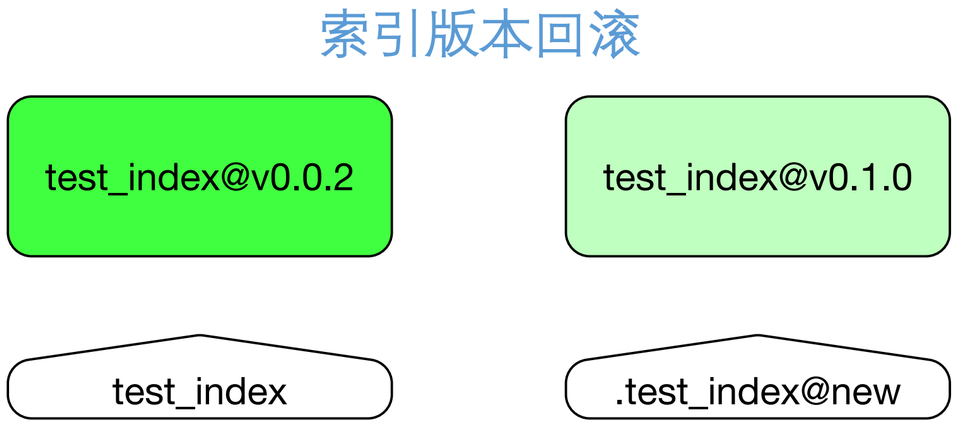
完成写入 索引版本切换



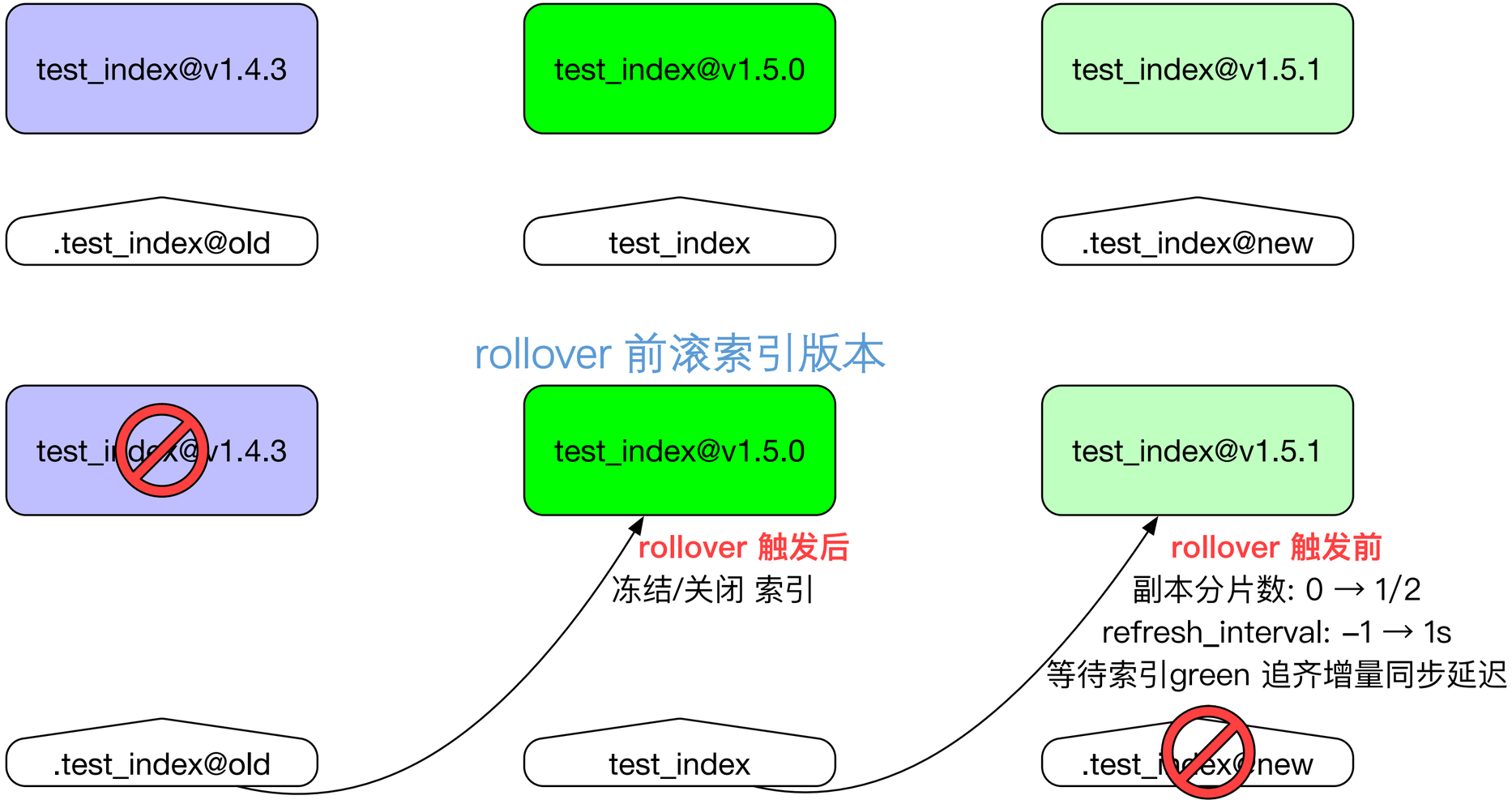
数据库加速、搜索、推荐类应用场景



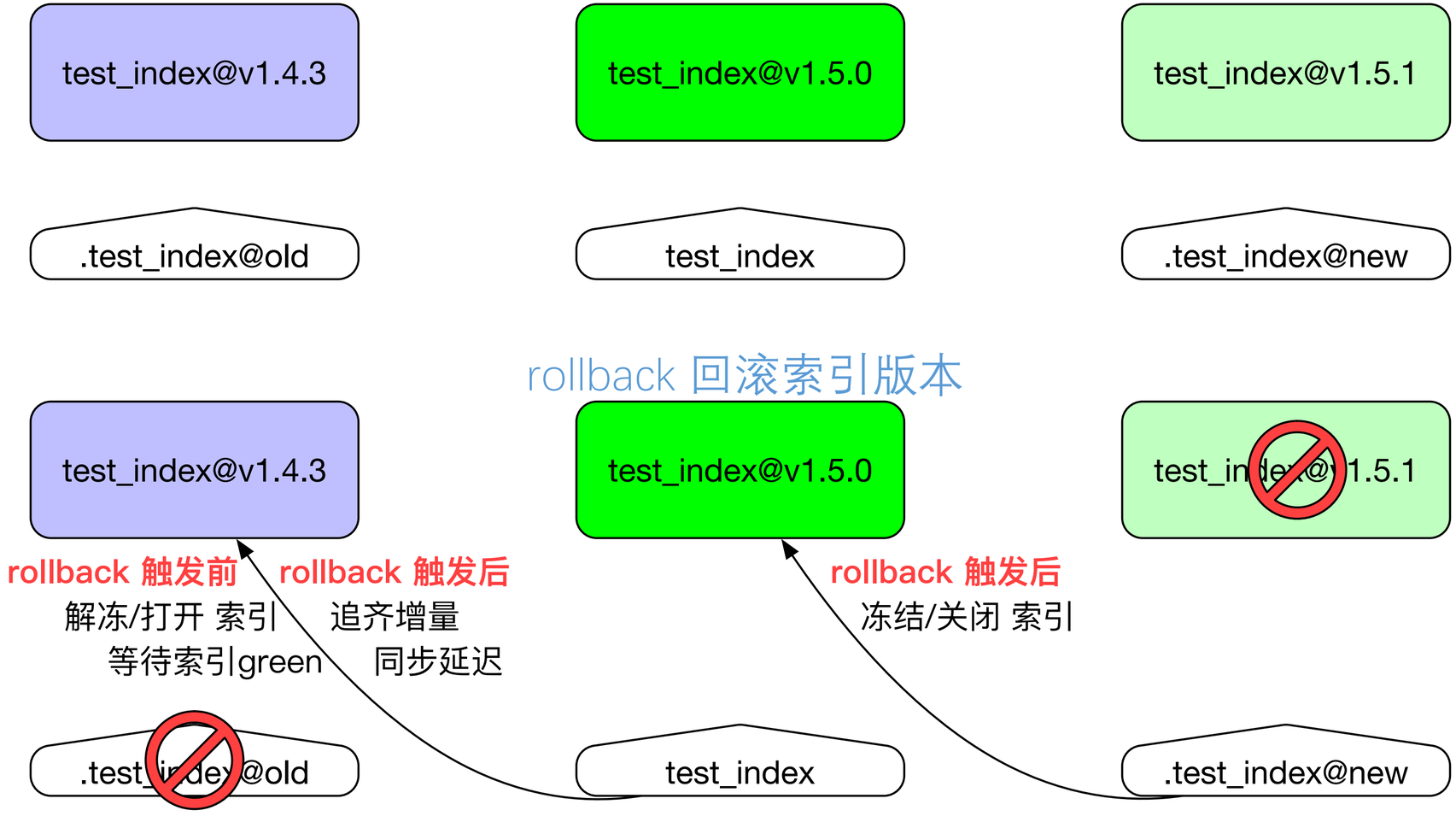




数据库加速、搜索、推荐类应用场景



数据库加速、搜索、推荐类应用场景



搜索、增量同步动作，将随别名切换到老索引。且切换别名为原子性操作

目录概要

ES支持的doc的增改方式

- index
  - 给定全新 _id：创建新文档
  - 给定已有 _id：覆盖更新已有文档 (先删除、再创建)
  - 不给 _id：自动生成 _id
- create
  - 给定全新 _id：创建新文档
  - 给定已有 _id：拒绝该次写入
  - 不给 _id：自动生成 _id
- delete
  - 删除文档不会立即将文档从磁盘中删除，只是将文档标记为已删除状态

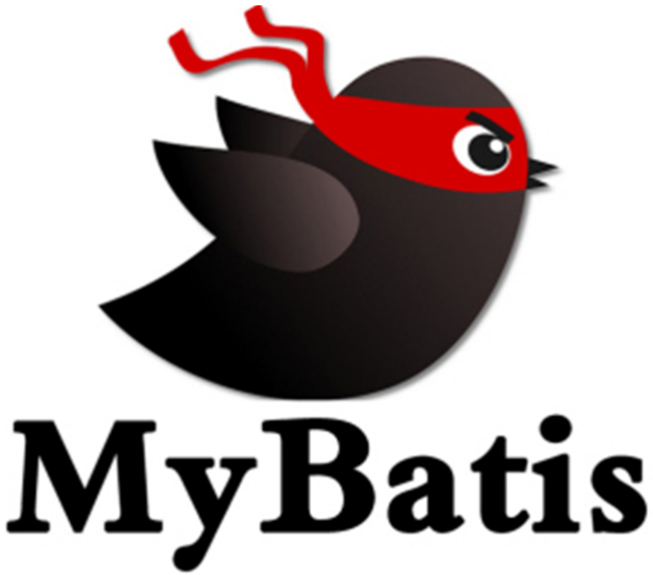
ES支持的doc的增改方式

- update
  - 给定已有 _id ：
    1. 以旧文档的JSON，合并新JSON（更新或添加传入字段），得到新版的完整JSON
    2. 删除旧doc
    3. 以新版JSON创建新文档
  - 给定全新 _id：拒绝该次写入
- upsert
  - 给定全新 _id：以JSON内容创建
- script
  - 使用Painless脚本，使用旧文档和新传参，动态更新JSON内容

目录概要

大索引的高效开发管理方式

DB成熟的ORM开发管理方式，十分高效



开源社区也涌现了很多针对ES的ORM工具包
但流传度不广 用户群体较少

https://github.com/gitchennan/elasticsearch-mapper

https://github.com/muzin/ebatis

https://github.com/spring-projects/spring-data-elasticsearch

https://github.com/bbossgroups/bboss-elasticsearch

……

大索引的高效开发管理方式

JAVA DEMO 示例

1. 定义Field注解，对应生成es该字段的mapping，未配置的会根据类型自动生成字段mapping

2. 定义Trans注解，描述该字段是由哪个类的哪些字段演化而来，并标注了其中的逻辑，便于大量字段情况下的维护

3. 结合lombok、jackson等支持注解的成熟工具包，将ES索引相关的代码精简到极致，统一一处维护，避免各种错误

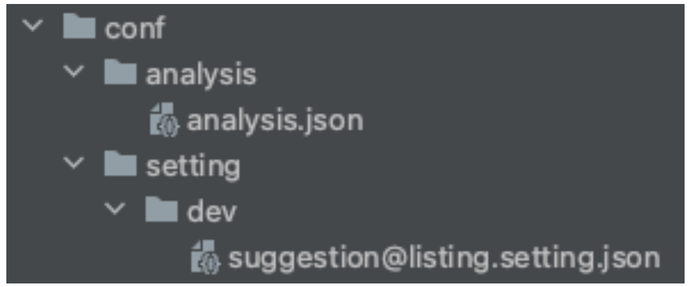
4. 可以直接使用该ORM模型类进行json序列化/反序列化，并实现了一个Doc类描述多个配置的功能

```java
@EqualsAndHashCode(callSuper = true)
@FieldNameConstants
@Data
@JsonInclude(JsonInclude.Include.NON_EMPTY)
public class SuggestionDoc extends Doc {

    @FieldTrans(source = SuggestionInfo.class, trans = ToListTrans.class)
    private List<String> [redacted];

    @FieldTrans(source = SuggestionInfo.class, trans = ToListTrans.class)
    private List<String> [redacted];

    private String country;

    @FieldTrans(trans = IgnoreTrans.class)
    private String type;

    @FieldTrans(trans = IgnoreTrans.class)
    @SearchAsYouTypeField(analyzer = "suggestion_search_analyzer", copy_to = "valueKeyword")
    private String value;

    @FieldTrans(trans = IgnoreTrans.class)
    private String valueKeyword;

    public String id() { return StringUtils0.base64Encode(String.join( delimiter: "|", type, value)); }
}
```

大索引的高效开发管理方式

JAVA DEMO 示例

```java
@Test
public void testSuggestionDoc() throws IOException {
    XContentBuilder xContentBuilder = new CreateIndexRequestBuilder()
            .buildIndexMetaSetting(ListingSuggestionDoc.class,  env: "dev");
    System.out.println(Strings.toString(xContentBuilder));
}
```



统一管理维护分词器配置

+

索引特殊setting配置声明

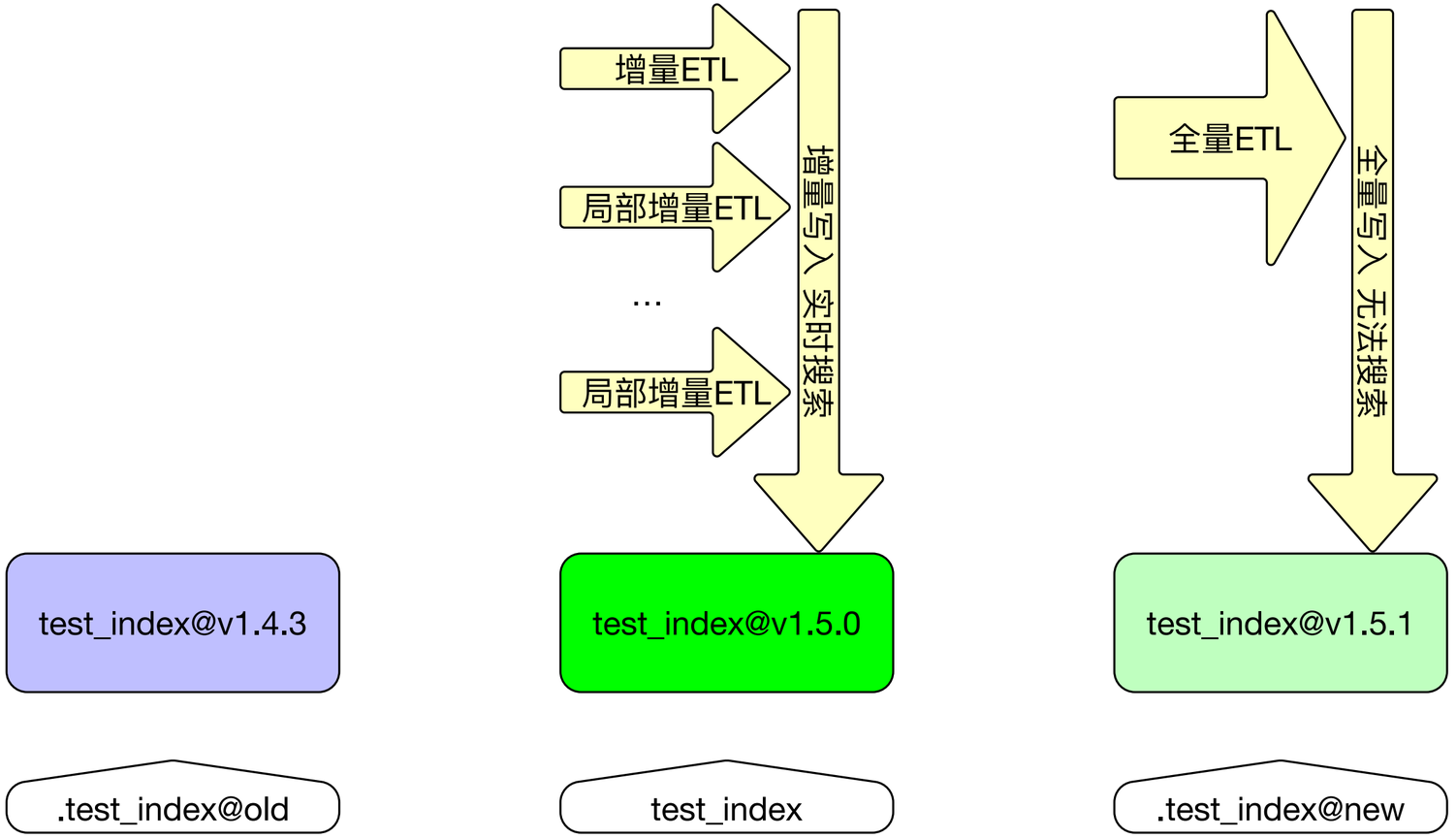
输出用于创建索引的setting+mapping配置

```
{
  "mappings" : {
    "dynamic" : false,
    "properties" : {
      "[illegible]" : {
        "type" : "keyword"
      },
      "[illegible]" : {
        "type" : "keyword"
      },
      "country" : {
        "type" : "keyword"
      },
      "type" : {
        "type" : "keyword"
      },
      "value" : {
        "type" : "search_as_you_type",
        "max_shingle_size" : 3,
        "analyzer" : "suggestion_search_analyzer",
        "copy_to" : "valueKeyword"
      },
      "valueKeyword" : {
        "type" : "keyword"
      }
    }
  },
  "settings" : {
    "index" : {
      "refresh_interval" : "-1",
      "analysis" : {…
      },
      "number_of_shards" : "2",
      "number_of_replicas" : "0"
    }
  },
  "aliases" : {
    ".suggestion@listing@new" : {
      "is_write_index" : null
    }
  }
}
```

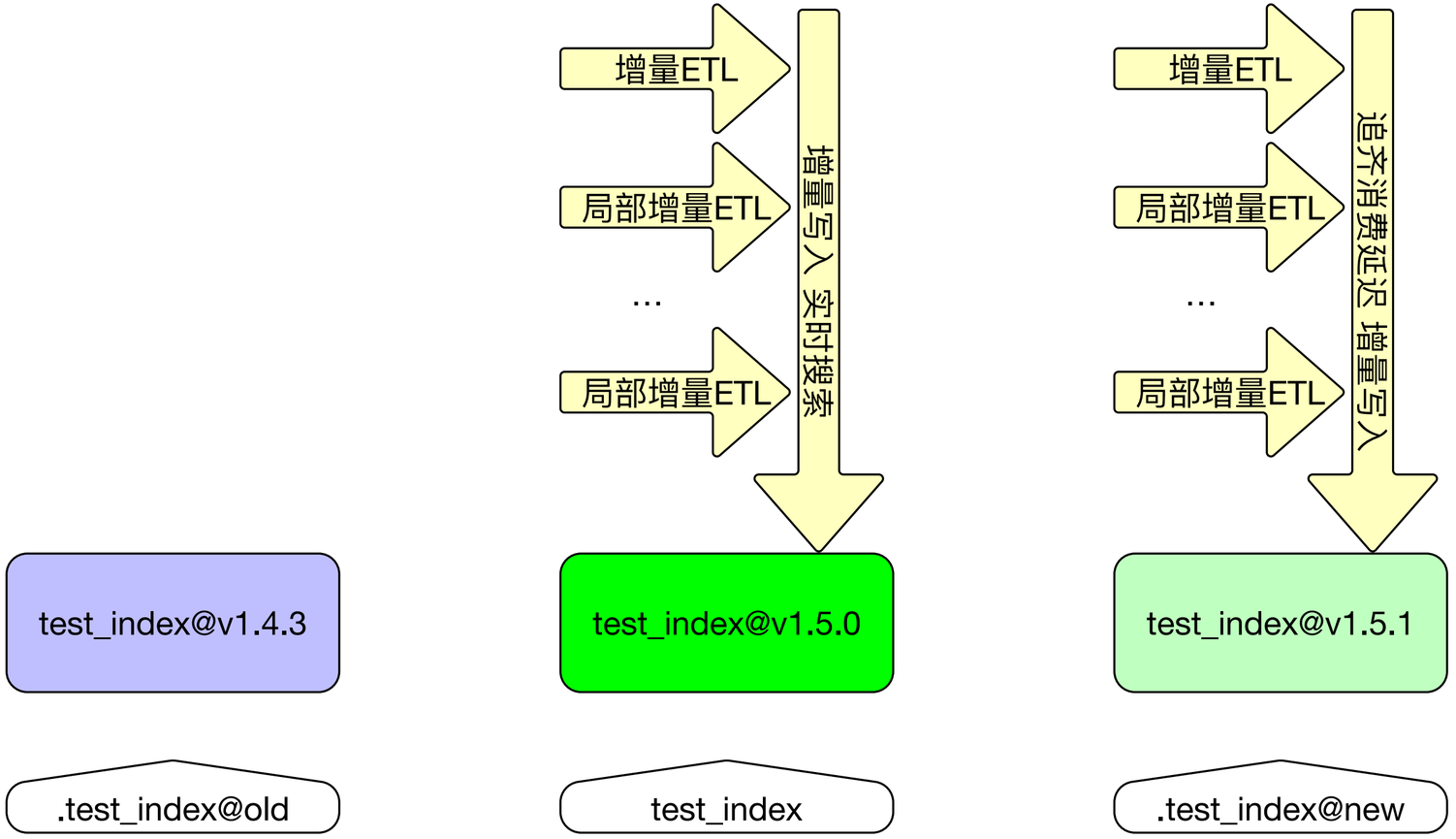
JAVA DEMO 示例

```
139     @PropertyView(LeadPart.Fields.LAST_TOUCH)
140     @FieldTrans(args = {"lastTouchDate.openedEmailTouch"})
141     private Long openedEmailTouch;
142
143     @PropertyView(LeadPart.Fields.LAST_TOUCH)
144     @FieldTrans(args = {"lastTouchDate.sellTouch"})
145     private Long sellTouch;
146
147     @PropertyView(LeadPart.Fields.LAST_TOUCH)
148     @FieldTrans(args = {"lastTouchDate.activityOtherTouch"})
149     private Long activityOtherTouch;
150
151     private Boolean emailOpened;
152
153     @StringField(type = StringType.Text, analyzer = "suggest_index_analyzer", search_analyzer = "suggest_search_analyzer")
154     @StringField(subField = "raw")
155     private String emailSuggest;
156
157     @StringField(type = StringType.Text, analyzer = "suggest_index_analyzer", search_analyzer = "suggest_search_analyzer")
158     @StringField(subField = "raw")
159     private String emailsSuggest;
160
161     private LocalDateTime esUpdateTime;
162
163     @StringField(type = StringType.Text, analyzer = "general_analyzer")
164     private String firstName;
```

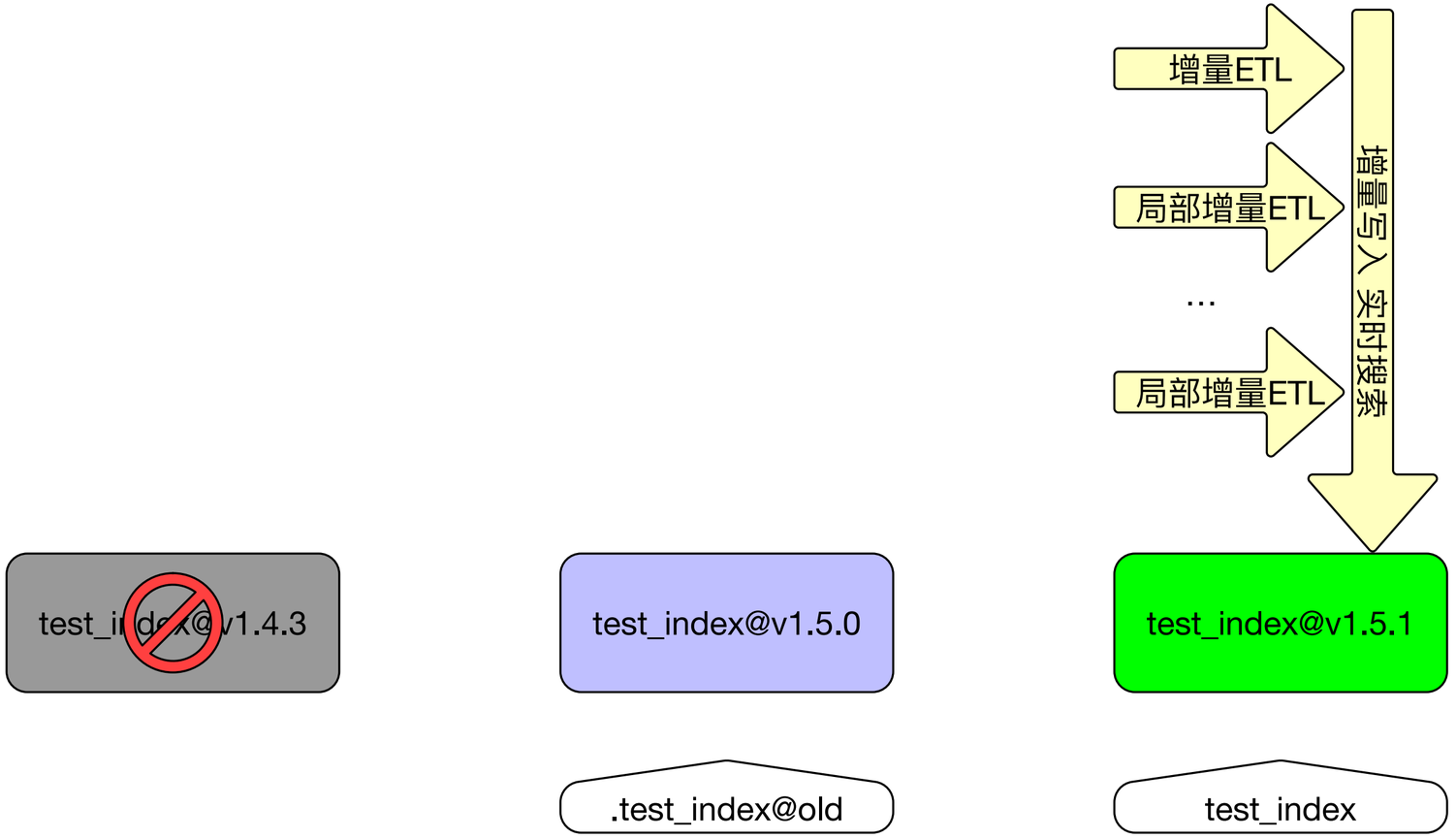
大索引的高效开发管理方式



大索引的高效开发管理方式



大索引的高效开发管理方式



谢 谢