



Elasticsearch

정희진

2024.08.27

Index

01 소개

- 1) Elasticsearch란
- 2) 용어 정리

02 검색 방법

- 1) Elasticsearch 검색 방법
- 2) Elasticsearch Index 구조

03 실습

- 1) CRUD
- 2) Bulk
- 3) Search
- 4) Aggregation

04 옵션

- 1) shards, replicas
- 2) mapping
- 3) routing

—

01

소개

1) Elasticsearch란

2) 용어 정리

Elasticsearch란

Apache Lucene 기반의 Java 오픈소스 분산 검색 엔진

기본적으로 데이터를 인덱싱하여 저장하고, 검색, 집계 등의 기능을 수행

기존 RDBMS에서 다루기 어려운 full text search 기능이 제공



용어 정리

Elasticsearch

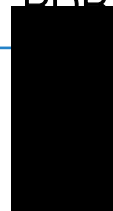


Document

Field

Mapping

DB



Row

Column

Schema

—
02

검색 방법

- 1) Elasticsearch 검색 방법
- 2) Elasticsearch Index 구조

Elasticsearch 검색 방법

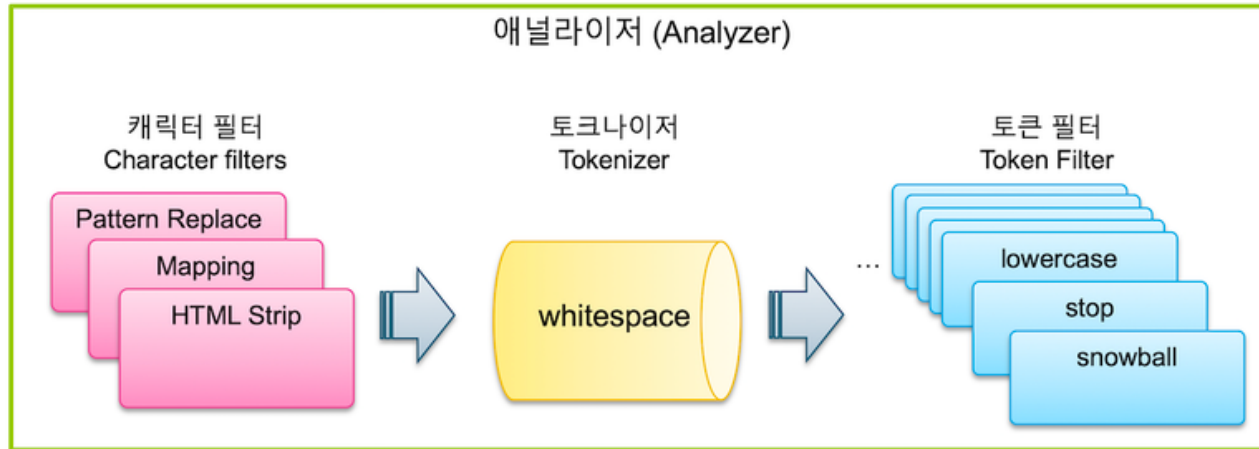
Inverted Index

ID	Text
doc1	The quick brown fox
doc2	The quick brown fox jumps over the lazy dog
doc3	The quick brown fox jumps over the quick dog
doc4	Brown fox brown dog
doc5	Lazy jumping dog

텀(Term)	ID	텀(Term)	ID
The	doc1, doc2, doc3	quick	doc1, doc2, doc3
brown	doc1, doc2, doc3, doc4	fox	doc1, doc2, doc3, doc4
jumps	doc2, doc3	over	doc2, doc3
the	doc2, doc3	lazy	doc2
dog	doc2, doc3, doc4, doc5	Brown	doc4
Lazy	doc5	jumping	doc5

Elasticsearch 검색 방법

Analyzer



Chracter filters

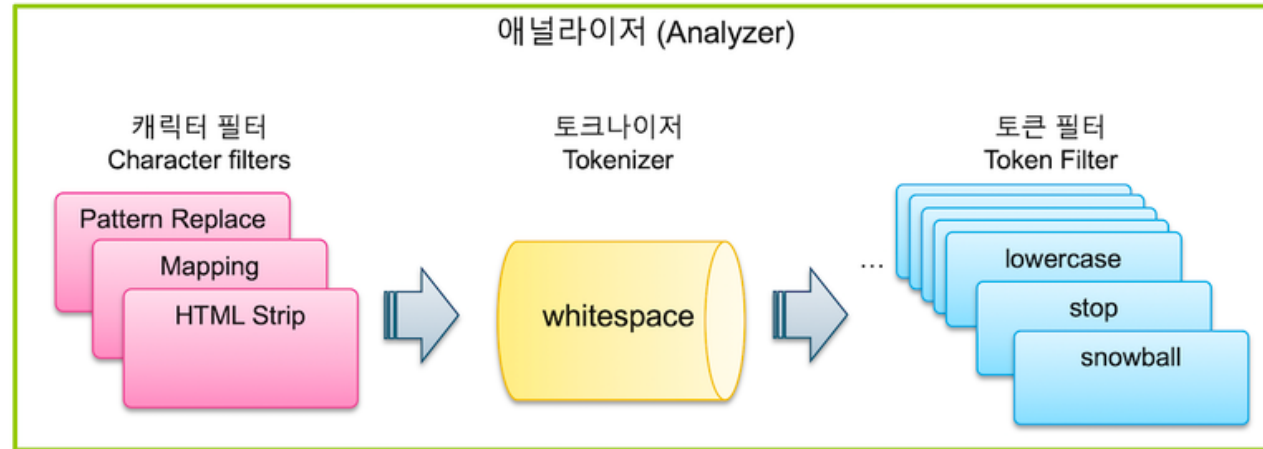
HTML Strip	<>태그 제거
Mapping	C++ → C_plus__plus_
Pattern Replace	정규식을 사용하여 치환

Tokenizer

Standard	공백으로 term 구분, 일부 특수문자 제거 ("jumped!" 처럼 문자 끝에 있는 ! 제거)
Letter	알파벳을 제외한 모든 공백, 숫자, 기호를 기준으로 term 분리
Whitespace	공백을 기준으로 term 분리

Elasticsearch 검색 방법

Analyzer



Token Filter

- lowercase 대소문자에 상관없이 검색이 가능, term을 모두 소문자로 변경
- Stop 불필요한 단어 제거, 불필요한 단어(stopwords) 설정 가능
- synonym 동의어 검색 제공 (“amazon”이라고 검색했을 때, “aws”도 함께 검색됨)
- snowball 형태소 분석 (~s, ~ing 제거)

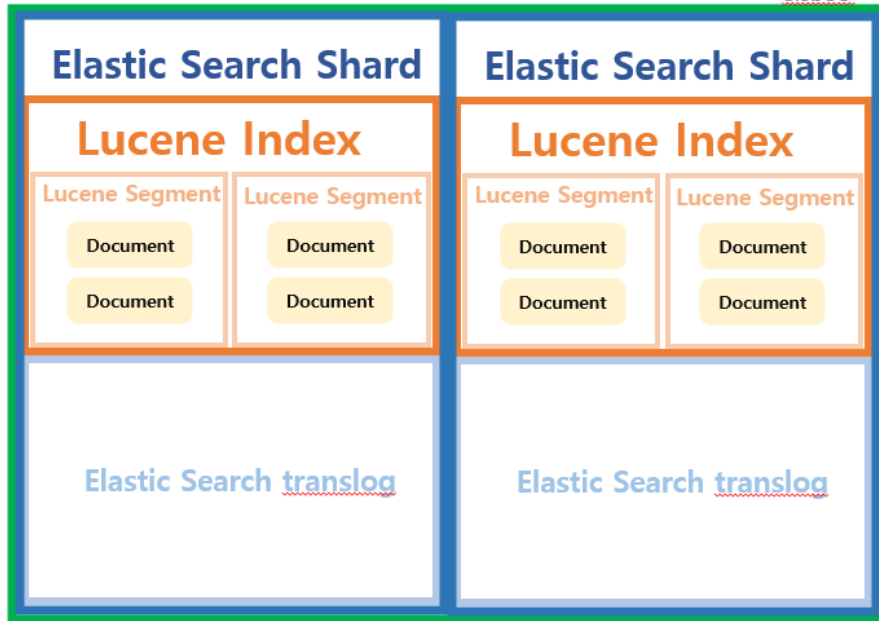
텀(Term)	ID	텀(Term)	ID
The	doc1, doc2, doc3	quick	doc1, doc2, doc3
brown	doc1, doc2, doc3, doc4	fox	doc1, doc2, doc3, doc4
jumps	doc2, doc3	over	doc2, doc3
the	doc2, doc3	lazy	doc2
dog	doc2, doc3, doc4, doc5	Brown	doc4
Lazy	doc5	jumping	doc5

텀(Term)	ID	텀(Term)	ID
the	doc1, doc2, doc3	quick	doc1, doc2, doc3
brown	doc1, doc2, doc3, doc4	fox	doc1, doc2, doc3, doc4
jumps	doc2, doc3	over	doc2, doc3
the	doc2, doc3	lazy	doc2
dog	doc2, doc3, doc4, doc5	brown	doc4
lazy	doc5	jumping	doc5

텀(Term)	ID	텀(Term)	ID
the	doc1, doc2, doc3	quick	doc1, doc2, doc3
brown	doc1, doc2, doc3, doc4	fox	doc1, doc2, doc3, doc4
jumps	doc2, doc3	over	doc2, doc3
dog	doc2, doc3, doc4, doc5	lazy	doc2, doc5
jumping	doc5		

Elasticsearch Index 구조

Elastic Search Index elsboo



Elastic search index \supset shard \supset lucene index \supset segment \supset document

./index

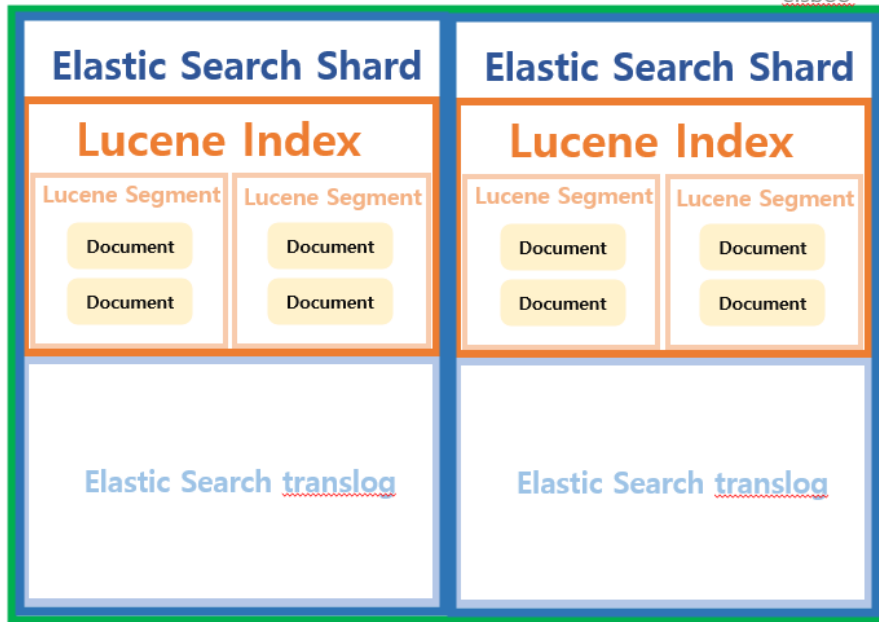
루씬이 관리하며, 세그먼트가 저장되는 공간

./translog

장애 발생 시 데이터 유실을 방지하기 위해 엘라스틱 서치가 도입한 방식
문서 색인 직후 세그먼트가 바로 생성되지 않고 translog와 메모리 버퍼에 먼저 기록
루씬이 디스크에 물리적으로 기록하기 전에 장애가 발생했을 때 translog 기록을 보며 데이터 복구
flush API 호출 시 translog 내용 삭제 (루씬 commit이 수행되고 데이터가 물리적으로 디스크에 쓰여지
기 때문)

Elasticsearch Index 구조

Elastic Search Index elsboo

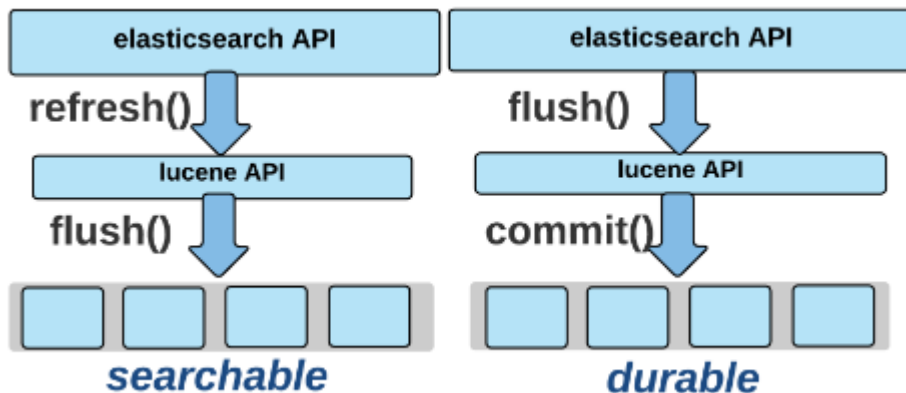


lucene flush – elasticsearch refresh

페이지 캐시에 데이터를 씀
디스크에 바로 쓰는 것보다 비교적 비용이 적음
데이터 유실 위험
translog에 데이터가 남아있음
세그먼트가 생성되었으므로 검색은 가능

lucene commit – elasticsearch flush

물리 디스크에 데이터를 씀
안전한 상태
translog 데이터는 지워짐



—

03

실습

- 1) CRUD
- 2) Bulk
- 3) Search
- 4) Aggregation

Elasticsearch 실습

The screenshot shows the Kibana console interface. The browser address bar indicates the URL is `localhost:5601/app/dev_tools#/console`. The Kibana header includes the 'elastic' logo, a search bar with the placeholder 'Find apps, content, and more.', and navigation icons. Below the header, the 'Dev Tools' tab is active, with sub-tabs for 'Console', 'Search Profiler', 'Grok Debugger', and 'Painless Lab (BETA)'. A notification banner at the top of the console area states 'Help us improve the Elastic Stack' and 'Usage collection is enabled', with a 'Dismiss' button. The console interface includes tabs for 'History', 'Settings', 'Variables', and 'Help'. The 'History' tab shows a single request: `1 GET my_index/_doc/1`. The response is displayed in a JSON format:

```
{
  "_index": "my_index",
  "_id": "1",
  "_version": 1,
  "_seq_no": 0,
  "_primary_term": 1,
  "found": true,
  "_source": {
    "message": "hello"
  }
}
```

 The status bar at the bottom right of the console indicates a `200 - OK` status and a response time of `17 ms`.

CRUD

문서 생성/전체 수정

```
PUT my_index/_doc/1
{
  "message": "hello"
}
```

문서 생성 (랜덤 id 지정)

```
POST my_index/_doc
{
  "message": "hello"
}
```

문서 생성

```
POST my_index/_create/1
{
  "message": "hello"
}
```

문서 생성

```
PUT my_index/_create/1
{
  "message": "hello"
}
```

CRUD

문서 조회 (메타데이터 포함)

GET my_index/_doc/1

문서 조회 (메타데이터 미포함)

GET my_index/_source/1

문서 부분 수정

```
POST my_index/_update/1
{
  "doc": {
    "message": "hello!!"
  }
}
```

문서 삭제

DELETE my_index/_doc/1

Bulk

POST _bulk

```
{ "index": { "_index": "my_index", "_id": "1" } }
{ "field": "value one" }
{ "index": { "_index": "my_index", "_id": "2" } }
{ "field": "value two" }
{ "delete": { "_index": "my_index", "_id": "2" } }
{ "create": { "_index": "my_index", "_id": "3" } }
{ "field": "value three" }
{ "update": { "_index": "my_index", "_id": "1" } }
{ "doc": { "field": "value two" } }
```

Index, create, update, delete 동작 가능

```
{
  "errors": false,
  "took": 426,
  "items": [
    {
      "index": {
        "_index": "my_index",
        "_id": "1",
        "_version": 1,
        "result": "created",
        "_shards": {
          "total": 2,
          "successful": 1,
          "failed": 0
        },
        "_seq_no": 0,
        "_primary_term": 1,
        "status": 201
      }
    },
    {
      "index": {
        "_index": "my_index",
        "_id": "2",
        "_version": 1,
        "result": "created",
        "_shards": {
          "total": 2,
          "successful": 1,
          "failed": 0
        },
        "_seq_no": 1,
        "_primary_term": 1,
        "status": 201
      }
    },
    {
      "delete": {
        "_index": "my_index",
        "_id": "2",
        "_version": 2,
        "result": "deleted",
        "_shards": {
          "total": 2,
          "successful": 1,
          "failed": 0
        },
        "_seq_no": 2,
        "_primary_term": 1,
        "status": 200
      }
    },
    {
      "create": {
        "_index": "my_index",
        "_id": "3",
        "_version": 1,
        "result": "created",
        "_shards": {
          "total": 2,
          "successful": 1,
          "failed": 0
        },
        "_seq_no": 3,
        "_primary_term": 1,
        "status": 201
      }
    }
  ]
}
```

```
{
  "took": 13,
  "timed_out": false,
  "_shards": {
    "total": 1,
    "successful": 1,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": {
      "value": 2,
      "relation": "eq"
    },
    "max_score": 1.0,
    "hits": [
      {
        "_index": "my_index",
        "_id": "3",
        "_score": 1,
        "_source": {
          "field": "value three"
        }
      },
      {
        "_index": "my_index",
        "_id": "1",
        "_score": 1,
        "_source": {
          "field": "value two"
        }
      }
    ]
  }
}
```


Search

URI 검색

```
GET my_index/_search?q=field:value
```

데이터 본문 검색

```
GET my_index/_search
{
  "query": {
    "match": {
      "field": "value"
    }
  }
}
```

Search

```
POST my_index/_bulk
{"index":{"_id":1}}
{"message":"The quick brown fox"}
{"index":{"_id":2}}
{"message":"The quick brown fox jumps over the lazy dog"}
{"index":{"_id":3}}
{"message":"The quick brown fox jumps over the quick dog"}
{"index":{"_id":4}}
{"message":"Brown fox brown dog"}
{"index":{"_id":5}}
{"message":"Lazy jumping dog"}
```

id	message
1	The quick brown fox
2	The quick brown fox jumps over the lazy dog
3	The quick brown fox jumps over the quick dog
4	Brown fox brown dog
5	Lazy jumping dog

Search

OR 조건 검색

```
GET my_index/_search
{
  "query": {
    "match": {
      "message": "quick dog"
    }
  }
}
```

id	message
1	The quick brown fox
2	The quick brown fox jumps over the lazy dog
3	The quick brown fox jumps over the quick dog
4	Brown fox brown dog
5	Lazy jumping dog

Search

AND 조건 검색

```
GET my_index/_search
{
  "query": {
    "match": {
      "message": {
        "query": "quick dog",
        "operator": "and"
      }
    }
  }
}
```

id	message
2	The quick brown fox jumps over the lazy dog
3	The quick brown fox jumps over the quick dog

Search

match_phrase: 구문 일치 검색

```
GET my_index/_search
{
  "query": {
    "match_phrase": {
      "message": "quick dog"
    }
  }
}
```

id	message
3	The quick brown fox jumps over the quick dog

Search

must: 쿼리가 참인 문서 검색 (거짓인 문서는 must not 사용)

```
GET my_index/_search
{
  "query": {
    "bool": {
      "must": [
        {
          "match": {
            "message": "quick"
          }
        },
        {
          "match_phrase": {
            "message": "lazy dog"
          }
        }
      ]
    }
  }
}
```

id	message
2	The quick brown fox jumps over the lazy dog

Search

```
GET my_index/_search
{
  "query": {
    "match": {
      "message": "quick dog"
    }
  }
}
```

Score

- 검색 결과가 조건과 얼마나 일치하는지를 나타냄
- Score가 높을수록 검색어와 가장 연관성 있음
- TF(Term Frequency): 검색 단어를 많이 포함할수록 score↑
- IDF(Inverse Document Frequency): 중요한 단어를 포함할수록 score↑
- Field Length: 필드의 길이가 짧을수록 score↑

id	score	message
3	0.87627405	The quick brown fox jumps over the quick dog
2	0.6744513	The quick brown fox jumps over the lazy dog
1	0.6173784	The quick brown fox
5	0.35847887	Lazy jumping dog
4	0.32951736	Brown fox brown dog

Search

shoud: score를 높임

GET my_index/_search

```
{
  "query": {
    "bool": {
      "must": [
        {
          "match": {
            "message": "fox"
          }
        }
      ],
      "should": [
        {
          "match": {
            "message": "lazy"
          }
        }
      ]
    }
  }
}
```

id	score	message
2	0.94896436	The quick brown fox jumps over the lazy dog
1	0.32951736	The quick brown fox
4	0.32951736	Brown fox brown dog
3	0.23470736	The quick brown fox jumps over the quick dog

Search

filter: score에 영향을 주지 않고 필터링

GET my_index/_search

```
{
  "query": {
    "bool": {
      "must": [
        {
          "match": {
            "message": "fox"
          }
        }
      ],
      "filter": [
        {
          "match": {
            "message": "quick"
          }
        }
      ]
    }
  }
}
```

id	score	message
1	0.32951736	The quick brown fox
2	0.23470736	The quick brown fox jumps over the lazy dog
3	0.23470736	The quick brown fox jumps over the quick dog

Search

range : 범위 검색 (날짜도 가능, score에 영향 X)

```
GET phones/_search
{
  "query": {
    "range": {
      "price": {
        "gte": 700,
        "lte": 900
      }
    }
  }
}
```

gte	이상
gt	초과
lte	이하
lt	미만

id	score	price
2	1	795
3	1	859

Aggregation

GET my_stations/_search

```
{
  "size": 0,
  "aggs": {
    "all_passangers": {
      "sum": {
        "field": "passangers"
      }
    }
  }
}
```

```
{
  "took": 2,
  "timed_out": false,
  "_shards": {
    "total": 1,
    "successful": 1,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": {
      "value": 4,
      "relation": "eq"
    },
    "max_score": null,
    "hits": []
  },
  "aggregations": {
    "all_passangers": {
      "value": 16071
    }
  }
}
```

id	date	line	station	passangers
1	2024-04-22	2호선	홍대입구	2314
2	2024-04-22	3호선	양재	4121
3	2024-04-22	2호선	강남	5724
4	2024-04-22	2호선	신촌	3912

all_passangers

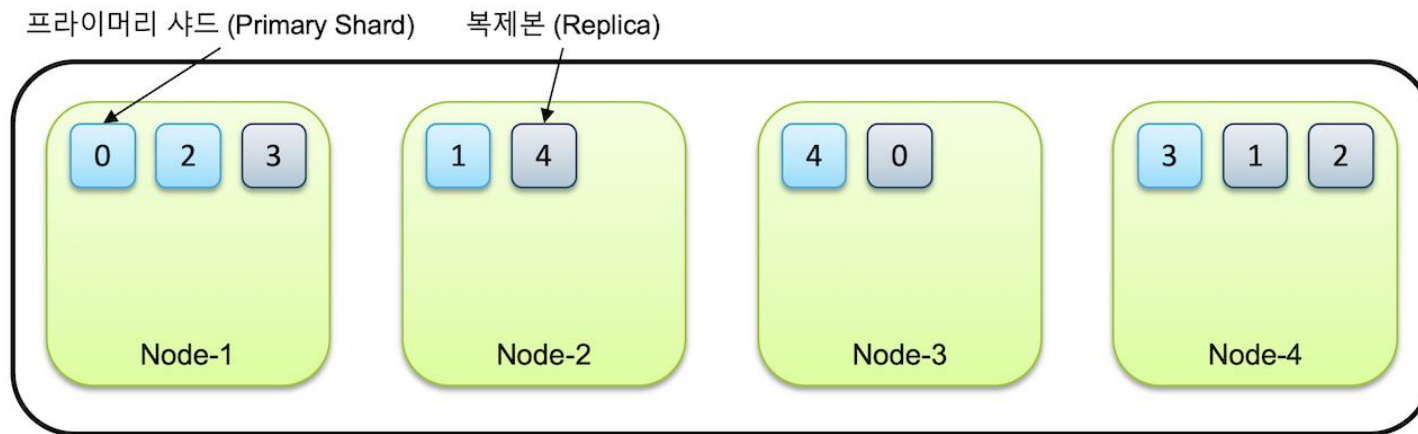
16071

—
04

옵 셴

- 1) shards, replicas
- 2) mapping
- 3) routing

shards, replicas



```
PUT my_index
{
  "settings": {
    "index": {
      "number_of_shards": 4,
      "number_of_replicas": 1
    }
  }
}
```

mapping

```
PUT my_index
{
  "mappings": {
    "properties": {
      "category": {
        "type": "text"
      },
      "date": {
        "type": "date",
        "format": "yyyy/MM/dd"
      }
    }
  }
}
```

keyword	text
date	Integer
boolean	float
object	nested
...	...

routing

GET routing/_search

```
{
  "took": 31,
  "timed_out": false,
  "_shards": {
    "total": 5,
    "successful": 5,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": {
      "value": 1,
      "relation": "eq"
    },
    "max_score": 1,
    "hits": [
      {
        "_index": "routing",
        "_id": "1",
        "_score": 1,
        "_routing": "\"myid\"",
        "_source": {
          "login_id": "myid",
          "comment": "hello"
        }
      }
    ]
  }
}
```

GET routing/_search?routing=myid

```
{
  "took": 2,
  "timed_out": false,
  "_shards": {
    "total": 1,
    "successful": 1,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": {
      "value": 1,
      "relation": "eq"
    },
    "max_score": 1,
    "hits": [
      {
        "_index": "routing",
        "_id": "1",
        "_score": 1,
        "_routing": "\"myid\"",
        "_source": {
          "login_id": "myid",
          "comment": "hello"
        }
      }
    ]
  }
}
```

routing

동일 routing끼리 같은 샤드에 위치시켜 검색 성능을 끌어 올림

라우팅 값을 항상 올바르게 지정해야 함

라우팅 지정이 올바르게 명시되지 않으면 색인된 문서가 존재함에도 불구하고 문서가 없다는 응답을 받음

라우팅 지정에 대한 정책을 세워 지켜야 함

```
PUT routing {
  "settings": {
    "number_of_shards": 5,
    "number_of_replicas": 1
  },
  "mappings": {
    "_routing": {
      "required": true
    }
  }
}
```




감사합니다.