03. Search API using NestClient

NestClient , .

- : https://www.elastic.co/guide/en/elasticsearch/client/net-api/7.x/introduction.html
- : https://github.com/psmon/searchapi/blob/master/SearchApi/Services/SearchIndex.cs

```
        public async Task<int> ReindexAll()
        {
            await _elasticClient.DeleteByQueryAsync<SearchGoods>(q => q.MatchAll());
            var allGoods = await FindAll();
            foreach (var item in allGoods)
            {
                //
                item.terms = $"{item.nameKr} {item.category1} {item.category2} {item.category3} {item.terms}";
                await _elasticClient.IndexDocumentAsync(item);
            }
            return allGoods.Count;
        }
```

2 .

- 1 (Upset)
- 1

1

```
SearchGoods item= new SearchGoods
{
    something1="",
    something2="",
}

//  ,   ID  Search.
// ID  ,     .
var searchResponse = await _elasticClient.SearchAsync<SearchGoods>(sd => sd
    .Index(_indexName)
    .Size(1)
    .Query(q => q
        .Match(m => m.Field("id").Query(item.no.ToString())
        )));
var hitIds = searchResponse.HitsMetadata.Hits.Select(h => h.Id);

await _elasticClient.UpdateAsync<SearchGoods>(DocumentPath<SearchGoods>.Id(hitIds.First()), descriptor =>
descriptor
        .Index(_indexName)
        //.DocAsUpsert()    -- ,     .
        .Doc(item)
    );
```

1 . .

.

, View 0 .

.

```
        public async Task<int> UpdateItem(SearchGoods item)
        {
            await _elasticClient.UpdateAsync<SearchGoods>(item.no, u => u.Doc(item));

            await _elasticClient.UpdateByQueryAsync<SearchGoods>(u => u
            .Query(q => q
            .Match(m => m
                .Field(p => p.goodsNo)
                .Query(item.goodsNo)
            ))
            .Script($"ctx._source.price = {item.price}")
            .Conflicts(Conflicts.Proceed)
            .Refresh(true));

            await _searchRepository.SaveChangesAsync();
            return 0;
        }
```

, .

```
public async Task<SearchResult> FindByFilter(SearchFilter filterOpt)
        {
            var result = new SearchResult();
            int page = filterOpt.paging == null ? 0 : filterOpt.paging.page;
            int limit = Math.Min(100,filterOpt.paging == null ? 10 : filterOpt.paging.limit);

            //Paging
            var searchDes = new SearchDescriptor<SearchGoods>()
                .From(page)
                .Size(limit);

            //Sort
            var sortDescriptor = new SortDescriptor<SearchGoods> ();

                        ..... Sort

            searchDes = searchDes.Sort( s=>sortDescriptor);

            //Filter
            var filters = new List<Func<QueryContainerDescriptor<SearchGoods>, QueryContainer>>();
                        ..... Filter

            searchDes = searchDes.Query(q=>q
                .Bool(bq => bq.Filter(filters)));

            var engine_result = await _elasticClient.SearchAsync<SearchGoods>(searchDes);
            result.list = engine_result.Documents.ToList<SearchGoods>();
            result.total = (int)engine_result.Total;
            result.size = result.list.Count;
        }
```

++ .

,

DSL .

```
           if(!String.IsNullOrWhiteSpace(filterOpt.sort?.price))
            {
                if (filterOpt.sort.price == "desc")
                {
                    sortDescriptor.Field(f => f.price, Nest.SortOrder.Descending);
                }
                else if(filterOpt.sort.price == "asc")
                {
                    sortDescriptor.Field(f => f.price, Nest.SortOrder.Ascending);
                }
            }
```

```
            if ( !String.IsNullOrWhiteSpace(filterOpt.filters?.category1) )
            {
                filters.Add(fq => fq.Match(t => t.Field(f => f.category1).Query(filterOpt.filters.
category1)));
            }
```

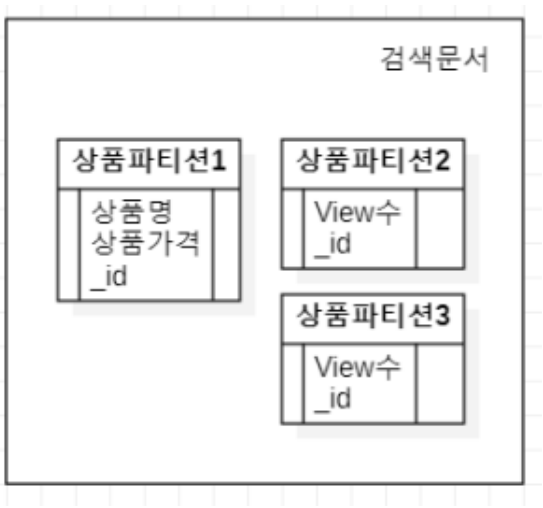
```
            //Price
            if (filterOpt.filters?.minPrice > 0 && filterOpt.filters?.maxPrice > 0)
            {
                filters.Add(fq => fq.Range(c => c
                    .Name("named_query")
                    .Boost(1.1)
                    .Field(p => p.price)
                    .GreaterThanOrEquals(filterOpt.filters.minPrice)
                    .LessThanOrEquals(filterOpt.filters.maxPrice)
                    .Relation(RangeRelation.Within)
                ));
            }
```

Text (terms)

```
            //Query
            if (!String.IsNullOrWhiteSpace(filterOpt.keyword))
            {
                var keywords = filterOpt.keyword.Split(' ');
                filters.Add(fq => fq.Terms(t => t.Field(f => f.terms).Terms(keywords)));
            }
```

, Fulltext Multi .

Split , .



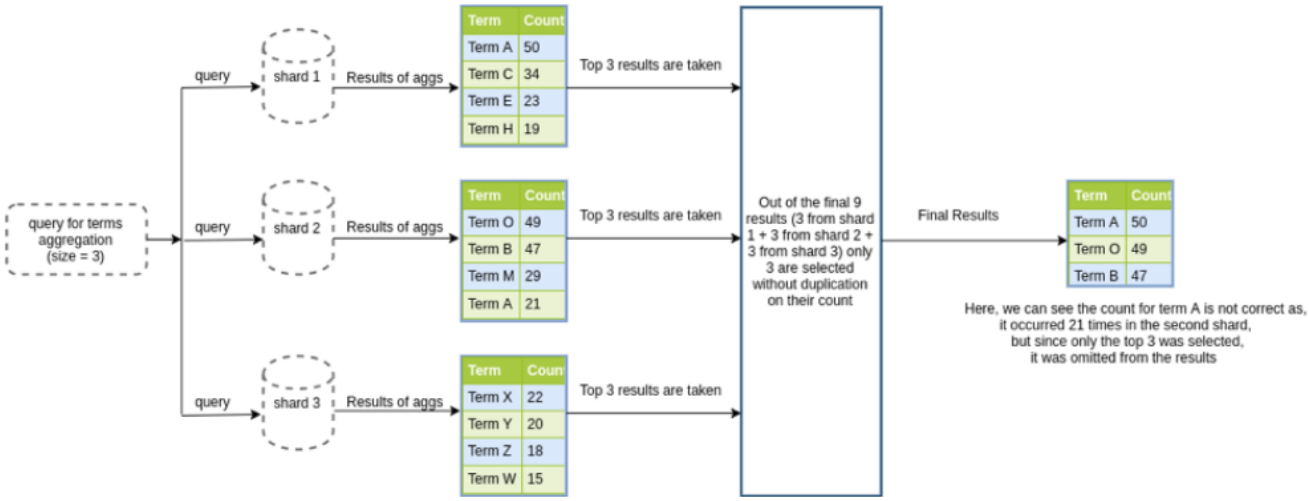
View

.

View 1 , Nosql .

( ) ( )

Insert View View .



.

- , Swap API
-

Link

- https://ojava.tistory.com/129 -
-
- https://qbox.io/blog/how-to-download-all-unique-terms-field-elasticsearch