ElasticSearch ilk kurulum ( version : 6.5.3 )

Gerekenler

1. Java jdk kurulu olmalı (Drive’a eklenen sürüm için <https://drive.google.com/file/d/1OSO2XD8G5PNJxE9_BtkvFB__W7wngJXN/view?usp=sharing>)

Not : Üst sürüme güncellediğinizde mevcut elastic sürümü çalışmayacaktır.

1. elasticsearch indirilecek

(<https://www.elastic.co/downloads/elasticsearch>)

1. Bilgisayar Özellikleri > Gelişmiş Sistem Ayarları > Ortam Değişkenleri > sistem ve user değişkenleri (her ikisine de)

Yeni değişken

Değişken adı : JAVA\_HOME

Değişken değeri : C:\Program Files\Java\jdk1.8.0\_221 (örnek)

Not : Ortam yolunu eklerken görünmeyen karakter olmamasına dikkat edin.

1. C:\elasticsearch\bin\elasticsearch.bat çalıştırılır (servis)
2. <http://localhost:9200/> de elasticsearch görüntülenir (Json çıktısı verir)

Kibana ilk kurulum ( version : 6.5.3 )

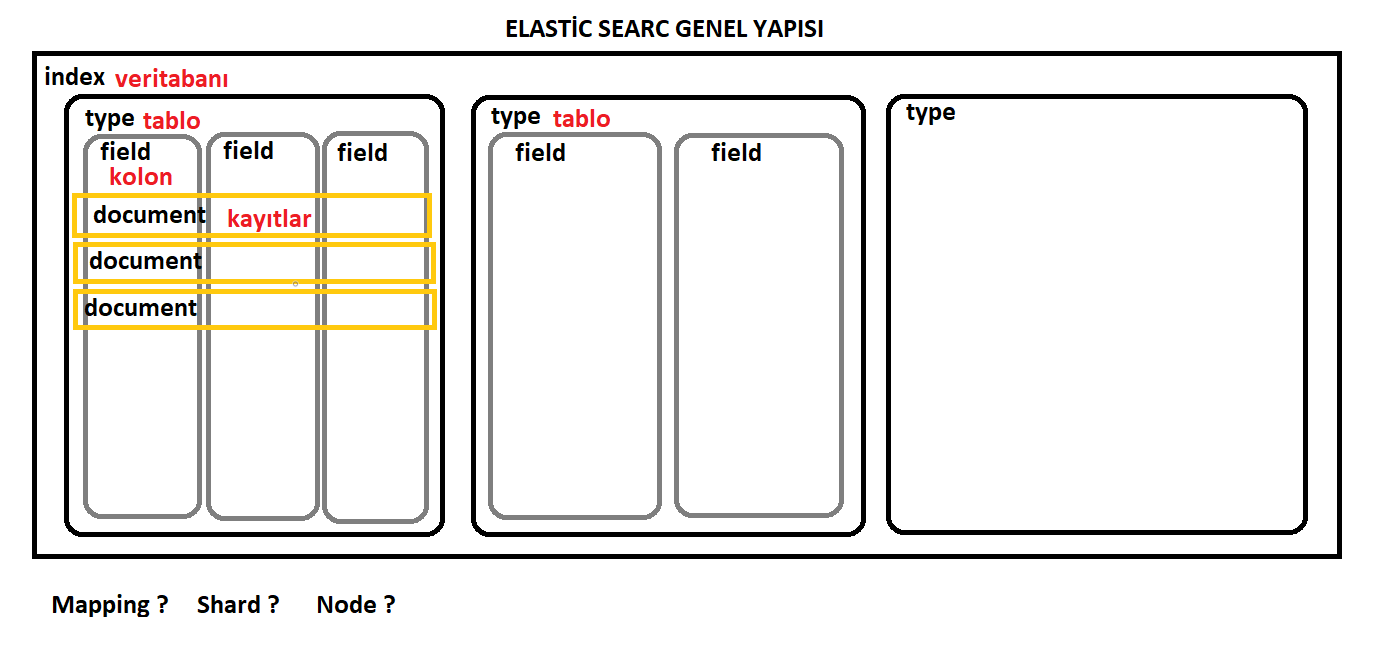
kibana indirilecek (elasticsearch ile sürümleri aynı olmak zorunda, yoksa çalışmıyor.

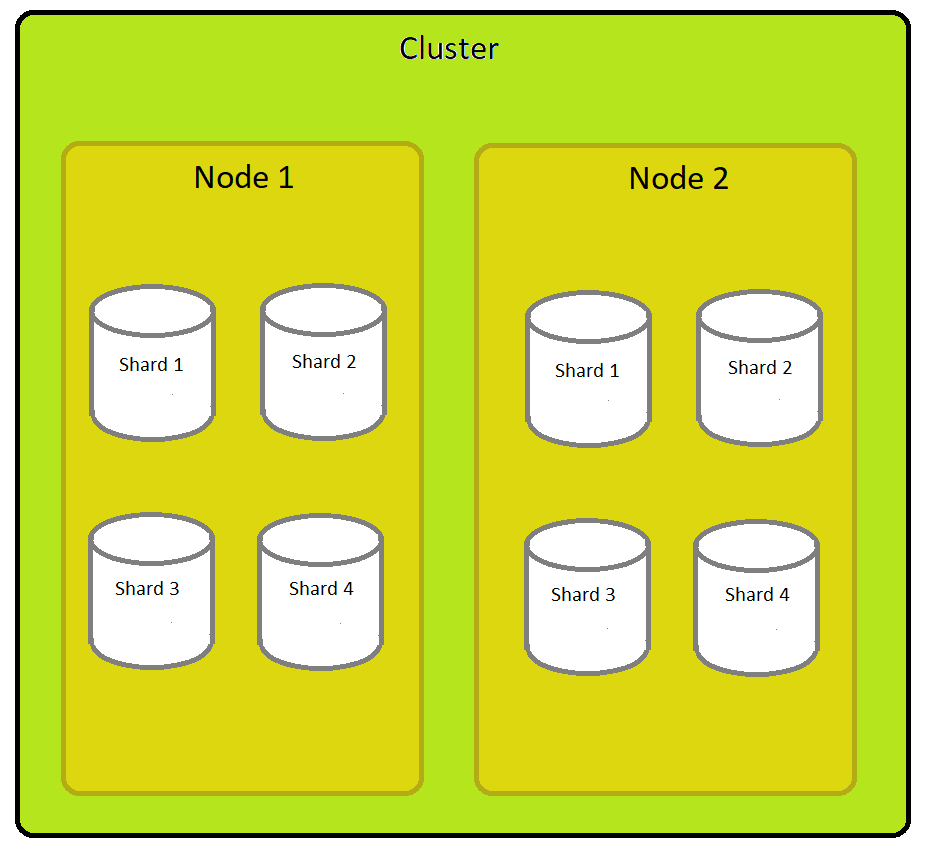
<https://www.elastic.co/downloads/kibana> )

elasticdeki gibi JAVA\_HOME tarzından bi ortam değişkeni tanımlamaya ihtiyaç yok.

elasticsearch servisi başlatıldıktan sonra C:\kibana\bin\kibana.bat **yönetici olarak** çalıştırılır (servis)

<http://localhost:5601/> de elasticsearch görüntülenir (panel görüntülenir)





|  |  |
| --- | --- |
| **Cluster Ayarları** | |
| PUT \_cluster/settings  {  "transient": {  "cluster.routing.allocation.disk.watermark.low": "100gb",  "cluster.routing.allocation.disk.watermark.high": "50gb",  "cluster.routing.allocation.disk.watermark.flood\_stage": "10gb",  "cluster.info.update.interval": "1m"  }  } | En düşük disk ve En yüksek disk limiti değiştirilebilir.  Saltokunur mod için olan kritik değer ayarlanabilir  Not : Default ayarlarda disk boyutu %5 nin altına geçtiğinde Elasticsearch saltokunur moda geçer.  Kaynak :  <https://stackoverflow.com/a/50609418/10475544> |

|  |  |
| --- | --- |
| **Elastic kibana Dev Tools Sorgu Tipleri** | |
| DELETE /bills\_en | # bills\_en Indexini siler |
| GET /bills\_en/\_search  {  "query": {  "match\_all": {}  }  } | # Indexteki tüm kayıtları getirir. |
| GET /bills\_en/\_count | # Indexteki tüm kayıtların toplam sayısını verir. |
| GET /bills\_en/\_search  {  "query": {  "match": {  "importPortCode": "DZALG"  }  }  } | # importPortCode 'DZALG' olan kayıtları getirir. |
| GET /bills\_en/\_search  {  "query": {  "bool": {  "must": [  {  "multi\_match": {  "query": "FOODS",  "fields": [  "buyerName",  "supplierName"  ]  }  }  ]  }  },  "from":3,  "size": 5  } | # 'multi\_match' sayesinde hem buyerName'de, hem de supplierName'de "FOODS" içeren kayıtları getirir.  # must, Sql de AND'e karşılık gelir. birden fazla sorguda tümleşik arama yapar. |
| GET /bills\_en/\_search  {  "query": {  "range": {  "weight": {  "gt": 1,  "lt": 10  }  }  },  "sort": [  {  "weight": {  "order": "asc"  }  }  ]  } | # 'Weight' alanında 1 den büyük 10dan küçük olan kayıtları, düz sıralayarak getirir. |
| GET /bills\_en/\_search  {  "query": {  "wildcard": {  "importPortCode": {  "value": "DZ\*"  }  }  }  } | # importPortCode 'DZ' ile başlayan tüm kayıtları getirir. |
| GET /bills\_en/\_search  {  "query": {  "range": {  "date": {  "gte": "2015-01-01",  "lte": "2015-01-02"  }  }  }  ,"sort": [  {  "date": {  "order": "desc"  }  }  ]  } | # tarih aralığındaki kayıtları, ters sıralayarak getirir. |
| GET /bills\_en/\_search  {  "query": {  "bool": {  "must": [  {  "match": {  "buyerName": "ANCHOR"  }  },  {  "match": {  "hsCodeDescription": "PARTS"  }  },  {  "range": {  "date": {  "gte": "2015-01-10",  "lte": "2015-02-10"  }  }  }  ]  }  }  } | # buyerName "ANCHOR" olan, hsCodeDescription "PARTS" olan ve 2015-02-10 ile 2015-01-10 tarih arası verileri getirir. |
| GET /bills\_en/\_search  {  "query": {  "bool": {  "must": [  {  "match": {  "hsCode": "870006"  }  },  {  "match": {  "importCountry": "indonesia"  }  },  {  "range": {  "date": {  "gte": "2015-01-01",  "lte": "2020-01-09"  }  }  }  ]  }  }  } | Konşimento üzerinde, üstteki sorgunun örneği  # http://localhost:61282/Bills/BillSearch?gtip=020850&country=&dtp1=01.01.2015&dtp2=09.01.2020 |
| DELETE /bills\_en/  {  "query": {  "bool": {  "must": [  {  "match": {  "importCountryName": "Singapore"  }  }  ]  }  }  } | # importCountryName “Singapore” olan verileri sil  Not: Bi hata var. **Bu sorgu tüm indexi sildi!!!!!** |
| POST /bills\_en/\_delete\_by\_query  {  "query": {  "bool": {  "must": [  {  "match": {  "importCountryName": "Singapore"  }  }  ]  }  }  } | # importCountryName “Singapore” olan verileri sil  Not : Doğru sorgu bu fakat API öneri desteği yok |
| ***Gruplama Örnekleri ( Aggregation )*** | |
| GET /bills\_en/\_search  {  "from": 0,  "size": 0,  "aggs": {  "countries": {  "terms": {  "field": "importCountryName",  "size": 100  }  }  }  } | # importCountryName alanına göre tüm dataların gruplanmış 100 kaydını getirir. |
| GET /bills\_en/\_search  {  "from": 0,  "size": 5,  "aggs": {  "PortCode": {  "terms": {  "field": "importPortCode",  "size": 1000  }  }  }  } | # Tüm datalardan ilk 5 kaydı ve importPortCode alanına göre tüm dataların gruplanmış 1000 kaydını getirir. |
| ***IS NULL ve IS NOT NULL Çalışmaları*** | |
| GET /bills\_en/\_search  {  "query": {  "bool": {  "must": [  {  "exists": {  "field": "importCountryName"  }  }  ]  }  }  } | # importCountryName kaydı varolanları getirir. |
| GET /bills\_en/\_search  {  "query": {  "bool": {  "must\_not": [  {  "exists": {  "field": "exportCountryName"  }  }  ]  }  }  } | # exportCountryName kaydı varolmayanları getirir. |
| GET /bills\_en/\_search  {  "query": {  "query\_string": {  "default\_field": "exportCountryName",  "query": "\_exists\_:exportCountryName"  }  }  } | # exportCountryName kaydı varolanları getirir. |
|  |  |

|  |  |
| --- | --- |
| **Gruplama ve Grup Eleman Sayısı** | |
|  | |
| GET /company\_statistic\_requests/\_search  {  "query":{  "match":{  "targetCompanyId":{  "query":"96075"  }  }  },  "size":0,  "aggs":{  "userIpAddressAgg":{  "cardinality":{  "field":"userIpAddress"  }  },  "userIpAddressAggTerms":{  "terms": {  "field": "userIpAddress",  "size": 100  }  }  }  } | {  "took" : 1,  "timed\_out" : false,  "\_shards" : {  "total" : 5,  "successful" : 5,  "skipped" : 0,  "failed" : 0  },  "hits" : {  "total" : 43,  "max\_score" : 0.0,  "hits" : [ ]  },  "aggregations" : {  "userIpAddressAgg" : {  "value" : 38  },  "userIpAddressAggTerms" : {  "doc\_count\_error\_upper\_bound" : 0,  "sum\_other\_doc\_count" : 0,  "buckets" : [  {  "key" : "::1",  "doc\_count" : 6  },  {  "key" : "105.191.91.222",  "doc\_count" : 1  },  {  "key" : "113.145.189.253",  "doc\_count" : 1  },  {  "key" : "121.85.85.243",  "doc\_count" : 1  },  {  "key" : "122.8.140.102",  "doc\_count" : 1  },  …..#100 item  ]  }  }  } |

|  |  |
| --- | --- |
| ***Elastic Nest Sınıfı Sorguları C#*** | |
| **Index oluşturma** | |
| static ElasticClient elasticClient = new ElasticClient(new ConnectionSettings(new Uri("http://localhost:9200")));  static string targetIndexName = $"bills\_en";  private void EnsureEnglishIndexCreated()  {  if (!elasticClient.IndexExists(targetIndexName).Exists)  {  var result = elasticClient.CreateIndex(targetIndexName,  i => i  .Settings(s => s  .Analysis(a => a  .Tokenizers(tk => tk.Pattern("comma\_analyzer", p => p.Pattern(",")))  .TokenFilters(tf => tf  .AsciiFolding("my\_ascii\_folding", af => af.PreserveOriginal(true))  .Stop($"english\_stop", st => st.StopWords($"\_english\_"))  .Stemmer($"english\_stemmer", st => st.Language("english"))  .Stemmer($"english\_possessive\_stemmer", st => st.Language("possessive\_english"))  )  .Analyzers(analyzer => analyzer  .Custom($"english\_analyzer", ca => ca  .Tokenizer("standard")  .Filters("my\_ascii\_folding", "lowercase", $"english\_stop", $"english\_stemmer", "english\_possessive\_stemmer"))  )  ))  //.Mappings(m => m.Map<ESCompany>(mp => mp.AutoMap())) \***!!!\***  .Mappings(m => m.Map<ESBill>(mp => mp.Properties<ESBill>(p => p  .Number(nm => nm.Name(n => n.Id).Index())  .Keyword(k => k.Name(n => n.BLNumber).Index(true))  .Date(d => d.Name(n => n.Date).Index(true))  .Text(t => t.Name(n => n.BuyerName).Index(true)  .Analyzer("english\_analyzer").Fielddata(true)  .Fields(fs => fs.Keyword(fsk => fsk.Name("raw").Index(true))))  .Keyword(t => t.Name(n => n.BuyerAddress).Index(false))  .Text(t => t.Name(n => n.SupplierName).Index(true)  .Analyzer("english\_analyzer").Fielddata(true)  .Fields(fs => fs.Keyword(fsk => fsk.Name("raw").Index(true))))  .Keyword(t => t.Name(n => n.SupplierAddress).Index(false))  .Text(t => t.Name(n => n.Consignee).Index(true)  .Analyzer("english\_analyzer").Fielddata(true)  .Fields(fs => fs.Keyword(fsk => fsk.Name("raw").Index(true))))  .Keyword(t => t.Name(n => n.ConsigneeAddress).Index(false))  .Keyword(k => k.Name(n => n.HsCode).Index(true))  .Text(t => t.Name(n => n.HsCodeDescription).Index(true)  .Analyzer("english\_analyzer").Fielddata(true)  .Fields(fs => fs.Keyword(fsk => fsk.Name("raw").Index(true))))  .Text(t => t.Name(n => n.HsCodeDetailedDescription).Index(true)  .Analyzer("english\_analyzer").Fielddata(true)  .Fields(fs => fs.Keyword(fsk => fsk.Name("raw").Index(true))))  .Number(nm => nm.Name(n => n.PackingContainerQuantity).Index(true))  .Keyword(k => k.Name(n => n.PackingUnit).Index(true))  .Keyword(k => k.Name(n => n.TrunkVVD).Index(true))  .Keyword(k => k.Name(n => n.VesselName).Index(true))  .Keyword(k => k.Name(n => n.FullNameOfCarrier).Index(true))  .Number(nm => nm.Name(n => n.TEU).Index(true))  .Number(nm => nm.Name(n => n.FEU).Index(true))  .Number(nm => nm.Name(n => n.MEA).Index(true))  .Number(nm => nm.Name(n => n.Weight).Index(true))  .Keyword(t => t.Name(n => n.ExportCountryName).Index(true))  .Keyword(t => t.Name(n => n.ExportPortName).Index(true))  .Keyword(k => k.Name(n => n.ExportPortCode).Index(true))  .Keyword(t => t.Name(n => n.ImportCountryName).Index(true))  .Keyword(t => t.Name(n => n.ImportPortName).Index(true))  .Keyword(k => k.Name(n => n.ImportPortCode).Index(true))  ).AutoMap()  )).Aliases(a => a.Alias($"{targetIndexName}\_alias"))  );  }  }  \***!!!\*** = Burada belirtildiği gibi de otomatik mapping oluşturulabilir. Fakat büyük verilerde performansı iyi düzeyde tutmak adına İstenilen alanları indexlemek veya keyword olarak tanımlamak daha iyi olacaktır. | |
| **Ekleme veya güncelleme** | |
| ElasticProcesses elastic = new ElasticProcesses();  elastic.EnsureEnglishIndexCreated(); // => burada, üstteki IndexCreate’i çağırıyor.  ***Ekleme***  var response = \_elasticClient.Index(esBillModel, i => i.Index(\_targetIndex)  .Refresh(Elasticsearch.Net.Refresh.False));  ***Güncelleme***  var response = elasticClient.Update<ESBill, object>(DocumentPath<ESBill>  .Id(esBillModel.BLNumber), u => u  .Index(targetIndexName)  .Doc(esBillModel)  .DocAsUpsert(true)  .RetryOnConflict(8)); | |
| **Tek Şarta göre sorgu** | |
| public ESBill DetailsBill(string id)  {  var searchResult = \_elasticClient.Search<ESBill>(e => e.Index(\_targetIndex)  .Query(q =>  q.Match(m =>  m.Field(f => f.BLNumber).Query(id)  )  )  );  var hitModel = new ESBill();  if (!searchResult.IsValid) return null;  {  if (searchResult != null && searchResult.Hits.Count > 0)  {  hitModel = new ESBill  {  Id = searchResult.Hits.SingleOrDefault().Id,  BLNumber = searchResult.Hits.SingleOrDefault().Source.BLNumber,  HsCode = searchResult.Hits.SingleOrDefault().Source.HsCode,  Date = searchResult.Hits.SingleOrDefault().Source.Date  };  }  }  return hitModel;  } | |
|  | |

|  |  |
| --- | --- |
| **HATALAR** | |
|  | |
| üstte resimde görüldüğü gibi Index ekleme (insert) işleminde bu tarz bi hata görülmesi, diskin dolu olduğundan dolayı elastic in kendini korumaya almasına sebep oluyor. | |
| Çözüm 1 : Elastic hata vermeyene kadar disk boşaltmak, yer açmak | Çözüm 2 : index ayarlarından bloklamayı devredışı bırakmak |
| Çözüm 2 için gereken kod ve kaynak :  Çözüm 2 için gereken kod ve kaynak :  PUT \_settings  {  "index":{  "blocks":{  "read\_only\_allow\_delete":"false"  }  }  } | <https://discuss.elastic.co/t/forbidden-12-index-read-only-allow-delete-api/110282/3>  Buradaki yazıda disk doluluğu ile ilgili yazılmış.  .kibana olarak yazıldığında genel ayar, (çalışmazsa) indexin adı verildiğinde ise indexin adı verilir.  <https://stackoverflow.com/a/50609418>  <https://discuss.elastic.co/t/forbidden-12-index-read-only-allow-delete-api/110282/5> |
| PUT \_all/\_settings  {  "index":{  "blocks":{  "read\_only\_allow\_delete":"false"  }  }  } |  |
| PUT \_settings  {  "index":{  "blocks":{  "read\_only\_allow\_delete":"false"  }  }  } |  |
| PUT /bills\_en/\_settings  {  "index":{  "blocks":{  "read\_only\_allow\_delete":"false"  }  }  } |  |
| PUT /bills\_en/\_settings  {  "index.blocks.read\_only\_allow\_delete": null  } |  |
| PUT \_cluster/settings  {  "transient": {  "cluster.routing.allocation.disk.watermark.low": "100gb",  "cluster.routing.allocation.disk.watermark.high": "50gb",  "cluster.routing.allocation.disk.watermark.flood\_stage": "10gb",  "cluster.info.update.interval": "1m"  }  } |  |
|  | |