**Elasticsearch DSL Tutorial with Examples**

**1️⃣ Introduction to DSL**

Elastic search DSL is a powerful query language based on JSON used to search, filter, and analyze data in Elasticsearch.

A typical search query looks like this:

GET my-index/\_search

{

"query": {

"match": {

"field\_name": "value"

}

}

}

**2️⃣ Sample Data to Ingest**

Create an index called students and ingest the following documents:

**Create Index:**

PUT students

{

"mappings": {

"properties": {

"name": { "type": "text" },

"age": { "type": "integer" },

"grade": { "type": "keyword" },

"subjects": { "type": "keyword" },

"score": { "type": "float" },

"joined\_date": { "type": "date" }

}

}

}

**Bulk Insert Sample Data:**

POST \_bulk

{ "index": { "\_index": "students" } }

{ "name": "Alice", "age": 20, "grade": "A", "subjects": ["Math", "English"], "score": 85.5, "joined\_date": "2023-01-15" }

{ "index": { "\_index": "students" } }

{ "name": "Bob", "age": 22, "grade": "B", "subjects": ["Science", "English"], "score": 78.0, "joined\_date": "2023-03-12" }

{ "index": { "\_index": "students" } }

{ "name": "Charlie", "age": 21, "grade": "C", "subjects": ["Math", "History"], "score": 65.3, "joined\_date": "2023-06-10" }

{ "index": { "\_index": "students" } }

{ "name": "Diana", "age": 23, "grade": "A", "subjects": ["Science", "Math"], "score": 92.0, "joined\_date": "2023-02-05" }

{ "index": { "\_index": "students" } }

{ "name": "Ethan", "age": 20, "grade": "B", "subjects": ["History", "English"], "score": 72.4, "joined\_date": "2023-05-18" }

**3️⃣ Basic Queries**

**Match Query (Full-text search):**

json

CopyEdit

GET students/\_search

{

"query": {

"match": {

"name": "Alice"

}

}

}

**Term Query (Exact match for keyword fields):**

json

CopyEdit

GET students/\_search

{

"query": {

"term": {

"grade": "A"

}

}

}

**4️⃣ Filtered Queries and Compound Queries**

**Range Query (Filter by score):**

json

CopyEdit

GET students/\_search

{

"query": {

"range": {

"score": {

"gte": 80

}

}

}

}

**Bool Query (Combine multiple conditions):**

json

CopyEdit

GET students/\_search

{

"query": {

"bool": {

"must": [

{ "term": { "grade": "A" } },

{ "range": { "score": { "gte": 85 } } }

]

}

}

}

**5️⃣ Aggregations**

**Average Score:**

GET students/\_search

{

"size": 0,

"aggs": {

"average\_score": {

"avg": {

"field": "score"

}

}

}

}

**Students Count by Grade:**

json

CopyEdit

GET students/\_search

{

"size": 0,

"aggs": {

"grades\_count": {

"terms": {

"field": "grade"

}

}

}

}

**6️⃣ Sorting and Pagination**

**Sort by Score Descending:**

json

CopyEdit

GET students/\_search

{

"sort": [

{ "score": "desc" }

]

}

**Pagination with from and size:**

json

CopyEdit

GET students/\_search

{

"from": 0,

"size": 2

}

**7️⃣ Real-World Use Cases**

* Filter top performers: score >= 90
* Count how many students chose "Math"
* Average score per subject (use nested fields if subject scores are stored separately)
* Students who joined after a certain date