

HW

Task 1 - Indexing and CRUD operations (20 Points)

Task 2 - Querying and aggregating data (40 Points)

1. Searching

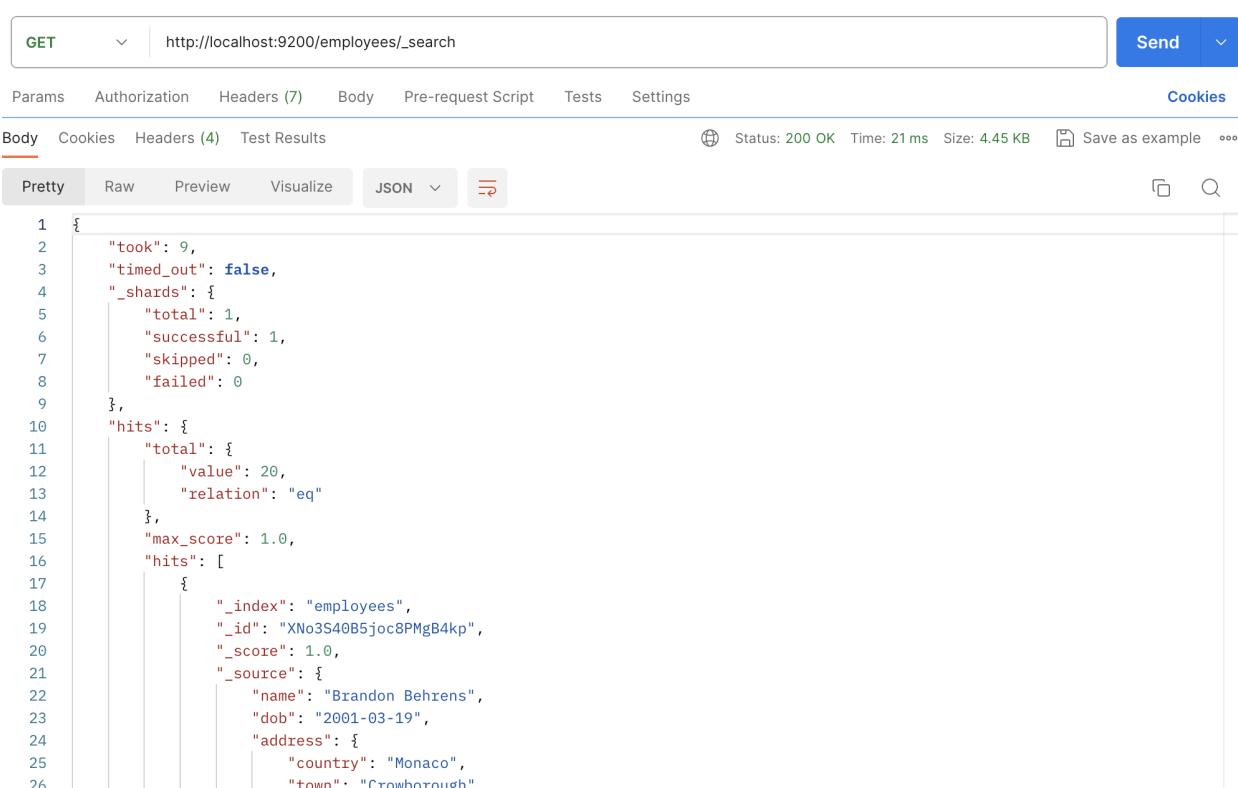
2. Aggregating

3. Joining

4. Suggesting

Task 1 - Indexing and CRUD operations (20 Points)

1. Get all records (you may starts with POST http://localhost:9200/employees/_search to see ids assigned to the records. It was possible to assign ids explicitely while doing bulk upload, but we let Elasticsearch to set ids randomly).



The screenshot shows a Postman interface with the following details:

- Method: GET
- URL: http://localhost:9200/employees/_search
- Headers: (7) including Content-Type: application/json
- Body: (raw JSON response)
- JSON Response (Pretty):

```
1 {
2   "took": 9,
3   "timed_out": false,
4   "_shards": {
5     "total": 1,
6     "successful": 1,
7     "skipped": 0,
8     "failed": 0
9   },
10  "hits": {
11    "total": {
12      "value": 20,
13      "relation": "eq"
14    },
15    "max_score": 1.0,
16    "hits": [
17      {
18        "_index": "employees",
19        "_id": "XNo3S40B5joc8PMgB4kp",
20        "_score": 1.0,
21        "_source": {
22          "name": "Brandon Behrens",
23          "dob": "2001-03-19",
24          "address": {
25            "country": "Monaco",
26            "town": "Crowborough"
27          }
28        }
29      }
30    ]
31  }
32 }
```

2. Get an employee by id.

The screenshot shows a Postman request for a GET operation on the URL `http://localhost:9200/employees/_doc/X9o3S40B5joc8PMgB4ks`. The response status is 200 OK, with a time of 59 ms and a size of 486 B. The response body is a JSON object representing an employee document, displayed in a pretty-printed format. The employee has the name "Marcus Speer", was born on "1975-02-13", and lives in "Birtley, China". They have skills in Python and AZURE, and an experience level of 4. Their rating is 7.4, and their salary is 18660. The response also includes metadata like _index, _id, _version, _seq_no, _primary_term, found, and source.

```
1 {  
2   "_index": "employees",  
3   "_id": "X9o3S40B5joc8PMgB4ks",  
4   "_version": 1,  
5   "_seq_no": 3,  
6   "_primary_term": 1,  
7   "found": true,  
8   "_source": {  
9     "name": "Marcus Speer",  
10    "dob": "1975-02-13",  
11    "address": {  
12      "country": "China",  
13      "town": "Birtley"  
14    },  
15    "email": "marcusspeer6@lender.omi.niigata.jp",  
16    "skills": [  
17      "Python",  
18      "AZURE"  
19    ],  
20    "experience": 4,  
21    "rating": 7.4,  
22    "description": "yearly discussion example politics rocky replacing insertion addressed replacing eve galaxy  
competitions pipes ceremony chips",  
23    "verified": true,  
24    "salary": 18660  
25  }  
26}
```

3. Create a new employee and put in into the index.

Specify id in the url. Query the new employee by its id.

POST http://localhost:9200/employees/_doc/cNrfS40B5joc8PMgRon0

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL **JSON** [Beautify](#)

```

1 {
2   "name": "Ana Brown",
3   "dob": "1993-03-19",
4   "address": {
5     "country": "Belarus",
6     "town": "Gomel"
7   },
8   "email": "anabrown9@gmail.com",
9   "skills": [
10    "Java",
11    "AWS"

```

Body Cookies Headers (5) Test Results Status: 201 Created Time: 66 ms Size: 338 B [Save as example](#)

Pretty Raw Preview Visualize **JSON** [Copy](#) [Search](#)

```

1 {
2   "_index": "employees",
3   "_id": "cNrfS40B5joc8PMgRon0",
4   "_version": 1,
5   "result": "created",
6   "_shards": {
7     "total": 2,
8     "successful": 1,
9     "failed": 0
10   },
11   "_seq_no": 21,
12   "_primary_term": 1
13 }

```

GET http://localhost:9200/employees/_doc/cNrfS40B5joc8PMgRon0 Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Body Cookies Headers (4) Test Results Status: 200 OK Time: 12 ms Size: 472 B Save as example ⚙️

Pretty Raw Preview Visualize JSON 🔍

```

1 {
2   "_index": "employees",
3   "_id": "cNrfS40B5joc8PMgRon0",
4   "_version": 1,
5   "_seq_no": 21,
6   "_primary_term": 1,
7   "found": true,
8   "_source": {
9     "name": "Ana Brown",
10    "dob": "1993-03-19",
11    "address": {
12      "country": "Belarus",
13      "town": "Gomel"
14    },
15    "email": "anabrown9@gmail.com",
16    "skills": [
17      "Java",
18      "AWS"
19    ],
20    "experience": 10,
21    "rating": 9.2,
22    "description": "confident, ambitious, highly motivated Java experience interview learning python",
23    "verified": true,
24    "salary": 30000
25  }
26 }
```

4. Add one more employee and try multi-get request. It may look like: GET
[http://localhost:9200/employees/_mget {"docs": \[{"_id": "1"}, {"_id": "2"}\]}}](http://localhost:9200/employees/_mget?{)

GET http://localhost:9200/employees/_mget

Params Authorization Headers (9) Body **JSON** Pre-request Script Tests Settings Cookies Beautify

```

1 {
2   "docs": [
3     {
4       "_id": "cNrfS40B5joc8PMgRon0"
5     },
6     {
7       "_id": "Xdo3S40B5joc8PMgB4ks"
8     }
9   ]
10 }

```

Body Cookies Headers (4) Test Results

Status: 200 OK Time: 12 ms Size: 687 B Save as example

Pretty Raw Preview Visualize JSON

```

1 {
2   "docs": [
3     {
4       "_index": "employees",
5       "_id": "cNrfS40B5joc8PMgRon0",
6       "_version": 2,
7       "_seq_no": 22,
8       "_primary_term": 1,
9       "found": true,
10      "_source": {
11        "name": "Ana Brown",
12        "dob": "1993-03-19",
13        "address": {
14          "country": "Belarus",
15          "town": "Gomel",
16          "email": "anabrown@gmail.com",
17          "skills": [
18            "Java",
19            "C++",
20            "experience": 10,
21            "rating": 9.2,
22            "description": "confident, ambitious, highly motivated Java experience interview learning python",
23            "verified": true,
24            "salary": 8000
25          ],
26          "age": 25
27        },
28        "name": "Oleksandr Grin",
29        "dob": "1985-06-01",
30        "address": {
31          "country": "Poland",
32          "town": "Olsztyn",
33          "email": "oleksandr.grin@yahoo.com",
34          "skills": [
35            "Java",
36            "C++",
37            "experience": 10,
38            "rating": 9.2,
39            "description": "lithuanian wall reasons fwreware ntcc cho defensive pressed reality printable ctrl transition alumni groundwater planner",
40            "verified": true,
41            "salary": 8000
42          ],
43        }
44      }
45    }
46  ]
47 }

```

GET http://localhost:9200/employees/_mget

Params Authorization Headers (9) Body **JSON** Pre-request Script Tests Settings Cookies

Status: 200 OK Time: 12 ms Size: 687 B Save as example

Pretty Raw Preview Visualize JSON

```

1 {
2   "docs": [
3     {
4       "_index": "employees",
5       "_id": "cNrfS40B5joc8PMgRon0",
6       "_version": 2,
7       "_seq_no": 22,
8       "_primary_term": 1,
9       "found": true,
10      "_source": {
11        "name": "Ana Brown",
12        "dob": "1993-03-19",
13        "address": {
14          "country": "Belarus",
15          "town": "Gomel",
16          "email": "anabrown@gmail.com",
17          "skills": [
18            "Java",
19            "C++",
20            "experience": 10,
21            "rating": 9.2,
22            "description": "confident, ambitious, highly motivated Java experience interview learning python",
23            "verified": true,
24            "salary": 8000
25          ],
26          "age": 25
27        },
28        "name": "Oleksandr Grin",
29        "dob": "1985-06-01",
30        "address": {
31          "country": "Poland",
32          "town": "Olsztyn",
33          "email": "oleksandr.grin@yahoo.com",
34          "skills": [
35            "Java",
36            "C++",
37            "experience": 10,
38            "rating": 9.2,
39            "description": "lithuanian wall reasons fwreware ntcc cho defensive pressed reality printable ctrl transition alumni groundwater planner",
40            "verified": true,
41            "salary": 8000
42          ],
43        }
44      }
45    }
46  ]
47 }

```

5. Learn mapping, automatically created by Elasticsearch.

GET http://localhost:9200/employees/_mapping

Pay attention to string values types. They are both "text" and "keyword". Learn about the difference.

The screenshot shows the Postman application interface. At the top, there is a header bar with 'GET' and a dropdown menu. Below it, the URL 'http://localhost:9200/employees/_mapping' is entered. To the right of the URL is a 'Send' button. Underneath the URL, there are tabs for 'Params', 'Authorization', 'Headers (7)', 'Body', 'Pre-request Script', 'Tests', and 'Settings'. The 'Body' tab is currently selected. On the far right of the header, there is a 'Cookies' section. Below the tabs, there are buttons for 'Pretty', 'Raw', 'Preview', 'Visualize', 'JSON', and a copy icon. The main content area displays the JSON response from the Elasticsearch mapping. The response is a large block of JSON code with line numbers 1 through 28 on the left. The JSON structure shows the 'employees' type with 'mappings' for 'address' and 'town' fields, each having 'text' and 'keyword' types with specific settings like 'ignore_above': 256.

```
1 {
2   "employees": {
3     "mappings": {
4       "properties": {
5         "address": {
6           "properties": {
7             "country": {
8               "type": "text",
9               "fields": {
10                 "keyword": {
11                   "type": "keyword",
12                   "ignore_above": 256
13                 }
14               }
15             },
16             "town": {
17               "type": "text",
18               "fields": {
19                 "keyword": {
20                   "type": "keyword",
21                   "ignore_above": 256
22                 }
23               }
24             }
25           }
26         },
27         "description": {
28           "type": "text"
29         }
30       }
31     }
32   }
33 }
```

6. Update one of employee data. Add one more skill.

POST http://localhost:9200/employees/_update/cNrfS40B5joc8PMgRon0

Body

```

1 {
2   "script": {
3     "source": "ctx._source.skills.add(params.new_skill)",
4     "lang": "painless",
5     "params": {
6       "new_skill": "Python"
7     }
8   }

```

Status: 200 OK Time: 112 ms Size: 288 B Save as example

Body

```

1 {
2   "_index": "employees",
3   "_id": "cNrfS40B5joc8PMgRon0",
4   "_version": 3,
5   "result": "updated",
6   "_shards": {
7     "total": 2,
8     "successful": 1,
9     "failed": 0
10   },
11   "_seq_no": 23,
12   "_primary_term": 1
13 }

```

7. Delete one of employees by its id.

DELETE http://localhost:9200/employees/_doc/cNrfS40B5joc8PMgRon0

Body

```

1 {
2   "_index": "employees",
3   "_id": "cNrfS40B5joc8PMgRon0",
4   "_version": 4,
5   "result": "deleted",
6   "_shards": {
7     "total": 2,
8     "successful": 1,
9     "failed": 0
10   },
11   "_seq_no": 24,
12   "_primary_term": 1
13 }

```

Status: 200 OK Time: 87 ms Size: 288 B Save as example

Task 2 - Querying and aggregating data (40 Points)

1. Searching

1. Get employees with "description":"motivated" using Match query.

The screenshot shows the Postman interface with a GET request to `http://localhost:9200/employees/_search`. The Body tab is selected, displaying the following JSON query:

```
1 {
2     "query": {
3         "match": {
4             "description": "motivated"
5         }
6     }
7 }
```

The response status is 200 OK with a total time of 37 ms and a size of 1.58 KB. The response body is displayed in Pretty mode:

```
1 {
2     "took": 20,
3     "timed_out": false,
4     "_shards": {
5         "total": 1,
6         "successful": 1,
7         "skipped": 0,
8         "failed": 0
9     },
10    "hits": {
11        "total": {
12            "value": 3,
13            "relation": "eq"
14        },
15        "max_score": 2.2264347,
16        "hits": [
17            {
18                "_index": "employees",
19                "_score": 2.2264347,
20                "_source": {
21                    "id": 1,
22                    "name": "Carey Catlett",
23                    "description": "motivated"
24                }
25            }
26        ]
27    }
28 }
```

2. Get employees with "description":"hightly motivated" using Match query.

Do you still get Carey Catlett in the response? He has only "motivated" in his description.

It's because by default Elastic uses OR condition between the two words in Match condition.

Add "operator": AND to the condition and see how it works now. You need to restructure Match condition a little bit.

The screenshot shows a Postman request configuration. The method is GET, the URL is `http://localhost:9200/employees/_search`, and the Body is set to JSON. The request body contains the following Elasticsearch query:

```

1 {
2   "query": {
3     "match": {
4       "description": {
5         "query": "highly motivated",
6         "operator": "and"
7       }
8     }
9   }
10 }

```

The response status is 200 OK with a time of 48 ms and a size of 750 B. The response body is:

```

1 {
2   "took": 40,
3   "timed_out": false,
4   "_shards": {
5     "total": 1,
6     "successful": 1,
7     "skipped": 0,
8     "failed": 0
9   },
10  "hits": {
11    "total": {
12      "value": 1,
13      "relation": "eq"
14    }
}

```

3. Run multi_match query to get employees with "python" either in "skills" or in "description" fields.

GET http://localhost:9200/employees/_search

Send

Params Authorization Headers (9) Body **JSON** Pre-request Script Tests Settings Cookies Beautify

```

1 {
2   ...
3   "query": {
4     ...
5     "multi_match": {
6       ...
7         "query": "python",
8         ...
9         "fields": ["skills", "description"]
10      ...
11    }
12  ...
13}
14 ...
15

```

Body Cookies Headers (4) Test Results

Status: 200 OK Time: 61 ms Size: 3.26 KB Save as example

Pretty Raw Preview Visualize JSON

```

1 {
2   "took": 35,
3   "timed_out": false,
4   "_shards": {
5     "total": 1,
6     "successful": 1,
7     "skipped": 0,
8     "failed": 0
9   },
10  "hits": {
11    "total": {
12      "value": 7,
13      "relation": "eq"
14    },
15    "max_score": 3.192822,
16    ...
17  }
18 ...
19

```

4. Get employee with "name" : "stepanie" using Term query.

Try the same with "name" : "Stepanie" and "name" : "Stepanie Spain". Did you get the result? Why?

The screenshot shows a Postman request to `http://localhost:9200/employees/_search`. The request method is GET. The Body tab contains the following JSON query:

```

1 {
2   ...
3     "query": {
4       ...
5         "term": {
6           ...
7             "name": "stepanie"
8           ...
9         }
10    ...
11   }
12 }
```

The response status is 200 OK, with a total of 1 hit. The hit details are as follows:

```

6   "successful": 1,
7   "skipped": 0,
8   "failed": 0
9 },
10 "hits": {
11   "total": {
12     "value": 1,
13     "relation": "eq"
14   },
15   "max_score": 2.6855774,
16   "hits": [
17     {
18       ...
19       "_index": "employees",
20       "_id": "bdo3S40B5joc8PMgB4ks",
21       "_score": 2.6855774,
22       "_source": {
23         ...
24         "name": "Stepanie Spain",
25         ...
26       }
27     }
28   ]
29 }
```

My answer:

only “stepanie” resulted in non-empty result set.

This is because “name” is text field which means that text there was tokenized and tokens were normalized and stemmed. And term query is used for finding exact matches. It doesn’t preprocess the query. So when we pass “stepanie”, we basically give a stemmed version of that token and this is why we have a hit. And when we pass “Stepanie”, we don’t get a hit, because it doesn’t match the stemmed version of original string in “name” field.

5. Get employee with "name.keyword" : "stepanie" using Terms query.

Try "name.keyword" : "Stepanie Spain". Review the difference between Match and Term queries as well as between text and keyword types search.

GET Send

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {  
2   "query": {  
3     "term": {  
4       "name.keyword": "stepanie"  
5     }  
6   }  
7 }
```

Body Cookies Headers (4) Test Results

Pretty Raw Preview Visualize JSON

```
1 {  
2   "took": 7,  
3   "timed_out": false,  
4   "_shards": {  
5     "total": 1,  
6     "successful": 1,  
7     "skipped": 0,  
8     "failed": 0  
9   },  
10  "hits": {  
11    "total": {  
12      "value": 0,  
13      "relation": "eq"  
14    },  
15    "max_score": null,  
16    "hits": []  
17  }  
18 }
```

The screenshot shows a Postman request for a GET operation on the URL `http://localhost:9200/employees/_search`. The 'Body' tab is selected, containing the following JSON query:

```

1  {
2    ...
3      "query": {
4          "term": {
5              "name.keyword": "Stepanie Spain"
6          }
7      }
8  }

```

The response status is 200 OK, with a time of 28 ms and a size of 748 B. The response body is displayed in Pretty mode:

```

1 {
2   "took": 13,
3   "timed_out": false,
4   "_shards": {
5       "total": 1,
6       "successful": 1,
7       "skipped": 0,
8       "failed": 0
9   },
10  "hits": {
11      "total": {
12          "value": 1,
13          "relation": "eq"
14      },
15      "max_score": 2.6855774,
16      "hits": [
17          {
18              "index": "employees"
19          }
20      ]
21  }
22}

```

My answer:

keyword search is exact search, this is why term query was successful only when we specified full name “Stepanie Spain”

6. Get employees by several names using Terms query with array of values.

The screenshot shows a Postman interface with the following details:

- Method:** GET
- URL:** http://localhost:9200/employees/_search
- Body:** JSON (selected)
- Request Body:**

```

1 {
2   ...
3     "query": {
4       ...
5         "terms": {
6           ...
7             "name.keyword": ["Stepanie Spain", "Rochell Grimm"]
8           ...
9         }
10    }
11 }
```
- Response Status:** 200 OK
- Response Time:** 39 ms
- Response Size:** 1.13 KB
- Response Preview:**

```

1 {
2   "took": 20,
3   "timed_out": false,
4   "_shards": {
5     "total": 1,
6     "successful": 1,
7     "skipped": 0,
8     "failed": 0
9   },
10  "hits": {
11    "total": {
12      "value": 2,
13      "relation": "eq"
14    },
15    "max_score": 1.0,
16    "hits": [
17      {
18        ...
19      }
20    ]
21  }
22 }
```

7. Get employees with experience from 5 to 20 years using Range query.

GET http://localhost:9200/employees/_search

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```

1 {
2   "query": {
3     "range": {
4       "experience": {
5         "gte": 5,
6         "lt": 20
7       }
8     }
9   }
10 }
```

Body Cookies Headers (4) Test Results

Status: 200 OK Time: 65 ms Size: 4.45 KB Save as example

Pretty Raw Preview Visualize JSON

```

1 {
2   "took": 57,
3   "timed_out": false,
4   "_shards": {
5     "total": 1,
6     "successful": 1,
7     "skipped": 0,
8     "failed": 0
9   },
10  "hits": {
11    "total": {
12      "value": 15,
13      "relation": "eq"
14    },
15  }
```

8. Get employees younger than 1,Jan,2000 using Range query.

GET http://localhost:9200/employees/_search Send

Params Authorization Headers (9) Body **JSON** Pre-request Script Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```

1 {
2   "query": {
3     "range": {
4       "dob": {
5         "gte": "2000-01-01"
6       }
7     }
8   }
9 }
```

Body Cookies Headers (4) Test Results Status: 200 OK Time: 10 ms Size: 3.16 KB Save as example

Pretty Raw Preview Visualize JSON

```

1 {
2   "took": 3,
3   "timed_out": false,
4   "_shards": {
5     "total": 1,
6     "successful": 1,
7     "skipped": 0,
8     "failed": 0
9   },
10  "hits": {
11    "total": {
12      "value": 7,
13      "relation": "eq"
14    },
15    "max_score": 1.0,
```

9. Perform Wildcard search, for example "address.town":{"value": "b*y"}.

The screenshot shows a Postman interface with the following details:

- Method:** GET
- URL:** http://localhost:9200/employees/_search
- Body:** JSON (selected)
- Query Body:**

```

1 {
2   ...
3     "query": {
4       ...
5         "wildcard": {
6           ...
7             "address.town": "b*"
8           ...
9         }
10    }
11  }
12 }
```
- Response Status:** 200 OK
- Response Time:** 101 ms
- Response Size:** 1.12 KB
- Response Content (Pretty):**

```

1 {
2   "took": 76,
3   "timed_out": false,
4   "_shards": {
5     "total": 1,
6     "successful": 1,
7     "skipped": 0,
8     "failed": 0
9   },
10  "hits": {
11    "total": {
12      "value": 2,
13      "relation": "eq"
14    },
15    "max_score": 1.0,
16    "hits": [
17      {
18        ...
19      }
20    ]
21  }
22 }
```

10. Replace it by Regexp to get only "Batley".

GET Send

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL **JSON**

```

1 {
2   ...
3     "query": {
4       ...
5         "regexp": {
6           ...
7             "address.town": "ba.*y"
8           ...
9         }
10    ...
11  }
12 }
```

Body Cookies Headers (4) Test Results

Pretty Raw Preview Visualize **JSON**

```

1 {
2   "took": 6,
3   "timed_out": false,
4   "_shards": {
5     "total": 1,
6     "successful": 1,
7     "skipped": 0,
8     "failed": 0
9   },
10  "hits": {
11    "total": {
12      "value": 1,
13      "relation": "eq"
14    },
15    "max_score": 1.0,
16    "hits": [
17      {
18        ...
19      }
20    ]
21  }
22 }
```

GET http://localhost:9200/employees/_search Send

Params Authorization Headers (9) Body **JSON** Pre-request Script Tests Settings Cookies Beautify

```

1  {
2   ...
3   "query": {
4     ...
5     "regexp": {
6       ...
7       "address.town": "ba.*y"
7     }
}

```

Body Cookies Headers (4) Test Results Status: 200 OK Time: 13 ms Size: 711 B Save as example ⋮

Pretty Raw Preview Visualize JSON ⚡

```

15  "max_score": 1.0,
16  "hits": [
17    {
18      "_index": "employees",
19      "_id": "ado3S40B5joc8PMgB4ks",
20      "_score": 1.0,
21      "_source": {
22        "name": "Alise Creed",
23        "dob": "2002-05-29",
24        "address": {
25          "country": "Ivory Coast",
26          "town": "Batley"
27        },
28        "email": "alise_creed98@decrease.net.gg",
29        "skills": [
30          "Javascript",
31          "GCP"

```

Postbot Runner Start Proxv Cookies Trash ⌂

11. Investigate how Fuzzy query works. Make two spelling mistakes in the town, set "fuzziness": "2", and check you get the town in the response anyway.

GET http://localhost:9200/employees/_search

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```

1 {
2   ...
3     "query": {
4       ...
5         "fuzzy": {
6           ...
7             "address.town": {
8               ...
9                 "value": "Karsley",
10                "fuzziness": 2
11             }
12           ...
13         }
14       ...
15     }
16   ...
17 }
```

Body Cookies Headers (4) Test Results Status: 200 OK Time: 53 ms Size: 736 B Save as example

Pretty Raw Preview Visualize JSON

```

19   ...
20     ...
21       ...
22         ...
23           ...
24             ...
25               ...
26                 ...
27                   ...
28                     ...
29                       ...
30                         ...
31                           ...
32                             ...
33                               ...
34                                 ...
35                                   ...
36                                     ...
37                                       ...
38                                         ...
39                                           ...
40                                             ...
41                                               ...
42                                                 ...
43                                                   ...
44                                                     ...
45                                                       ...
46                                                         ...
47                                                       ...
48                                                       ...
49                                                       ...
50                                                       ...
51                                                       ...
52                                                       ...
53                                                       ...
54                                                       ...
55                                                       ...
56                                                       ...
57                                                       ...
58                                                       ...
59                                                       ...
60                                                       ...
61                                                       ...
62                                                       ...
63                                                       ...
64                                                       ...
65                                                       ...
66                                                       ...
67                                                       ...
68                                                       ...
69                                                       ...
70                                                       ...
71                                                       ...
72                                                       ...
73                                                       ...
74                                                       ...
75                                                       ...
76                                                       ...
77                                                       ...
78                                                       ...
79                                                       ...
80                                                       ...
81                                                       ...
82                                                       ...
83                                                       ...
84                                                       ...
85                                                       ...
86                                                       ...
87                                                       ...
88                                                       ...
89                                                       ...
90                                                       ...
91                                                       ...
92                                                       ...
93                                                       ...
94                                                       ...
95                                                       ...
96                                                       ...
97                                                       ...
98                                                       ...
99                                                       ...
100                                                       ...
101                                                       ...
102                                                       ...
103                                                       ...
104                                                       ...
105                                                       ...
106                                                       ...
107                                                       ...
108                                                       ...
109                                                       ...
110                                                       ...
111                                                       ...
112                                                       ...
113                                                       ...
114                                                       ...
115                                                       ...
116                                                       ...
117                                                       ...
118                                                       ...
119                                                       ...
120                                                       ...
121                                                       ...
122                                                       ...
123                                                       ...
124                                                       ...
125                                                       ...
126                                                       ...
127                                                       ...
128                                                       ...
129                                                       ...
130                                                       ...
131                                                       ...
132                                                       ...
133                                                       ...
134                                                       ...
135                                                       ...
136                                                       ...
137                                                       ...
138                                                       ...
139                                                       ...
140                                                       ...
141                                                       ...
142                                                       ...
143                                                       ...
144                                                       ...
145                                                       ...
146                                                       ...
147                                                       ...
148                                                       ...
149                                                       ...
150                                                       ...
151                                                       ...
152                                                       ...
153                                                       ...
154                                                       ...
155                                                       ...
156                                                       ...
157                                                       ...
158                                                       ...
159                                                       ...
160                                                       ...
161                                                       ...
162                                                       ...
163                                                       ...
164                                                       ...
165                                                       ...
166                                                       ...
167                                                       ...
168                                                       ...
169                                                       ...
170                                                       ...
171                                                       ...
172                                                       ...
173                                                       ...
174                                                       ...
175                                                       ...
176                                                       ...
177                                                       ...
178                                                       ...
179                                                       ...
180                                                       ...
181                                                       ...
182                                                       ...
183                                                       ...
184                                                       ...
185                                                       ...
186                                                       ...
187                                                       ...
188                                                       ...
189                                                       ...
190                                                       ...
191                                                       ...
192                                                       ...
193                                                       ...
194                                                       ...
195                                                       ...
196                                                       ...
197                                                       ...
198                                                       ...
199                                                       ...
200                                                       ...
201                                                       ...
202                                                       ...
203                                                       ...
204                                                       ...
205                                                       ...
206                                                       ...
207                                                       ...
208                                                       ...
209                                                       ...
210                                                       ...
211                                                       ...
212                                                       ...
213                                                       ...
214                                                       ...
215                                                       ...
216                                                       ...
217                                                       ...
218                                                       ...
219                                                       ...
220                                                       ...
221                                                       ...
222                                                       ...
223                                                       ...
224                                                       ...
225                                                       ...
226                                                       ...
227                                                       ...
228                                                       ...
229                                                       ...
230                                                       ...
231                                                       ...
232                                                       ...
233                                                       ...
234                                                       ...
235                                                       ...
236                                                       ...
237                                                       ...
238                                                       ...
239                                                       ...
240                                                       ...
241                                                       ...
242                                                       ...
243                                                       ...
244                                                       ...
245                                                       ...
246                                                       ...
247                                                       ...
248                                                       ...
249                                                       ...
250                                                       ...
251                                                       ...
252                                                       ...
253                                                       ...
254                                                       ...
255                                                       ...
256                                                       ...
257                                                       ...
258                                                       ...
259                                                       ...
260                                                       ...
261                                                       ...
262                                                       ...
263                                                       ...
264                                                       ...
265                                                       ...
266                                                       ...
267                                                       ...
268                                                       ...
269                                                       ...
270                                                       ...
271                                                       ...
272                                                       ...
273                                                       ...
274                                                       ...
275                                                       ...
276                                                       ...
277                                                       ...
278                                                       ...
279                                                       ...
280                                                       ...
281                                                       ...
282                                                       ...
283                                                       ...
284                                                       ...
285                                                       ...
286                                                       ...
287                                                       ...
288                                                       ...
289                                                       ...
290                                                       ...
291                                                       ...
292                                                       ...
293                                                       ...
294                                                       ...
295                                                       ...
296                                                       ...
297                                                       ...
298                                                       ...
299                                                       ...
300                                                       ...
301                                                       ...
302                                                       ...
303                                                       ...
304                                                       ...
305                                                       ...
306                                                       ...
307                                                       ...
308                                                       ...
309                                                       ...
310                                                       ...
311                                                       ...
312                                                       ...
313                                                       ...
314                                                       ...
315                                                       ...
316                                                       ...
317                                                       ...
318                                                       ...
319                                                       ...
320                                                       ...
321                                                       ...
322                                                       ...
323                                                       ...
324                                                       ...
325                                                       ...
326                                                       ...
327                                                       ...
328                                                       ...
329                                                       ...
330                                                       ...
331                                                       ...
332                                                       ...
333                                                       ...
334                                                       ...
335                                                       ...
336                                                       ...
337                                                       ...
338                                                       ...
339                                                       ...
340                                                       ...
341                                                       ...
342                                                       ...
343                                                       ...
344                                                       ...
345                                                       ...
346                                                       ...
347                                                       ...
348                                                       ...
349                                                       ...
350                                                       ...
351                                                       ...
352                                                       ...
353                                                       ...
354                                                       ...
355                                                       ...
356                                                       ...
357                                                       ...
358                                                       ...
359                                                       ...
360                                                       ...
361                                                       ...
362                                                       ...
363                                                       ...
364                                                       ...
365                                                       ...
366                                                       ...
367                                                       ...
368                                                       ...
369                                                       ...
370                                                       ...
371                                                       ...
372                                                       ...
373                                                       ...
374                                                       ...
375                                                       ...
376                                                       ...
377                                                       ...
378                                                       ...
379                                                       ...
380                                                       ...
381                                                       ...
382                                                       ...
383                                                       ...
384                                                       ...
385                                                       ...
386                                                       ...
387                                                       ...
388                                                       ...
389                                                       ...
390                                                       ...
391                                                       ...
392                                                       ...
393                                                       ...
394                                                       ...
395                                                       ...
396                                                       ...
397                                                       ...
398                                                       ...
399                                                       ...
400                                                       ...
401                                                       ...
402                                                       ...
403                                                       ...
404                                                       ...
405                                                       ...
406                                                       ...
407                                                       ...
408                                                       ...
409                                                       ...
410                                                       ...
411                                                       ...
412                                                       ...
413                                                       ...
414                                                       ...
415                                                       ...
416                                                       ...
417                                                       ...
418                                                       ...
419                                                       ...
420                                                       ...
421                                                       ...
422                                                       ...
423                                                       ...
424                                                       ...
425                                                       ...
426                                                       ...
427                                                       ...
428                                                       ...
429                                                       ...
430                                                       ...
431                                                       ...
432                                                       ...
433                                                       ...
434                                                       ...
435                                                       ...
436                                                       ...
437                                                       ...
438                                                       ...
439                                                       ...
440                                                       ...
441                                                       ...
442                                                       ...
443                                                       ...
444                                                       ...
445                                                       ...
446                                                       ...
447                                                       ...
448                                                       ...
449                                                       ...
450                                                       ...
451                                                       ...
452                                                       ...
453                                                       ...
454                                                       ...
455                                                       ...
456                                                       ...
457                                                       ...
458                                                       ...
459                                                       ...
460                                                       ...
461                                                       ...
462                                                       ...
463                                                       ...
464                                                       ...
465                                                       ...
466                                                       ...
467                                                       ...
468                                                       ...
469                                                       ...
470                                                       ...
471                                                       ...
472                                                       ...
473                                                       ...
474                                                       ...
475                                                       ...
476                                                       ...
477                                                       ...
478                                                       ...
479                                                       ...
480                                                       ...
481                                                       ...
482                                                       ...
483                                                       ...
484                                                       ...
485                                                       ...
486                                                       ...
487                                                       ...
488                                                       ...
489                                                       ...
490                                                       ...
491                                                       ...
492                                                       ...
493                                                       ...
494                                                       ...
495                                                       ...
496                                                       ...
497                                                       ...
498                                                       ...
499                                                       ...
500                                                       ...
501                                                       ...
502                                                       ...
503                                                       ...
504                                                       ...
505                                                       ...
506                                                       ...
507                                                       ...
508                                                       ...
509                                                       ...
510                                                       ...
511                                                       ...
512                                                       ...
513                                                       ...
514                                                       ...
515                                                       ...
516                                                       ...
517                                                       ...
518                                                       ...
519                                                       ...
520                                                       ...
521                                                       ...
522                                                       ...
523                                                       ...
524                                                       ...
525                                                       ...
526                                                       ...
527                                                       ...
528                                                       ...
529                                                       ...
530                                                       ...
531                                                       ...
532                                                       ...
533                                                       ...
534                                                       ...
535                                                       ...
536                                                       ...
537                                                       ...
538                                                       ...
539                                                       ...
540                                                       ...
541                                                       ...
542                                                       ...
543                                                       ...
544                                                       ...
545                                                       ...
546                                                       ...
547                                                       ...
548                                                       ...
549                                                       ...
550                                                       ...
551                                                       ...
552                                                       ...
553                                                       ...
554                                                       ...
555                                                       ...
556                                                       ...
557                                                       ...
558                                                       ...
559                                                       ...
560                                                       ...
561                                                       ...
562                                                       ...
563                                                       ...
564                                                       ...
565                                                       ...
566                                                       ...
567                                                       ...
568                                                       ...
569                                                       ...
570                                                       ...
571                                                       ...
572                                                       ...
573                                                       ...
574                                                       ...
575                                                       ...
576                                                       ...
577                                                       ...
578                                                       ...
579                                                       ...
580                                                       ...
581                                                       ...
582                                                       ...
583                                                       ...
584                                                       ...
585                                                       ...
586                                                       ...
587                                                       ...
588                                                       ...
589                                                       ...
590                                                       ...
591                                                       ...
592                                                       ...
593                                                       ...
594                                                       ...
595                                                       ...
596                                                       ...
597                                                       ...
598                                                       ...
599                                                       ...
600                                                       ...
601                                                       ...
602                                                       ...
603                                                       ...
604                                                       ...
605                                                       ...
606                                                       ...
607                                                       ...
608                                                       ...
609                                                       ...
610                                                       ...
611                                                       ...
612                                                       ...
613                                                       ...
614                                                       ...
615                                                       ...
616                                                       ...
617                                                       ...
618                                                       ...
619                                                       ...
620                                                       ...
621                                                       ...
622                                                       ...
623                                                       ...
624                                                       ...
625                                                       ...
626                                                       ...
627                                                       ...
628                                                       ...
629                                                       ...
630                                                       ...
631                                                       ...
632                                                       ...
633                                                       ...
634                                                       ...
635                                                       ...
636                                                       ...
637                                                       ...
638                                                       ...
639                                                       ...
640                                                       ...
641                                                       ...
642                                                       ...
643                                                       ...
644                                                       ...
645                                                       ...
646                                                       ...
647                                                       ...
648                                                       ...
649                                                       ...
650                                                       ...
651                                                       ...
652                                                       ...
653                                                       ...
654                                                       ...
655                                                       ...
656                                                       ...
657                                                       ...
658                                                       ...
659                                                       ...
660                                                       ...
661                                                       ...
662                                                       ...
663                                                       ...
664                                                       ...
665                                                       ...
666                                                       ...
667                                                       ...
668                                                       ...
669                                                       ...
670                                                       ...
671                                                       ...
672                                                       ...
673................................................................
```

2. Aggregating

- Aggregate employees by skills. You shoud use "skills.keyword" here.

GET http://localhost:9200/employees/_search

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL **JSON**

Beautify

```

1  {
2    "size": 0,
3    "aggs": {
4      "skills_aggregation": {
5        "terms": {
6          "field": "skills.keyword"
7        }
8      }
9    }
10 }
```

Body Cookies Headers (4) Test Results Status: 200 OK Time: 24 ms Size: 647 B Save as example

Pretty Raw Preview Visualize JSON

```

18 "aggregations": {
19   "skills_aggregation": {
20     "doc_count_error_upper_bound": 0,
21     "sum_other_doc_count": 0,
22     "buckets": [
23       {
24         "key": "GCP",
25         "doc_count": 8
26       },
27       {
28         "key": "AZURE",
29         "doc_count": 7
30       },
31     ]
32 }
```

2. Filter out employees by adding filter condition to the previous aggregation query.
 Aggregate verified employees only.

GET http://localhost:9200/employees/_search Send

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```

1  {
2   ...
3   "size": 0,
4   "query": {
5     "term": {
6       "verified": true
7     }
8   },
9   "aggs": {
10    "skills_aggregation": {
11      "terms": {
12        "field": "skills.keyword"
13      }
14    }
15  }

```

Body Cookies Headers (4) Test Results Status: 200 OK Time: 98 ms Size: 647 B Save as example

Pretty Raw Preview Visualize JSON

```

1  {
2   "took": 72,
3   "timed_out": false,
4   "_shards": {
5     "total": 1,
6     "successful": 1,
7     "skipped": 0,
8     "failed": 0
9   },
10  "hits": {
11    "total": {
12      "value": 17,

```

3. Add average metric on "field": "rating" to every bucket from the previous aggregation.
 You'll need to add nested avg aggregation.

GET http://localhost:9200/employees/_search

Send

Params Authorization Headers (9) Body **JSON** Pre-request Script Tests Settings Cookies Beautify

```

1  {
2    "size": 0,
3    "query": {
4      "term": {
5        "verified": true
6      }
7    },
8    "aggs": {
9      "skills_aggregation": {
10        "terms": {
11          "field": "skills.keyword"
12        },
13        "aggs": {
14          "avg_rating": {
15            "avg": {
16              "field": "rating"
17            }
18          }
19        }
20      }
21    }
22  }

```

Pretty Raw Preview Visualize JSON

Status: 200 OK Time: 109 ms Size: 975 B Save as example

```

10   "hits": {
11     "total": {
12       "value": 17,
13       "relation": "eq"
14     },
15     "max_score": null,
16     "hits": []
17   },
18   "aggregations": {
19     "skills_aggregation": {
20       "doc_count_error_upper_bound": 0,
21       "sum_other_doc_count": 0,
22       "buckets": [
23         {
24           "key": "GCP",
25           "doc_count": 7,
26           "avg_rating": {
27             "value": 6.428571428571429
28           }
29         },
30         {
31           "key": "AWS",
32           "doc_count": 5,
33           "avg_rating": {
34             "value": 6.219999885559082
35           }
36       ]
37     }
38   }
39 }
```

4. Replace average aggregation by stats aggregation. Sort skills buckets by average stats metric DESC.

"order": {"rating_stats.avg": "desc"}.

GET Send

Params Authorization Headers (9) **Body** ● Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
1 {
2   "size": 0,
3   "query": {
4     "term": {
5       "verified": true
6     }
7   },
8   "aggs": {
9     "skills_aggregation": {
10      "terms": {
11        "field": "skills.keyword",
12        "order": {
13          "rating_stats.avg": "desc"
14        }
15      },
16      "aggs": {
17        "rating_stats": {
18          "stats": {
19            "field": "rating"
20          }
21        }
22      }
23    }
24  }
25 }
```

GET http://localhost:9200/employees/_search

Params Authorization Headers (9) Body ● Pre-request Script Tests Settings Cookies

Body Cookies Headers (4) Test Results Status: 200 OK Time: 58 ms Size: 1.52 KB Save as example ⚙️

Pretty Raw Preview Visualize JSON 🔍

```

18 "aggregations": {
19   "skills_aggregation": {
20     "doc_count_error_upper_bound": 0,
21     "sum_other_doc_count": 0,
22     "buckets": [
23       {
24         "key": "Scala",
25         "doc_count": 2,
26         "rating_stats": {
27           "count": 2,
28           "min": 4.599999904632568,
29           "max": 9.399999618530273,
30           "avg": 6.999999761581421,
31           "sum": 13.999999523162842
32         }
33       },
34       {
35         "key": "Python",
36         "doc_count": 5,
37         "rating_stats": {
38           "count": 5,
39           "min": 5.5,
40           "max": 7.400000095367432,
41           "avg": 6.619999980926513,
42           "sum": 33.09999990463257
43         }
44       },
45       {
46         "key": "GCP",
47         "doc_count": 7,
48       }
49     ]
50   }
51 }

```

3. Joining

1. Query the child's documents and then returns the parents associated with them as the results.

Suppose we need to query for the term “music” in the field “comments_description” in the child documents,

and to get the parent documents corresponding to the search results. Use has_child query.

The screenshot shows the Postman interface with a GET request to `localhost:9200/posts-comments/_search`. The Body tab is selected, displaying the following JSON query:

```

1 {
2   "query": {
3     "has_child": {
4       "type": "comment",
5       "query": {
6         "match": {
7           "comment_description": "music"
8         }
9       }
10    }
11  }
12 }

```

The response status is 200 OK, with a total time of 74 ms and a size of 455 B. The response body is:

```

10 "hits": {
11   "total": {
12     "value": 1,
13     "relation": "eq"
14   },
15   "max_score": 1.0,
16   "hits": [
17     {
18       "_index": "posts-comments",
19       "_id": "2",
20       "_score": 1.0,
21       "_source": {
22         "_join_type": {
23           "name": "post"
24         },
25         "post_title": "Beauty and the beast - a nice movie"
26       }
27     }
28   ]
29 }

```

2. Has_parent query would perform the opposite of the has_child query, that is it will return the child documents of the parent documents that matched the query.
 Search for the word “Beauty” in the parent document and return the child documents for the matched parents.

GET Send

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL **JSON** Beautify

```

1  {
2   ...
3   "query": {
4     ...
5     "has_parent": {
6       ...
7       "parent_type": "post",
8       ...
9       "query": {
10         ...
11         "match": {
12           ...
13           "post_title": "Beauty"
14         }
15       }
16     }
17   }
18 }
```

Body Cookies Headers (4) Test Results

Pretty Raw Preview Visualize **JSON**

Status: 200 OK Time: 25 ms Size: 1021 B

```

1 {
2   "took": 12,
3   "timed_out": false,
4   "_shards": {
5     "total": 1,
6     "successful": 1,
7     "skipped": 0,
8     "failed": 0
9   },
10  "hits": {
11    "total": {
12      "value": 2,
13      "relation": "eq"
14    },
15    "max_score": 1.0,
16    "hits": [
17      {
18        ...
19        "_index": "posts-comments",
20        "_id": "C",
21        "_score": 1.0,
22        ...
23      }
24    ]
25  }
26 }
```

```

15      "max_score": 1.0,
16      "hits": [
17          {
18              "_index": "posts-comments",
19              "_id": "C",
20              "_score": 1.0,
21              "_routing": "2",
22              "_source": {
23                  "_join_type": {
24                      "name": "comment",
25                      "parent": "2"
26                  },
27                  "comment_author": "Emma Cochrane",
28                  "comment_description": "There's the sublime beauty of a forgotten world and the promise of happily-ever-after to draw you to one
29                  of your favourite fairy tales, once again. Give it an encore."
30              }
31          },
32          {
33              "_index": "posts-comments",
34              "_id": "D",
35              "_score": 1.0,
36              "_routing": "2",
37              "_source": {
38                  "_join_type": {
39                      "name": "comment",
40                      "parent": "2"
41                  },
42                  "comment_author": "Common Sense Media Editors",
43                  "comment_description": "Stellar music, brisk storytelling, delightful animation, and compelling characters make this both a great
44                  animated feature for kids and a great movie for anyone"
45              }
46          }
47      ]

```

4. Suggesting

1. Run a "suggest" request on "description" field. Provide a text (or a single word) which is supposed to exist in the description field with 1-2 mistakes.

See if suggester returns the original text/word. Change number of mistakes and observe the result.

with small amount of mistakes:

GET http://localhost:9200/employees/_search Send

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```

1  {
2   ...
3   "suggest": {
4     ...
5     "my-suggest": {
6       ...
7       "text": "we had discusssion politicum",
8       ...
9       "term": {
10      ...
11      "field": "description"
12      ...
13    }
14  }
15}

```

Body Cookies Headers (4) Test Results Status: 200 OK Time: 16 ms Size: 622 B Save as example ...

Pretty Raw Preview Visualize JSON

```

32   ...
33   "suggest": {
34     ...
35     "my-suggest": {
36       ...
37       "text": "discusssion",
38       ...
39       "offset": 7,
40       ...
41       "length": 11,
42       ...
43       "options": [
44         ...
45         {
46           ...
47           "text": "politicum",
48           ...
49           "offset": 19,
50           ...
51           "length": 9,
52           ...
53           "options": [
54             ...
55             {
56               ...
57               "text": "politics",
58               ...
59               "score": 0.9,
60               ...
61               "freq": 1
62             }
63           ]
64         }
65       ]
66     }
67   }

```

with big amount of mistakes:

GET http://localhost:9200/employees/_search Send

Params Authorization Headers (9) Body **JSON** Pre-request Script Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```

1  {
2   .... "suggest": {
3     .... "my-suggest": {
4       .... "text": "we had desscusssion poliiticum",
5       .... "term": {
6         .... "field": "description"
7       }
8     }
9   }
10 }
```

Body Cookies Headers (4) Test Results

Status: 200 OK Time: 45 ms Size: 543 B Save as example

Pretty Raw Preview Visualize JSON

```

25   },
26   {
27     "text": "had",
28     "offset": 3,
29     "length": 3,
30     "options": []
31   },
32   {
33     "text": "descussion",
34     "offset": 7,
35     "length": 12,
36     "options": []
37   },
38   {
39     "text": "poliiticum",
40     "offset": 20,
41     "length": 10,
42     "options": []
43   }
44 ]
45 }
```