

# 06 - Mapping

---

## Datové typy

---

Každé pole indexu má určitý datový typ, přičemž:

- Datový typ pole je neměnný v celém indexu - není možné uložit kód produktu jednou jako číslo a jednou jako string (druhá operace skončí chybou).
- Datový typ se detekuje automaticky na základě ukládaného dokumentu (pokud ještě pole v indexu není).
- Datový typ pole není možné změnit, jediným řešením je vytvoření nového indexu a následná reindexace.

Nejpoužívanější datové typy, se kterými se budeme setkávat:

- `keyword`: Text, ve kterém se nebude fulltextově vyhledávat - optimalizováno pro filtrování a agregace
- `text`: Text určený pro fulltextové vyhledávání
- `long`, `integer`: Celé číslo
- `double`, `float`: Desetinné číslo
- `date`: Pole pro uložení data nebo času. Pokud chceme použít jiné, než předdefinované formátování ( `2019-01-01`, `2019-01-01T12:10:30Z` nebo `1420070400001` ), je třeba ho uvést při vytváření pole.
- `boolean`: Hodnoty `true` a `false`
- `object`, `nested`, `join`: Pomocí speciálních datových typů je možné uložit do Elasticsearch relaci mezi dokumenty
- `geo_point`, `geo_shape`: Uložení geografických dat
- `array`: Pokud chceme uložit pole, musí být všechny prvky stejného typu. Jako typ se neuvádí array, ale typ jednoho prvku pole.

Veškeré podporované datové typy jsou specifikovány v [dokumentaci](#).

## Mapping

---

Při ukládání dokumentů do Elasticsearch zároveň vznikají metadata obsahující informace o tom, jaké pole má jaký datový typ. Tomu se říká mapping a zobrazit si jej lze pomocí endpointu `_mapping`:

POST product/\_doc

```
{
  "code": "APP549396996",
  "name": "iPhone 13 128GB Midnight",
  "description": "iPhone 13. The most advanced dual-camera system ever on iPhone.
Lightning-fast A15 Bionic chip. A big leap in battery life. Durable design. Superfast
5G. 1 And a brighter Super Retina XDR display.",
  "price": 799.99,
  "category": "Electronics",
  "stock": true
}
```

GET product

The screenshot shows a web application interface with a top navigation bar containing 'Console', 'Search Profiler', 'Grok Debugger', 'Painless Lab', and a 'BETA' badge. Below the navigation bar, there are tabs for 'History', 'Settings', and 'Help'. The 'History' tab is active, showing a list of requests. The first request is a POST to 'product/\_doc' with a JSON body. The second request is a GET to 'product'. The GET request is selected, and its response is displayed on the right. The response is a JSON object with a 'product' field containing a mapping of the document's fields to their types. A red box highlights the 'product' field in the response, and a red arrow points from the GET request in the history to this box. The response status is '200 - success' and the response time is '78 ms'.

```
1 POST product/_doc
2 {
3   "code": "APP549396996",
4   "name": "iPhone 13 128GB Midnight",
5   "description": "iPhone 13. The most advanced dual-camera system ever on iPhone.
6     Lightning-fast A15 Bionic chip. A big leap in battery life. Durable design. Superfast
7     5G. 1 And a brighter Super Retina XDR display.",
8   "price": 799.99,
9   "category": "Electronics",
10  "stock": true
11 }
12 GET product

1 {
2   "product": {
3     "aliases": { },
4     "mappings": {
5       "properties": {
6         "category": {
7           "type": "text",
8           "fields": {
9             "keyword": {
10              "type": "keyword",
11              "ignore_above": 256
12            }
13           }
14         },
15         "code": {
16           "type": "text",
17           "fields": {
18             "keyword": {
19              "type": "keyword",
20              "ignore_above": 256
21            }
22           }
23         },
24         "description": {
25           "type": "text",
26           "fields": {
27             "keyword": {
28              "type": "keyword",
29              "ignore_above": 256
30            }
31           }
32         },
33         "name": {
34           "type": "text",
35           "fields": {
36             "keyword": {
37              "type": "keyword",
38              "ignore_above": 256
39            }
40           }
41         },
42         "price": {
43           "type": "float"
44         },
45         "stock": {
46           "type": "boolean"
47         }
48       }
49     },
50     "settings": {
51       "index": {
52         "routing": {
53           "allocation": {
54             "include": {
55               "_tier_preference": "data_content"
56             }
57           }
58         },
59         "number_of_shards": "1",
60         "provided_name": "product",
61         "creation_date": "1644748959634",
```

V příkladu výše je mapování vytvořeno automaticky na základě ukládaných dat.

Nicméně ne vždy je Elasticsearch schopný správně detekovat datové typy. Například pole `code` detekované jako `text` nebude pravděpodobně využito pro fulltextové hledání, takže by vhodným type bylo spíše `keyword`. V tomto případě je nutné mapování definovat předem. Vytvořme tedy nový prázdný index s správným mapováním:

```
PUT product-2
{
  "mappings": {
    "properties": {
      "code": {                // <= název pole
        "type": "keyword"    // <= datový typ
      }
    }
  },
  // volitelně lze také uvést nastavení indexu:
  "settings": {
    "number_of_replicas": 0
  }
}
```

Následně lze přidat dokument do tohoto indexu:

```
POST product-2/_doc
{
  "code": "APP549396996",
  "name": "iPhone 13 128GB Midnight",
  "description": "iPhone 13. The most advanced dual-camera system ever on iPhone. Lightning-fast A15 Bionic chip. A big leap in battery life. Durable design. Superfast 5G. 1 And a brighter Super Retina XDR display.",
  "price": 799.99,
  "category": "Electronics",
  "stock": true
}
```

Nyní při zobrazení mapování pomocí endpointu `_mapping`:

D

Dev Tools

Console

Search Profiler

Grok Debugger

Painless Lab

BETA

History

Settings

Help

200 - success70 ms

```

1 PUT product-2
2 {
3   "mappings": {
4     "properties": {
5       "code": {
6         "type": "keyword"
7       }
8     }
9   }
10 }
11
12 POST product-2/_doc
13 {
14   "code": "APP549396996",
15   "name": "iPhone 13 128GB Midnight",
16   "description": "iPhone 13. The most advanced dual-camera system ever on iPhone.
17     Lightning-fast A15 Bionic chip. A big leap in battery life. Durable design.
18     Superfast 5G. 1 And a brighter Super Retina XDR display.",
19   "price": 799.99,
20   "category": "Electronics",
21   "stock": true
22 }
23
24 GET product-2/_mapping

```

```

1 {
2   "product-2": {
3     "mappings": {
4       "properties": {
5         "category": {
6           "type": "text",
7           "fields": {
8             "keyword": {
9               "type": "keyword",
10              "ignore_above": 256
11            }
12          }
13        },
14        "code": {
15          "type": "keyword"
16        },
17        "description": {
18          "type": "text",
19          "fields": {
20            "keyword": {
21              "type": "keyword",
22              "ignore_above": 256
23            }
24          }
25        },
26        "name": {
27          "type": "text",
28          "fields": {
29            "keyword": {
30              "type": "keyword",
31              "ignore_above": 256
32            }
33          }
34        },
35        "price": {
36          "type": "float"
37        },
38        "stock": {
39          "type": "boolean"
40        }
41      }
42    }
43  }
44 }

```

Nová pole mohou být do mapování přidána pomocí endpointu `_mapping`. Nová pole budou dostupná pod klíčem `properties`:

```

PUT product-2/_mapping
{
  "properties": {
    "rating": {
      "type": "half_float"
    }
  }
}

```

Dev Tools

Console Search Profiler Grok Debugger Painless Lab BETA

History Settings Help 200 - success 73 ms

```
1 PUT product-2
2 {
3   "mappings": {
4     "properties": {
5       "code": {
6         "type": "keyword"
7       }
8     }
9   }
10 }
11
12 POST product-2/_doc
13 {
14   "code": "APP549396996",
15   "name": "iPhone 13 128GB Midnight",
16   "description": "iPhone 13. The most advanced dual-camera system ever on iPhone.
17   Lightning-fast A15 Bionic chip. A big leap in battery life. Durable design.
18   Superfast 5G. 1 And a brighter Super Retina XDR display.",
19   "price": 799.99,
20   "category": "Electronics",
21   "stock": true
22 }
23
24 GET product-2/_mapping
25
26 PUT product-2/_mapping
27 {
28   "properties": {
29     "rating": {
30       "type": "half_float"
31     }
32   }
33 }
34
35 GET product-2/_mapping
```

```
1 {
2   "product-2": {
3     "mappings": {
4       "properties": {
5         "category": {
6           "type": "text",
7           "fields": {
8             "keyword": {
9               "type": "keyword",
10              "ignore_above": 256
11            }
12          }
13        },
14        "code": {
15          "type": "keyword"
16        },
17        "description": {
18          "type": "text",
19          "fields": {
20            "keyword": {
21              "type": "keyword",
22              "ignore_above": 256
23            }
24          }
25        },
26        "name": {
27          "type": "text",
28          "fields": {
29            "keyword": {
30              "type": "keyword",
31              "ignore_above": 256
32            }
33          }
34        },
35        "price": {
36          "type": "float"
37        },
38        "rating": {
39          "type": "half_float"
40        },
41        "stock": {
42          "type": "boolean"
43        }
44      }
45    }
46  }
47 }
```

Při ukládání dokumentů do Elasticsearch musí všechna pole dodržovat existující nastavení. V příkladu výše jsme nastavili pole `rating` jako číslo pomocí typu `half_float`. Pokud se ale pokusíme uložit text do pole `rating`:

```
POST product-2/_doc
{
  "name": "OEM 4K HDMI Cable for TV",
  "rating": "UNKNOWN"
}
```

Dev Tools

Console Search Profiler Grok Debugger Painless Lab BETA

History Settings Help 400 - error 86 ms

```
1 POST product-2/_doc
2 {
3   "name": "OEM 4K HDMI Cable for TV",
4   "rating": "UNKNOWN"
5 }
```

```
1 {
2   "error": {
3     "root_cause": [
4       {
5         "type": "mapper_parsing_exception",
6         "reason": "failed to parse field [rating] of type [half_float] in
7         document with id 'LPe88n4BW4jVxzZGj_W6'. Preview of field's value:
8         'UNKNOWN'"
9       }
10      ],
11      "type": "mapper_parsing_exception",
12      "reason": "failed to parse field [rating] of type [half_float] in document
13      with id 'LPe88n4BW4jVxzZGj_W6'. Preview of field's value: 'UNKNOWN'",
14      "caused_by": {
15        "type": "number_format_exception",
16        "reason": "For input string: '\\UNKNOWN\\'"
17      }
18    },
19    "status": 400
20  }
21 }
```

Request skončí s chybou.

# Mapování pole více způsoby

Občas je nutné jedno pole ukládat více způsoby.

Například si představme výrobce produkt. Ten bude využit pro fulltextové vyhledávání (takže bude typu `text`). Ale v případě, že bychom chtěli vypsát všechny dostupné výrobce pomocí agregací (které spolupracují s neanalyzovanými poli), bude třeba použít typ `keyword`.

V tomto případě by mapování vypadalo následovně:

```
PUT product-3
{
  "mappings": {
    "properties": {
      "name": {
        "type": "text",
        "fields": {
          "keyword": {
            "type": "keyword"
          }
        }
      }
    }
  }
}
```

## Dynamic templates

Pokud nechcete specifikovat mapování pro každé pole zvlášť, lze využít `dynamic_templates`.

Takto by například vypadalo pravidlo, které vždy uloží string jako `keyword`:

```
PUT my_logs-2022.02.09
{
  "mappings": {
    "dynamic_templates": [
      {
        "strings": {
          "match_mapping_type": "string",
          "mapping": {
            "type": "keyword"
          }
        }
      }
    ]
  }
}
```

# Index templates

Šablony Elasticsearch obecně umožňují nastavit pravidla, která se použijí při splnění určitých podmínek. Například **index templates** definují nastavení indexů a mapování pro nově vytvořené indexy (pokud jejich názvy odpovídají vzoru).

To je užitečné při vytváření denních indexů se stejným nastavením (což je typické pro Logstash).

Řekněme, že každý nově vytvořený index s názvem začínajícím na `test` by neměl mít žádné replika shards, protože není určen pro produkční použití. Toho dosáhneme vytvořením šablony:

```
PUT _index_template/test_no_replicas
{
  "index_patterns": ["test*"],
  "template": {
    "settings": {
      "number_of_replicas": 0
    }
  }
}
```

Nyní, pokud vytvoříte index `test_123`, neměl by mít žádné repliky. Ale pokud vytvoříte index `123_test` Měl by mít defaultní nastavení.

The screenshot shows the Elasticsearch Dev Tools interface. On the left, the console history shows the following commands:

```
1 PUT _index_template/test_no_replicas
2 {
3   "index_patterns": ["test*"],
4   "template": {
5     "settings": {
6       "number_of_replicas": 0
7     }
8   }
9 }
10
11 PUT test_123
12
13 GET test_123
```

A red arrow points to the `"number_of_replicas": 0` line in the first command.

On the right, the response for the `GET test_123` command is displayed, showing the index settings:

```
1 {
2   "test_123": {
3     "aliases": { },
4     "mappings": { },
5     "settings": {
6       "index": {
7         "routing": {
8           "allocation": {
9             "include": {
10              "_tier_preference": "data_content"
11            }
12          }
13        },
14        "number_of_shards": "1",
15        "provided_name": "test_123",
16        "creation_date": "1645311128091",
17        "number_of_replicas": "0",
18        "uuid": "bM4vufPMTAq-Cp_XUIFv1A",
19        "version": {
20          "created": "8000099"
21        }
22      }
23    }
24  }
25 }
```

A red arrow points to the `"number_of_replicas": "0"` line in the response.

Index templates mohou být také vytvořeny v Kibana UI: Stack Management > Index Management > Index Templates. Po kliknutí na `Create template` se zobrazí průvodce:

Stack Management

Index Management

Management

Ingest

Data

Alerts and Insights

Indices

Data Streams

Index Templates

Component Templates

Use composable index templates to automatically apply settings, mappings, and aliases to indices.

View 1

Reload

Search...

Create template

Name	Index patterns	Components	Data stream	Content	Actions
ilm-history	ilm-history-5*			M S A	
logs	logs-*	logs-mappings, data-strea		None	
logs-apm.app	logs-apm.app*	logs-apm.app@mappings,		M S A	
logs-apm.error	logs-apm.error*	logs-apm.error@mappings,		M S A	
logs-app_search.analytics	logs-app_search.analytics	logs-mappings, data-strea		M S A	
logs-app_search.search_r	logs-app_search.search_r	logs-mappings, data-strea		M S A	

Zvolte název a index pattern:

Stack Management

Index Management

Templates

Create template

Management

Ingest

Data

Alerts and Insights

Security

Kibana

Logistics

Component templates

Index settings

Mappings

Aliases

Review template

Logistics

Name

Index patterns

Data stream

Priority

Version

\_meta field

Name

test\_no\_replicas

Index patterns

test\*

Create data stream

Priority (optional)

Version (optional)

Add metadata

Next

Pokud chcete použít "Component templates" (znovupoužitelné bloky šablon), můžete je vybrat v druhém kroku.

Ve třetím kroku uveďte JSON reprezentující nastavení indexu:



Stack Management Index Management Templates Create template

## Create template

Logistics Component templates Index settings Mappings Aliases Review template

### Index settings (optional)

Define the behavior of your indices.

Index settings

```
{
  "number_of_replicas": 0
}
```

Use JSON format: {"number\_of\_replicas":1}

< Back Next > Preview index template

Ve čtvrtém kroku lze definovat mappings. Například pro pole `uuid`, které bude typu `keyword`:

Stack Management Index Management Templates Create template

## Create template

Logistics Component templates Index settings Mappings Aliases Review template

### Mappings (optional)

Define how to store and index documents.

Mapped fields Runtime fields Dynamic templates Advanced options

Define the fields for your indexed documents. [Learn more.](#)

Search fields

Field name Field type

uuid Keyword

Keyword documentation

Add field

Add field

< Back Next > Preview index template

V pátém kroku lze indexu přiřadit alias

V posledním kroku si můžete zkontrolovat request, který bude následně spuštěn:

Stack Management

Index Management

Templates

Create template

Management

Ingest 

Ingest Pipelines

Logstash Pipelines

Data 

Index Management

Index Lifecycle Policies

Snapshot and Restore

Rollup Jobs

Transforms

Cross-Cluster Replication

Remote Clusters

Alerts and Insights 

Rules and Connectors

Reporting

Machine Learning Jobs

Watcher

Security 

Users

Roles

API keys

Role Mappings

Kibana 

Data Views

Saved Objects

Tags

Search Sessions

Spaces

Advanced Settings

Create template

Logistics

Component templates

Index settings

Mappings

Aliases

Review template

Review details for 'test\_no\_replicas'

Summary

Preview

Request

This request will create the following index template.

```
PUT _index_template/test_no_replicas
{
  "template": {
    "settings": {
      "number_of_replicas": 0
    },
    "mappings": {
      "dynamic": true,
      "numeric_detection": false,
      "date_detection": true,
      "dynamic_date_formats": [
        "strict_date_optional_time",
        "yyyy/MM/dd HH:mm:ss Z||yyyy/MM/dd Z"
      ],
      "_source": {
        "enabled": true,
        "includes": [],
        "excludes": []
      },
      "_routing": {
        "required": false
      },
      "dynamic_templates": [],
      "properties": {
        "uuid": {
          "type": "keyword"
        }
      }
    },
    "index_patterns": [
      "test*"
    ]
  }
}
```

< Back

Create template

## Úkol: Mappings

1. Vytvořte index template nazvanou `comments`, která bude používat následující mappings pro index nazvaný `comments` (index bude vytvořen v následujícím kroku):

Field	Type
title	text
text	text
author	text + keyword
email	keyword
visible	boolean
score	decimal number
created	date accepting only format <code>yyyy-MM-dd</code>

2. Uložte následující dokument do indexu `comments`:

```
{
  "title": "Test comment",
  "text": "Hello world!",
  "author": "Hicham Sharp",
  "email": "hicham@sharp-family.zzz",
  "visible": true,
  "score": 12.7,
  "created": "2022-02-18"
}
```

3. Nyní zkontrolujte mapování indexu `comments`. Mělo by odpovídat požadavkům z prvního kroku.