

# 03 - Cluster, SQL, CRUD operace

## Kibana

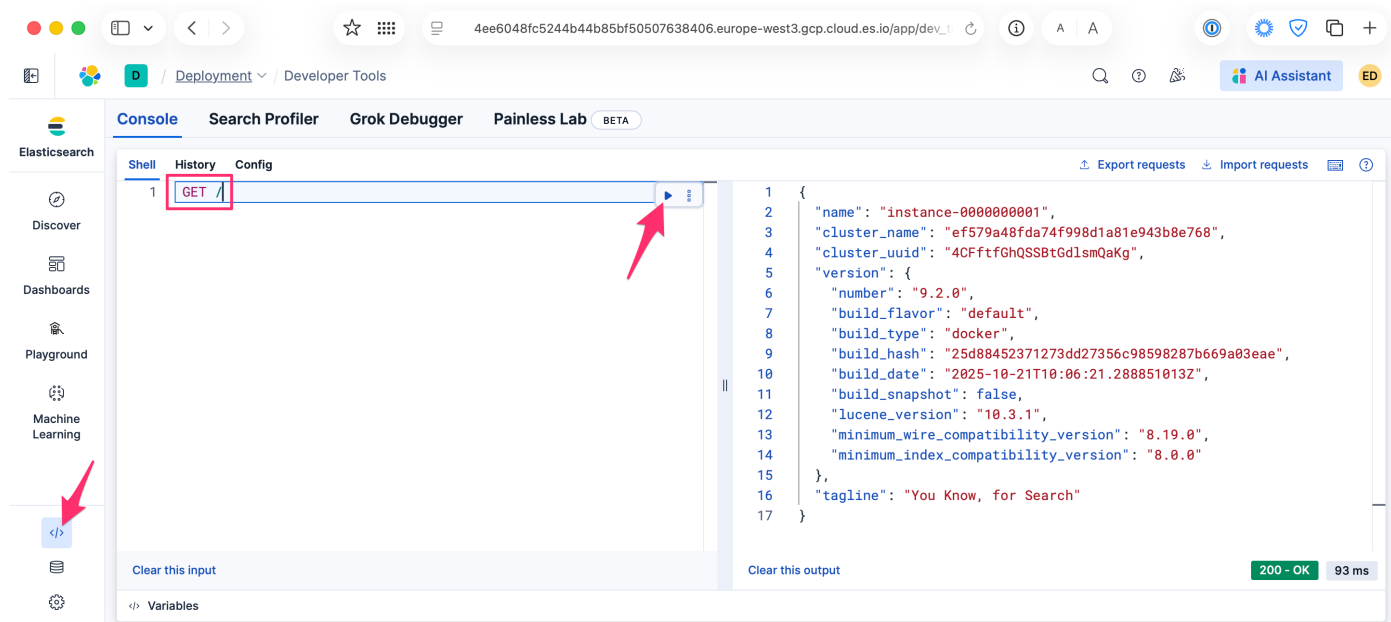
Pokud bychom měli Elasticsearch spuštěný lokálně, nejjednodušší request by vypadal následovně:

```
GET http://localhost:9200
```

Vzhledem k tomu, že budeme ke spuštění dotazů používat Kibanu, není nutné zadávat adresu Elasticsearch. Ta je již nastavena v konfiguraci Kibany. Stejný request jako výše bude mít v Kibaně následující podobu:

```
GET /
```

Dotaz spustíme šipkou vedle dotazu, výsledek se objeví v pravé části okna:



Kromě toho lze příkaz spustit klávesovou zkratkou `ctrl + enter` (respektive `cmd + enter` na Apple). Je také možné nechat kód přeformátovat kliknutím na tři tečky a možnost `Auto indent`.

## Prozkoumávání clusteru

Prvním příkazem, kterým lze zjistit stav celého clusteru je odeslání `GET` requestu na endpoint `_cat/health?` (počáteční lomítko můžeme z dotazu vynechat). Zjistíte tak název a status clusteru (tedy stav všech shardů v clusteru). Pokud je vše v pořádku, status je `green`. Pokud existují shardy, které není možné přiřadit na žádný node (například pokud lokálně nastavíte počet replik větší než 1), bude status `yellow`. Pokud nejsou některá data vůbec dostupná, bude status `red`.

epoch	timestamp	cluster	status	node.total	node.data	shards	pri	relo	init	unassign	pending_tasks	max_task_wait_time	active_shards_percent
1	1638662725	00:05:25	86d0f9893a9c4f1089eaf8588a5262	green	3	2	180	90	0	0	0	0	100.0%

Alternativně lze stav clusteru vypsat i v formátu JSON: `GET _cluster/health`

Cluster se skládá z nodů, vypsat je lze příkazem `GET _cat/nodes?v`:

ip	heap.percent	ram.percent	cpu	load_1m	load_5m	load_15m	node.role	master	name
10.45.255.79	53	65	1	2.80	2.82	2.50	himrst	-	instance-0000000000
10.45.255.60	21	92	0	1.62	1.96	1.68	mv	-	tiebreaker-0000000002
10.45.255.24	37	66	2	1.41	1.58	1.75	himrst	*	instance-0000000001

Dále budeme pracovat s **indexy**, jejichž seznam je možné získat requestem `GET _cat/indices?v`. V cloudu jsou předvytvořené především systémové indexy začínající tečkou. Je zde také index s vzorovými daty:

health	status	index	uuid	pri	rep	docs.count	docs.deleted	store.size	pri.store.size
green	open	.ent-search-actastic-workplace_search_accounts_v16	Ggs76PX1QnqUH9jIBYWMRQ	1	1	1	0	12.1kb	6kb
green	open	.ent-search-actastic-ouch_access_tokens-refresh-token-unique-constraint	h6M0Fv00RDyFRC24y_1lQ	1	1	0	0	416b	208b
green	open	.ent-search-actastic-app_search_crawler_content_metadata-content_hash-engine-old-unique-constraint	4dND5bA1YlCF-j5U1p12KA	1	1	0	0	416b	208b
green	open	.ent-search-ab-lock-28200304	1CAY0E1cQF53LulUp12KA	1	1	0	0	75.7kb	19.6kb
green	open	.kibana-event-log-7.15.2-000001	7NA-vrkDRucpd2ZucwM0CA	1	1	1	0	12.1kb	6kb
green	open	.ent-search-actastic-ouch_access_tokens-token-unique-constraint	pCCZML5Stof1FvDUaCQR	1	1	1	0	7.3kb	3.6kb
green	open	.ent-search-actastic-workplace_search_search_groups_v4-name-unique-constraint	AdQnL2Zb57r86Sot1R653w	1	1	0	0	7kb	3.5kb
green	open	.ent-search-actastic-synonyms	7LvFAPHC66awo6mHv8YA	1	1	0	0	416b	208b
green	open	.ent-search-actastic-document_types_v2	oMK_NC3XTG-F8x7o0HTLIQ	1	1	0	0	416b	208b
green	open	.ent-search-actastic-reindex_jobs	FaxTo8K8PheJASRLHCqLw	1	1	0	0	416b	208b
green	open	.ent-search-actastic-app_search_accounts_v9-key-unique-constraint	SL3ccYFYTolVfWQ5Qjrhg	1	1	0	0	7kb	3.5kb
green	open	.ent-search-actastic-crawler_crawl_requests_v4	m5948e6v6m0Z0ACylEbt8w	1	1	0	0	416b	208b
green	open	.ent-search-actastic-workplace_search_content_source_user_identities_v3	85byLZKJ8R6GvY9JW1L35Q	1	1	0	0	416b	208b
green	open	.kibana_task_manager_7.15.2-000001	XP4ALZ16bvehH7J0P1Pw	1	16	26	654.2kb	330.3kb	330.3kb
green	open	.ent-search-actastic-workplace_search_role_mappings_v5	jJ6jY_1sQ5C-xoi-pgdz8RA	1	1	0	0	416b	208b
green	open	.opm-custom-link	12Ln-DyXQ5KzU6vuvuVMQZ	1	1	0	0	416b	208b
green	open	.ent-search-actastic-document_types_v2-engine-old-slug-unique-constraint	Gt4eHl1s65fmx7gBK35M9kg	1	1	0	0	416b	208b
green	open	.opm-7.15.2-span-000001	8L1H9d55Q8e-1TCA7vEdA	1	1	0	0	416b	208b
green	open	.ent-search-actastic-connectors_jobs_v5	v-mVvd7HTTep85gdpdpw	1	1	0	0	416b	208b
green	open	.ent-search-actastic-telemetry_status_v2	0HL0o8HV5m9nQz0et08a	1	1	0	0	416b	208b
green	open	.kibana_7.15.2-0001	0201118yKCy11s8Cqemng	1	1685	19	15.1mb	5mb	5mb
green	open	.ent-search-actastic-actastic-status_v2	3p-gLzLImay15HfzfsiGA	1	1	0	0	8.2kb	4.1kb
green	open	.ent-search-actastic-workplace_search_content_sources_v21	bFLv8wAHRayiut1d5Gfx1Q	1	1	0	0	416b	208b
green	open	.ent-search-actastic-togo_migrations_v1	6qK8bopCR-5ERb1W7a13yQ	1	1	95	0	23.5kb	11.7kb
green	open	.fleet-enrollent-api-keys-7	x1x0a0WQ_27tp1Q87rVQ	1	1	2	0	23.3kb	11.6kb
green	open	.opm-agent-configuration	VX1YncmqY6C1p1M1L1kg	1	1	0	0	416b	208b
green	open	.ent-search-actastic-crawler_process_crawls	DYbATcY2SHGLdCjogHmqw	1	1	0	0	416b	208b
green	open	.ent-search-actastic-app_search_crawler_content_metadata	13k1VW853kqWm05Poc6A	1	1	0	0	416b	208b
green	open	.ent-search-actastic-app_search_api_tokens_v3-authentication_token-unique-constraint	1Hb4Lss5Zerrrrr1m1Y5Q	1	1	0	0	416b	208b
green	open	.ent-search-actastic-workplace_search_organizations_v16	yRQYksoqC1K8D8Cq5Q	1	1	1	0	508kb	24.5kb
green	open	.ent-search-actastic-app_search_role_mapping_engines_v3-engine-old-loc-to-go_role_mapping_id-unique-constraint	LE8aTFvUuSe6y2h1FrEg	1	1	0	0	416b	208b
green	open	.ent-search-actastic-crawler_domains_v4-engine-old-name-unique-constraint	gaVyyeg5IPWGa_1JqKcw	1	1	0	0	416b	208b
green	open	.ent-search-actastic-ecs-ilm-logs-production-2021.12.04-000001	1H85-VYVT_6v5X1Aw1D1g	1	1	0	0	416b	208b
green	open	.ent-search-actastic-ecs-ilm-logs-production-2021.12.04-000001	tIDm_U1RQz112v_9dk3Q	1	1	1	0	27.8kb	13.9kb
green	open	.ent-search-actastic-app_search_crawler_content_url_metadata	e18Z2LAIQn6dHosq18-w	1	1	0	0	416b	208b
green	open	.ent-search-esqueues-me_queue_v1-process_crawl	MaxLu1XQZGjJW8K18Fong	1	1	0	0	416b	208b
green	open	.ent-search-actastic-users_v6	0WNRJK_oT1mFBycc9MLFTw	1	1	1	0	16kb	8kb
green	open	.ent-search-esqueues-me_worker_v1	L9xkryZg080q7VEJmPeyA	1	1	0	0	37.2kb	14.7kb
green	open	.ent-search-actastic-app_search_roles_v2	1H85-VYVT_6v5X1Aw1D1g	1	1	0	0	416b	208b
green	open	.kibana_sample_data_logs	8186d015SasQP1m1K3Kdg	1	1	14074	0	17.5mb	8.8mb
green	open	.ent-search-workplace-search-analytics-ecs-ilm-logs-production-2021.12.04-000001	CPDRKZ11-1KXQZvU8m9KA	1	1	0	0	416b	208b
green	open	.ent-search-workplace-search-analytics-ecs-ilm-logs-production-2021.12.04-000001	07ubk108F1C1U67E7cADn	1	1	0	0	416b	208b

Endpointy `_cat` jsou užitečné při zkoumání stavu clusteru. Pokud byste je ale chtěli použít ve skriptech nebo automatizaci, bude lepší použít strojově čitelné varianty vracějící JSON, jako `_cluster/health`, `_cluster/stats`, `_nodes` nebo `_nodes/stats`.

# SQL

Přestože je primárním způsobem komunikace s Elasticsearch odesílání JSON dokumentů pomocí REST rozhraní, je možné použít i SQL. Jeho možnosti jsou v Elasticsearch omezené, pokud ale neznáte Elasticsearch query language, může být právě SQL rychlým způsobem, jak s Elasticsearch.

K dispozici je [CLI nástroj](#), který ve staženém Elasticsearch spustíte příkazem:

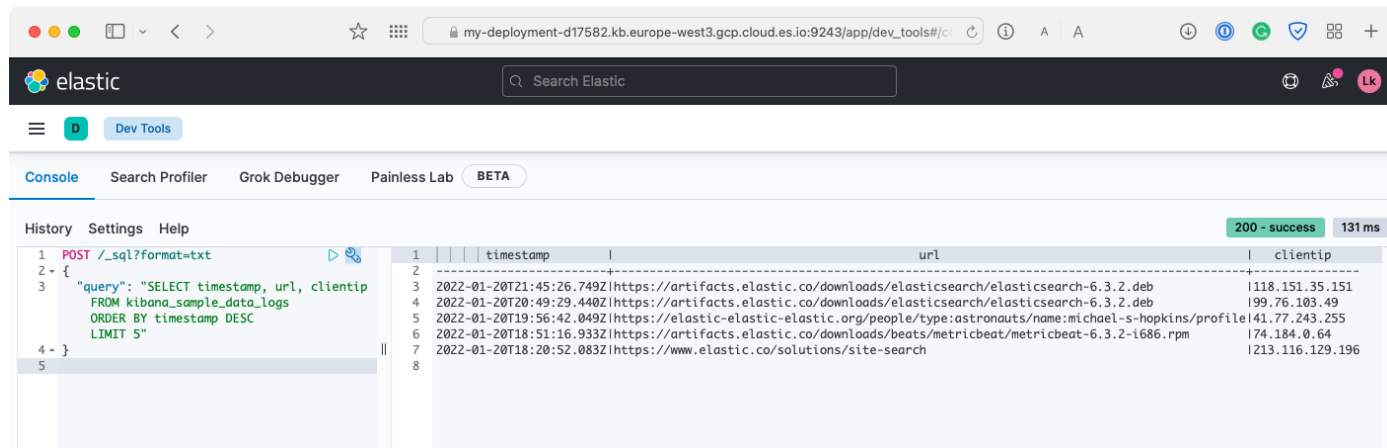
```
/bin/elasticsearch-sql-cli
```

Následně se spustí konzole, do které je možné SQL příkazy zadávat.

V rámci Kibany je nutné SQL příkaz zabalit do JSON dokumentu:

```
POST /_sql?format=txt
{
  "query": "SELECT timestamp, url, clientip FROM kibana_sample_data_logs ORDER BY
timestamp DESC LIMIT 5"
}
```

Při použití formátu `txt` se tento výsledek vyhledávání ve vzorových datech zobrazí v tabulce:



The screenshot shows the Elastic Dev Tools interface. The console tab is active, displaying a POST request to `/_sql?format=txt` with a JSON body containing a SQL query. The response is a 200 status code, indicating success, and the result is displayed in a table format. The table has three columns: `timestamp`, `url`, and `clientip`. The results are ordered by `timestamp` in descending order, limited to 5 rows.

timestamp	url	clientip
2022-01-20T21:45:26.749Z	https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-6.3.2.deb	118.151.35.151
2022-01-20T20:49:29.440Z	https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-6.3.2.deb	199.76.103.49
2022-01-20T19:56:42.049Z	https://elastic-elastic-elastic.org/people/type:astronauts/name:michael-s-hopkins/profile	141.77.243.255
2022-01-20T18:51:16.933Z	https://artifacts.elastic.co/downloads/beats/metricbeat/metricbeat-6.3.2-i686.rpm	174.184.0.64
2022-01-20T18:20:52.083Z	https://www.elastic.co/solutions/site-search	1213.116.129.196

Podpora SQL umožňuje Elasticsearch napojit (pomocí JDBC respektive ODBC knihovny) na další nástroje, například MS Excel nebo Tableau. Dá se také využít k učení se Elasticsearch query language. Každý SQL dotaz je totiž interně transformován na Elasticsearch query a tu si můžeme nechat zobrazit:

```
POST /_sql/translate
{
  "query": "SELECT timestamp, url, clientip FROM kibana_sample_data_logs ORDER BY
timestamp DESC LIMIT 5"
}
```

Použití SQL má však své limity, především je použitelné jen pro získávání dat, ne pro jejich změnu. Dále je problematické použití s některými datovými typy, především s vnořenými objekty a poli.

# Vytvoření dokumentu

Nový dokument do Elasticsearch uložíme následujícím příkazem:

```
PUT user/_doc/1
{
  "name": "Oliver Johnson"
}
```

V tomto requestu `user` značí název indexu, do kterého bude dokument uložen, `1` je jednoznačný identifikátor dokumentu a `{"name": "Oliver Johnson"}` je samotný dokument.

Všimněte si, že jsme nikde nedefinovaly, jaká pole bude dokument obsahovat. To je rozdíl oproti RDBMS, kde je nutné všechny sloupce tabulky předem definovat. V Elasticsearch můžeme rovnou dokument vytvořit, není třeba předem definovat, jaké pole bude obsahovat.

Pokud bychom tento dokument uložili znovu, původní se celý přepíše. Pokud neznáme ID, můžeme nechat Elasticsearch vygenerovat UUID, které dokumentu přidělí, pak také nikdy nedojde k přepsání stávajícího dokumentu:

```
POST user/_doc
{
  "name": "James Williams"
}
```

Druhý přístup se v praxi využívá u dat, která už zpravidla nebude třeba nijak modifikovat, nebo ID vůbec nevíme - například u logů. V Kibaně se v tomto případě setkáte s pojmenováním `Data Streams`. Používat ID má naopak smysl například při ukládání produktů nebo zákazníků.

V případě, že by dosud index `user` neexistoval, automaticky se vytvoří. A pokud existuje šablona odpovídající názvu nového indexu, převezme z ní nově vytvářený index nastavení.

Po vytvoření dokumentu obdržíme response, která obsahuje výsledek, zda se dokument podařilo uložit, do jakého indexu a s jakým ID. Zároveň je vidět jeho verze, tedy zda došlo k prvnímu uložení dokumentu s daným ID, nějakému dalšímu.

The screenshot shows the Elastic Dev Tools interface. The 'Console' tab is active, displaying a list of requests. The first request is a `PUT user/_doc/1` with a JSON body: `{ "name": "Oliver Johnson" }`. The response is a 201 status code, indicating success, with a response time of 77 ms. The response body is a JSON object: `{ "_index": "user", "_type": "_doc", "_id": "hTh1wX4BIC3Q38hFRWM4", "_version": 1, "result": "created", "_shards": { "total": 2, "successful": 2, "failed": 0 }, "_seq_no": 1, "_primary_term": 1 }`.

## Získání dokumentu

Pokud známe index a ID dokumentu, je možné jej dohledat pomocí příkazu:

```
GET user/_doc/1
```

V response je dokument dostupný pod klíčem `_source`:

The screenshot shows the Elastic Dev Tools interface. The 'Console' tab is active, displaying a list of requests. The first request is a `GET user/_doc/1`. The response is a 200 status code, indicating success, with a response time of 78 ms. The response body is a JSON object: `{ "_index": "user", "_type": "_doc", "_id": "1", "_version": 1, "_seq_no": 0, "_primary_term": 1, "found": true, "_source": { "name": "Oliver Johnson" } }`. The `_source` field is highlighted with a red box.

Přestože jsou dokumenty v Elasticsearch vyhledatelné až po nějaké době (standardně 1s), pro request s ID platí, že je dokument dostupný ihned.

Pokud chcete získat dokument přesně tak, jak byl uložen (obsah klíče `_source`), je to možné dotazem:

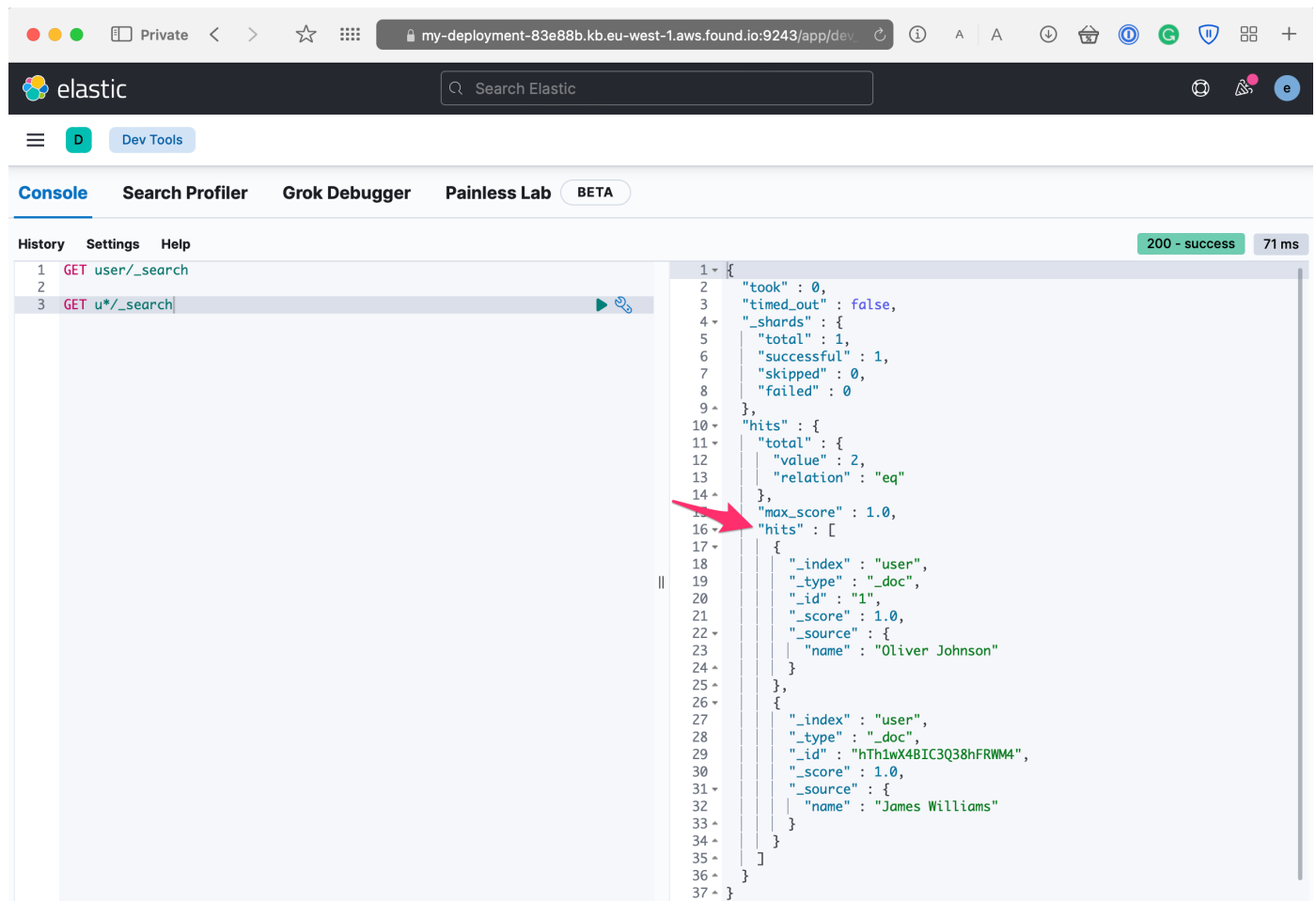
```
GET user/_source/1
```

# Výpis dokumentů

Pro zobrazení všech dokumentů v indexu slouží následující dotaz:

```
GET user/_search
```

Je možné název indexu kompletně vynechat, pak se bude vyhledávat ve všech indexech v clusteru. V názvu indexu je také možné použít `*`, což značí libovolné znaky. Pokud bychom chtěli vyhledat ve všech indexech začínajících písmenem `u`, vypadalo by to následovně:



Tento endpoint budeme používat pro vyhledávání. Narozdíl od získání dokumentu dle ID obsahuje response více informací:

- `took`: doba vykonání requestu v ms
- `timed_out`: příznak, zda se stihl request vykonat
- `_shards`: počty shradů, na kterých byl request vykonán
- `hits`: výsledek vyhledávání, který dále obsahuje:
  - `hits.total`: Celkový počet nalezených výsledků (maximálně 10.000)
  - `hits.max_score`: Skóre nejrelevantnějšího výsledku
  - `hits.hits`: Nalezené dokumenty (ve výchozím stavu prvních 10)

Počet nalezených dokumentů v `hits.hits` je ve výchozím stavu omezen na 10 záznamů. Počet v `hits.total` odpovídá celkovému počtu dokumentů, které odpovídají dotazu, což nemusí odpovídat počtu dokumentů v `hits.hits`.

Maximální počet nalezených záznamů v `hits.total` je z výkonostních důvodů omezen na 10.000. Pokud je v indexu více dokumentů a chceme znát přesný počet nalezených záznamů, je třeba upravit vyhledávací dotaz:

```
GET user/_search
{
  "track_total_hits": true
}
```

## Modifikace dokumentu

Nejjednodušším způsobem modifikace dokumentu je jeho nahrazení novější verzí. Je k tomu nutné ukládat dokument se shodným ID:

```
PUT user/_doc/3
{
  "name": "Patricia Williams"
}

PUT user/_doc/3
{
  "name": "Jennifer Miller"
}

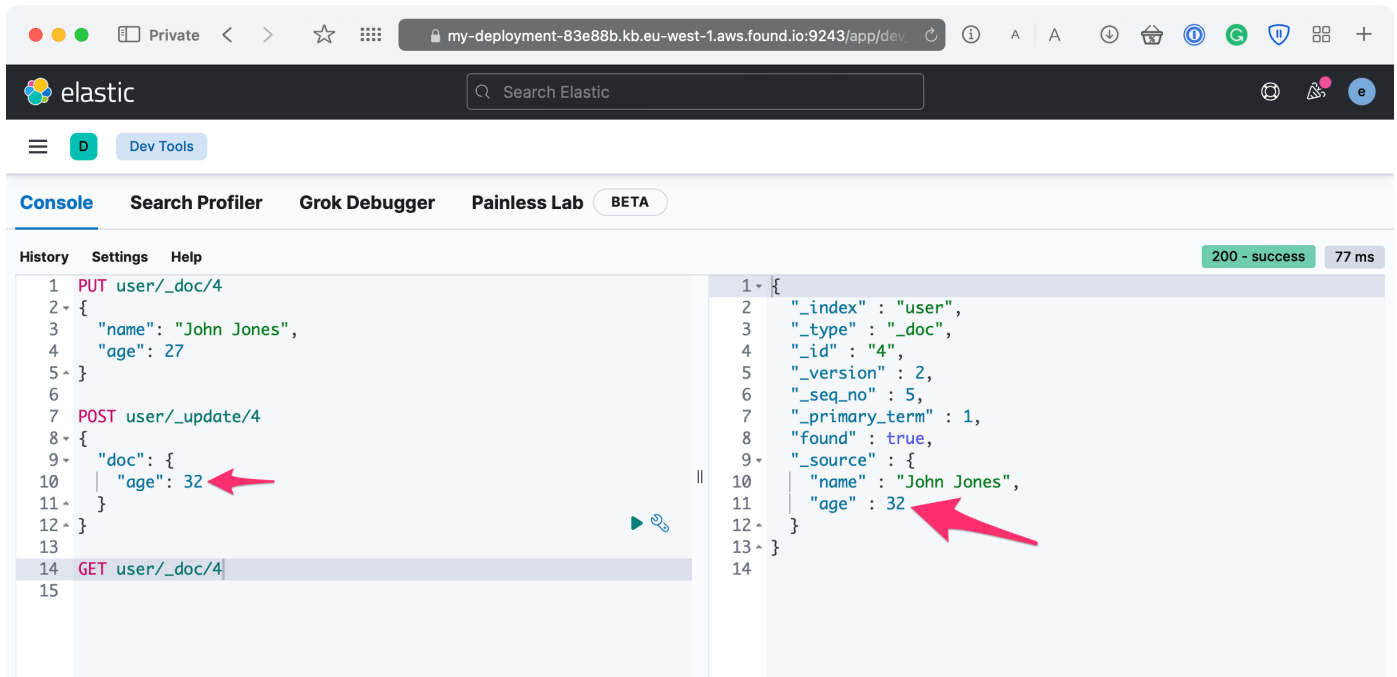
GET user/_doc/3
# "name": "Jennifer Miller"
```

Je však možné i aktualizovat pouze konkrétní pole dokumentu. Nejprve vytvoříme nový dokument:

```
PUT user/_doc/4
{
  "name": "John Jones",
  "age": 27
}
```

Pokud bychom chtěli nyní změnit pouze věk uživatele s ID `4`, bylo by to možné takto:

```
POST user/_update/4
{
  "doc": {
    "age": 32
  }
}
```



Pomocí endpointu `_update` je možné vytvořit nebo upravit pole existujícího dokumentu.

Pokud potřebujete k aktualizaci dokumentu využít stávající hodnoty (například zvýšení věku uživatele o 2), bude nutné využít skript:

```
POST user/_update/4
{
  "script" : "ctx._source.age += 2"
}
```

Samotný skript využívá jazyk [painless](#), který poskytuje proměnné, podmínky nebo cykly. Stávající hodnoty dokumentu jsou dostupné pod `ctx._source`.

## Mazání dokumentu

Dokument se maže uvedením stejné URL, jako kdybychom chtěli přidat nový dokument podle ID, pouze se použije HTTP metoda `DELETE`:

```
DELETE user/_doc/1
```

## Úkol: CRUD

1. Vytvořte následující dokumenty v indexu `subscription_list`:



```
{
  "name": "First customer",
  "e-mail": "first-customer@icloud.com",
  "age": 31
}
```

```
{
  "name": "Second customer",
  "e-mail": "second-customer@gmail.com",
  "age": 47
}
```

2. vypište všechny dokumenty v indexu `subscription_list`

History Settings Help 200 - success 81 ms

```
1 {
2   "took" : 1,
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        "_index" : "subscription_list",
19        "_type" : "_doc",
20        "_id" : "hjihwX48IC3Q38hFeG0a",
21        "_score" : 1.0,
22        "_source" : {
23          "name" : "First customer",
24          "e-mail" : "first-customer@icloud.com",
25          "age" : 31
26        }
27      },
28      {
29        "_index" : "subscription_list",
30        "_type" : "_doc",
31        "_id" : "hzihwX48IC3Q38hFhmOI",
32        "_score" : 1.0,
33        "_source" : {
34          "name" : "Second customer",
35          "e-mail" : "second-customer@gmail.com",
36          "age" : 47
37        }
38      }
39    ]
40  }
41 }
```

3. Změňte e-mailovou adresu prvního zákazníka na `first@outlook.com` a zobrazte výsledek:

```
History Settings Help 200 - success 543 ms

1 {
2   "took" : 400,
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        "_index" : "subscription_list",
19        "_type" : "_doc",
20        "_id" : "hzihwX48IC3Q38hFm0I",
21        "_score" : 1.0,
22        "_source" : {
23          "name" : "Second customer",
24          "e-mail" : "second-customer@gmail.com",
25          "age" : 47
26        }
27      },
28      {
29        "_index" : "subscription_list",
30        "_type" : "_doc",
31        "_id" : "hjiihwX48IC3Q38hFeG0a",
32        "_score" : 1.0,
33        "_source" : {
34          "name" : "First customer",
35          "e-mail" : "first@outlook.com",
36          "age" : 31
37        }
38      }
39    ]
40  }
41 }
```

4. Smažte druhého zákazníka a znovu vypiště všechny dokumenty v indexu `subscription_list`:

```
History Settings Help 200 - success 847 ms

1 {
2   "took" : 764,
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        "_index" : "subscription_list",
19        "_type" : "_doc",
20        "_id" : "hjiihwX48IC3Q38hFeG0a",
21        "_score" : 1.0,
22        "_source" : {
23          "name" : "First customer",
24          "e-mail" : "first@outlook.com",
25          "age" : 31
26        }
27      }
28    ]
29  }
30 }
31 }
```