



# TP3 Gestion d'index + CRUD

# Les Opération de lecture/écriture

- `curl -X <VERBE> '<PROTOCOL>: // <HÔTE>: <PORT> / <PATH>? <QUERY_STRING>' -d '<CORPS>'`
- Où:
- VERBE: méthode ou verbe HTTP approprié: GET, POST, PUT, HEAD ou DELETE
- PROTOCOLE: http ou https (si vous avez un proxy https devant Elasticsearch.)
- HOST: nom d'hôte de n'importe quel nœud de votre cluster Elasticsearch ou localhost pour un nœud sur votre ordinateur local.
- PORT: port exécutant le service HTTP Elasticsearch, dont la valeur par défaut est 9200.
- PATH: API Endpoint (par exemple, `_count` renverra le nombre de documents dans le cluster). Le chemin peut contenir plusieurs composants, tels que `_cluster / stats` ou `_nodes / stats / jvm`
- QUERY\_STRING: Tous les paramètres de chaîne de requête facultatifs (par exemple, "pretty" impriment la réponse JSON pour la rendre plus lisible).
- BODY: Un corps de requête encodé en JSON (si la requête en a besoin.)

# Les Opération de lecture/écriture

## \* Créer un index

```
→ Module4 curl -XPUT 'http://localhost:9200/test?pretty'
{
  "acknowledged" : true,
  "shards_acknowledged" : true,
  "index" : "test"
}
```

## Liste tous les indices

```
→ Module4 curl -XGET 'http://localhost:9200/_cat/indices?v'
health status index      uuid                                pri rep docs.count docs.deleted store.size pri.store.size
green open   .geoip_databases  08ji6kHPTiqLPreopIFaIw          1  0         40             0      38.4mb      38.4mb
yellow open   movies           2eSldP45QT-NZTY0Aj-5dw          1  1        4999             0       4.1mb       4.1mb
green open   .apm-custom-link  Fe06UTUwRyymSF9QkrtlMA         1  0          0             0       226b       226b
yellow open   test             lq00Z_WxSM-9cn8KwadSTg          1  1          0             0       226b       226b
green open   .kibana_task_manager_7.17.7_001 zoNGpeAfsU2HYM7THMpS7A         1  0         17        13761       10mb       10mb
yellow open   personne         w_FqSla9SIeyKE4CL4F7Jw          1  1          0             0       226b       226b
green open   .apm-agent-configuration GyvihWH0Q16fMpMHQxvclw          1  0          0             0       226b       226b
green open   .async-search    lMxx6lT2TsC7zX9j6bRJFw          1  0          0             0       252b       252b
green open   .kibana_7.17.7_001 0_sDsZyuRa0UbZWzZEiXug          1  0         698             9       2.4mb       2.4mb
yellow open   movies2          Q-5YxBVwRjSpYcPUSaeibw          1  1        4849             0       3.6mb       3.6mb
green open   .tasks           jTuo9JLYTtmLb3ZLeXQyug          1  0          2             0       7.8kb       7.8kb
```

```
→ Module4 curl -XPUT 'http://localhost:9200/myIndexName?pretty'
{
  "error" : {
    "root_cause" : [
      {
        "type" : "invalid_index_name_exception",
        "reason" : "Invalid index name [myIndexName], must be lowercase",
        "index_uuid" : "_na_",
        "index" : "myIndexName"
      }
    ],
    "type" : "invalid_index_name_exception",
    "reason" : "Invalid index name [myIndexName], must be lowercase",
    "index_uuid" : "_na_",
    "index" : "myIndexName"
  },
  "status" : 400
}
```

# Les Opération de lecture/écriture

## \* Supprimer un index

```
+ Module4 curl -XGET 'http://localhost:9200/_cat/indices?v'
health status index      uuid                                pri rep docs.count docs.deleted store.size pri.store.size
green open  .geoip_databases        08ji6kHPTiqLPreopIFaIw  1  0         40           0    38.4mb    38.4mb
yellow open  movies                  2eSldP45QT-NZTY0Aj-5dw  1  1        4999           0     4.1mb     4.1mb
green open  .apm-custom-link        Fe06UTUwRyymSF9Qkrt1MA  1  0           0           0     226b     226b
yellow open  test                    lq00Z_WxSM-9cn8KwadSTg  1  1           0           0     226b     226b
green open  .kibana_task_manager_7.17.7_001 zoNGpeAfsU2HYM7THMpS7A  1  0          17        13761    10mb     10mb
yellow open  personne                w_FqSla9SIeykE4Cl4F7Jw  1  1           0           0     226b     226b
green open  .apm-agent-configuration GyvihWH0Q16fMpMHQxvclw  1  0           0           0     226b     226b
green open  .async-search           lMxx6LT2TsC7zX9j6bRJFw  1  0           0           0     252b     252b
green open  .kibana_7.17.7_001      0_sDsZyuRa0UbZWzZEiXug  1  0          698           9     2.4mb     2.4mb
yellow open  movies2                  Q-5YxBVwRjSpycPUsaebw  1  1        4849           0     3.6mb     3.6mb
green open  .tasks                   jTuo9JLYTtmLb3ZLeXQyug  1  0           2           0     7.8kb     7.8kb
```

```
+ Module4 curl -XDELETE 'http://localhost:9200/test?pretty'
```

```
{
  "acknowledged" : true
}
```

```
+ Module4 curl -XGET 'http://localhost:9200/_cat/indices?v'
```

```
health status index      uuid                                pri rep docs.count docs.deleted store.size pri.store.size
green open  .geoip_databases        08ji6kHPTiqLPreopIFaIw  1  0         40           0    38.4mb    38.4mb
yellow open  movies                  2eSldP45QT-NZTY0Aj-5dw  1  1        4999           0     4.1mb     4.1mb
green open  .apm-custom-link        Fe06UTUwRyymSF9Qkrt1MA  1  0           0           0     226b     226b
green open  .kibana_task_manager_7.17.7_001 zoNGpeAfsU2HYM7THMpS7A  1  0          17        13824    9.9mb     9.9mb
yellow open  personne                w_FqSla9SIeykE4Cl4F7Jw  1  1           0           0     226b     226b
green open  .apm-agent-configuration GyvihWH0Q16fMpMHQxvclw  1  0           0           0     226b     226b
green open  .async-search           lMxx6LT2TsC7zX9j6bRJFw  1  0           0           0     252b     252b
green open  .kibana_7.17.7_001      0_sDsZyuRa0UbZWzZEiXug  1  0          698           9     2.4mb     2.4mb
yellow open  movies2                  Q-5YxBVwRjSpycPUsaebw  1  1        4849           0     3.6mb     3.6mb
green open  .tasks                   jTuo9JLYTtmLb3ZLeXQyug  1  0           2           0     7.8kb     7.8kb
```

# Les Opération de lecture/écriture

\* Ajouter un document

```
PUT /personnes/personne/1?pretty
```

```
{  
  "name": "Jean Dupont"  
}
```

```
{  
  "_index" : "personnes",  
  "_type" : "personne",  
  "_id" : "1",  
  "_version" : 1,  
  "result" : "created",  
  "_shards" : {  
    "total" : 2,  
    "successful" : 1,  
    "failed" : 0  
  },  
  "_seq_no" : 0,  
  "_primary_term" : 1  
}
```

```
POST /personnes/personne?pretty
```

```
{  
  "first_name" : "Danielle",  
  "last_name" : "Dupont",  
  "age" : 32,  
  "interests" : [ "music" ]  
}
```

```
{  
  "_index" : "personnes",  
  "_type" : "personne",  
  "_id" : "YE9scYQBhQKHToL8SADM",  
  "_version" : 1,  
  "result" : "created",  
  "_shards" : {  
    "total" : 2,  
    "successful" : 1,  
    "failed" : 0  
  },  
  "_seq_no" : 1,  
  "_primary_term" : 1  
}
```

# Les Opération de lecture/écriture

- \* Récupérer un document par identifiant

```
GET /personnes/personne/1?pretty
```

```
1  #! Elasticsearch built-in security
   be accessible to anyone. See
   -minimal-setup.html to enable
2  #! [types removal] Specifying
   endpoint instead.
3  {
4    "_index" : "personnes",
5    "_type" : "personne",
6    "_id" : "1",
7    "_version" : 1,
8    "_seq_no" : 0,
9    "_primary_term" : 1,
10   "found" : true,
11   "_source" : {
12     "name" : "Jean Dupont"
13   }
14 }
```

# Les Opération de lecture/écriture

- \* Liste tous les documents d'un index

```
→ Module4 curl -X GET http://localhost:9200/personnes/_search?pretty
{
  "took" : 0,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 2,
      "relation" : "eq"
    },
    "max_score" : 1.0,
    "hits" : [
      {
        "_index" : "personnes",
        "_type" : "personne",
        "_id" : "1",
        "_score" : 1.0,
        "_source" : {
          "name" : "Jean Dupont"
        }
      },
      {
        "_index" : "personnes",
        "_type" : "personne",
        "_id" : "YE9scYQBhQKHToL8SADM",
        "_score" : 1.0,
        "_source" : {
          "first_name" : "Danielle",
          "last_name" : "Dupont",
          "age" : 32,
          "interests" : [

```



# Les Opération de lecture/écriture

- \* compter le nombre de documents dans le cluster

```
→ Module4 curl -X GET http://localhost:9200/personnes/_count?pretty
{
  "count" : 2,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  }
}
```



# Modifier vos données elasticsearch

Mettre à jour un document avec PUT:

Supposons que nous ayons déjà indexé un document personne, la commande PUT remplace le document précédent.

```
PUT /personnes/personne/1?pretty
* {
  "name": "John Dupont"
* }
```

```
{
  "_index" : "personnes",
  "_type" : "personne",
  "_id" : "1",
  "_version" : 2,
  "result" : "updated",
  "_shards" : {
    "total" : 2,
    "successful" : 1,
    "failed" : 0
  },
  "_seq_no" : 2,
  "_primary_term" : 1
}
```

```
GET /personnes/personne/1?pretty
```

```
{
  "_index" : "personnes",
  "_type" : "personne",
  "_id" : "1",
  "_version" : 2,
  "_seq_no" : 2,
  "_primary_term" : 1,
  "found" : true,
  "_source" : {
    "name" : "John Dupont"
  }
}
```

# Modifier vos données elasticsearch

Mettre à jour un document avec POST:

pour indexer le document sans un identifiant, nous utilisons la requête POST au lieu de PUT .

RQ: Si on ne fournit pas d'identifiant, elasticsearch générera un identifiant aléatoire et l'utilisera ensuite pour indexer le document.

```
POST /personnes/personne/1/_update?pretty|
{
  "doc": {"name": "Johnny Dupont"}
}
```

```
GET personnes/_search
```

```
{
  "_index": "personnes",
  "_type": "personne",
  "_id": "1",
  "_score": 1.0,
  "_source": {
    "name": "Johnny Dupont"
  }
}
```

```
{
  "_index": "personnes",
  "_type": "personne",
  "_id": "1",
  "_version": 3,
  "result": "updated",
  "_shards": {
    "total": 2,
    "successful": 1,
    "failed": 0
  },
  "_seq_no": 4,
  "_primary_term": 1
}
```

# Modifier vos données elasticsearch

\* mettre à jour le document avec des informations supplémentaires:

```
POST /personnes/personne/1/_update?pretty
{
  "doc": {"name": "Johnny Dupont", "age": 40}
}
```

```
{
  "_index" : "personnes",
  "_type" : "personne",
  "_id" : "1",
  "_version" : 4,
  "_seq_no" : 5,
  "_primary_term" : 1,
  "found" : true,
  "_source" : {
    "name" : "Johnny Dupont",
    "age" : 40
  }
}
```

mettre à jour le document en utilisant des scripts simples:

```
POST /personnes/personne/1/_update?pretty
{
  "script": "ctx._source.age += 2"
}
```

```
{
  "_index" : "personnes",
  "_type" : "personne",
  "_id" : "1",
  "_version" : 5,
  "_seq_no" : 6,
  "_primary_term" : 1,
  "found" : true,
  "_source" : {
    "name" : "Johnny Dupont",
    "age" : 42
  }
}
```

# Supprimer vos données elasticsearch

\* Supprimer un document:

```
DELETE /personnes/personne/1?pretty
```

```
GET /personnes/personne/1?pretty|
```

```
{
  "_index" : "personnes",
  "_type" : "personne",
  "_id" : "1",
  "found" : false
}
```

# le traitement par Lot

Outre l'indexation de la mise à jour et de la suppression du document, elasticsearch fournit également la possibilité d'effectuer des opérations de lecture et d'écriture par lots à l'aide de l'API `_bulk` .

- pour mettre à jour plusieurs documents en utilisant l'API `_bulk`

```
POST /personnes/personne/_bulk?pretty
{"index":{"_id":"1"}}
{"name": "John Dupont" }
{"index":{"_id":"2"}}
{"name": "Michelle Dupont" }
```

```
{
  "took" : 4,
  "errors" : false,
  "items" : [
    {
      "index" : {
        "_index" : "personnes",
        "_type" : "personne",
        "_id" : "1",
        "version" : 11,
        "result" : "updated",
        "shards" : {
          "total" : 2,
          "successful" : 1,
          "failed" : 0
        },
        "_seq_no" : 19,
        "_primary_term" : 1,
        "status" : 200
      }
    },
    {
      "index" : {
        "_index" : "personnes",
        "_type" : "personne",
        "_id" : "2",
        "version" : 1,
        "result" : "created",
        "shards" : {
          "total" : 2,
          "successful" : 1,
          "failed" : 0
        }
      }
    }
  ]
}
```

# le traitement par Lot

- pour mettre à jour et supprimer les documents en utilisant l'API `_bulk`

```
POST /personnes/personne/_bulk?pretty
{"update":{"_id":"1"}}
{"doc": { "name": "Johnny Dupont" } }
{"delete":{"_id":"2"}}
```

```
{
  "_index" : "personnes",
  "_type" : "personne",
  "_id" : "1",
  "_score" : 1.0,
  "_source" : {
    "name" : "Johnny Dupont"
  }
}
```

```
{
  "took" : 3,
  "errors" : false,
  "items" : [
    {
      "update" : {
        "_index" : "personnes",
        "_type" : "personne",
        "_id" : "1",
        "_version" : 12,
        "result" : "updated",
        "_shards" : {
          "total" : 2,
          "successful" : 1,
          "failed" : 0
        }
      },
      "_seq_no" : 21,
      "_primary_term" : 1,
      "status" : 200
    },
    {
      "delete" : {
        "_index" : "personnes",
        "_type" : "personne",
        "_id" : "2",
        "_version" : 2,
        "result" : "deleted",
        "_shards" : {
          "total" : 2,
          "successful" : 1,
          "failed" : 0
        }
      }
    }
  ]
}
```

# Gestion d'index + CRUD

Vous êtes mnt prêt pour le TP3 => voir Notebook