

SERLi



Premiers Pas



De quoi s'agit-il?

- Moteur de recherches orienté documents
 - JSON, basé sur [Apache Lucene](#)
- Sans schéma
- Distribué
- Scalable, Haute disponibilité
- Multi-tenant
 - Création/Suppression dynamique d'index
- Centré sur l'[API](#)
 - REST



Pourquoi faire?

- Recherches non structurées
 - Tous les documents qui contiennent le mot “moteur”
- Recherche structurée
 - Tous les documents créés entre le 01/10/2014 et le 31/12/2014
- Agrégations
 - Répartition des documents par mois et par type
- Combinaison de plusieurs types de recherches
- Temps-réel



Références



WIKIPEDIA



Des centaines de références... Pour tous usages



Principaux concepts

- Document
 - Élément stocké dans ElasticSearch, au format JSON
- Noeud
 - Une instance d'ElasticSearch
- Cluster
 - Un ensemble de noeuds
 - Le cluster contient un unique master node



Principaux concepts II

- **Index**
 - Ensemble de documents
 - Tout document appartient à un index
- **Type**
 - Sous-ensemble de documents
 - Tout document a un type
- **Shard**
 - Fragment d'un index (scalabilité)



Installation/Prise en main



- Avoir java (récent) installé sur ça machine
- Télécharger la version (zip) sur download.elasticsearch.org
- Extraire l'archive
- Lancer [bin/elasticsearch](#)

Le cluster est démarré!!!
sur <http://localhost:9200>





Un peu de configuration

Dans le fichier `config/elasticsearch.yml`

`node.name` : nom du noeud, nom aléatoire à chaque démarrage

`cluster.name` : le nom du cluster, tous les noeuds qui partagent le **même nom de cluster** appartiennent au même cluster

`discovery.zen.ping.multicast.enabled` : active la découverte automatique des noeuds

Disabled en production



Les données

Par défaut, les données sont **directement stockées dans l'arborescence** d'ElasticSearch

Dans le répertoire **data**

Chaque index est stocké indépendemment
data/elasticsearch/nodes/n/indices

Il est possible de spécifier un autre répertoire



Plugins

ElasticSearch propose un système de plugins

Des plugins pour :

- ajouter de nouvelles fonctionnalités
- la supervision
- écrire et concevoir des requêtes



Installation d'un plugin

On utilise la commande `bin/plugin`
`plugin install <nom du plugin>`

Par exemple :

```
plugin install mobz/elasticsearch-head
```

Les plugins peuvent avoir une IHM (`_site`)
`http://localhost:9200/_plugin/head/`



Deux protocoles

HTTP : 92XX

Expose l'API REST

Par défaut sur le port 9200

Transport : 93XX

Communication entre les noeuds (échanges de données, gossip)

Mais aussi pour certains clients (Java)

Par défaut sur le port 9300



Data in/Data out



Documents

Documents au format JSON

```
{  
  "firstName": "Tony",  
  "lastName": "Stark",  
  "aka": "Iron Man",  
  "team": "Avengers",  
  "age": 45  
}
```



Types de données

Types primitifs

string, number, boolean, datetime, binaire

Types complexes

Array, Object

Autres types

geo_point, geo_shape, ip, multi-field

Inférence de type à l'insertion



Conventions de l'API

`http://[host]:[port]/index/type/_action|id`

Associé à un verbe HTTP :

- GET
- POST
- PUT
- DELETE
- HEAD



API multi-tenants

POST /**heroes**/_search : tout **heroes**

POST /_search : tous les index

POST /**heroes,vilains**/_search

Recherche **heroes** et **vilain**

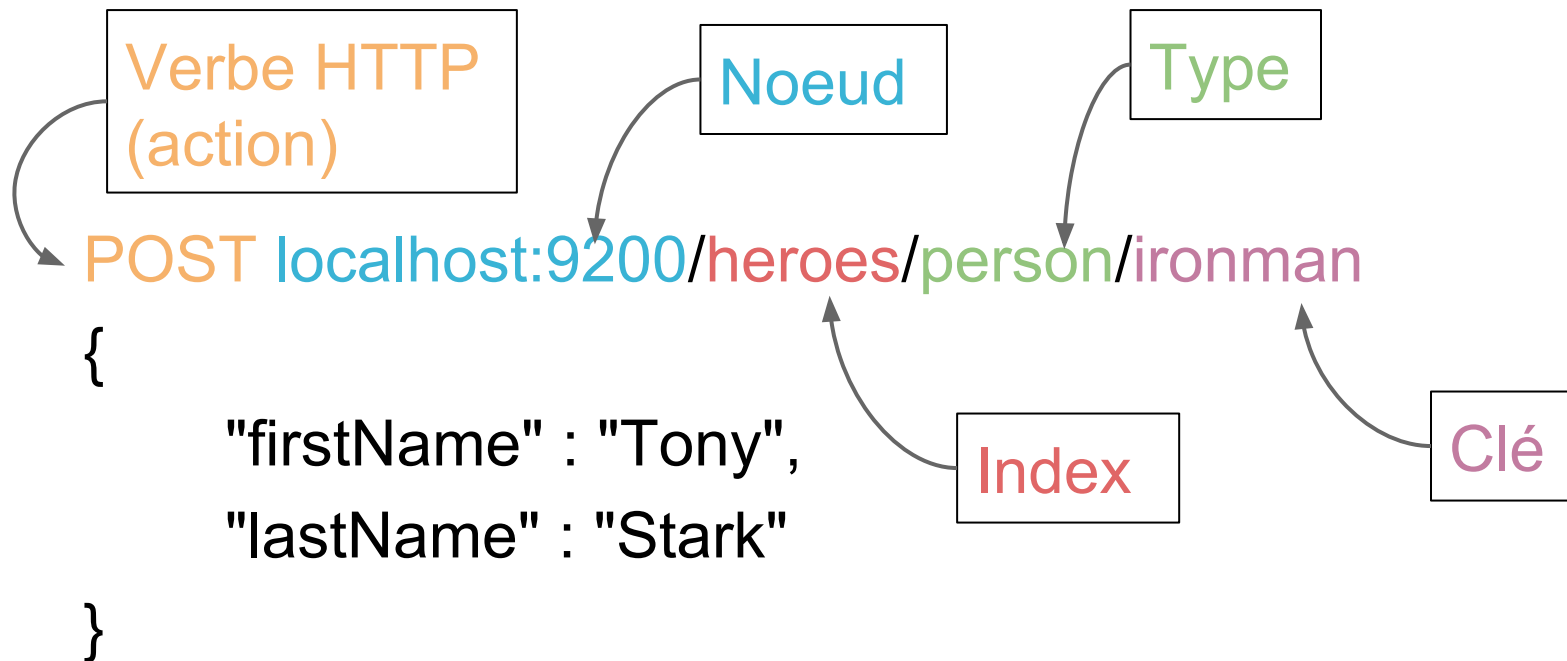
POST /**h***/_search

Recherche sur tous les index qui commencent par un **h**



Indexer un document

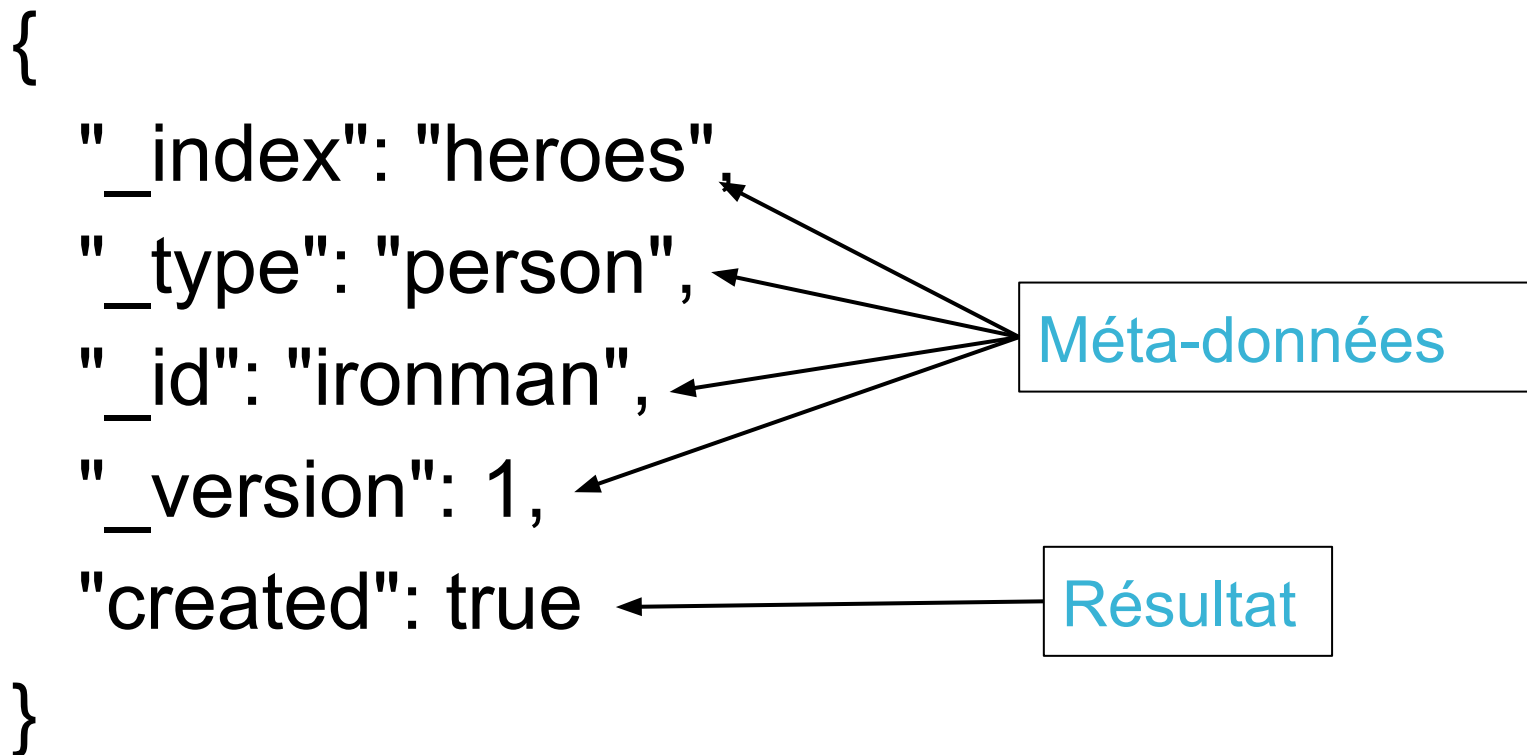
Pour indexer un document, on utilise l'API REST





Retour de l'insertion

Suite à l'insertion on obtient la **réponse** suivante :





Indexation : POST vs PUT

Lors de l'indexation d'un document, **POST** et **PUT** sont équivalents

POST permet de créer des documents sans renseigner la clé

POST /heroes/person/

```
{  
  "firstName" : "Charles",  
  "lastName" : "Xavier"  
}
```



```
{  
  "_index": "heroes",  
  "_type": "person",  
  "_id": "AUuFm0z0oSZRHss7_tP7",  
  "_version": 1,  
  "created": true  
}
```



Extraction d'un document

Pour extraire un document, on utilise **GET**
GET /**heroes**/**person**/**ironman**

La réponse :

```
{  
  "_index": "heroes",  
  "_type": "person",  
  "_id": "ironman",  
  "_version": 11,  
  "found": true,  
  "_source": {  
    "firstName": "Tony",  
    "lastName": "Stark"  
  }  
}
```

_found indique si le document a été trouvé

_source contient le document extrait



Mettre à jour un document

On peut utiliser un **POST** ou **PUT** pour mettre à jour l'intégralité du document

Ou l'API **_update**

POST /heroes/person/ironman/_update

```
{  
  "doc" : {  
    "firstName" : "Tomy"  
  }  
}
```



Exists/Delete

Pour vérifier l'existence d'un document, on utiliser le verbe **HEAD**

HEAD /heroes/person/ironman

Le statut de la réponse :

200 : le document existe

404 : le document n'existe pas

Pour supprimer on utilise le verbe **DELETE**

DELETE /heroes/person/ironman



Verrouillage optimiste

L'attribut `_version` permet de mettre en oeuvre un **verrouillage optimiste**

`PUT /heroes/person/ironman?version=1`

Si la version du document n'est pas **1**, l'opération échoue et retourne une **erreur**



Création des indexes

Dans les **exemples précédents** les index ont été créés **à la volée** à l'insertion des documents
Il est possible de créer des index explicitement

PUT /heroes/

Nom de l'index

{

"settings" : { ... },

Configuration

"mappings" : { ... }

Mappings

}



Suppression d'un index

Pour supprimer un index, on utilise le verbe
DELETE

DELETE */heroes/*