

Cahier de TP « Apache Solr »

Pré-requis

- Système d'exploitation recommandé : Linux ou Windows 10
- JDK8
- Bon éditeur (Notepad++, Vscode, ...)

TP1 : Installation et démo

Récupérer une distribution de Apache Solr et la dézipper dans un répertoire de travail : (\$SOLR_HOME)

Démarrer la configuration cloud

```
cd $SOLR_HOME
```

```
./bin/solr start -e cloud
```

Répondre à l'assistant en choisissant :

- 2 nœuds
- les ports proposés
- une collection de données nommée **techproducts**
- 2 shards, 2 répliques
- **sample_techproducts_configs** comme configuration

Ouvrir et découvrir l'interface d'administration : <http://localhost:8983/solr/>

Indexer des données avec :

```
./bin/post -c techproducts example/exampledocs/*
```

Ensuite visualiser les documents via l'interface d'admin en effectuant des recherche

- Tous les documents
- Les documents contenant le terme **foundation**
- Les documents dont le champ **cat** contient **electronics**
- Les documents contenant la phrase « **CAS latency** »

Vous pouvez également utiliser *curl* pour effectuer ses requêtes

Supprimer la collection via :

```
bin/solr delete -c techproducts
```

Arrêter le cluster

```
bin/solr stop -all
```

Démarrer la configuration **techproducts** et accéder à l'interface exemple :

<http://localhost:8983/solr/techproducts/browse>

TP2 : Mise en place de coeur

2.1 Création de coeurs

Démarrer une configuration standalone de Solr

Créer un coeur nommé ***formation_managed*** (Schéma managé et mode schemaless)

Créer un autre coeur nommé ***formation***, se positionner en mode éditable

Visualiser les fichiers de configuration créés.

2.2 Comprendre l'analyse

Testez l'analyse sur la chaîne « Une formation débutant sur SolR » :

- Avec le type de champ *text_ws*
- Avec le type de champ *text_general*
- Avec le type de champ *text_fr*

Que constatez-vous ?

Lisez les commentaires associés à ces types de champs dans *schema.xml*

Essayez avec :

« Overview of Documents, Fields, and Schema Design » et le champ phonétique *english*

Exécuter le test fourni pour visualiser les effets des annotations

2.3 Analyseur phonétique français

Définir un nouveau type de champs effectuant une analyse phonétique en français, y associer un champ dynamique et tester l'analyse

TP3 : Indexation

3.1 Indexation XML

Visualiser le fichier XML fourni et modifier le fichier ***schema.xml*** pour être le plus précis sur les champs utilisés

Effectuer la bonne configuration dans ***solrconfig.xml*** pour fixer les champs

Utiliser le fichier XML fourni pour alimenter les 2 coeurs

3.2 Indexation JSON

Reprendre le fichier ***slides.json*** fourni et effectuer une requête permettant d'insérer le document dans le schéma maîtrisé précédent

3.3 Indexation CSV (Optionnel)

Ajouter un document au format CSV dans les coeurs précédents

TP4 : Importation de documents bureautique

Utiliser l'utilitaire *posttool* ou *curl* pour indexer les documents bureautiques fournis.

TP5 : Importation base de données

TP6 : Configuration request handler et 1ères recherches

6.1 Configuration

Base bureautique :

Configurer le request handler pour :

- travailler par défaut sur le champ **content**
- forcer une sortie en json
- utiliser par défaut le parseur lucene
- Désactiver les searchcomponent facet, morelikethi et highlight
- Ajouter le_score dans les champs retournés
- Informations de debug

6.2 Requêtes

Effectuer les recherches suivantes en utilisant la syntaxe lucene :

- Documents répondant à « Java »
- Documents ne répondant pas à « Java »
- Limiter les documents retournés de la première requête
- Documents dont le contenu répond à « Java »
- Documents PDF dont le contenu répond à « Java »
- Documents dont le contenu répond à « SolR»
- Documents dont le champ titre contient administration
- Document créés après une date particulière
- Document créés après une date particulière et dont le contenu répondant « Java Elastic Search » mais pas « Administration »

TP7 : Différents types de recherches

7.1 Spell-check

Voir la configuration du spell-check

7.2 Highlight

Utiliser les paramètres de highlight

7.3 auto-complétion

TP8 : Agrégation de documents

8.1 Facettes

Utiliser les paramètres liés aux facettes

8.2 Regroupement

TP9 : Recherche géo-graphique et client Java

Compléter l'application SpringBoot afin de pouvoir importer des données de géo-location dans un coeur SolR :

- Ajout des dépendances vers SolrJ
- Classe de configuration SolR créant un bean HttpClient
- Implémentation d'une classe service ajoutant un document dans un coeur à partir de la classe du modèle Position.java

Préparer un coeur définissant des champs géographique avec les types :

- LatLonPointSpatialField
- SpatialRecursivePrefixTreeFieldType

Effectuer ensuite les requêtes suivantes :

- Rechercher les documents via un rectangle
- Distance à partir d'un point central
- idem avec en plus des facettes
- Agrégation de type heatmap sur le champ RPT