

Installation de Solr

Installer une machine virtuelle Java 5 ou ultérieur: <http://www.java.com/fr/download>

Aller dans votre dossier de classe et télécharger l'archive contenant Solr: `cours-solr.zip`

Décompresser l'archive.

Démarrage de Solr

Bien que Solr puisse être démarré directement en double-cliquant sur le fichier "start.jar", l'idéal est cependant de le démarrer via une ligne de commandes. Ainsi un accès direct aux logs est possible via la fenêtre de la ligne de commandes. En tous les cas, les fichiers logs sont automatiquement stockés dans le répertoire "logs" du dossier Solr.

Dans la ligne de commande, aller dans le répertoire "cours-solr" et exécuter la commande:

```
java -jar start.jar
```

Solr va alors automatiquement démarrer et pourra être accédé, dès le démarrage terminé, via un navigateur Web (de préférence FireFox) à l'adresse suivante:

<http://localhost:8983/solr/admin>

Sous Windows, il conviendra encore juste d'indiquer que Solr n'est pas le dernier cheval de Troyes à la mode, en débloquent "Java(TM) Platform SE Binary" lorsque le message d'alerte de Windows apparaît.

Données de test

L'instance de Solr installée contient comme données l'ensemble des localités de Suisse.

Requêtes Lucene

- ☐ Rechercher quelques localités, en utilisant le formulaire de requête.
- ☐ Rechercher les localités dont le numéro postal est compris entre 1500 et 2200 (inclus)
- ☐ Rechercher les localités dont le numéro postal est compris entre 3000 et 4500 (exclus)
- ☐ Rechercher les localités dont le numéro postal est 1000, 2000, 3000, 4000 ou 5000 classées par ordre décroissant des acronymes de cantons puis croissant des noms de villes.
- ☐ Rechercher les localités dont le numéro postal est plus petit que 2000 (exclus), classées par nom de ville.
- ☐ Rechercher les localités dont le numéro postal est plus grand que 7000 (inclus).
- ☐ Rechercher les localités contenant "sur", "derrière" ou "devant" dans leur nom.
- ☐ Rechercher les localités des cantons romands.
- ☐ Rechercher les localités dont le nom commence par la lettre "J".
- ☐ Rechercher les localités dont le nom commence par une lettre entre "L" et "Z", classées une fois par ordre croissant des noms de localités et l'autre par ordre décroissant.
- ☐ Rechercher toutes les localités, classées par canton.
- ☐ Rechercher les localités dont le numéro postal est compris entre 2000 et 2100. Puis entre 3230 et 3250.
- ☐ Effectuer les 2 recherches précédentes (2000-2100 et 3230-3250) en une seule requête.
- ☐ Rechercher les localités dont le numéro postal est compris entre 1500 et 2000 (exclus) mais n'étant ni dans le canton de Fribourg ni dans celui de Vaud. N'afficher dans le résultat que le numéro postal, le nom de la ville et l'acronyme du canton en veillant à ce que tous les résultats soient retournés.
- ☐ Rechercher les localités du canton de Berne dont le numéro postal est compris entre 2500 et 2750 mais dont le nom ne contient pas "Bienne".
- ☐ Rechercher les localités dont le nom contient à peu près le terme "Lussy".
- ☐ Rechercher les localités dont le nom contient "am" mais pas "See".
- ☐ Rechercher les localités dont le nom contient "am See".

Paramètres

- ☐ Tester les paramètres 'start' et 'rows' en affichant successivement les résultats 0-99, 100-199, 200-299 de toutes les localités du canton de Vaud.
- ☐ Tester le paramètre 'sort' et effectuant des tris sur les différents champs des localités du canton de Neuchâtel: par numéro postal, par nom de localité.
- ☐ Tester le paramètre 'fl' et affichant successivement que les champs "score", "city.name", "city.code", "city.region.name" et "city.region.code".
- ☐ Tester le paramètre 'wt' en modifiant le format de sortie du résultat.
- ☐ Tester le paramètre 'tr' en utilisant une des stylesheets disponibles dans le répertoire "solr/conf/xslt".
- ☐ Créer dans le répertoire "solr/conf/xslt" une stylesheet XSLT basique affichant le nombre de résultats trouvés par une requête Solr et la tester.
- ☐ Tester le paramètre 'debugQuery' et l'ajoutant à des requêtes déjà effectuées précédemment. Particulièrement sur les requêtes complexes agissant sur plusieurs champs. L'élément 'parsedquery' donne beaucoup d'informations intéressantes sur la manière dont Lucene interprète les requêtes.

Recherche par facettes

- ☐ Rechercher toutes les localités en affichant les facettes des noms de cantons.
- ☐ Rechercher toutes les localités dont le nom contient "la" ou "le" en affichant les facettes des acronymes de cantons.
- ☐ Rechercher toutes les localités en affichant une facette pour les localités dont le nom est entre A et E, une autre entre F et J, une autre entre K et O, une autre entre P et T, une autre entre U et Z.

Correction orthographique

- ☐ Tester la correction orthographique en activant le composant spellchecker pour des requêtes sur des noms de ville mal orthographiés.

Soulignement

- ☐ Tester le soulignement en activant le composant highlighting pour des requêtes sur certains mots.