



Licence 3 D.A.N.T.

U.E. Nouvelles technologies du WEB



Moteur d'indexation distribué

Description technique et fonctionnelle

Sommaire

- Qu'est-ce qu'un moteur d'indexation ?
- Qu'est-ce qu'une API REST ?
- Distribution du moteur d'indexation
- Objectif
- Technologies utilisées
- Créer une table
- Insérer / charger des données
- Technique de stockage
- Créer un index
- Recherche de données
- Tests
- Démonstration



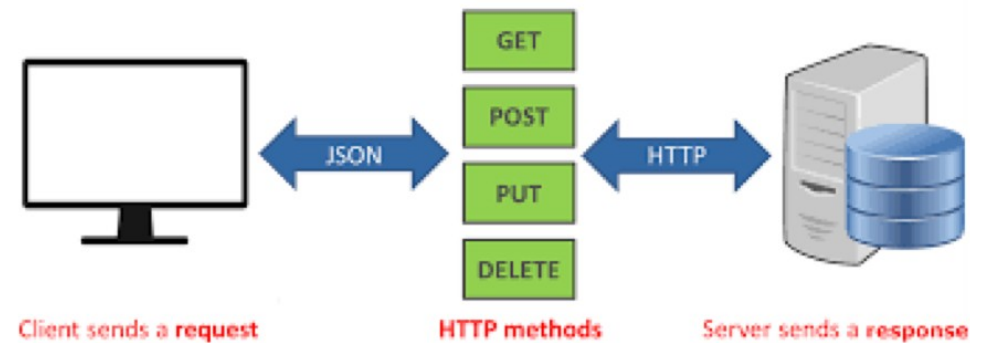
Qu'est-ce qu'un moteur d'indexation ?

- Création d'index
- Ingérer des données
- Retrouver des lignes
- Distribuer le Dataset



Qu'est-ce qu'une API REST ?

- Sans Etat
- Orienté Client - Serveur
- Avec une interface uniforme
- Avec un système de couche
- Un code scalable



Distribution du moteur d'indexation

- **Avantages :**
 - Pas de nœud master
 - Très scalable (pas besoin de grosse machine)
 - Distribuer les requêtes sur différents nœuds
 - Amélioration des performances
 - Résistance aux pannes



Objectif



- L'objectif est de créer un moteur d'index qui permet de créer des tables composées de colonnes nommées-typées en pouvant indexer une ou plusieurs colonnes pour accélérer les opérations de recherche sur des grosses masses de données.
- Le moteur est manipulé à travers une API dont on peut instancier plusieurs noeuds qui communiquent lors des opérations d'insertion et de recherche pour distribuer les tâches entre eux.

Technologies utilisées

- Environnement de développement : IntelliJ IDEA
- Gestionnaire de package : Maven
- Gestionnaire de version : Git
- Test Unitaire : JUnit
- Pour l'API REST : REST Easy

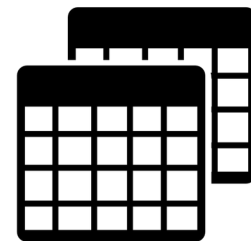




Fonctionnalités



Créer une table



- **Il est possible de créer une table par les moyens suivants :**
 - JSON décrivant les colonnes (nom et type) dans le body, ou par un fichier json.
 - Fichier CSV dont on crée la table à partir de son header, et on insère par la même occasion ses données.
 - Les types possibles de colonnes sont Int, String, Long et Float.
- **Remarque :**
 - Quand une table est créée à partir du header d'un CSV, toutes les colonnes prennent par défaut le type String.
 - Il est possible de créer une table de manière distribuée ou non, à préciser en request param. Par défaut la création n'est pas distribuée.

Insérer / charger de la donnée

- Une fois la table créée, pour y insérer des données, les moyens suivants sont possibles :
 - Requête similaire à SQL dans le body en texte : `INSERT INTO [Table name] VALUES ([Column name] : [value]), ([Column name] : [value])... ([Column name] : [value])`. La requête est insensible à la casse.
 - Insérer des données par un fichier CSV dont le séparateur est la virgule.
- Remarque :
 - Quand une table est créée à partir du header d'un CSV, toutes les colonnes prennent par défaut le type String.

LOADING...



Technique de stockage



- **Les données brutes sont stockées dans le disque dur de la manière suivante:**
 - Une table a une liste de FileManager.
 - Un FileManager gère le stockage d'un nombre de lignes (borné par une constante).
 - Le FileManager stocke ses données dans 2 fichiers binaires : l'un stockant les lignes en byte[] brutes, et l'autre stockant la position de chacune des lignes pour faciliter l'accès aléatoire.
 - Un FileManager a 2 états : Loaded ou pas.
 - Loaded veut dire que la donnée est chargée en RAM (donc accessible et modifiable rapidement), sinon la donnée n'est pas chargée en RAM → Lecture et écriture coûteuse.

Créer un index

- En donnant la liste des colonnes qu'on veut indexer
- On peut choisir le type de la structure de données de l'index :

- **B+Tree :**

- Full disk stored
- Semi disk stored
- Full RAM stored

- **Hash table :**

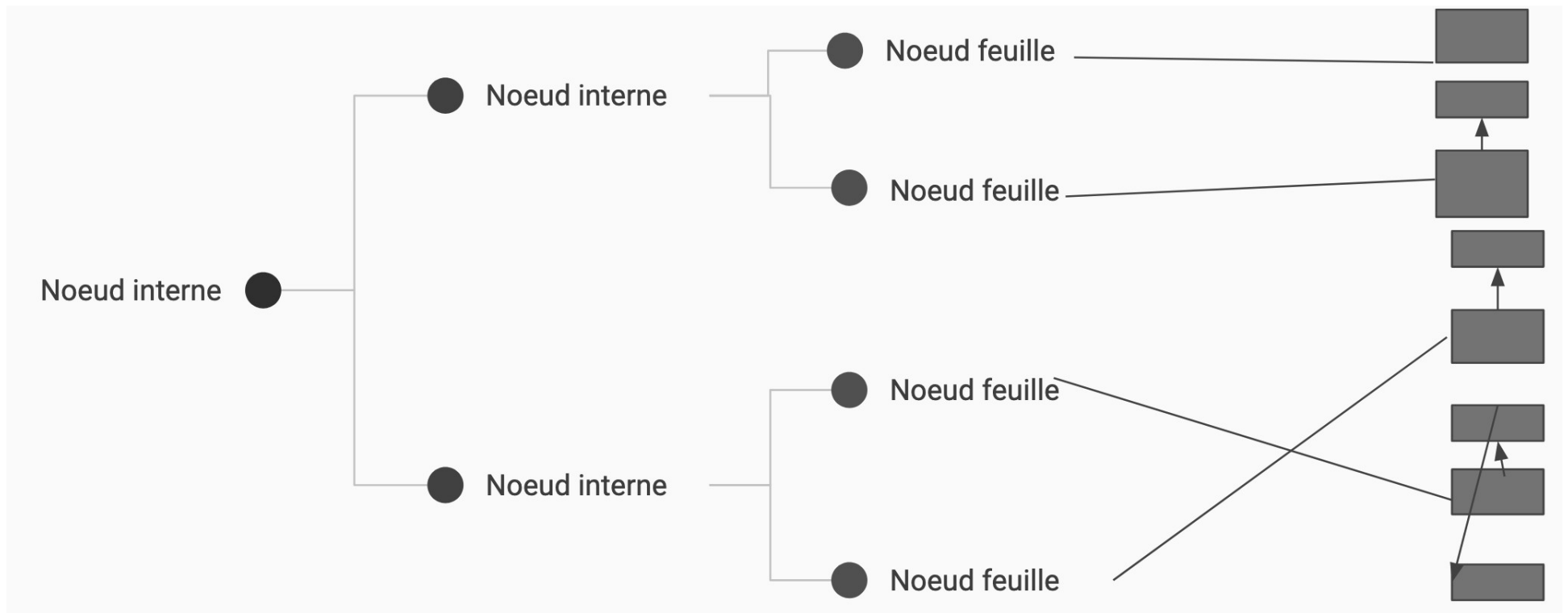
- Full RAM Stored

⇒ par défaut B+Tree - Semi disk stored.

⇒ Pour pouvoir faire des requêtes avec des where > ou where < l'idéal est d'utiliser B+Tree.



Technique de stockage



Technique de stockage



- **B+Tree :**

- On a des noeuds et chaque noeud feuille pointe sur la tête d'une liste représentée par un grand tableau d'entiers pointant sur des positions logiques de lignes de la table d'origine. On a donc les noeuds et les listes.

- **Types de stockage :**

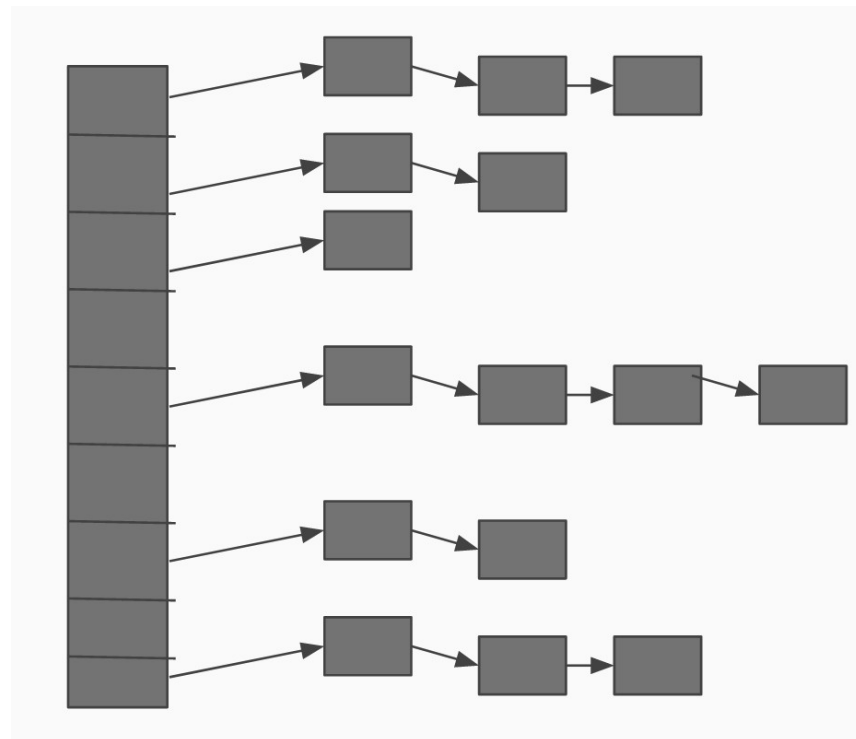
- Full RAM : Noeuds et listes en RAM
 - Semi Disk : Noeuds en RAM et listes en DD
 - Full Disk : Noeuds et listes en DD
- ⇒ En général le plus efficace est le Semi Disk

Technique de stockage



- **Hash table :**

- Index basé sur une table de hachage, les clés pointent directement sur la liste correspondant.
- Stocké en RAM sous forme de `HashMap<Integer, List<Integer>>`



Recherche de données



- **Via une requête similaire à SQL donnée en Body :**
 - Par défaut, il y a une clause limit fixée à 10000, mais précisable via query param.
 - Possibilité de sélectionner toutes les colonnes (par select *) ou une liste de colonnes séparées par des virgules.
- **Stratégie de la sélection des index :**
 - Si la requête contient que des Where = liés par des AND uniquement ET il existe un index sur toutes ces colonnes en même temps : On utilise cet index.
 - Sinon, Parcourir chacun des Where : S'il existe un index sur cette colonne on l'utilise sinon on recherche de manière naïve (séquentielle).

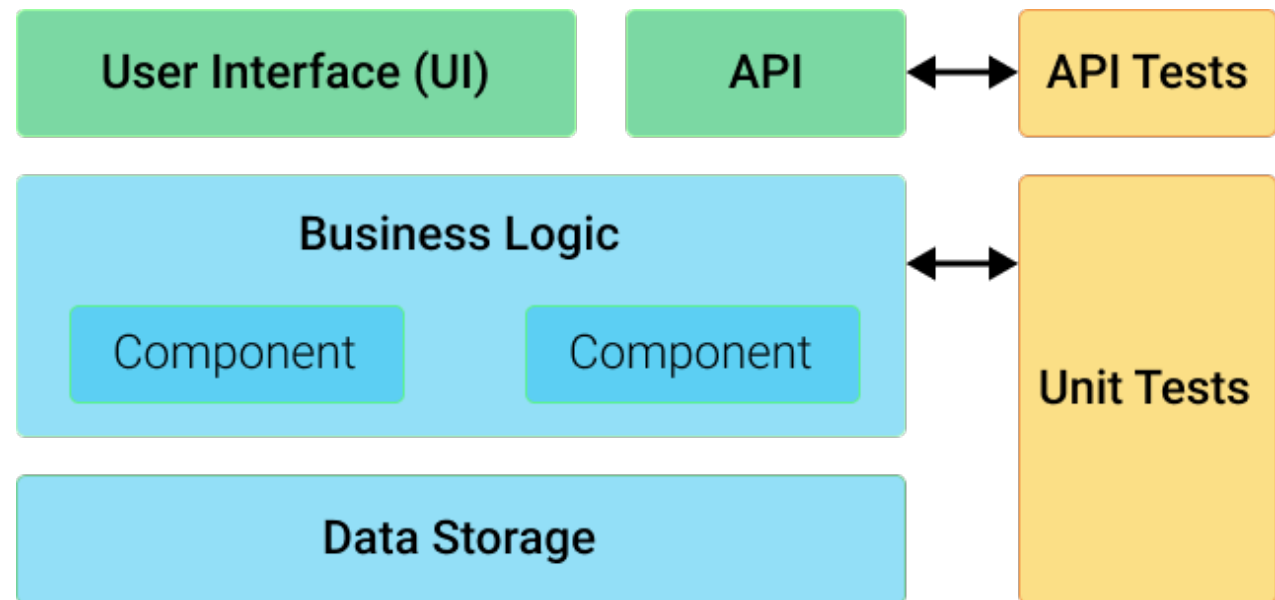


Tests



REST API Testing

- Les tests d'API (application programming interface) permettent d'effectuer des vérifications fonctionnelles directement au niveau de l'API.
- C'est une partie de la phase de tests d'intégration permettant de vérifier si l'API vérifie les attentes en terme de :
 - Fonctionnalité,
 - Fiabilité,
 - Performance,
 - Sécurité.



Test unitaires

- Tester chaque partie individuelle du programme et vérifier leur bon fonctionnement.
- Peut être assimilé au « white box testing » dans l'approche « box testing », le code est visible au testeur.
- Peut être effectué à n'importe quel moment.
- Test uniquement les fonctionnalités elles-mêmes, peut ne pas attraper les erreurs d'intégration et/ou systèmes.
- Commence par la spécification du module, et fait attention au comportement de celui-ci.
- Ne permet pas de vérifier si le code fonctionne correctement avec les dépendances externes correctement.
- Exécuté généralement par le développeur.
- Trouver les erreurs est assez rapide.
- La maintenance de ces tests est peu coûteuse.





Démonstration

