Projet Open Food Facts

Description

Vous connaissez peut-être l'application **Yuka,** disponible sur smartphone. **Yuka** fournit des informations nutritionnelles sur pratiquement tous les produits alimentaires commercialisés en France. En plus d'informations, elle fournit également un score nutritionnel, de A (excellent) à F (mauvais).

Cette application à succès s'est construite sur une **base de données open source** appelée **Open Food Facts**.

La base de données **Open Food Facts** est une base de données **mondiale**, qu'on peut télécharger sous la forme d'un fichier **CSV**. Le fichier que je vais vous demander de traiter dans le cadre de ce projet est le même que celui sur lequel s'est basé Yuka. Il ne concerne que **les produits alimentaires fabriqués en France**.

Dans ce TP vous allez créer une application qui met en base ce fichier.

Description du contenu du fichier

Le fichier open food facts, au format CSV, comporte 13 432 références de produits avec 30 informations associées par produit. Dans ce fichier le caractère séparateur est le **|**. Si on découpe selon ce caractère séparateur, on obtient le tableau d'éléments suivant :

- Index 0 : catégorie du produit
- Index 1 : marque du produit
- Index 2 : nom du produit
- Index 3 : score nutritionnel : A (excellent) à F (mauvais)
- Index 4 : liste des ingrédients séparés la plupart du temps par des virgules, mais pas toujours.
- Index 5 : énergie pour 100g (en joules)
- Index 6 : quantité de graisse pour 100g
- etc.
- Index 28 : liste des allergènes séparés la plupart du temps par des virgules
- Index 29 : liste des additifs séparés la plupart du temps par des virgules

Objectifs

Le but est de **concevoir** et **développer** une application basée sur l'API JPA pour mettre en base toutes ces données.

Les règles de gestion à respecter :

- Une catégorie doit être unique en base de données
- Une marque doit être unique en base de données
- Un ingrédient doit être unique en base de données.
- Un allergène doit être unique en base de données
- Un additif doit être unique en base de données
- Cas particulier du traitement des ingrédients :
  - Dans le fichier, les ingrédients sont séparés par un séparateur. Exemple : Sucre, farine, banane
  - Le séparateur le plus courant est la virgule mais il est possible que d'autres séparateurs aient été utilisés comme le tiret. En effet comme il s'agit d'une base mondiale, certains utilisateurs ne respectent pas toujours les consignes. Attention donc à prévoir plusieurs cas de figures.
  - De manière à éviter les ingrédients du type Sucre* ou _Sucre_ par exemple, il faudra prévoir une suppression des caractères parasites.
  - Certains ingrédients possèdent une description entre parenthèses : il est demande de supprimer toutes les informations présentes entre parenthèses.
  - Certains ingrédients possèdent également des %. De manière à éviter d'avoir des ingrédients du type Sucre 50%, il faudra veiller à supprimer tous les pourcentages.

Cas d'exemples pour les traitements des ingrédients, des allergènes et des additifs :
Cas 1 – les caractères spéciaux :
Avant traitement : Sucre*, farine, _Maïs_
Après traitement : Sucre, farine, Maïs

Cas 2 – les pourcentages :
Avant traitement : Sucre 15%, farine 50%, Maïs 35%
Après traitement : Sucre, farine, Maïs

Cas 3 – les parenthèses :
Avant traitement : Sucre, banane, Pâte (Farine 50%, Sucre 20%, Œufs 30%)
Après traitement : Sucre, banane, Pâte

Objectif n°1 : Réaliser un dossier de conceptions contenant :

- Un diagramme de classes métier
- Un modèle physique de données
- Prévoyez 1 journée pour ce travail sinon vous n'aurez pas le temps de réaliser le développement.

- Vous placerez le dossier de conception dans un répertoire appelé **conception** et situé à la racine de votre projet Git.

Objectif n°2 : Développer l'application :

- Mettre en en place les entités JPA avec les annotations
- Développer de manière professionnelle en tenant compte des bonnes pratiques de codage : javadoc, pas de code redondant, etc.
- Mettez en place une couche DAO, acronyme de Data Access Object, (une DAO par entité) pour les accès à la base de données.
    - Si vous voulez en savoir plus sur le pattern DAO : https://www.baeldung.com/java-dao-pattern voir notamment le **chapitre 3.1**.

Objectif n°3 : mettre en place des optimisations afin que votre traitement soit le plus rapide possible. Vous pourrez essayer par exemple de mettre en place **ehcache**.
Record à battre : 35 minutes en respectant les règles de gestion bien sûr.

Objectif n°4 : Si vous avez terminé l'objectif 3, développez une application de restitution avec un menu et permettant :

- de rechercher les N meilleurs produits pour une Marque donnée. La marque et la valeur de N sont demandées à l'utilisateur.
- de rechercher les N meilleurs produits pour une Catégorie donnée. La catégorie et la valeur de N sont demandées à l'utilisateur.
- de rechercher les N meilleurs produits par Marque et par Catégorie. La marque, la catégorie et la valeur de N sont demandées à l'utilisateur.
- d'afficher les N ingrédients les plus courants avec le nb de produits dans lesquels ils apparaissent. La valeur de N est demandée à l'utilisateur.
- d'afficher les N allergènes les plus courants avec le nb de produits dans lesquels ils apparaissent. La valeur de N est demandée à l'utilisateur.
- d'afficher les N additifs les plus courants avec le nb de produits dans lesquels ils apparaissent. La valeur de N est demandée à l'utilisateur.

Instructions techniques complémentaires

- Créez un nouveau projet **STS/GitHub/Maven** nommé **traitement-fichier**
- Créez un répertoire **conception** à la racine de votre projet
    - Placez-y le dossier de conception.
- Réalisez les développements.
- Mettez en place les optimisations afin de battre le record !
- Développez l'application de restitution des informations.
- Bon courage !
- COMMITEZ !!