

## TP n°8 – Implémentation Jpa sur le projet open food facts

### Objectifs

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 TP est le même que celui sur lequel s'est basé Yuka. Il ne concerne que les **les produits alimentaires fabriqués en France**.

Dans ce TP vous allez mettre en base de données un fichier open food facts contenant près de 13432 produits commercialisés en France.

### Description

Développez une nouvelle application permettant d'intégrer en base de données le fichier **open food facts**.

Le fichier fourni comporte 13 432 références avec 30 informations associées par produit :

- catégorie
- marque
- nom
- score nutritionnel : A (excellent) à F (Faible)
- liste des ingrédients
- énergie pour 100g (en joules)
- quantité de graisse pour 100g
- liste des allergènes
- liste des additifs
- présence d'huile de palme
- toutes les vitamines,
- calcium, magnésium, etc..

Au total chaque référence de produit a 30 attributs.

### Type de fichier

Le fichier fourni est un fichier CSV.

Un fichier CSV est un fichier dans lesquelles les informations sont séparées par des séparateurs. Ici le séparateur utilisé est le | :

- la première ligne fait la liste des attributs séparés par des caractères |
- chaque ligne suivante correspond à un article et les valeurs des attributs sont séparées par des |
  - o la liste des ingrédients est séparée par des virgules, parfois des ; ou encore des tirets.
  - o la liste des additifs et des allergènes également
  - o Attention certaines données, notamment chiffrées, ne sont pas toujours renseignées.

## Instructions

- Créez un nouveau **projet maven** nommé **traitement-fichier-jpa-off**
- Poussez-le sur GitHub
- Créez une nouvelle base de données appelée **open-food-facts**.
- Créez une classe exécutable **IntegrationOpenFoodFacts**.
- Vous devrez mettre en place des **DAO** afin de gérer vos diverses entités. Pas de requête en dehors des DAO !!
- **Précautions à prendre :**
  - o Réfléchissez aux classes que vous allez mettre en place avant de vous lancer. Attention aux cardinalités entre classes et l'impact sur le modèle de données
  - o Il n'est pas question d'avoir des doublons dans les tables.
  - o Votre programme doit être rejouable, c'est-à-dire qu'il est capable de vérifier si une donnée existe déjà ou non avant de l'intégrer.
  - o Si un produit ne satisfait pas aux contrôles, par exemple si vous n'arrivez pas à parser la liste des ingrédients, vous devez le stocker dans une table des erreurs.
  - o **Plus difficile :** Votre programme est capable de reprendre là où il en était en cas de plantage.
  - o **Exigences qualités :** Vous devez mettre en place la javadoc.
- **Réalisez également une application permettant à un utilisateur de sélectionner dans un menu :**
  - Affichez les 10 produits les mieux notés d'une catégorie
  - Affichez les 10 produits les mieux notés d'une marque
  - Affichez les 10 produits les mieux notés d'une catégorie et qui ne contiennent pas un ingrédient donné
    - Affichez les 10 produits les mieux notés d'une marque et qui ne contiennent pas un ingrédient donné
    - Affichez les 10 produits les mieux notés d'une catégorie et qui ne contiennent pas un allergène donné
    - Affichez les 10 produits les mieux notés d'une marque et qui ne contiennent pas un allergène donné

## Livrables attendus

- Les 2 applications
- Le fichier SQL contenant la structure des tables.

## COMMITTEZ