

Entretien technique Développeur Sénior

Développer une API permettant de rechercher et gérer des produits alimentaire

Consignes

- Les recherches de produits et de substitution doivent requêter <https://fr.openfoodfacts.org>
- Utilisation de la version **5.4 de Symfony**
- Utilisation de **PHP 7.4** minimum
- **Idéalement** développer ce projet en suivant les principes d'**architecture DDD** avec 3 domaines (**Si l'architecture DDD n'est pas possible, respecter le pattern MVC de Symfony**)
 - Recherche
 - Liste
 - Substitution
- **Idéalement** développer le projet en suivant les **principes TDD**
 - Si le TDD n'est pas possible, il faut a minima que le code soit entièrement testé
- La base de données doit être sous moteur **MariaDB**
- **Aucune lib autorisée sauf** les suivantes :
 - Symfony
 - Doctrine
 - Lexik/jwt-bundle
 - Monolog
 - nelmio/apidocbundle
 - overblog/GraphQLBundle (si réalisation du bonus GraphQL)
 - Lib de test parmi les suivantes
 - PHPUnit
 - Codeception
 - Prophetize
- **Commits réguliers** mettant en évidence les phases d'écriture de tests et de conception
- **Dockerizer** l'intégralité de la stack via **Docker ou Docker-compose**
- Le rendu doit se faire via un **repository Git** (GitHub, GitLab, Bitbucket) de votre choix
- Respecter toutes les **bonnes pratiques** PHP (PSR) et Symfony

Bonus

Les bonus sont purement optionnels et aucune pénalisation si les bonus ne sont pas effectués

- Afin d'améliorer les performances, un cache de type **Redis** peut être ajouté à la stack pour cacher les requêtes déjà effectuées
- Afin d'améliorer la résilience, il est possible d'articuler la communication entre les services avec un **service de messages**.
- Développer ce test sous **PHP 8**
- Ecrire les endpoints API en GraphQL

Enoncé

Vous devez développer une API sécurisée par JWT qui permettra de :

- Rechercher un ou des produit(s) alimentaire(s) par
 - Nom
 - Critères
 - Allergènes (exemple) :
 - Lait
 - Fruits à coque
 - Soja
 - ...
 - EAN
 - Marque
 - Catégorie

A noter : *Les champs de recherche doivent fonctionner aussi bien pour des mots exacts que pour des portions de mots*

L'API doit retourner au format JSON les résultats paginés par lot de 20, triés par pertinence avec les éléments suivants :

- Nom du produit
- EAN
- Marque
- Liste des ingrédients
- Liste des allergènes
- Nutri-score
- Valeurs nutritionnelles
- Meilleur substitut trouvé à ce produit

L'API doit exposer les endpoints suivants :

- /save/ean_du_produit

Lors de la requête /save, l'API doit sauvegarder en base de données le favori pour l'utilisateur qui a requis.

- /save/substitution/ean_du_produit

Lors de la requête /save/substitution/ean_du_produit, l'API doit sauvegarder en base de données la meilleure substitution trouvée pour l'EAN donné pour l'utilisateur qui a requis.

- /exclude/ean_du_produit

Lors de la requête /exclude/ean_du_produit, l'API doit sauvegarder le produit en base de données en tant que liste d'exclusion pour l'utilisateur qui a demandé.

- /delete/ean_du_produit

Lors de la requête /delete, l'API doit supprimer en base de données le favori pour l'utilisateur qui a demandé.

- /clear

Lors de la requête /clear, l'API doit vider en base de données les favoris pour l'utilisateur qui a demandé.

- /search/name?name=nom_du_produit&ean=...&marque=...

Au moins un critère de recherche est obligatoire.

Lors de la requête /search, l'API doit retourner les produits correspondants aux critères en excluant la liste d'exclusion de l'utilisateur s'il y en a une.