

Fiche d'investigation de fonctionnalité

Fonctionnalité : Recherche principale

Problématique : Développer un algorithme de recherche performant fournissant aux utilisateurs une recherche rapide, presque instantanée

Option 1 : Version en programmation fonctionnelle avec filter ()

On utilise filter() pour filtrer le contenu d'un tableau selon un critère donné. Ici on parcourt le tableau des recettes selon le titre, les ingrédients ou la description.

Avantages

- Lisibilité du code

Inconvénients

- Plus d'opérations à faire pour filtrer les résultats

Nombre de champs minimum pour lancer la recherche : 3

Option 2 : Version utilisant la boucle native for

Utiliser une boucle sur l'ensemble des recettes pour voir s'il y a un match avec la valeur de la recherche

Avantages

- Rapidité de l'exécution du code

Inconvénients

- Boucle native moins lisible à première vue

Nombre de champs minimum pour lancer la recherche : 3

Solution retenue : Nous avons retenu l'option 1 utilisant filter().

En effet, après avoir comparé les deux options via JSBench nous nous sommes aperçus que l'option 1 effectue un nombre de calculs supérieur à celui de l'option 2 mais cette différence est minime et n'entraîne pas d'impacts sur la performance de la recherche compte tenu du nombre limité de recettes dans le fichier JSON. Ainsi la recherche est quasi instantanée et l'utilisateur ne voit aucune différence entre les deux versions. Hors la lisibilité et simplicité du code de l'option 1 en font un choix judicieux pour la lisibilité du code, sa simplicité ou son refactoring.

Si on avait eu un nombre beaucoup plus important de recettes il aurait fallu choisir l'option 2 qui aurait alors été significativement plus rapide.

Les résultats des comparaisons sont disponibles ici : <https://jsben.ch/9IkPS>

