

Fiche d'investigation

Fiche d'investigation de fonctionnalité

Fonctionnalité : Recherche avancée de recettes

Problématique : Optimiser l'algorithme de recherche de recettes sur une base de données structurée en objets JS — comparatif entre implémentation native et fonctionnelle.

Option 1 : Implémentation native (for loops)

Dans cette approche, on parcourt la liste de recettes à l'aide de boucles `for` classiques imbriquées :

- Boucle principale sur les recettes
- Boucle imbriquée sur les ingrédients
- Boucle imbriquée sur les ustensiles
- Concaténation progressive des champs texte
- Comparaison du texte final normalisé avec le terme recherché (`includes()`)

Avantages :

- (+) Contrôle total sur les boucles et les opérations
- (+) Facile à déboguer pas à pas
- (+) Très performant (moins d'appels de fonctions internes JS)

Inconvénients :

- (−) Code verbeux et répétitif
 - (−) Difficile à maintenir si la structure de données évolue
 - (−) S'éloigne du paradigme fonctionnel moderne
-

Option 2 : Implémentation fonctionnelle (filter/map)

Dans cette approche, on utilise les fonctions natives de manipulation de tableaux :

- `filter()` pour filtrer directement les résultats
- `map()` pour extraire les noms d'ingrédients
- `join()` pour fusionner tous les champs texte
- Conversion en lowercase puis comparaison avec `includes()`

Avantages :

- (+) Code concis et lisible
- (+) Conforme aux bonnes pratiques fonctionnelles modernes
- (+) Facilement maintenable et modifiable
- (+) Meilleure expressivité métier

Inconvénients :

- (-) Léger surcoût lié aux fonctions internes JS (`map` , `join` , `spread`)
- (-) Contrôle limité sur l'optimisation bas niveau

Résultat du test de performance (JS Bench)

Version	Performance mesurée
Native	100%
Fonctionnelle	98.44%

Solution retenue

Nous avons retenu l'approche **fonctionnelle (filter/map)**.

Malgré une performance légèrement inférieure, sa concision, sa lisibilité et son évolutivité en font un meilleur choix pour la production. La différence de performance est négligeable au regard des avantages en matière de maintenabilité.

Résumé rapide

Critère	Native (for)	Fonctionnelle (filter/map)
Lisibilité	✗	✓
Contrôle bas-niveau	✓	✗

Modernité	✗	✓
Maintenabilité	✗	✓
Performance pure	✓ (100%)	Légère perte (98.44%)