



Fiche d'investigation de fonctionnalité

Fonctionnalité : Recherche et filtrage des recettes dans l'interface utilisateur	Fonctionnalité #2
Problématique : Optimiser la recherche tout en gardant un comportement fluide côté utilisateur	

Option 1 : Algorithme 1 – Méthodes de tableau (Array methods) utilisation des méthodes natives de tableau JavaScript (filter, some, every, map, etc.) pour appliquer successivement les critères de recherche (texte) puis l'intersection des tags (ustensile ...) sélectionnés.

Avantages <ul style="list-style-type: none"> ⊕ Code concis, lisible et proche des bonnes pratiques modernes JavaScript. ⊕ Très bonnes performances mesurées sur jsben.ch : Bloc 1 (Array methods) : 22 585 ops/sec 	Inconvénients <ul style="list-style-type: none"> ⊖ Empilement d'appels de méthodes qui peut être moins parlant pour un débutant
Nombre de champs minimum: 1 recette	

Option 2 : Algorithme 2 – Boucles natives (for / for...of)

réécriture du même algorithme avec des boucles impératives (for, for...of) pour parcourir les recettes et tester manuellement le texte et les tags.

Avantages <ul style="list-style-type: none"> ⊕ Contrôle très explicite du flux (break, continue, etc.). ⊕ Logique facilement compréhensible pour un profil débutant en JavaScript 	Inconvénients <ul style="list-style-type: none"> ⊖ Code plus verbeux, plus difficile à factoriser ⊖ Plus de risques d'erreurs (boucle infinis). ⊖ Performances inférieures dans nos tests jsben.ch : Bloc 2 (boucles natives) : 7 678 ops/sec.
Nombre de champs minimum 1 recette	

Solution retenue : l'algorithme 1

- Performances nettement meilleures, environ trois fois plus rapide que la version boucles natives.
 - Code plus idiomatique, plus simple à maintenir et à étendre.
 - Le comportement respecte strictement le cas d'utilisation de filtrage des recettes : recherche texte à partir de 3 caractères, filtrage par intersection de tous les tags sélectionnés (ingrédients, ustensiles, appareils), et affichage d'un message dédié lorsqu'aucune recette ne correspond pas.
- Les deux implémentations ayant été validées sur jsben.ch, nous conservons l'algorithme 1 comme solution principale et gardons l'algorithme 2 comme référence pédagogique et de comparaison de performances.

