

Fiche d'investigation de fonctionnalité

Fonctionnalité : Moteur de recherche de recettes	Fonctionnalité #1
Problématique : Fournir une expérience utilisateur fluide pour rechercher des recettes par mots-clés et filtres dynamiques.	

Option 1 : Méthodes fonctionnelles (array-method)

Dans cette option, nous utilisons les méthodes natives de l'objet Array, telles que `.filter()`, `.map()`, `.some()`, `.includes()`. Ces méthodes permettent de chaîner les traitements et d'écrire du code plus lisible et expressif. Elles sont particulièrement adaptées pour manipuler des tableaux comme la liste des recettes.

Avantages <ul style="list-style-type: none">⊕ Syntaxe concise, lisible et moderne⊕ Code facilement maintenable et réutilisable⊕ Optimisé pour des interactions courantes avec les tableaux⊕ Adapté aux manipulations DOM (génération HTML avec <code>.map()</code>)	Inconvénients <ul style="list-style-type: none">⊖ Moins de contrôle étape par étape⊖ Difficile à déboguer pas à pas pour un débutant⊖ Moins performant dans certains cas extrêmes (ex : traitements lourds sur 100 000 éléments)
--	--

Option 2 : Méthodes fonctionnelles (loop-method)

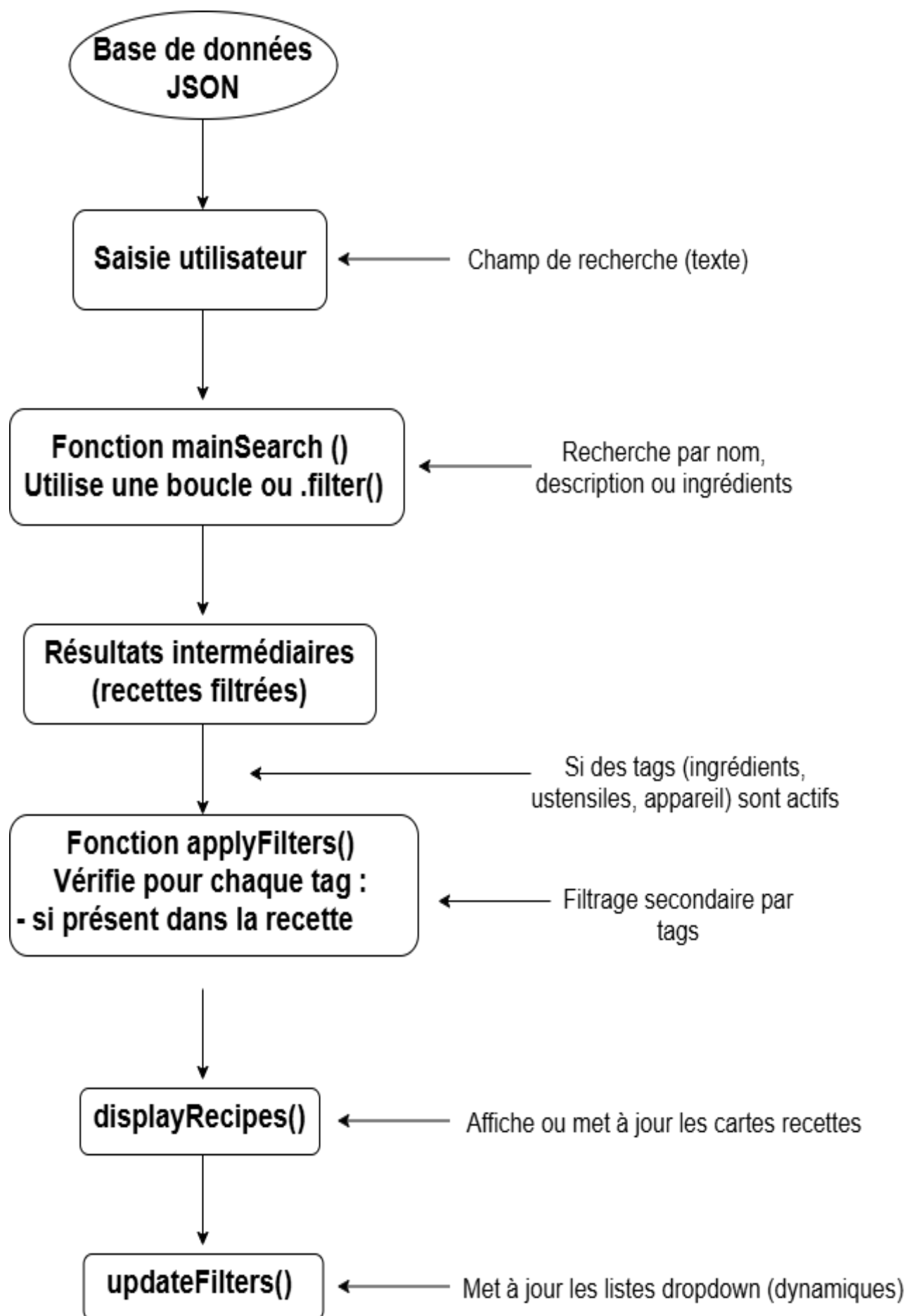
Dans cette option, nous utilisons des **boucles classiques**, comme `for`, `for...of`, et `while`, ainsi que des instructions `if`, `break`, et `push()` pour construire les résultats. Ce choix permet une meilleure maîtrise de chaque étape du traitement et une plus grande flexibilité dans les cas personnalisés.

Avantages <ul style="list-style-type: none">⊕ Permet une gestion très fine des étapes⊕ Plus pédagogique et plus facile à suivre pour un développeur débutant⊕ Meilleur pour optimiser manuellement certaines portions de code	Inconvénients <ul style="list-style-type: none">⊖ Syntaxe plus longue et plus verbeuse⊖ Moins lisible sur des traitements complexes imbriqués⊖ Risque de duplication de logique (boucles imbriquées fréquentes)
---	---

Solution retenue :

La version array-method a été retenue pour sa **lisibilité**, sa **modularité** et sa **concision**, permet une mise en œuvre rapide et maintenable du moteur de recherche, tout en répondant aux besoins du projet.

La version loop-method est utile pour former de nouveaux développeurs ou pour des cas où un **contrôle très fin des performances** est requis.



Description

Setup block (useful for function initialization. it will be run before every test, and is not part of the benchmark)

```
1 const recipes = [
2   { name: "Tarte au citron", ingredients: [{ ingredient: "citron" }, { ingredient: "sucre" } ],
3   { name: "Salade", ingredients: [{ ingredient: "tomate" }, { ingredient: "concombre" } ] },
4   { name: "Citronnade", ingredients: [{ ingredient: "eau" }, { ingredient: "citron" } ] },
5   { name: "Omelette", ingredients: [{ ingredient: "oeuf" }, { ingredient: "beurre" } ] },
6 ];
7
8 const query = "citron";
```

ADD LIBRARY

Boilerplate block (code will be executed before every block and is part of the benchmark. use it for data initializing)

array-method

-

loop-method

```
1 const result = [];
2 for (let i = 0; i < recipes.length; i++) {
3   const ingredients = recipes[i].ingredients;
4   for (let j = 0; j < ingredients.length; j++) {
5     if (ingredients[j].ingredient.toLowerCase().includes(query)) {
6       result.push(recipes[i]);
7     }
8   }
9 }
10 }
```

-

+

result

array-method (7424584)

100%

loop-method (7326823)

98.68%

no title (put title and/or keywords here, which describes your test)

RUN TESTS

GENERATE PAGE URL

NEW BENCHMARK

Description

Setup block (useful for function initialization. it will be run before every test, and is not part of the benchmark)

```
1 const recipes = [
2   { name: "Tarte au citron", ingredients: [{ ingredient: "citron" }, { ingredient: "sucre" } ],
3   { name: "Salade", ingredients: [{ ingredient: "tomate" }, { ingredient: "concombre" } ] },
4   { name: "Citronnade", ingredients: [{ ingredient: "eau" }, { ingredient: "citron" } ] },
5   { name: "Omelette", ingredients: [{ ingredient: "oeuf" }, { ingredient: "beurre" } ] },
6 ];
7
8 const query = "citron";
```

ADD LIBRARY

Boilerplate block (code will be executed before every block and is part of the benchmark. use it for data initializing)

array-method

```
1 const result = recipes.filter(recipe =>
2   recipe.ingredients.some(ing => ing.ingredient.toLowerCase().includes(query))
3 );
```

-

loop-method

-

+

result

array-method (7424584)

100%

loop-method (7326823)

98.68%