

Fiche d'investigation fonctionnalité

Fonctionnalité : Search Recipes

Fonctionnalité #2

Problématique : Pour rechercher des recettes, l'utilisateur peut utiliser une barre de recherche principale ainsi que des menus déroulants pour filtrer les recettes affichées (cf. figure 1). Nous voulons optimiser la recherche principale et déterminer quel type d'algorithme utiliser.

Option 1 : Utilisation des boucles natives

Dans cette option, nous utilisons des boucles natives (while, for...) pour la recherche principale.

Code :

```
// filters the recipes based on the main criteria (search input)
this._sortedRecipes = [];

for (let i = 0; i < this._recipes.length; i++) {
  const recipe = this._recipes[i];
  if (
    recipe.search(this._criteria) ||
    recipe.searchIngredient(this._criteria) ||
    recipe.searchUstensil(this._criteria) ||
    recipe.searchAppliance(this._criteria) ||
    recipe.searchDescription(this._criteria)
  ) {
    this._sortedRecipes.push(recipe);
  }
}
```

Option 2 : Utilisation des méthodes de l'objet array

Dans cette option, nous utilisons des méthodes de l'objet array (foreach, filter, map, reduce) pour la recherche principale.

Code :

```
// filters the recipes based on the main criteria (search input)
this._sortedRecipes = this._recipes.filter(recipe =>
  recipe.search(this._criteria) ||
  recipe.searchIngredient(this._criteria) ||
  recipe.searchUstensil(this._criteria) ||
  recipe.searchAppliance(this._criteria) ||
  recipe.searchDescription(this._criteria)
);
```

Solution retenue :

L'option 1 donne de meilleurs résultats de performances pour la recherche principale. Ce sera donc la solution utilisée. (cf. figure 2)

Annexes

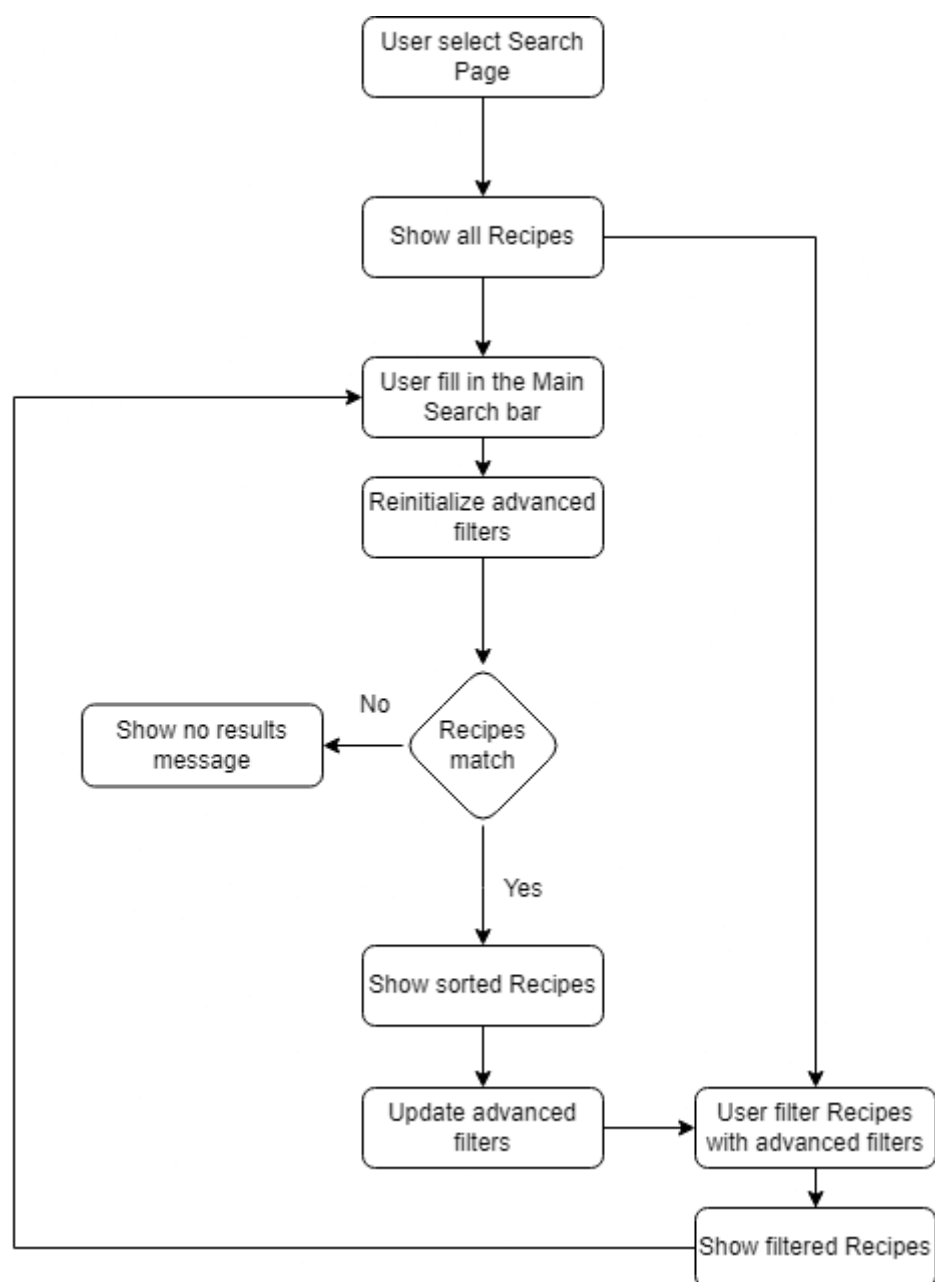


Figure 1 : Diagramme pour la recherche des recettes

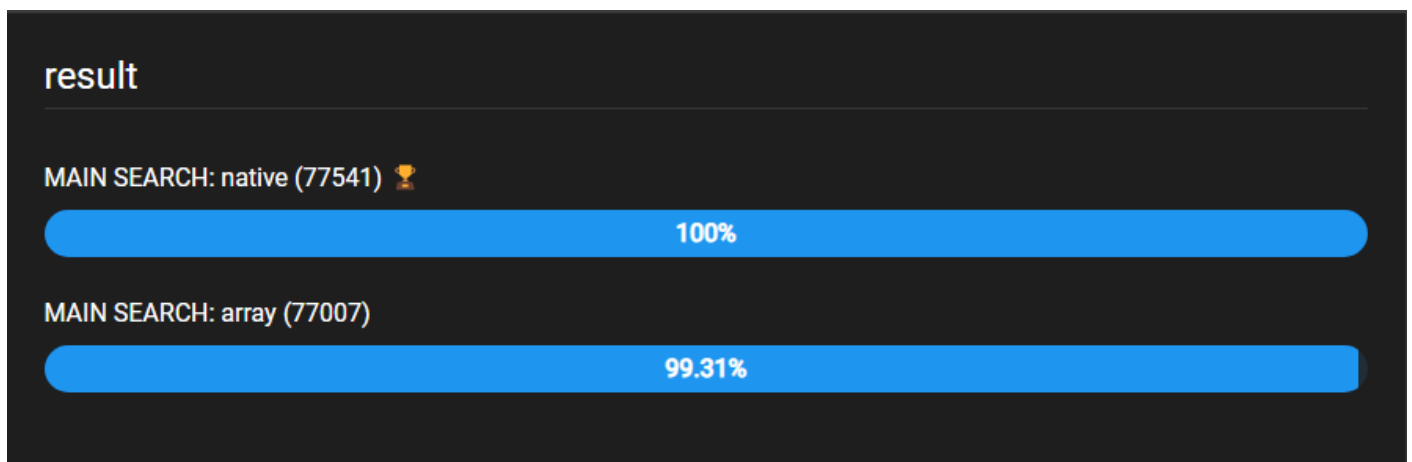


Figure 2 : Résultats du test de performance des deux options