

## Fiche d'investigation de fonctionnalité

**Fonctionnalité :** Fiche d'investigation de fonctionnalité de recherche générale

**Problématique :** Développer des algorithmes de recherche performants qui permettent une recherche de recette rapide et quasi instantanée a ces utilisateurs.

### Option 1 : MainSearchV1 en programmation fonctionnelle avec filter()

Effectue une recherche de recettes en filtrant un tableau de recettes "recipesArray" en fonction de mots clés donnés par l'utilisateur dans une zone de recherche.

#### Avantages

- Plus simple à comprendre et à maintenir en raison de la structure du code

#### Inconvénients

- Plus d'operations ce qui peut rendre le processus de filtrage plus lent.

**Nombre de champs minimum pour lancer la recherche : 3**

### Option 2 : mainSearchV2 en algorithme de tri avec boucle for

La fonction utilise une boucle for pour parcourir un tableau de recettes. "ismatch" vérifie si tous les mots recherchés par l'utilisateur se trouvent dans les informations d'une recette. Si oui, alors la recette est considérée comme une correspondance et est ajoutée à un tableau de recettes filtrées.

#### Avantages

- L'algorithme de tri rend la recherche plus rapide a executer.

#### Inconvénients

- Plus dur à comprendre en raison de la structure plus complexe de l'algorithme.
- Le code n'est pas très flexible et peut être difficile à maintenir si vous voulez ajouter de nouveaux éléments de recherche

**Nombre de champs minimum pour lancer la recherche : 3**

### Solution retenue :

Je choisis d'utiliser la fonction mainSearchV1 car bien que les résultats de performance soient légèrement inférieurs à l'option 2, la différence est si minime (voir resultat JSBen.ch) qu'elle n'a aucun impact significatif pour 50 recettes et est imperceptible pour les utilisateurs. Cependant, à grande échelle, il serait préférable d'utiliser mainSearchV2.

En outre, le code de mainSearchV1 est plus facile à comprendre et plus lisible car il n'y a aucun algorithme complexe visible dans la fonction, ce qui la rend plus simple et plus maintenable..

## Annexes

Figure 1 - Diagramme d'activité Classic login Workflow

**Figure 2 : Approche “Email First” et usage de Google Identity Toolkit**

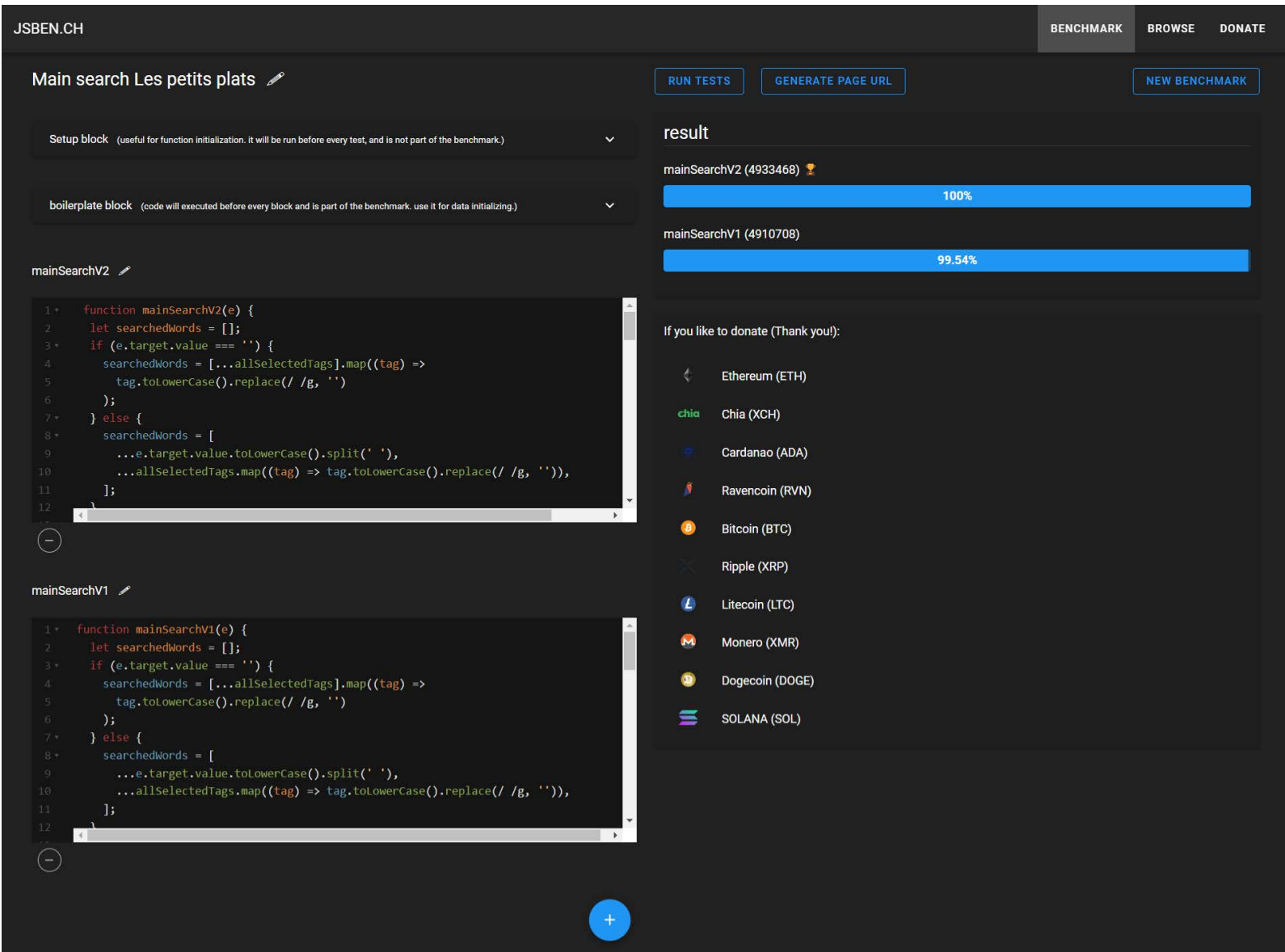


Figure 3 : Comparatif de performance des options 1 et 2 sur JSBench.js