Les petits plats

Fiche d'investigation de fonctionnalité

Fonctionnalité : Algorithme de recherche	**Fonctionnalité #1**

Problématique : Dans le souci de satisfaire au mieux les attentes des utilisateurs du site « Les petits plats » on cherche à implémenter un moteur de recherche le plus fluide et rapide possible, adapté à la base de données du site.

La recherche s'effectue à la fois dans le titre, la description et les ingrédients de chaque recette sur le mot clé tapé dans la barre de recherche principale, ainsi que sur les tags sélectionnés (cf. Annexe 2). Les résultats sont affinés à chaque changement dans les critères de recherche(ajout/suppression).

Option 1 : Approche fonctionnelle

Dans cette option, nous proposons une approche fonctionnelle avec les méthodes d'object array « foreach », « filter » et « includes » pour parcourir les données et filtrer les recettes correspondantes à tous les critères de recherche. La fonction retourne un tableau avec les ids des recettes correspondantes ou bien le tableau vide le cas échéant (cf. Annexe 3).

Avantages	**Inconvénients**
+ code flexible, concis, très lisible + meilleure maintenabilité + efficace sur petits/moyens volumes de données	- disponible nativement à partir de Firefox 13, Chrome 38, Opera25, Edge12, Safari7 et Node0.12, mais transpilable via Babel pour le reste - performance est altérée sur de très grands tableaux (1M+ éléments)

Option 2 : Approche avec les boucles natives « for », « for…in »

Dans cette option, nous proposons une approche 'classique' avec la séquence d'instructions qui est composée de boucles « for » imbriquées, de structures conditionnelles « if » et d'affectations de variables de valeurs booléennes. La fonction retourne un tableau créé avec les ids des recettes correspondantes à tous les critères de recherche ou le tableau vide le cas échéant (cf. Annexe 3).

Avantages	**Inconvénients**
+ compatible avec tous les navigateurs + meilleure performance d'itérations + plus efficace sur de très grands volumes de données	- code plus verbeux - code plus complexe à comprendre - impact négatif sur la maintenabilité

Solution retenue :

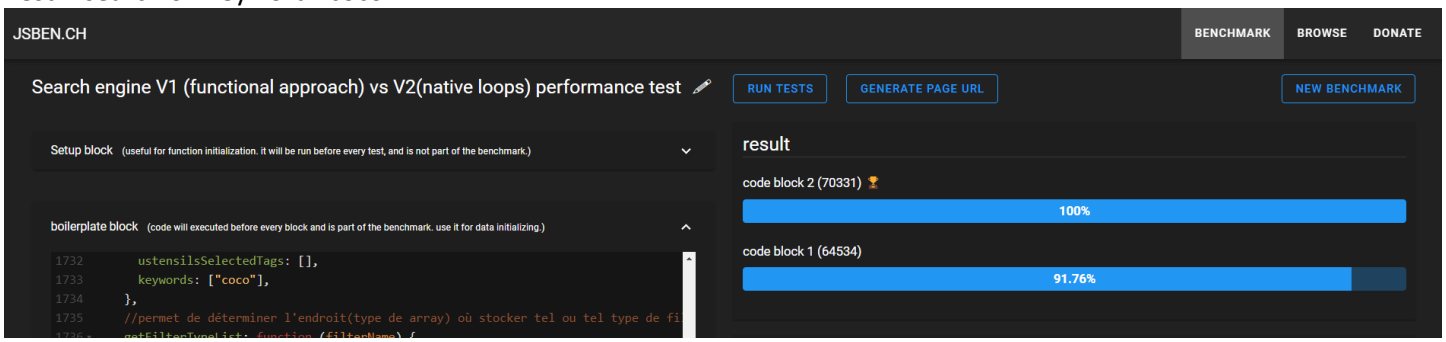
Bien que l'échantillon de 50 recettes soit assez réduit pour comparer les différences de performances entre 2 approches, compte tenu des objectifs imposés pour ce projet sur la rapidité de la fonctionnalité de recherche, l'approche avec les boucles natives sera retenue pour son meilleur taux de performance (cf. Annexe 1), en dépit des avantages non négligeables offerts par l'approche fonctionnelle.

Annexe 1

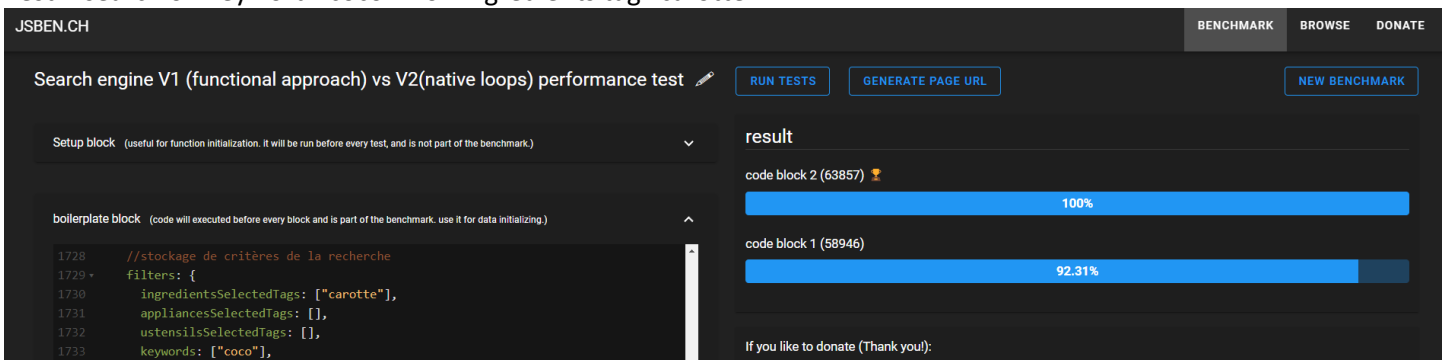
Benchmark test results on JSBEN.ch

Context: code block 1: search function with array methods
code block 2: search function with native loops

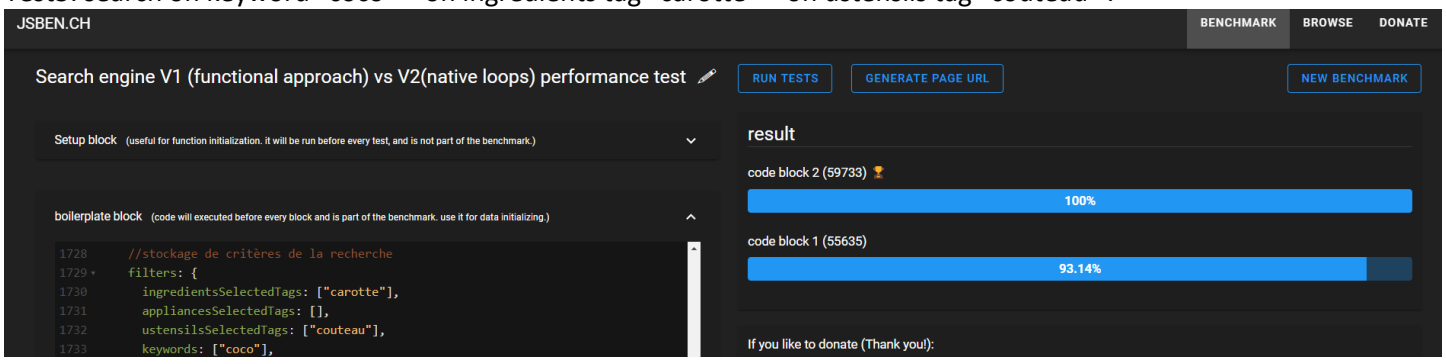
Test1: search on keyword "coco":



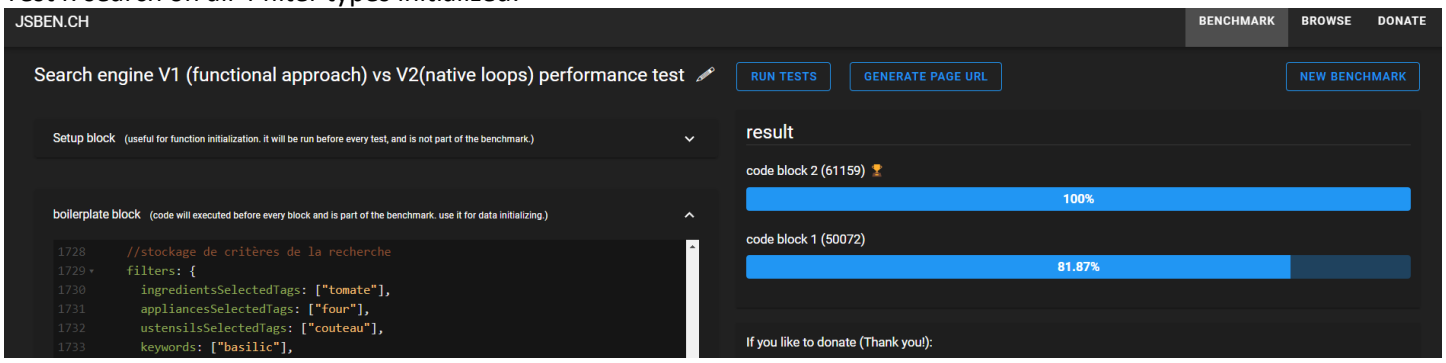
Test2: search on keyword "coco" + on ingredients tag "carotte":



Test3: search on keyword "coco" + on ingredients tag "carotte" + on ustensils tag "couteau" :

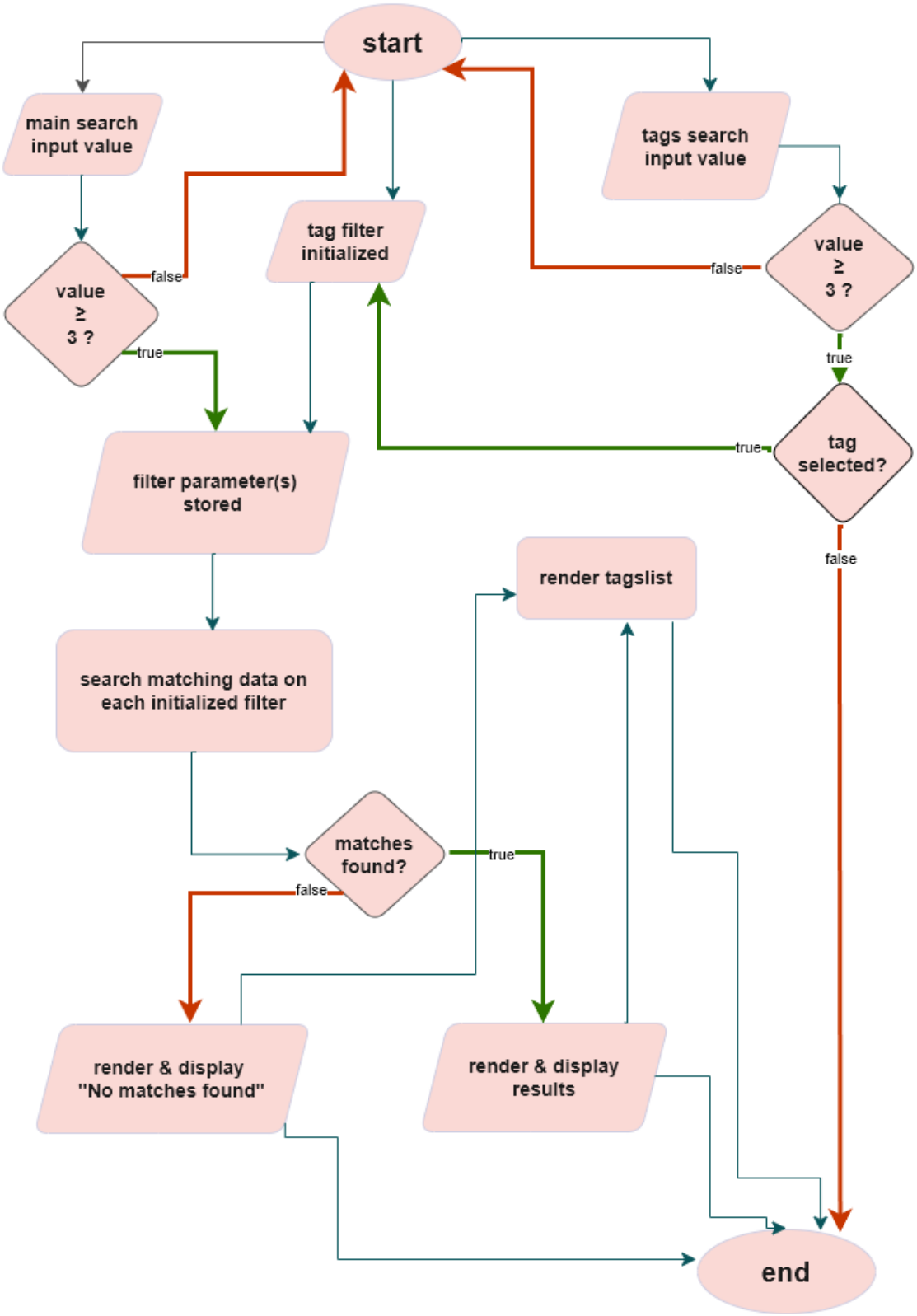


Test4: search on all 4 filter types initialized:



Annexe 2

Search engine algorithm logic



Annexe 3

Search engine function details