

Fiche d'investigation de fonctionnalité

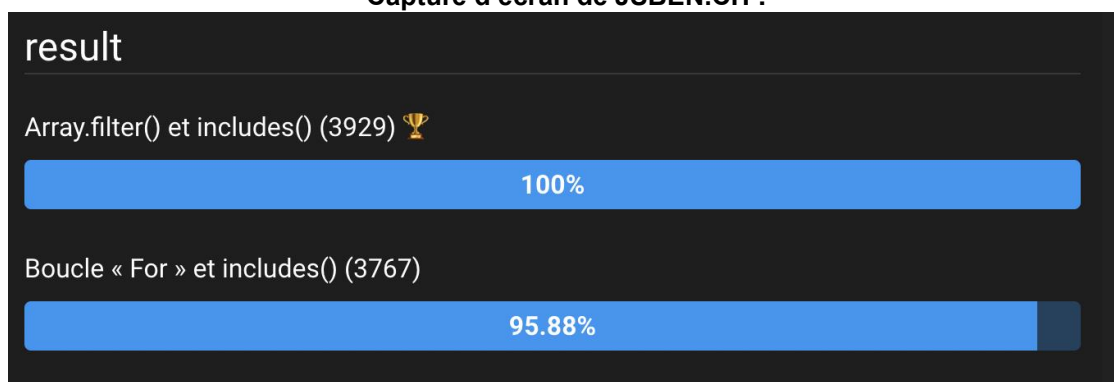
Fonctionnalité : Algorithme de recherche des recettes	Fonctionnalité #1
Problématique : Afin de faire la différence avec les sites de recettes concurrents, nous cherchons à avoir une recherche la plus rapide et fluide possible.	

Option 1 : Array.filter() et includes() Dans cette option, nous utilisons la fonction array.filter() pour directement trier les recettes des tableaux. L'avantage est que le code est court et simple à comprendre, de plus celui-ci est le plus rapide des deux algorithmes.	
Avantages <ul style="list-style-type: none"> • Code très court et simple • Algorithme le plus rapide (~3929 ops selon jsben.ch) 	Inconvénients <ul style="list-style-type: none"> • RAS

Option 2 : Boucle « For » et includes() Dans cette option, nous utilisons une boucle « for » pour vérifier chaque recette présente dans les tableaux, si l'élément correspond, il est envoyé dans un nouveau tableau qui sera utilisé pour le rendu.	
Avantages <ul style="list-style-type: none"> • RAS 	Inconvénients <ul style="list-style-type: none"> • Ajoute un nouveau tableau • Code un peu moins court et simple • Algorithme moins rapide (~3767 ops selon jsben.ch)

Solution retenue : Etant donné que nous recherchons l'algorithme le plus performant, le choix se porte donc sur l'option 1, c'est-à-dire « Array.filter() et includes() », car celle-ci est en moyenne légèrement plus rapide selon les différents tests effectués sur Jsben.ch. https://jsben.ch/mPXfH

Capture d'écran de JSBEN.CH :



Annexes

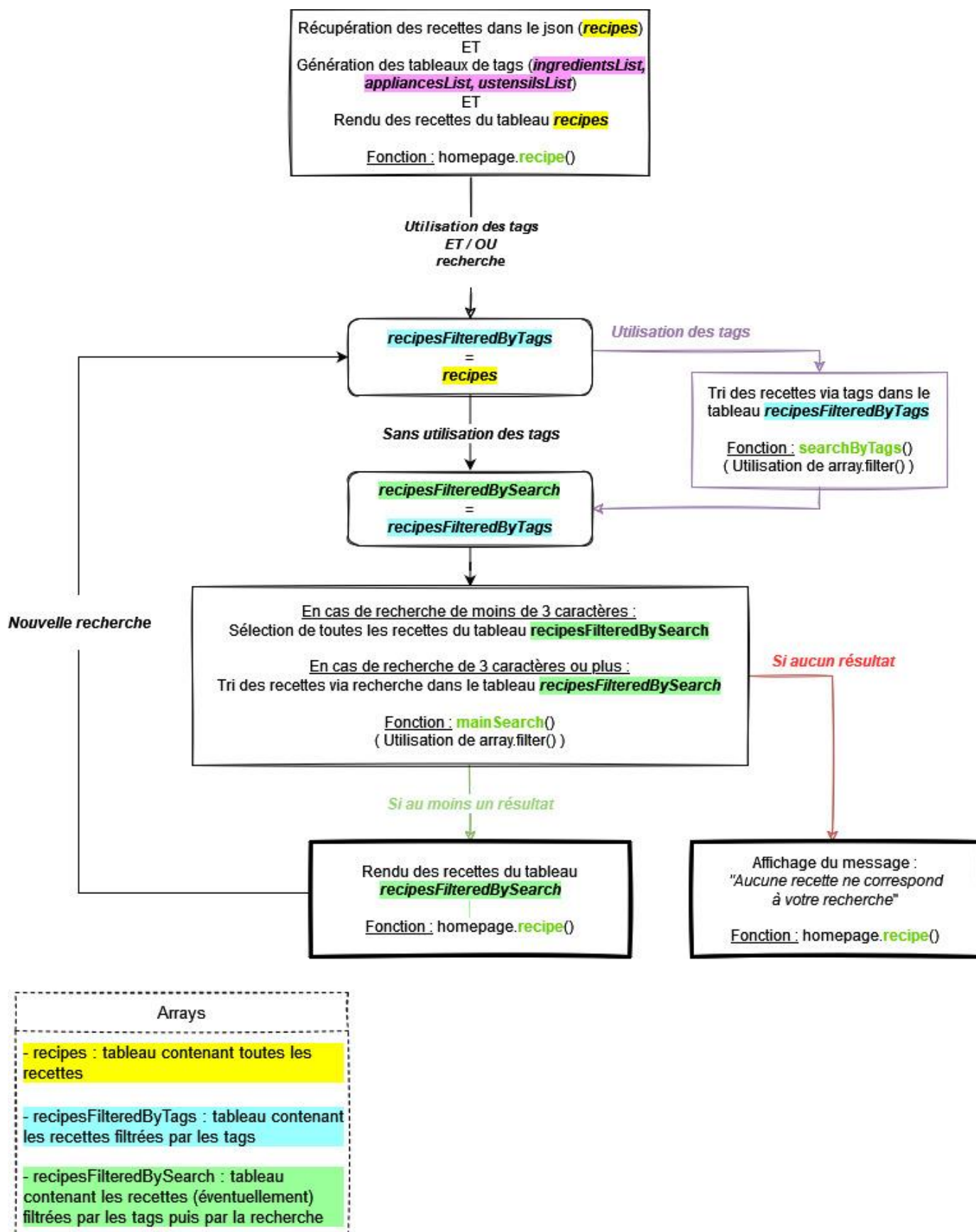


Figure 1 - Array.filter() et includes()

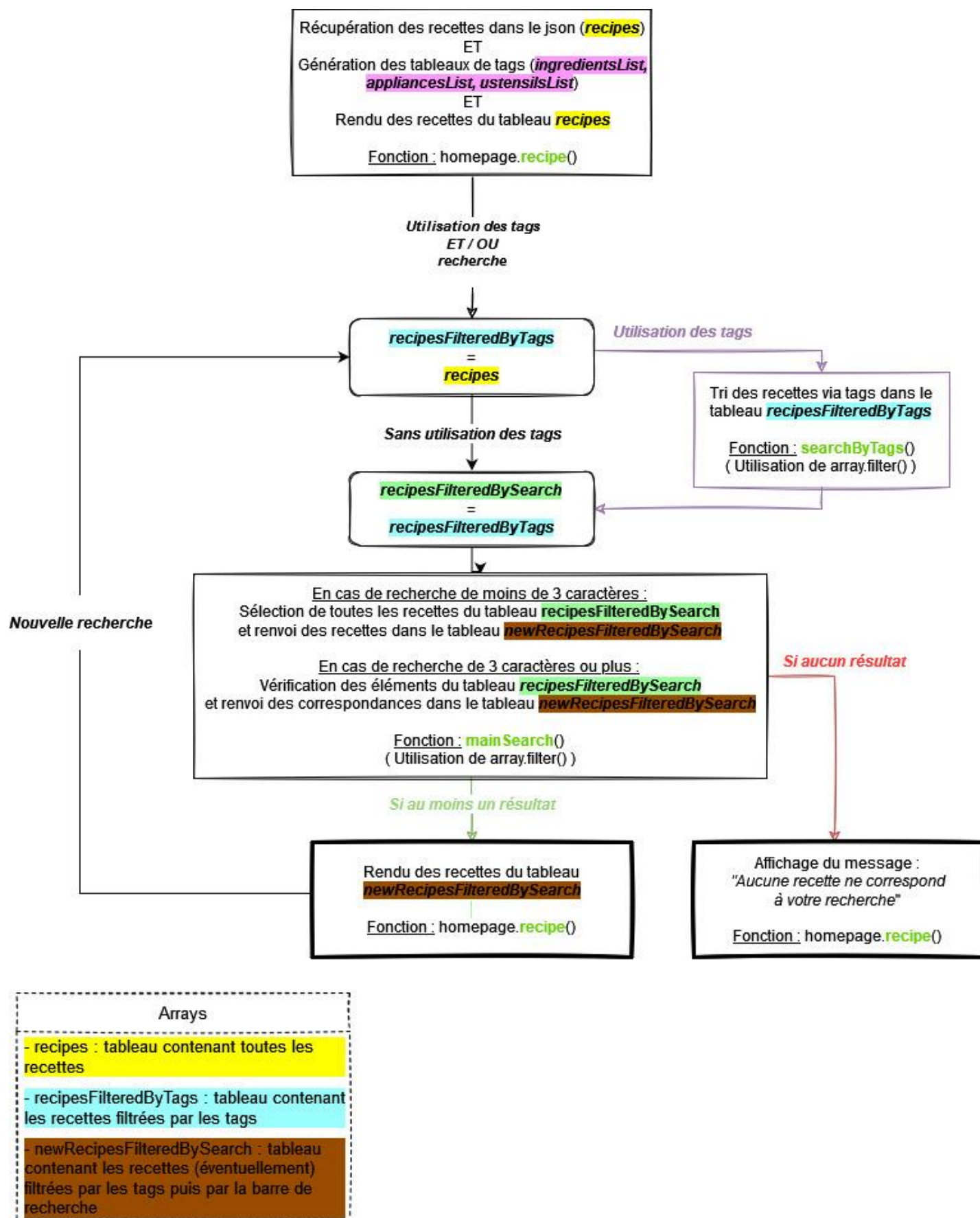


Figure 2 - Boucle « For » et includes()