

Fiche d'investigation de fonctionnalité

Fonctionnalité : Filtrer des recettes	Fonctionnalité #1
Problématique : Pour proposer à l'utilisateur les recettes correspondantes à ses requêtes le plus rapidement possible, une investigation est mise en place afin de comparer des fonctions native de javascript : la méthode .filter et la boucle for. La boucle correspondante à la recherche globale a été isolé pour être soumise aux comparateurs du nombre d'opérations par seconde javascript : jsben.ch. Enfin, nous allons voir que le contexte, c'est-à-dire le nombre de données à traiter impacte sur les résultats des tests.	

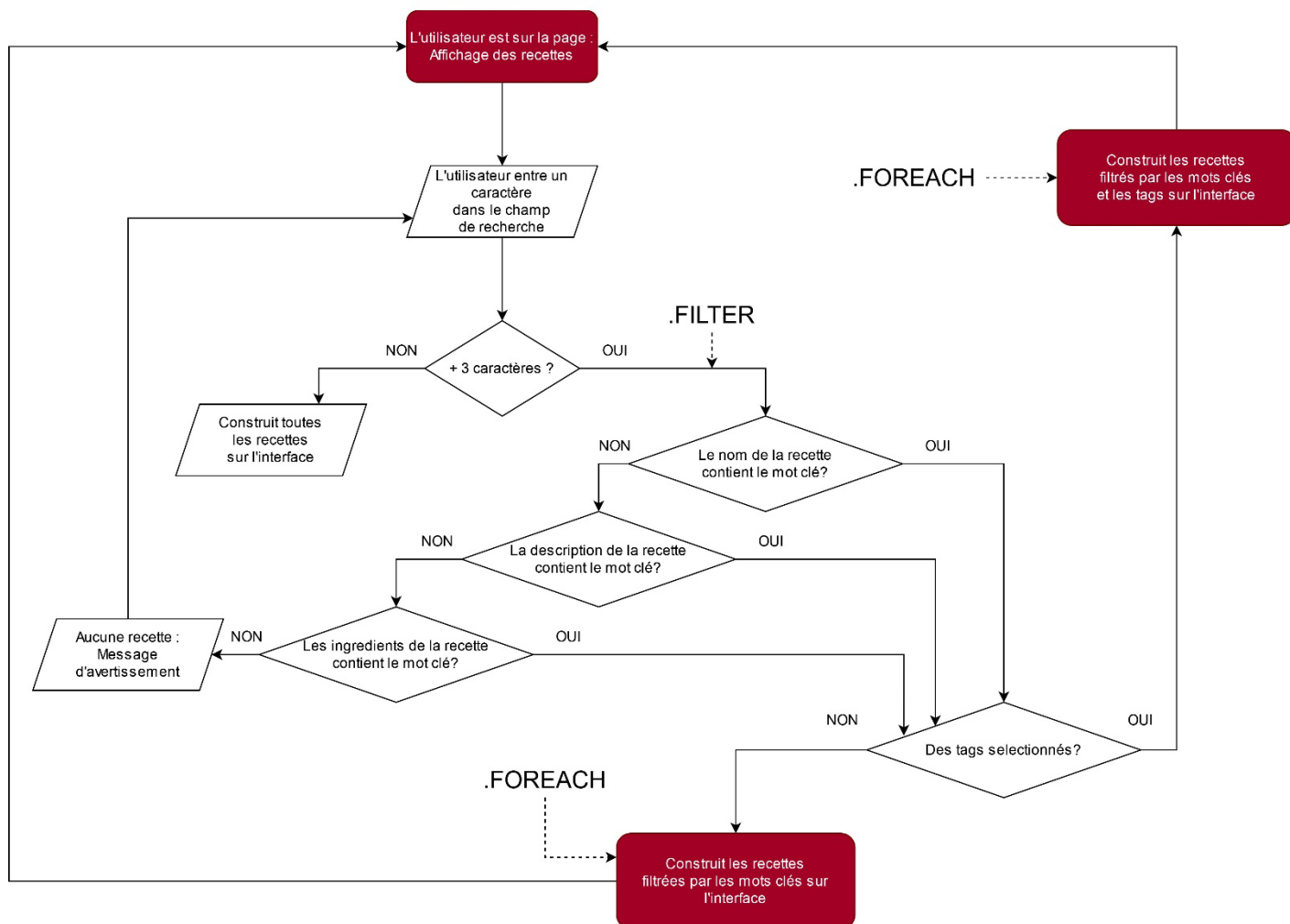
Option 1 : La méthode .FILTER de Javascript Les premiers résultats sont radicales, .filter devance largement la boucle FOR sur les cinquante recettes. Le fonctionnement de .FILTER permet de retourner un tableau filtré avec une fonction fléchée indiquant la condition a respectée. Des callbacks sont effectués sur chaque élément parcouru. Valeur, index, tableau parcouru sont ainsi retournés et utilisables à l'intérieur de la boucle elle-même.	
Avantages <ul style="list-style-type: none">⊕ rapide sur de petits fichier de données⊕ lisibilité facilitée	Inconvénients <ul style="list-style-type: none">⊖ performance diminuée proportionnellement à la quantité de données en traitement
Nombre d'opérations très élevé sur de données limités à un certain nombre http://jsben.ch/xPQ16	

Option 2 : La Boucle FOR Au premier abord, la boucle For n'est pas aussi rapide que .filter. Au fil des tests, il est observé qu'au fur et à mesure que les données grandissent, son écart de score avec la méthode .filter augmente proportionnellement avec la taille des donnée. L'utilisation de la boucle For est très personnalisable et nous pouvons limiter ses retours qu'au stricte nécessaire.	
Avantages <ul style="list-style-type: none">⊕ Minimaliste⊕ Performante sur des fichiers de données conséquentes	Inconvénients <ul style="list-style-type: none">⊖ écriture plus lourde
Rapidité pas relevable sur des petits fichiers de données mais considérable sur des grands. http://jsben.ch/xPQ16	

Solution retenue : Sur cet algorithme et ce fichier de données, la méthode .filter devance la boucle FOR. Cependant, l'écart se réduit à mesure que les données s'enrichissent. Sur le long terme et sur des données conséquentes, la boucle FOR devance la methode .filter. Etant donné que ce projet est destiné à être rapide dans le traitement de beaucoup de données, il est judicieux de travailler des maintenant avec la boucle FOR plutôt que de changer d'avis en production pendant que la base de données s'étoffe. Ainsi la boucle FOR est une fonction native de javascript incontournable pour boucler sur des tableaux riches en contenu.
--

Annexes

Algorithme : .FILTER



Algorithme : BOUCLE FOR

