

Fiche d'investigation de fonctionnalité

Fonctionnalité : Filtre / Recherche personnalisée	Fonctionnalité #1
Problématique : Création d'un algorithme de recherche qui récupère des données json et offre à l'utilisateur la possibilité de filtrer l'affichage de recherche en fonction de mots clefs via un input ou des boutons.	

Option 1 : Filtrage à l'aide de boucle For Dans cette option, nous avons une page qui permet de filtrer de façon rapide et optimal les recettes via le input recherche et 3 boutons spécifiques. Ces couches de filtres sont constamment actualisé et mis à jour.	
Avantages <ul style="list-style-type: none">⊕ Filtrage dynamique⊕ Site répond aux besoins client	Inconvénients <ul style="list-style-type: none">⊖ Maintenance du site plus compliqué pour la compréhension du code
Création d'une classe qui reprend plusieurs données : 7 Nom des recettes, images, description, ingrédients, temps, appareils, ustensiles. Nombre de tableaux : 3 Ingrédients, appareils et ustensiles	

Option 2 : Filtrage à l'aide de boucle ForEach Dans cette option, nous avons une page qui permet de filtrer de façon rapide et optimal les recettes via le input recherche et 3 boutons spécifiques. Ces couches de filtres sont constamment actualisé et mis à jour.	
Avantages <ul style="list-style-type: none">⊕ Pas d'index donc plus lisible⊕ Adapter à la manipulation de tableau	Inconvénients <ul style="list-style-type: none">⊖ Moins performant
Création d'une classe qui reprend plusieurs données : 7 Nom des recettes, images, description, ingrédients, temps, appareils, ustensiles. Nombre de tableaux : 3 Ingrédients, appareils et ustensiles	

Solution retenue : Nous avons donc retenu l'approche ForEach bien que les performances en comparatif sont légèrement en dessous avec For ou While, le code est plus facilement lisible et facilite la manipulation avec les tableaux. Dans la mesure où ce projet n'est pas volumineux nous préconisons cette méthode.
--

Annexes

Figure 1 - Diagramme du fonctionnement de l'algorithme



