

Les petits plats

Fiche d’investigation fonctionnalité #1

Fiche d’investigation de fonctionnalité

|  |  |
| --- | --- |
| **fonctionnalité** : Algorithme de recherche en JavaScript | **Fonctionnalité #1** |
| **Problématique :** Afin d’implanter un moteur de recherche en JavaScript, nous cherchons l’algorithme le plus adapté.  L’algorithme devra faire des recherches à la fois dans les titres, description et les ingrédients de recettes.  Les recherches des utilisateurs se feront à travers un champ de recherche principal et également par intermédiaire de filtres. | |

|  |  |
| --- | --- |
| **Option 1 : Recherche linéaire avec une boucle For (cf Figure)**  Dans cette option, nous proposons une recherche linéaire (Linear Search) avec des boucles natives “for”. Elle a l’avantage d’être un algorithme classique de force brute, travaillant directement sur des tableaux non triés. Le principal inconvénient : le processus n’est pas adapté pour les grands tableaux. | |
| **Avantages**   * Travail directement sur des tableaux non triés. * Compatible avec tous les navigateurs. * Efficace sur de petites listes. | **Inconvénients**   * Code très verbeux. * N’utilise aucune logique pour réduire les éléments de recherche. * Très peu efficace pour des recherches répétées sur de grands tableaux. |
| **Capture d’écran 2022-05-05 à 13.42.06Test d’efficacité:** | |

|  |  |
| --- | --- |
| **Option 2 : Recherche avec la méthodes d’objet array “filter” et “some”**  Selon le revoie booléen de some(), filter() va créer et retourner un nouveau tableau, plutôt que de modifier les données initiales.  Ce n’est pas algorithme de recherche le plus performant mais il a l’avantage d’être flexible et lisible. | |
| **Avantages**   * Code plus flexible, concis et donc plus lisible. | **Inconvénients** |
|  | |

**Solution retenue :**