

Fiche d'investigation de fonctionnalité

Fonctionnalité : Recherche Principale	Fonctionnalité #1
Problématique : Obtenir l'algorithme de recherche le plus rapide	

Option 1 : La méthode natives

Dans cette option, nous utilisons, pour l'algorithme de notre recherche, les méthodes natives (boucles for, while ...). Cette méthode nous permet de voir comment se comporte la fonction et la façon dont les données sont découpées et traitées. L'inconvénient de cette méthode réside dans le fait qu'elle est plus fastidieuse à écrire.

Avantages

- Compréhension du découpage des données
- Meilleure prise en main pour les autres développeurs

Inconvénients

- Conception fastidieuse

Réalisation de 3 boucles + une fonction de comparaison des données

Option 2 : La méthode array

Dans cette option, nous utilisons, pour l'algorithme de notre recherche, les méthodes array (foreach, reduce ...). L'avantage de cette méthode est qu'elle est simple à comprendre et écrire. En revanche, c'est javascript qui s'occupe du traitement des données, il pourrait être difficile de savoir comment elles sont traitées jusqu'au résultat.

Avantages

- Conception simple

Inconvénients

- Si les concepts Javascript ne sont pas bien assimilés la méthode peut être mal utilisée.

Utilisation d'une seule boucle et d'un opérateur de comparaison

Solution retenue :

La solution retenue est celle de la méthode native. En effectuant un test JSBENCH, c'est cette méthode qui est apparue la plus rapide (cf ci dessous)

Méthode Array

finished

1,1 Md ops/s \pm 0.19%
0.51 % slower

```
function recherchePrincipale(data, input) {  
  // initialisation du nouveau tableau qui va accueillir les datas triées  
  const resultatRechcerche = [];  
  
  // Tous les résultats du tri sont transformés en minuscules pour éviter les différences de casse  
  const inputLowerCase = input.toLowerCase();  
  data.forEach((recette) => {  
    // On transforme les données en tableau avec flatMap  
    const lowerCaseIngredients = recette.ingredients.flatMap((recipe) =>  
      recipe.ingredient.toString().toLowerCase()  
    );  
    const lowerCaseTitre = recette.name.toString().toLowerCase();
```

☐ DEFER

Méthode Natives

finished

1,1 Md ops/s \pm 0.17%
Fastest

```
      correspondanceTrouvee = true,  
    }  
  }  
  
  if (correspondanceTrouvee) {  
    resultatRecherche.push(recette);  
    // Une correspondance a été trouvée dans cette recette, mais nous ne sortons pas de la boucle principale  
  }  
}  
  
return resultatRecherche;  
}
```

☐ DEFER

