

Fonctionnalité : Filtrer les recettes dans l'interface utilisateur

Problématique :

Pour permettre aux utilisateurs d'accéder rapidement à une recette correspondant à leurs besoins, nous cherchons à mettre en place un système de filtration des recettes dans l'interface utilisateur.

Option 1 : Utilisation de boucles natives (for)

Dans cette option, nous utilisons des boucles natives pour filtrer les recettes en fonction de la recherche de l'utilisateur.

Avantages :

- Contrôle précis sur le processus de filtrage.
- Facile à comprendre et à implémenter pour les développeurs familiers avec les boucles.

Inconvénients :

- Nécessite plus de code pour réaliser la même tâche qu'avec des méthodes de tableau.
- Moins élégant et plus sujet aux erreurs.

Option 2 : Utilisation de méthodes de l'objet array (filter, map, etc.)

Dans cette option, nous utilisons des méthodes de l'objet array telles que **filter**, **map**, etc., pour filtrer les recettes en fonction de la recherche de l'utilisateur.

Avantages :

- Code plus concis et lisible.
- Réduit le risque d'erreurs grâce à l'utilisation de méthodes spécifiquement conçues pour la manipulation des tableaux.
- Performances potentiellement meilleures grâce à l'optimisation interne des méthodes de tableau.

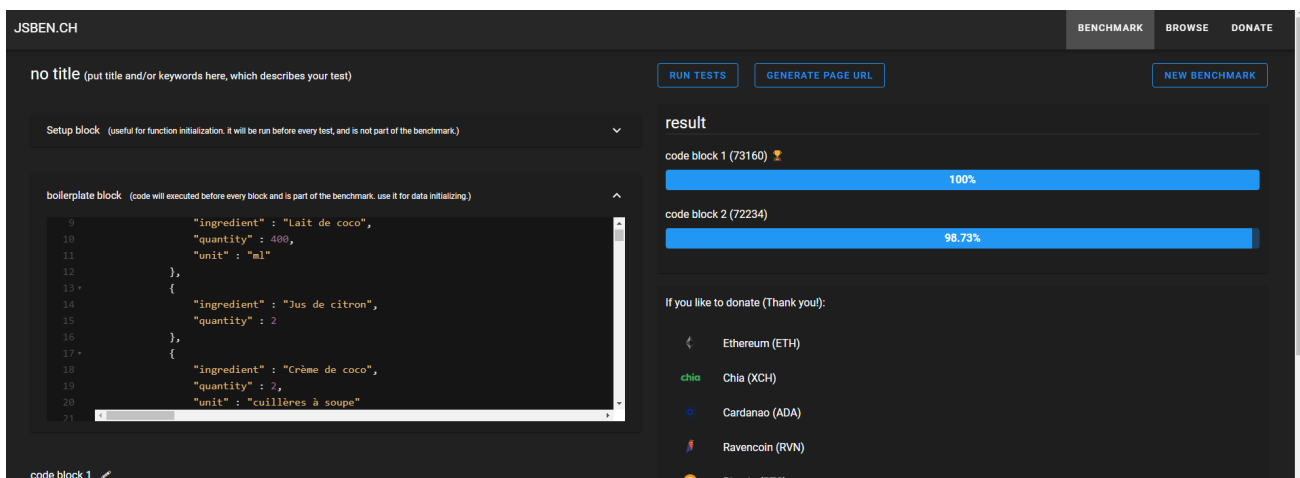
Inconvénients :

- Peut-être moins intuitif pour les développeurs moins familiers avec les méthodes de tableau.

Solution retenue :

Nous avons retenu l'utilisation de méthodes de l'objet array telles que **filter** pour filtrer les recettes dans l'interface utilisateur. Cette solution offre une combinaison de lisibilité, de concision et de performances, ce qui la rend idéale pour répondre aux besoins de notre application.

Comparaison de performance



```
code block 1
1 const recettesCorrespondantes = recipes.filter((recipe) => {
2   return (
3     recipe.name.toLowerCase().includes("coco") ||
4     recipe.description.toLowerCase().includes("coco") ||
5     recipe.ingredients.some((ingredient) =>
6       ingredient.ingredient.toLowerCase().includes("coco")
7     )
8   );
9 });

code block 2
1 let recettesCorrespondantes = [];
2 for (let i = 0; i < recipes.length; i++) {
3   const recipe = recipes[i];
4   if (
5     recipe.name.toLowerCase().includes("coco") ||
6     recipe.description.toLowerCase().includes("coco") ||
7     recipe.ingredients.some((ingredient) =>
8       ingredient.ingredient.toLowerCase().includes("coco")
9     )
10  ) {
11    recettesCorrespondantes.push(recipe);
12  }
13 }
```

