

Fiche d'investigation de fonctionnalité

Fonctionnalité: Filtrer les recettes

Problématique : Permettre de trouver une recette en prenant en compte 3 filtre (ingrédients, ustensiles et appareils) et une barre de recherche (qui sera prise en compte uniquement dans le cas ou 3 caractère ou plus sont dans la barre de recherche)

Option 1 : boucle for

Avantages :

- le code est plus détaillé
- plus familier

Inconvénients :

- beaucoup de ligne
- plus général (nécessité d'inspecter le code pour le comprendre)

Résultat du benchmark(<https://jsbench.me/>) :

- 5,6 k ops/s \pm 0.66%

Option 2 : map(), filter(), some()

Avantages :

- faible plus synthétique
- plus spécifique (on sait ce que fait le code dès qu'on voit la méthode)
- plus facile a maintenir (moins de variable nécessaire à l'extérieur de son scope)

Inconvénients :

-

Résultat du benchmark(<https://jsbench.me/>) :

- 115 k ops/s \pm 0.31%

Solution retenue :

Option 2

Les performances ont donnée des résultats montrant que l'option 1 est environs 95 % plus lente que l'option 2. De plus la lisibilité du code et la facilité de maintenance du code est également supérieur avec l'option 2, c'est donc elle qui sera retenu.

Setup HTML	<pre><div id="selected-keyword-ingredients">All</div> <div id="selected-keyword-appliances">Blender</div> <div id="selected-keyword-utensils"></div></pre>	
Setup JavaScript	<pre>const selectedUtensils = document.getElementById('selected-keyword-utensils'); const selectedIngredients = document.getElementById('selected-keyword-ingredients'); const selectedAppliances = document.getElementById('selected-keyword-appliances'); const actualRecipes = [{ "id": 1, "name": "Limónade de Coco", "servings": 1, "ingredients": ["1 litre d'eau", "1 citron", "100g de sucre", "100g de lait de coco"] }];</pre>	
for	<pre>function isIngredientInRecipe(wordsList, recipe) { let ingredientMatch = false; if (wordsList.length === 0) { return true; } if (wordsList.length > 0) { const recipeIngredients = recipe.ingredients; for (let j = 0; j < recipeIngredients.length; j++) { const ingredient = recipeIngredients[j].ingredient.toLowerCase(); if (wordsList.indexOf(ingredient) >= 0) { ingredientMatch = true; } } } }</pre>	<input type="checkbox"/> DEFER
finished		
5.6 k ops/s ± 0.66%		
95.12 % slower		
map	<pre>function isIngredientInRecipe(wordsList, recipe) { if (wordsList.length === 0) { return true; } const recipeIngredients = recipe.ingredients.map(ingredientObj => ingredientObj.ingredient.toLowerCase()); return wordsList.some(ingredient => recipeIngredients.includes(ingredient)); }</pre>	<input type="checkbox"/> DEFER
finished		
115 k ops/s ± 0.31%		
Fastest	<pre>function isUtensilsInRecipe(wordsList, recipe) { if (wordsList.length === 0) { return true; } }</pre>	
Test Case - click to add another test case		
Teardown JS - click to add teardown JavaScript		
Output (DOM) - click to monitor output (DOM) while test is running		
RUN again sort results		

