

Fiche d'investigation de fonctionnalité

Fonctionnalité : Recherche de recettes	Fonctionnalité #1
---	--------------------------

Problématique :

Filtrage des recettes dans l'interface utilisateur, l'utilisateur doit pouvoir accéder rapidement à la recette correspondant à sa recherche

Option 1 : Programmation fonctionnelle (annexe 1)

Utilisation des méthodes de l'objet Array (forEach, Filter...)

Emploi ici de la méthode « filter » qui filtre les recettes suivant la saisie effectuée et les correspondances trouvées dans le nom ou la description ou les ingrédients de la recette. La recette trouvée est ajoutée à un tableau qui servira à l'affichage des recettes.

De ce tableau, les différentes listes sont mises à jour.

Avantages

- code plus robuste et plus stable
- code plus court
- version plus rapide

Inconvénients

- code moins lisible

Saisie de 3 caractères minimum dans le champ de recherche principal

Option 2 : Programmation native (annexe 1)

Utilisation des boucles (while, for ...).

Ici utilisation de « for » qui itère sur le tableau des recettes et cherche s'il existe une correspondance entre la saisie, et le nom ou la description ou un des ingrédients de la recette.

Si oui, la recette en question est ajoutée à un nouveau tableau qui servira à l'affichage des recettes trouvées.

De ce tableau, les différentes listes sont mises à jour également

Avantages

- code plus lisible, plus facile à comprendre

Inconvénients

- code moins stable, plus long
- version plus lente

Saisie de 3 caractères minimum dans le champ de recherche principal

Solution retenue

Tests effectués sur 4000 recettes :

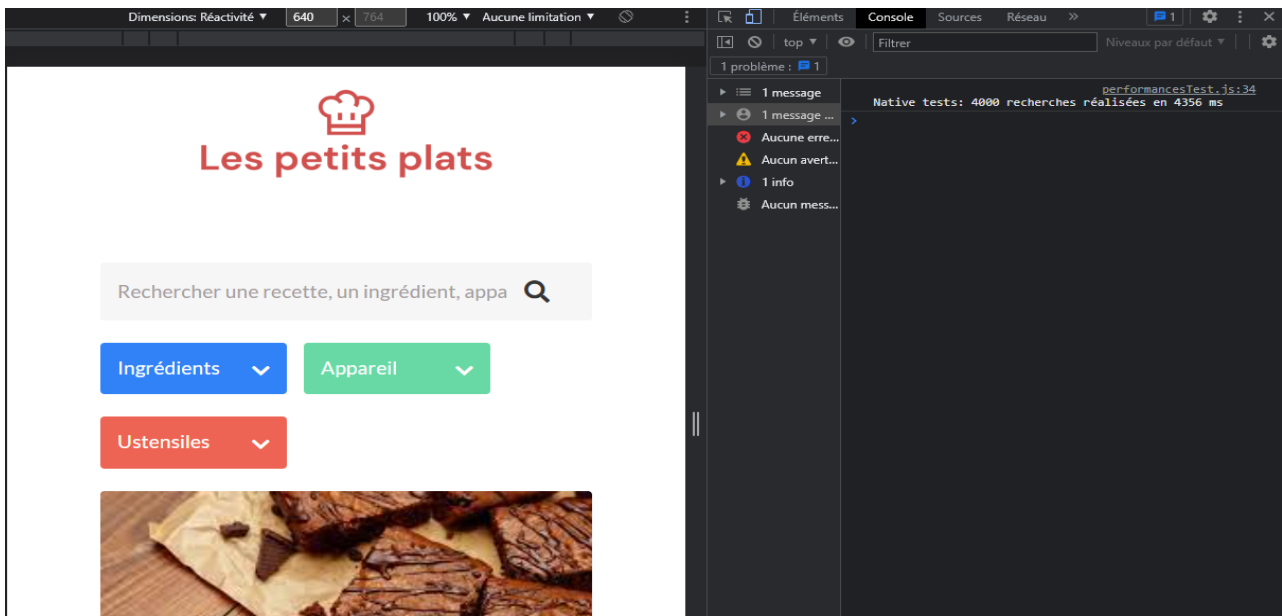
Tests avec console.time :

algorithmPerformance()

- programmation fonctionnelle avec filter : 4139 ms (en moyenne) - programmation native avec for : 4356 ms (en moyenne)

Notre choix se porte donc sur l'option 1, boucles avec Filter *et* forEach

The screenshot displays the 'Les petits plats' web application. The header features a red chef's hat icon and the text 'Les petits plats'. Below this is a search bar with the placeholder text 'Rechercher une recette, un ingrédient, appa' and a magnifying glass icon. Under the search bar are two buttons: 'Ingrédients' and 'Appareil'. To the right, the developer console is open, showing a message: 'Fonctionnel Test: 4000 recherches réalisées en 4139 ms'.



Rapport test JSBENSH :

JSBEN.CH

BENCHMARK

Setup block (useful for function initialization. It will be run before every test, and is not part of the benchmark.)

boilerplate block (code will executed before every block and is part of the benchmark. use it for data initializing.)

code block 1

```
1778 //----- Bensch N°1 -----//
1779 //*****
1780 let filteredRecipes = new Set(recipes);
1781
1782 const keywords = ["tarte", "beurre", "couteau"];
1783
1784 for (let keyword of keywords) {
1785   const recipesHasKeyWord = new Set();
1786
1787   for (let recipe of recipes) {
1788     if (
1789       recipe.name.toLowerCase().includes(keyword) ||
```

code block 2

```
1786
1787 keywords.forEach((keyword) => {
1788   let keywordHasRecipes = new Set();
1789
1790   keywordHasRecipes = recipes.filter((recipe) => {
1791     return (
1792       recipe.name.toLowerCase().includes(keyword) ||
1793       recipe.description.toLowerCase().includes(keyword) ||
```

result

code block 2 (19504) 100%

code block 1 (18452) 94.61%

AI Automatic Trader Bot up to %50 Profit On day

Signup and Get Free \$1 As Reward And Start Trade

ghooda.com

AI Automatic Trader Bot up to %50 Profit On day

Signup and Get Free \$1 As Reward And Start Trade

ghooda.com

AI Automatic Trader Bot up to %50 Profit On day

Signup and Get Free \$1 As Reward And Start Trade

ghooda.com

If you like to donate (Thank you!):

Ethereum (ETH)

Chia (XCH)

Cardanao (ADA)

Ravencoin (RVN)

Bitcoin (BTC)