Les petits plats

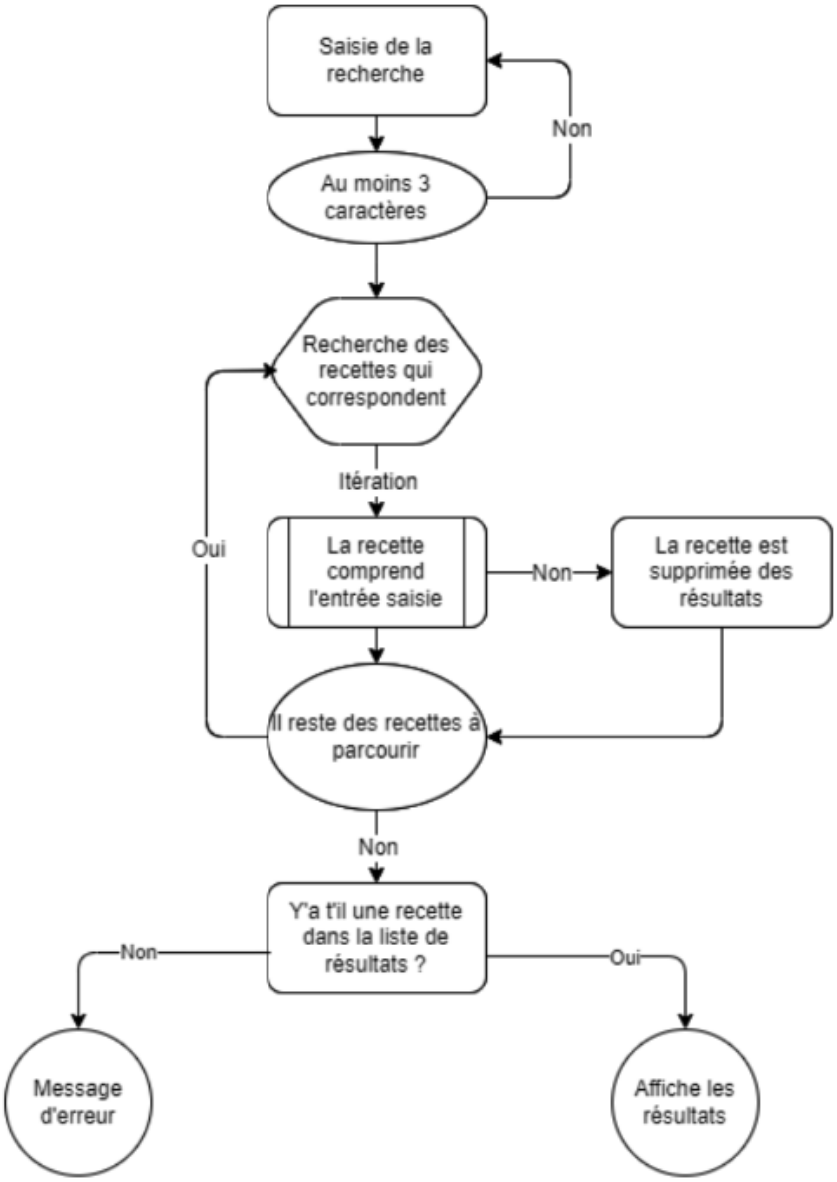
Fiche d'investigation de fonctionnalité

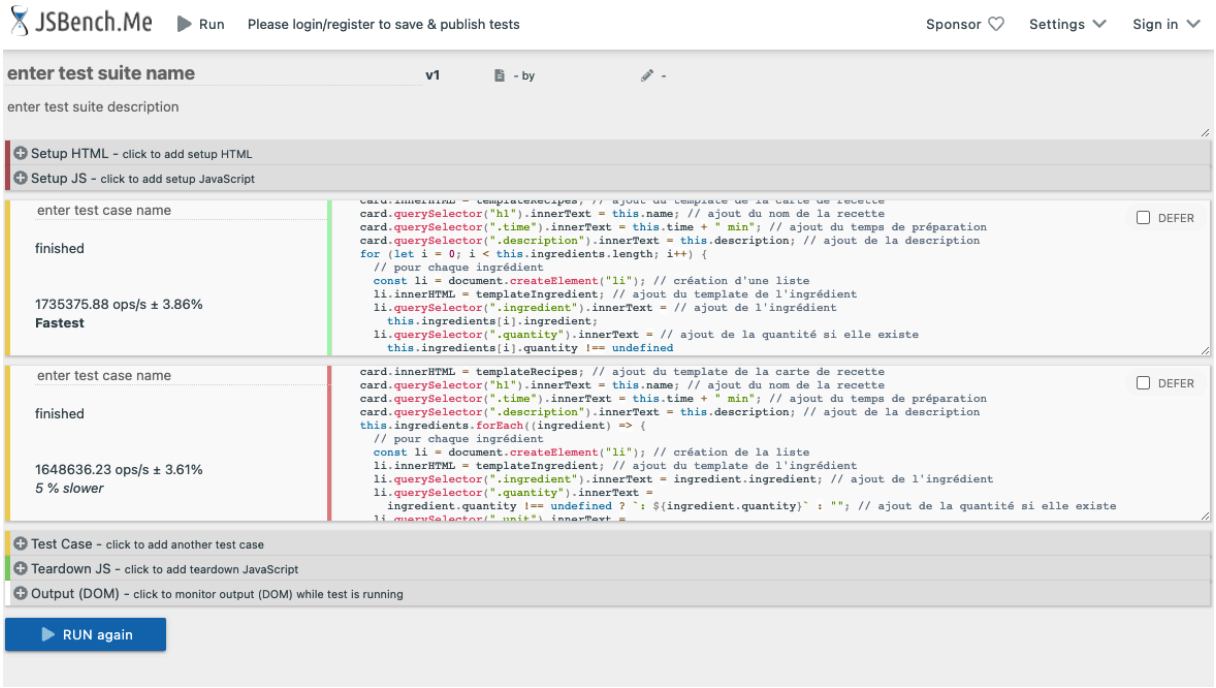
Fonctionnalité : Recherche principale	Fonctionnalité #1
Problématique : Afin de répondre aux besoins des utilisateurs, la recherche doit être la plus rapide possible.	

Option 1 : Programmation fonctionnelle Cette option utilise les méthodes de l'objet array (foreach, filter, map, reduce)	
Avantages : • Code plus compréhensible • Peu de changements en cas de nouveaux filtres	**Inconvénients :** • Incompatibilité avec certains navigateurs

Option 2 : Boucles natives Cette option utilise les boucles while et for	
Avantages : • Compatibiité avec tous les navigateurs	Inconvénients : • Code plus complexe, moins logique • Beaucoup de changements en cas de nouveaux filtres

Solution retenue :
Nous avons donc retenu l'approche fonctionnelle. Suite aux tests réalisés *(Annexe)* , nous avons pu constater que la programmation fonctionnelle est plus rapide que les boucles natives. De plus, la mise en place de nouveaux filtres étant simplifiée, maintenir le site sera facilité en cas de mises à jour.


Saisie de la recherche
Non
Au moins 3 caractères
Recherche des recettes qui correspondent
Itération
Oui
La recette comprend l'entrée saisie
Non
La recette est supprimée des résultats
Il reste des recettes à parcourir
Non
Y'a t'il une recette dans la liste de résultats ?
Non
Oui
Message d'erreur
Affiche les résultats


JSBench.Me
Run
Please login/register to save & publish tests
Sponsor
Settings
Sign in
enter test suite name
v1
- by
-
enter test suite description
Setup HTML - click to add setup HTML
Setup JS - click to add setup JavaScript
enter test case name
finished
1735375.88 ops/s ± 3.86%
Fastest
DEFER
card.querySelector("h1").innerText = this.name; // ajout du nom de la recette
card.querySelector(".time").innerText = this.time + " min"; // ajout du temps de préparation
card.querySelector(".description").innerText = this.description; // ajout de la description
for (let i = 0; i < this.ingredients.length; i++) {
// pour chaque ingrédient
const li = document.createElement("li"); // création d'une liste
li.innerHTML = templateIngredient; // ajout du template de l'ingrédient
li.querySelector(".ingredient").innerText = // ajout de l'ingrédient
this.ingredients[i].ingredient;
li.querySelector(".quantity").innerText = // ajout de la quantité si elle existe
this.ingredients[i].quantity !== undefined
enter test case name
finished
1648636.23 ops/s ± 3.61%
5 % slower
DEFER
card.innerHTML = templateRecipes; // ajout du template de la carte de recette
card.querySelector("h1").innerText = this.name; // ajout du nom de la recette
card.querySelector(".time").innerText = this.time + " min"; // ajout du temps de préparation
card.querySelector(".description").innerText = this.description; // ajout de la description
this.ingredients.forEach((ingredient) => {
// pour chaque ingrédient
const li = document.createElement("li"); // création de la liste
li.innerHTML = templateIngredient; // ajout du template de l'ingrédient
li.querySelector(".ingredient").innerText = ingredient.ingredient; // ajout de l'ingrédient
li.querySelector(".quantity").innerText =
ingredient.quantity !== undefined ? `: ${ingredient.quantity}` : ""; // ajout de la quantité si elle existe
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