

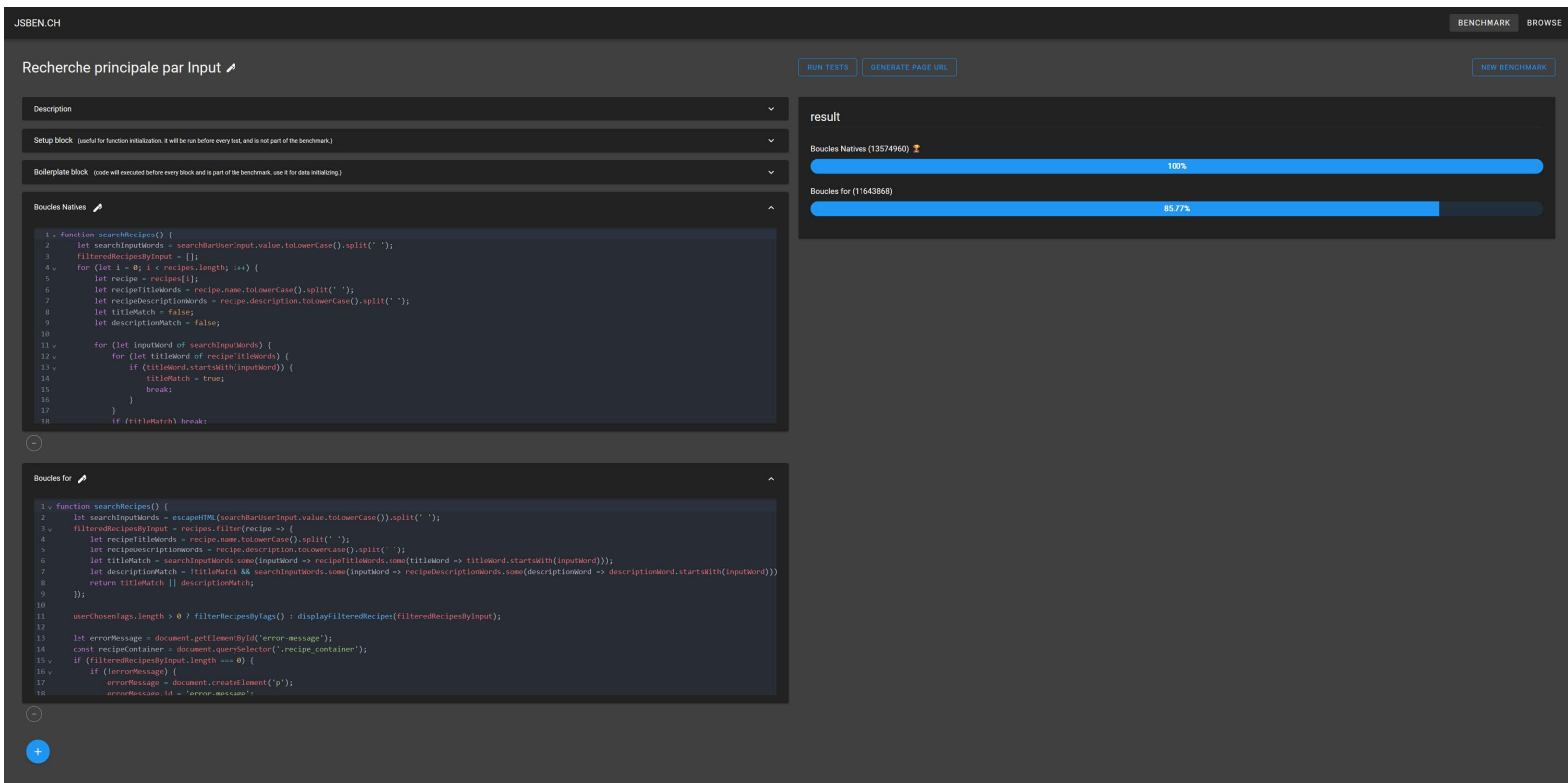
## Fiche d'investigation de fonctionnalité

<b>Fonctionnalité</b> : Filtrer les recettes dans l'interface utilisateur	<b>Fonctionnalité #1</b>
<b>Problématique</b> : Accéder rapidement à une recette correspondant à un besoin de l'utilisateur dans les recettes déjà reçues	
<b>Option 1 : Boucle Native</b> Cette option consiste à utiliser une boucle for pour parcourir la liste de recettes et identifier celles qui correspondent aux critères de recherche de l'utilisateur.	
<b>Avantages</b> <ul style="list-style-type: none"><li>⊕ Approche simple et facile à comprendre</li><li>⊕ Bonne performance pour la recherche par saisie utilisateur</li></ul>	<b>Inconvénients</b> <ul style="list-style-type: none"><li>⊖ Moins performant pour la recherche par filtres dans les menus déroulants</li></ul>

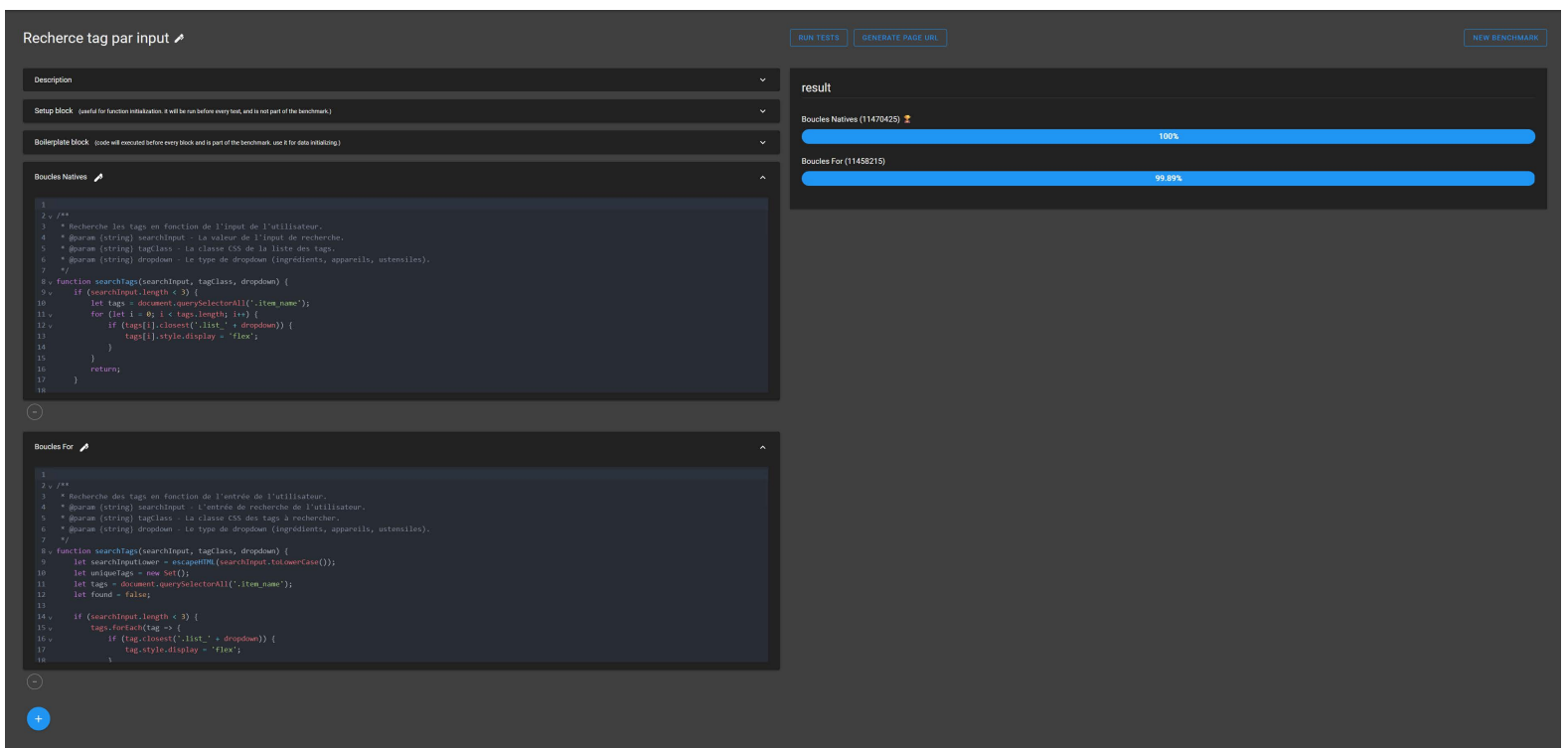
<b>Option 2 : Méthode "Filter"</b> Cette option consiste à utiliser la méthode filter() de JavaScript pour filtrer la liste de recettes en fonction des critères de recherche de l'utilisateur.	
<b>Avantages</b> <ul style="list-style-type: none"><li>⊕ Bonne performance pour la recherche par filtres dans les menus déroulants</li><li>⊕ Plus concis que l'utilisation d'une boucle for</li></ul>	<b>Inconvénients</b> <ul style="list-style-type: none"><li>⊖ Moins performant pour la recherche par input</li></ul>

<b>Solution retenue :</b> Nous avons donc retenu la solution des boucles utilisant la programmation fonctionnelle en raison de sa lisibilité, de sa maintenabilité. La programmation fonctionnelle offre des avantages significatifs par rapport à l'utilisation des boucles natives, en particulier pour des projets de grande envergure nécessitant un code propre et maintenable. Il serait possible d'utiliser les boucles natives pour les recherches par input qui sont plus performante ici tant que le nombre d'objet à rechercher reste limité.
---

# Recherche Principale par Input



# Recherche de tag par input



# Recherche avancée par Tag

JSBEN.CH

BENCHMARKBROWSE

Recherche par tag

Description

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

Boilerplate block (code will be executed before every block and is part of the benchmark, use it for data initializing)

Boucles For

```
1 ∨ /**
2  * Tableau des tags choisis par l'utilisateur.
3  * @type {Array<string>}
4  */
5  let userChosenTags = [];
6
7 ∨ /**
8  * Tableau des recettes filtrées.
9  * @type {Array<Object>}
10 */
11 let filteredRecipes = [];
12
13 ∨ /**
14  * Filtre les recettes en fonction des tags choisis par l'utilisateur.
15  */
16 ∨ function filterRecipesByTags() {
17     let sourceRecipes = (filteredRecipesByInput || filteredRecipesByInput.length === 0) ? recipes : filteredRecipesByInput;
18 }
```

Boucles Natives

```
1 ∨ /**
2  * Tableau global pour stocker les tags choisis par l'utilisateur.
3  * @type {Array}
4  */
5  let userChosenTags = [];
6
7 ∨ /**
8  * Tableau global pour stocker les recettes filtrées par tags.
9  * @type {Array}
10 */
11 let filteredRecipes = [];
12
13 ∨ /**
14  * Filtre les recettes en fonction des tags choisis par l'utilisateur.
15  */
16 ∨ function filterRecipesByTags() {
17     filteredRecipes = [];
18     resetDisplayAndPrintTV();
19 }
```

result

Boucles For (14057088) 🚩

100%

Boucles Natives (11381035)

80.96%