

Fiche d'investigation de fonctionnalité

Fonctionnalité : Algorithme de Recherche	Les Petits Plats – OpenClassRooms – P7
<u>Problématique</u> : Mettre en place un algorithme de Recherche qui offre les meilleures performances pour fournir la meilleure UX possible à l'utilisateur	

<u>Option 1</u> : Implémentation de l'algorithme avec des structures de contrôle « natives » <ul style="list-style-type: none">- for, if / else, while etc.- affectation d'une valeur à un index d'un tableau
<u>Remarques</u> : Les principaux inconvénients sont le manque de clarté dans le code (surtout lors des boucles imbriquées) et la longueur du code
<u>Benchmark Jsbench</u> : <ul style="list-style-type: none">- function triggerSearchAndUpdateDOM() 98.52 %- class ApplianceExtractor 93.32 %- class ExtractDataToObject 86.69 %

<u>Option 2</u> : Implémentation de l'algorithme avec des méthodes modernes propres aux tableaux <ul style="list-style-type: none">- forEach, map, push, some etc.
<u>Remarques</u> : Le code est réduit et on comprend directement la méthode utilisée
<u>Benchmark Jsbench</u> <ul style="list-style-type: none">- function triggerSearchAndUpdateDOM() 100 %- class ApplianceExtractor 100 %- class ExtractDataToObject 100 %

<u>Solution retenue</u> : L'option 2 est la plus performante. Globalement, les méthodes propres aux tableaux sont les plus rapides : elles permettent un gain de temps dans l'exécution du code qui est une des demandes clés sur la fonctionnalité demandée. L'option 2 améliore aussi grandement la lisibilité et la maintenabilité de la codebase.
