



Audit de performance de l'application web TodoListMe

10/05/2021

MATHIEU GAUBAL-VATILINGON

Projet 8 - parcours Développeur front-end

Cet audit a été réalisé avec l'outil devtools du navigateur Brave (chromium)

Sommaire


OPTIONS POUR L'AUDIT	2
RAPPORT DE L'AUDIT	2
LES PERFORMANCES	3
Opportunités	3
Diagnosticues	3
Points passées avec succès	5
ACCESSIBILITÉ	6
Contraste	7
Noms et étiquettes	7
Navigation	7
Internationalisation et localisation	7
Points passées avec succès	8
LES BONNES PRATIQUES	8
Confiance et sécurité	8
Expérience utilisateur	9
Général	9
Points passées avec succès	9
SEO	10
Mobile Friendly	10
Exploration et indexation	11
Bonnes pratiques relatives au contenu	11
VÉRIFICATEUR W3C	12
COMPARAISON RAPPORT	14
CONCLUSION	15

OPTIONS POUR L'AUDIT

L'audit depuis la devtools a été configuré avec les options suivantes :

☒ Clear storage ☒ Simulated throttling [Learn more](#)

Reset storage (localStorage, IndexedDB, etc) before auditing. (Good for performance & PWA testing)


Generate report
Identify and fix common problems that affect your site's performance, accessibility, and user experience. [Learn more](#)

Categories

- ☒ Performance
- ☐ Progressive Web App
- ☒ Best practices
- ☒ Accessibility
- ☒ SEO

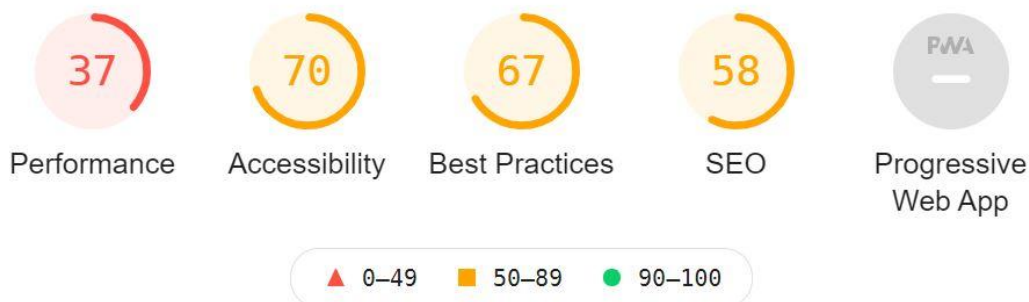
Device

- ☐ Mobile
- ☒ Desktop

Community Plugins (beta)

- ☒ Publisher Ads

RAPPORT DE L'AUDIT



Ce rapport nous indique les points faibles et les améliorations que nous pouvons apporter à l'application TodoListMe.


Légende :

- Conforme aux attentes des bonnes pratiques
- Les actions à mener pour optimiser
- ▲ Les points d'amélioration identifiés
- Astuces pouvant être utiles

LES PERFORMANCES

Opportunités

Ces suggestions peuvent aider votre page à se charger plus rapidement.

▲ Supprimer le JavaScript inutilisé  1,8 s ▼

Supprimez le JavaScript inutilisé pour réduire les octets consommés par l'activité du réseau

▲ Servez des images dans des formats de nouvelle génération  0.81 s ▼

Les formats d'image tels que JPEG 2000, JPEG XR et WebP offrent souvent une meilleure compression que PNG ou JPEG, ce qui signifie des téléchargements plus rapides et moins de consommation de données

■ Éliminez les ressources bloquant le rendu  0,57 s ▼

Les ressources bloquent la première peinture de votre page. Envisagez de fournir des JS / CSS critiques en ligne et de différer tous les styles / JS non critiques.

Diagnostiques

Davantage d'informations sur les performances de l'application.

▲ Assurez-vous que le texte reste visible pendant le chargement de la police Web ▼

Tirez parti de la fonctionnalité CSS d'affichage des polices pour vous assurer que le texte est visible par l'utilisateur pendant le chargement des polices Web.

▲ Réduire l'impact du code tiers Le code tiers a bloqué le thread principal pendant 580 ms ▼

Le code tiers peut avoir un impact significatif sur les performances de charge. Limitez le nombre de fournisseurs tiers redondants et essayez de charger du code tiers une fois le chargement de votre page terminé.

▲ Éviter de `document.write()` ▼

Pour les utilisateurs sur des connexions lentes, les scripts externes injectés dynamiquement via `document.write()` peuvent retarder le chargement de la page de dizaines de secondes.

▲ Éléments d'image ne sont pas explicites width et height

Définissez une largeur et une hauteur explicites sur les éléments de l'image pour réduire les changements de disposition.

▲ Minimiser le travail du thread principal — 5,7 s ▼

Envisagez de réduire le temps passé à analyser, compiler et exécuter JS.

▲ Servir des actifs statiques avec une stratégie de cache efficace — 31 ressources trouvées ▼

Une longue durée de vie du cache peut accélérer les visites répétées sur votre page.

■ Réduisez le temps d'exécution de JavaScript de — 3,4 s ▼

Envisagez de réduire le temps passé à analyser, compiler et exécuter JS.

● Évitez de chaîner les requêtes critiques — 12 chaînes trouvées ▼

Les chaînes de demandes critiques ci-dessous vous montrent quelles ressources sont chargées avec une priorité élevée. Envisagez de réduire la longueur des chaînes, de réduire la taille de téléchargement des ressources ou de différer le téléchargement des ressources inutiles pour améliorer le chargement de la page.

● Le chronométrage utilisateur marque et mesure — 8 chronomètres utilisateur ▼

Pensez à instrumenter votre application avec l'API User Timing pour mesurer les performances réelles de votre application lors d'expériences utilisateur clés.

- Limitez le nombre de demandes et réduisez la taille des transferts. — 94 demandes • 1 308 Kio

Pour définir des budgets pour la quantité et la taille des ressources de page, ajoutez un fichier budget.json.

- Le plus grand élément de peinture avec contenu — 1 élément trouvé

Il s'agit du plus grand élément de contenu peint dans la fenêtre.

- Évitez les grands décalages de mise en page — 5 éléments trouvés

Ces éléments DOM contribuent le plus au CLS de la page.

- Évitez les longues tâches du thread principal — 20 tâches longues trouvées

Répertorie les tâches les plus longues sur le thread principal, utile pour identifier les pires contributeurs au retard d'entrée.

- Évitez les animations non composées — 5 éléments animés trouvés

Les animations qui ne sont pas composées peuvent être saccadées et augmenter le CLS.

Points passés avec succès

- Images de bonne taille
- Différer les images hors écran
- Réduire le CSS
- Réduire le JavaScript Économies — potentielles de 44 Kio
- Supprimer le CSS inutilisé
- Encodez efficacement les images
- Activer la compression de texte
- Se connecter aux origines requises

- Le temps de réponse initial du serveur était court Le — document racine a pris 390 ms
- Évitez les redirections de plusieurs pages
- Précharger les demandes de clés
- Utilisez des formats vidéo pour le contenu animé
- Supprimer les modules en double dans les ensembles JavaScript
- Évitez de diffuser l'ancien JavaScript sur les navigateurs modernes Économies — potentielles de 31 Kio
- Précharger la plus grande image de peinture avec contenu
- Évite les énormes charges utiles du réseau — La taille totale était de 1308 Kio
- Évite une taille excessive du DOM — 372 éléments
- Chargement paresseux de ressources tierces avec façades
- Utilise des écouteurs passifs pour améliorer les performances de défilement

ACCESSIBILITÉ

Contraste

Ce sont des opportunités pour améliorer la lisibilité de votre contenu.

▲ Les couleurs d'arrière-plan et de premier plan n'ont pas un rapport de contraste suffisant. ▼

Le texte à faible contraste est difficile voire impossible à lire pour de nombreux utilisateurs.

Noms et étiquettes

Ce sont des opportunités pour améliorer la sémantique des contrôles de votre application. Cela peut améliorer l'expérience des utilisateurs de technologies d'assistance, comme un lecteur d'écran.

▲ <frame> ou les <iframe> éléments n'ont pas de titre

Les utilisateurs de lecteurs d'écran se fient aux titres des cadres pour décrire le contenu des cadres.

▲ Les éléments d'image n'ont pas d' [alt] attributs

Les éléments informatifs doivent viser un texte alternatif court et descriptif. Les éléments décoratifs peuvent être ignorés avec un attribut alt vide.

▲ Les éléments de formulaire n'ont pas d'étiquettes associées

Les libellés garantissent que les contrôles de formulaire sont correctement annoncés par les technologies d'assistance, comme les lecteurs d'écran

Navigation

▲ Les éléments d'en-tête ne sont pas dans un ordre séquentiel décroissant

Les en-têtes correctement ordonnés qui ne sautent pas de niveaux transmettent la structure sémantique de la page, ce qui facilite la navigation et la compréhension lors de l'utilisation de technologies d'assistance.

Internationalisation et localisation

▲ <html> l'élément n'a pas d' [lang] attribut

Si une page ne spécifie pas d'attribut lang, un lecteur d'écran suppose que la page est dans la langue par défaut choisie par l'utilisateur lors de la configuration du lecteur d'écran. Si la page n'est pas réellement dans la langue par défaut, le lecteur d'écran risque de ne pas annoncer correctement le texte de la page.

Points passés avec succès

- [aria-*] les attributs correspondent à leurs rôles
- [aria-hidden="true"] n'est pas présent sur le document <body>
- [aria-hidden="true"] les éléments ne contiennent pas de descendants focalisables
- [aria-*] les attributs ont des valeurs valides
- [aria-*] les attributs sont valides et ne sont pas mal orthographiés
- La page contient un en-tête, un lien de passage ou une région de référence
- Le document a un <title>élément
- [id] les attributs des éléments actifs et focalisables sont uniques
- Les identifiants ARIA sont uniques
- Les liens ont un nom discernable
- Les listes contiennent uniquement des éléments et des éléments de support de script (<script>et <template>).
- Les éléments de liste () sont contenus dans des éléments parents ou
- Aucun élément n'a une [tabindex]valeur supérieure à 0

LES BONNES PRATIQUES

Confiance et sécurité

- ▲ N'utilise pas HTTPS — 40 requêtes non sécurisées trouvées

Tous les sites doivent être protégés avec HTTPS, même ceux qui ne gèrent pas de données sensibles. Cela inclut d'éviter le contenu mixte, où certaines ressources sont chargées via HTTP malgré la demande initiale servie via HTTPS. HTTPS empêche les intrus de falsifier ou d'écouter passivement les communications entre votre application et vos utilisateurs, et est une condition préalable pour HTTP / 2 et de nombreuses nouvelles API de plate-forme Web.

- ▲ Comprend des bibliothèques JavaScript frontales avec des vulnérabilités de sécurité connues — 4 vulnérabilités détectées

Certains scripts tiers peuvent contenir des vulnérabilités de sécurité connues qui sont facilement identifiées et exploitées par les attaquants.

Expérience utilisateur

▲ Sert des images à basse résolution

Les dimensions naturelles de l'image doivent être proportionnelles à la taille d'affichage et au rapport de pixels pour maximiser la clarté de l'image

Général

▲ Enregistre un `unload` auditeur

L'événement ``unload`` ne se déclenche pas de manière fiable et son écoute peut empêcher les optimisations du navigateur comme le cache arrière-avant. Pensez à utiliser les événements «`pagehide`» ou «`visibilitéchange`» à la place.

▲ Les erreurs du navigateur ont été enregistrées dans la console

Les erreurs consignées dans la console indiquent des problèmes non résolus. Ils peuvent provenir d'échecs de demande de réseau et d'autres problèmes de navigateur.

Points passés avec succès

● Les liens vers des destinations d'origine croisée sont sûrs

● Évite de demander l'autorisation de géolocalisation lors du chargement de la page

● Évite de demander l'autorisation de notification lors du chargement de la page

● Permet aux utilisateurs de coller dans les champs de mot de passe

● Affiche les images avec un rapport hauteur / largeur correct

● La page a le doctype HTML

● Définit correctement le jeu de caractères

● Évite Application Cache

● Bibliothèques JavaScript détectées	▼
● Évite les API obsolètes	▼
● La page contient des cartes sources valides	▼
● Aucun problème dans le <code>Issues</code> panneau dans Chrome Devtools	▼

SEO

Mobile Friendly

Assurez-vous que vos pages sont adaptées aux mobiles afin que les utilisateurs n'aient pas à pincer ou à zoomer pour lire les pages de contenu

▲ N'a pas de <code><meta name="viewport"></code> balise avec <code>width</code> ou <code>initial-scale</code>	▼
Aucune balise <code><meta name =" viewport "></code> trouvée	

Ajoutez une balise `<meta name =" viewport ">` pour optimiser votre application pour les écrans mobiles.

▲ Le document n'utilise pas de tailles de police lisibles	▼
Le texte est illisible car il n'y a pas de balise Meta Viewport optimisée pour les écrans mobiles.	

Les tailles de police inférieures à 12 pixels sont trop petites pour être lisibles et obligent les visiteurs mobiles à «pincer pour zoomer» afin de lire. Efforcez-vous d'avoir > 60% du texte de la page $\geq 12\text{px}$.

▲ Les cibles tactiles ne sont pas dimensionnées de manière appropriée	▼
Les cibles tactiles sont trop petites, car aucune balise Meta Viewport n'est optimisée pour les écrans mobiles	

Les éléments interactifs tels que les boutons et les liens doivent être suffisamment grands (48x48px) et avoir suffisamment d'espace autour d'eux pour être assez faciles à toucher sans se chevaucher sur d'autres éléments.

Exploration et indexation

Pour apparaître dans les résultats de recherche, les robots d'exploration doivent accéder à votre application.

▲ Les liens ne sont pas explorables

Les moteurs de recherche peuvent utiliser les attributs «href» sur les liens vers des sites Web d'exploration. Assurez-vous que l'attribut `href` des éléments d'ancrage est lié à une destination appropriée, afin que davantage de pages du site puissent être découvertes.

Bonnes pratiques relatives au contenu

Formatez votre code HTML de manière à permettre aux robots d'exploration de mieux comprendre le contenu de votre application.

▲ Les éléments d'image n'ont pas d' [alt]attributs

Les éléments informatifs doivent viser un texte alternatif court et descriptif. Les éléments décoratifs peuvent être ignorés avec un attribut alt vide.

- Le document a un `<title>`élément
- Le document a une méta description
- La page a un code d'état HTTP réussi
- Les liens ont un texte descriptif
- L'indexation de la page n'est pas bloquée
- Le document a un `hreflang`
- Le document évite les plugins

VÉRIFICATEUR W3C

Nous voyons plusieurs erreurs lorsque nous mettons le code source du site web dans le vérificateur W3C.

Nous pouvons remarquer qu'il y a des balises ALT manquantes pour les images.

Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for <http://todolistme.net/>

Checker Input

Show ☐ source ☐ outline ☐ image report

Check by address

<http://todolistme.net/>

Use the Message Filtering button below to display options for hiding/showing particular messages, and to see total counts of errors and warnings.

Message Filtering 10 messages hidden by filtering

- Error** Bad value `///fonts.googleapis.com/css?family=Abel|Architects+Daughter` for attribute `href` on element `link`: Illegal character in query: `|` is not allowed.
 From line 11, column 2; to line 11, column 106

```
</title><link href="//fonts.googleapis.com/css?family=Abel|Architects+Daughter" rel="stylesheet" type="text/css">
```
- Error** An `img` element must have an `alt` attribute, except under certain conditions. For details, consult [guidance on providing text alternatives for images](#).
 From line 44, column 64; to line 44, column 106

```
span<h1><img src='./images/tick.png' id='logotick'><span
```
- Error** An `img` element must have an `alt` attribute, except under certain conditions. For details, consult [guidance on providing text alternatives for images](#).
 From line 47, column 419; to line 47, column 457

```
w.'><span><img src='./images/top_new_window.png'></span
```
- Error** An `img` element must have an `alt` attribute, except under certain conditions. For details, consult [guidance on providing text alternatives for images](#).
 From line 112, column 10; to line 112, column 38

```
<p><img src='./images/info.png'></p>
```
- Error** An `img` element must have an `alt` attribute, except under certain conditions. For details, consult [guidance on providing text alternatives for images](#).
 From line 147, column 7; to line 147, column 59

```
iv><img id='sortbutton' src='./images/sort_order.png' />
```
- Error** Element `div` not allowed as child of element `h2` in this context. (Suppressing further errors from this subtree.)
 From line 156, column 98; to line 156, column 120

```
e'></span><div id='todoprogess'></div>
```

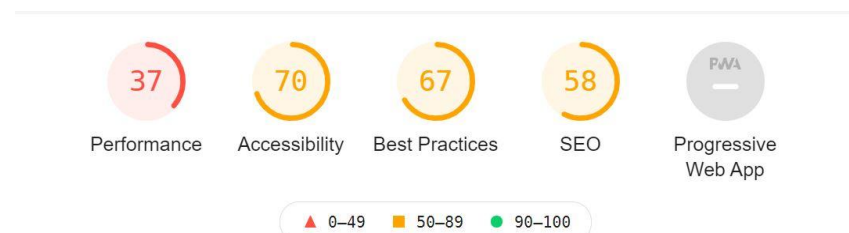
Contexts in which element `div` may be used:

 - Where [flow content](#) is expected.
 - As a child of a [div](#) element.

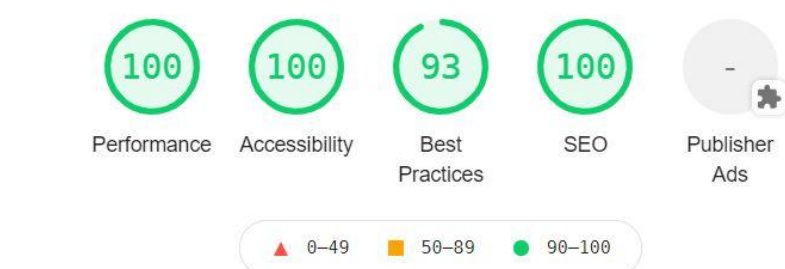
7. **Error** Element `div` not allowed as child of element `h3` in this context. (Suppressing further errors from this subtree.)
 From line 166, column 60; to line 166, column 82
`e :<div id='doneprogress'></div>`
 Contexts in which element `div` may be used:
 Where [flow content](#) is expected.
 As a child of a [dl](#) element.
8. **Error** An `img` element must have an `alt` attribute, except under certain conditions. For details, consult [guidance on providing text alternatives for images](#).
 From line 173, column 83; to line 173, column 112
`e items.'>←`
9. **Error** Element `div` not allowed as child of element `h3` in this context. (Suppressing further errors from this subtree.)
 From line 179, column 127; to line 179, column 148
`r'><div id='laterdialog'><div i`
 Contexts in which element `div` may be used:
 Where [flow content](#) is expected.
 As a child of a [dl](#) element.
10. **Error** Duplicate ID `later_number`.
 From line 179, column 257; to line 179, column 280
`</p></div>`
11. **Error** Element `div` not allowed as child of element `span` in this context. (Suppressing further errors from this subtree.)
 From line 202, column 342; to line 202, column 359
`pan><div id='fb-root'></div>`
 Contexts in which element `div` may be used:
 Where [flow content](#) is expected.
 As a child of a [dl](#) element.
12. **Error** Element `fb:like` not allowed as child of element `span` in this context. (Suppressing further errors from this subtree.)
 From line 202, column 438; to line 202, column 550
`></script><fb:like href='http://todolistme.net' send='false' layout='button_count' show_faces='false' font='lucida grande'></fb:l`
 Content model for element `span`:
[Phrasing content](#).
13. **Error** Element `g:plusone` not allowed as child of element `span` in this context. (Suppressing further errors from this subtree.)
 From line 202, column 589; to line 202, column 642
`plusspan'><g:plusone size='medium' href='http://todolistme.net'></g:pl`
 Content model for element `span`:
[Phrasing content](#).
14. **Error** An `img` element must have an `alt` attribute, except under certain conditions. For details, consult [guidance on providing text alternatives for images](#).
 From line 235, column 86; to line 235, column 116
`undo());' ></div>`

COMPARAISON RAPPORT

TodoListMe



Votre site todo-List



CONCLUSION

L'analyse indique des pistes d'amélioration en termes de performance. Attention toutefois aux connexions aux services tiers et veiller à produire un code minifié, épuré de redondances, optimisez le cache de l'application.

Pour rester dans les bonnes pratiques, favoriser le protocole HTTPS. Veillez aussi à ce que les scripts utilisent des versions stables afin d'éviter les failles de sécurité. Le code source devra être vérifié avant de mettre le site en production.

Le SEO n'est pas optimal et ne respecte pas les standards du web.

Inversement, votre application web respecte les normes du web en terme de performance, accessibilité, bonne pratique et SEO.

Cependant, l'application n'est pas en ligne donc n'est pas soumise à certains tests.