

Audit de performance

TODOS

todos

What needs to be done?

Double-click to edit a todo

Created by Oscar Godson

Refactored by Christoph Burgmer

Sommaire

1. **Fonctionnement de notre application**
2. **Analyse de qualité et performance web**
 1. **En général**
 2. **Accessibilité**
 3. **Nombre de requêtes**
 4. **Qualité**
 5. **Sécurité**
 6. **Volumes de données**
3. **Conclusion**

1. Fonctionnement de notre application

L'application TODOS a pour but de créer, éditer et gérer des listes de la façon la plus simple possible.

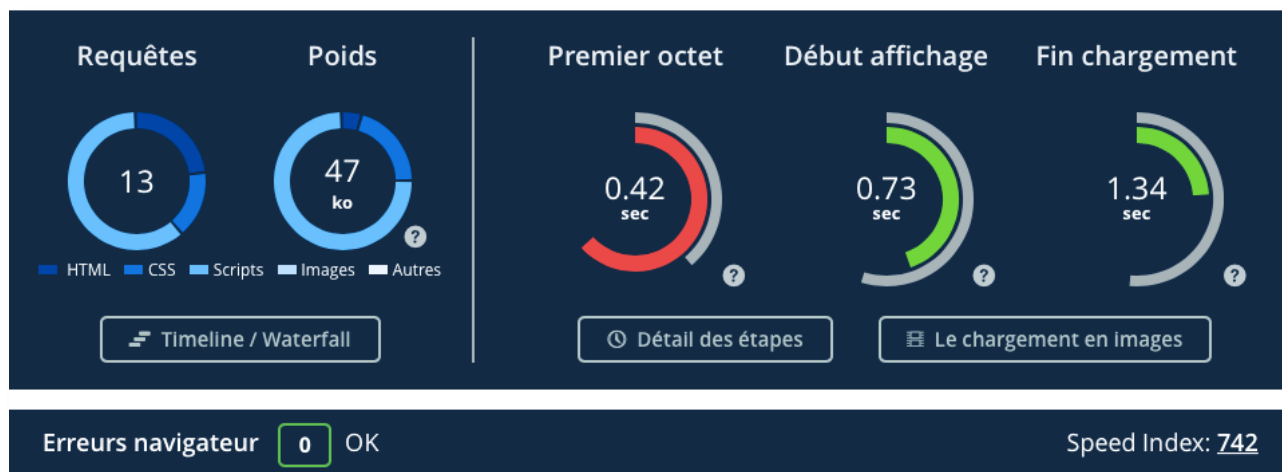
L'application est pour cela très épurée afin de conduire l'utilisateur à entrer sa première tâche le plus instinctivement que possible.

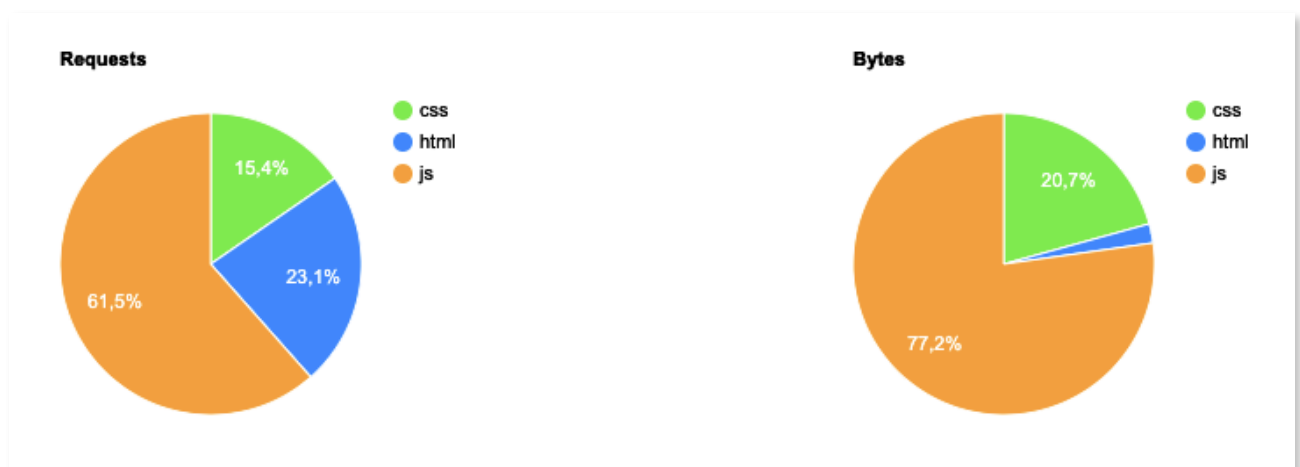
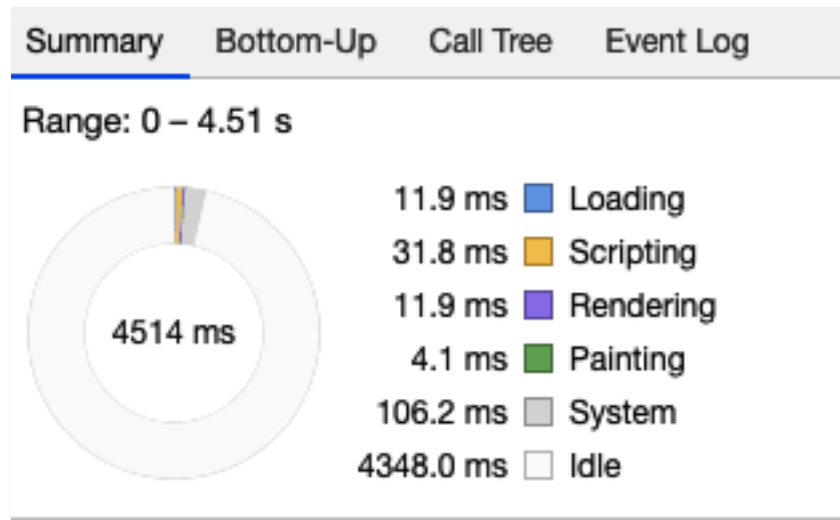
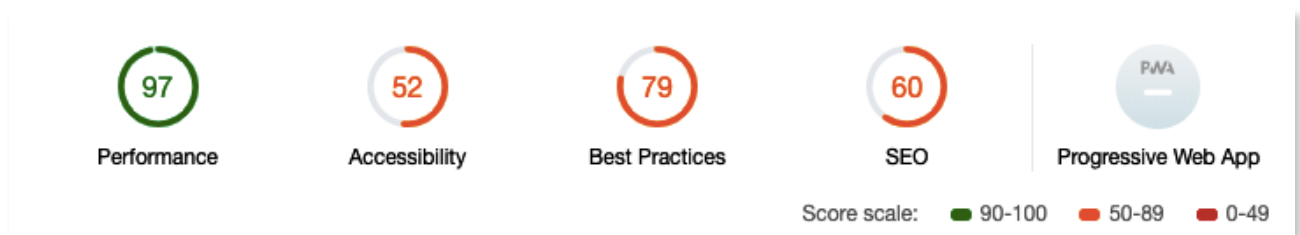
L'application s'adresse à un très large public.

2. Analyse de qualité et performance web

1. En général

Voici un ensemble de tableau bruts que nous allons analyser.





2. Accessibilité

Les problèmes observés:

- ▶ 1 label n'est pas lié à un élément
- ▶ 1 élément vide peut perturber les lecteurs d'écrans
- ▶ Aucune balise `<noscript>` n'est définie

L'optimisation à apporter:

- ▶ Expliquer le but de chaque champ de formulaire

Les bonnes pratiques:

- ▶ La langue de la page est spécifiée
- ▶ Aucun attribut "src" n'est vide
- ▶ Les liens ont des titres cohérents
- ▶ Le titre principal est bien le premier titre spécifié

EN BREF:

L'attribut **for** permet d'associer le label à un élément de la page, et aidera les lecteurs d'écran à mieux interpréter le contenu.

Les éléments `<p>`, ``, `<button>`, `<legend>`, `<caption>`, `<figcaption>` et `<quote>` ne doivent pas être vides car s'ils le sont, certains lecteurs d'écran auront des difficultés à interpréter leur présence.

Il faut supprimer ces éléments vides du code ou les décorer avec l'attribut **aria-hidden** pour que les lecteurs d'écran les ignorent.

Il faut améliorer l'expérience utilisateur sur le site en spécifiant une étiquette descriptive pour chaque champ de formulaire.

Lorsque la page utilise des scripts, il est souvent conseillé de définir au moins une balise **noscript**. Celle-ci est nécessaire pour afficher un message lorsque le JavaScript est désactivé par l'utilisateur.

La page possède un attribut **lang** qui permet aux liseuses d'écran de prendre correctement en charge le site.

L'attribut "src" permet d'accéder à une autre ressource. Rien ne justifie son utilisation avec une valeur vide.

Un lien est plus attractif si son texte décrit ce qui se cache derrière. On peut par ailleurs en profiter pour travailler ces textes avec des mots clés bien choisis, dans une démarche d'amélioration du référencement.

Il faut faciliter le travail des liseuses d'écran en indiquant le titre principal (**<h1>**) avant tout autre titre.

3. Nombre de requêtes

Les problèmes observés:

- ▶ 1 ressource est inaccessible

Les bonnes pratiques:

- ▶ Aucune redirection trop longue n'a été détectée
- ▶ Fichiers CSS bien répartis
- ▶ Aucune image n'utilise qu'une seule couleur

EN BREF:

Attention, certaines requêtes liées à cette page rencontrent des erreurs. Il est par exemple possible :

- qu'il y ait une erreur dans votre code HTML, CSS ou JavaScript
- que le serveur ait rencontré une erreur non gérée
- que certains services utilisés par le site ne répondent pas

Il est ainsi préférable de faire une requête vers un fichier de 50ko de données plutôt que 10 requêtes vers des fichiers de 5ko.

Les redirections déclenchent des allers-retours évitables sur le réseau, et allongent le temps de chargement de la page.

4. Qualité

Les problèmes observés:

- ▶ 1 propriété CSS est surchargée

L'optimisation à apporter:

- ▶ Aucune favicon n'est définie
- ▶ Attention aux commentaires au sein du code source
- ▶ 86% des couleurs définies ne sont utilisées qu'une seule fois
- ▶ 1 sélecteur CSS est dupliqué

Les bonnes pratiques:

- ▶ Les styles sont bien séparés du HTML
- ▶ Les extensions des ressources sont bonnes
- ▶ Toutes les ressources de la page définissent leur type
- ▶ Les propriétés CSS ne sont pas dupliquées

- ▶ Les identifiants utilisés dans votre HTML sont uniques
- ▶ Tous les sélecteurs CSS ne sont pas dupliqués
- ▶ Aucun sélecteur CSS n'est trop complexe

EN BREF:

Il faut veiller à transmettre aux internautes que les données dont ils ont besoin.

Il est souvent inutile de définir 2 fois la même propriété au sein d'une même règle : la seconde surchargera la première. Cela risque de nuire à la lisibilité du code CSS. Supprimer les propriétés surchargées inutilement réduira également le poids du fichier.

28 couleurs sont définies, et 24 d'entre elles ne sont utilisées qu'une seule fois (certaines peuvent être très proches, exemple : #FF0000 and #FE0000). Il serait mieux d'harmoniser au maximum les couleurs du site afin de simplifier le CSS mais aussi d'éviter des incohérences dans la charte graphique.

Utiliser plusieurs fois le même sélecteur sur des blocs déclaratifs différents au sein d'un même fichier CSS peut nuire à la lisibilité et à la maintenabilité du code. C'est également une source d'optimisation, puisque regrouper ces éléments au sein d'une même règle réduit le poids du fichier et réduit le temps nécessaire avant le rendu.

Utiliser un même identifiant pour 2 éléments différents peut entraîner des effets de bord, notamment lors de l'exécution de JavaScript ou lors de l'application des règles CSS.

Des sélecteurs CSS trop complexes nuisent à la lisibilité générale du code et constituent un gain possible en terme de performance.

5. Sécurité

Les problèmes observés:

- ▶ Il manque une politique de sécurité sur la provenance des ressources
- ▶ Cette page est exposée à des attaques du type "clickjacking"

L'optimisation à apporter:

- ▶ Bloquer la totalité du contenu lorsqu'une attaque XSS est suspectée
- ▶ 3 ressources tierces sont délivrées sans contrôle d'intégrité (SRI)

EN BREF:

Il ne faut pas permettre à des personnes malveillantes d'intégrer les pages de l'application sur leur site.

S'assurer que le navigateur du client fasse son maximum pour prévenir d'une attaque de type XSS.

La page charge des contenus provenant de tiers, il serait donc plus prévoyant de s'assurer de l'intégrité de ces données.

6. Volumes de données

L'optimisation à apporter:

- ▶ Activez la compression des ressources
- ▶ Réduire certains fichiers JavaScript (minification)
- ▶ Cette page charge 47 ko de données

Les bonnes pratiques:

- ▶ Les contenus sont bien appelés par des URLs uniques
- ▶ Pas besoin de minifier des ressources HTML
- ▶ Les ressources CSS sont minifiées
- ▶ Toutes les ressources ont un poids inférieurs à 1 Mo
- ▶ Cette page n'envoie pas trop de cookies
- ▶ Les réponses de redirection sont légères

EN BREF:

En compressant les ressources avec "gzip" ou "deflate", on peut réduire le nombre d'octets envoyés sur le réseau.

En compressant le code JavaScript, vous pouvez libérer de nombreux octets de données et réduire les délais de téléchargement, d'analyse et d'exécution.

Une page chargeant trop de données ralentit sa vitesse d'affichage, d'autant plus sur les connexions bas débit. Cela peut également générer de la frustration pour les internautes qui sont limités par des forfaits data.

Le contenu d'une redirection ne sera en aucun cas utilisé par les navigateurs web. C'est donc poids inutile qu'il convient d'éliminer.

3. Conclusion

Notre application TODOS possède de très bons points positifs à ce jour (cf. Conclusion de l'Audit de performance de notre concurrent).

En vue d'un éventuel scaling de l'application, plusieurs points seront à travailler.

Il faut rester vigilants sur l'accessibilité mobile. En effet, l'application doit être responsive afin de proposer une expérience utilisateur optimale sur tous les appareils.

Il faut également envisager la possibilité de créer un compte utilisateur afin de sauvegarder les listes et les synchroniser selon les appareils utilisés.

Passer aux requêtes sécurisées serait un atout en matière de sécurité des données.

Sur le marché très concurrentiel des todo-list, il serait nécessaire d'améliorer certains axes qui permettront d'optimiser la performance et l'expérience utilisateur afin de se démarquer de ce qui se fait sur le marché.

Il serait intéressant d'élaborer un plan de développement à moyen terme dans le but de développer l'application et ainsi augmenter le trafic. Cela nécessiterait une analyse marketing avec un Google Analytics par exemple.

Notre application TODOS (Conclusion Cf. fichier "audit de performance de notre concurrent"):

- ▶ Rapide
- ▶ Optimiser et performante
- ▶ Améliorer l'User experience
- ▶ Accessibilité: Design intéressant mais manque de contraste
- ▶ Une sous catégorie dans les liste pourrait être intéressante.
- ▶ Mettre en place des solution pour la sécurité
- ▶ Une version mobile serait un mieux
- ▶ Ajouter une page 404