

Rapport d’optimisation

Site Agence Web LA CHOUETTE AGENCE LYON.

Bécuwe Antoine

29/04/1996

PRÉFACE: Plan du rapport.

1. Introduction
2. Mise en place des axes d’amélioration
   1. Le “Black Hat” SEO
   2. Le contraste et l’accessibilité
   3. Un code validé par le W3C
   4. Réduction du poids des images
   5. Réduction poids JS et CSS
   6. Les balises Méta
   7. La prise en charge du responsive
   8. Appel aux scripts et stylesheets
   9. Accessibilité TA
   10. Allègement des fichiers et code
3. Conclusion

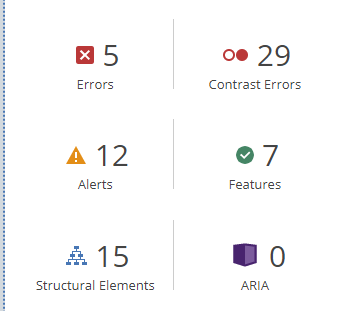
# INTRODUCTION

Voici le rapport d’optimisation SEO et Accessibilité concernant la réunion de lancement “Amélioration du référencement”.

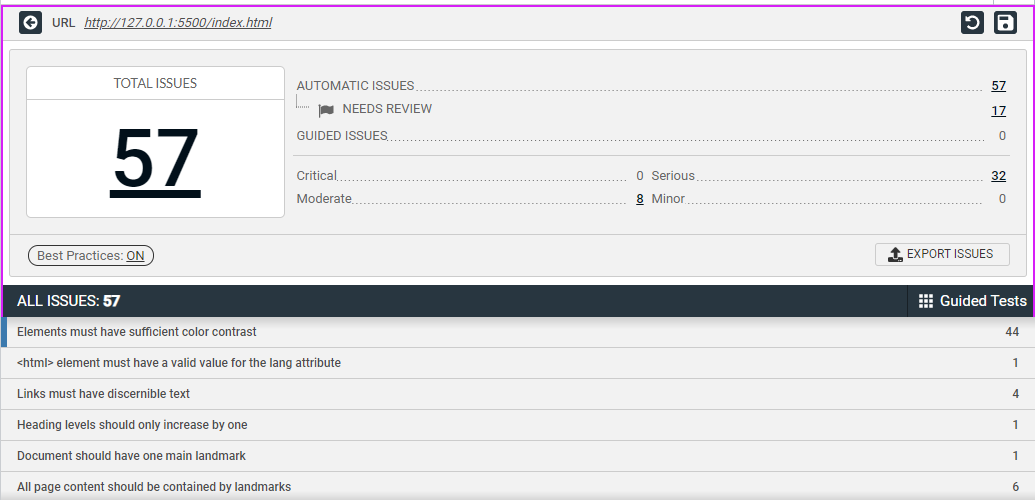
Le compte-rendu de cette réunion était le suivant:

* Analyse du SEO et du code du site
* Amélioration
* Comparaison des résultats

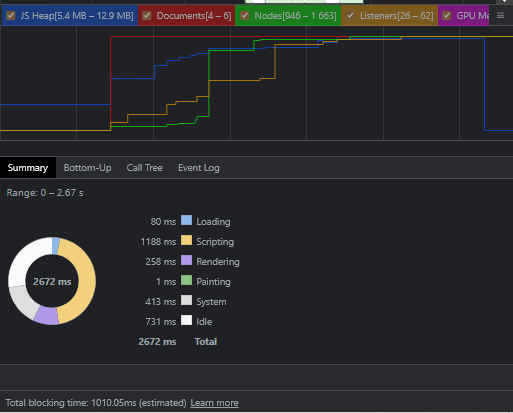
Lors de l’analyse du SEO et du code source j’ai pu trouver plein d’erreurs et des axes d’améliorations. En me basant sur le modèle fourni après la réunion j’ai pu les mettre en évidence. Avant de commencer ce rapport, voici quelques captures d’écrans du site original, avant modifications.



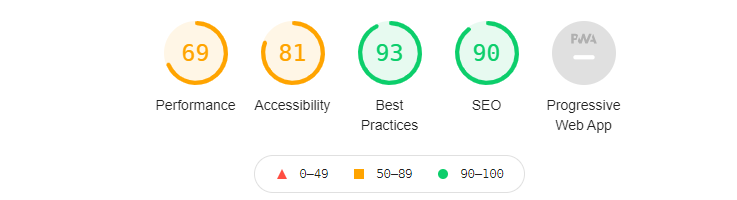
*Rapport WAVE, extension Chrome spécialisée pour l’accessibilité*

**

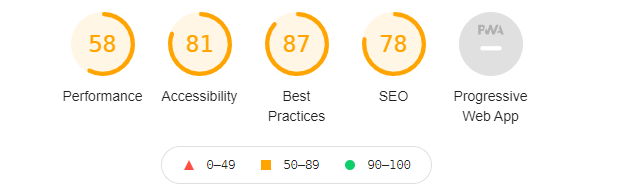
*Rapport Axe Developer Tools, également pour l’accessibilité*

**

*Rapport de performances au chargement de la page de l’inspecteur de la console de Google Chrome (on retire le Idle: 2672-731=1941ms)*

**

*Rapport Lighthouse, outil Google pour mesurer la performance, l’accessibilité, les bonnes pratiques et le SEO, version desktop*

**

*Le rapport Lighthouse, version mobile cette fois*

On peut remarquer de nombreuses erreurs ou performances à améliorer. La version mobile est préoccupante car c’est cette version du site qui sera lue par la Google Search Console, le robot qui s’occupera de l’affichage parmi les résultats de recherche.

Passons aux axes de développement.

1. MISE EN PLACE DES AXES D’AMÉLIORATION
2. LE BLACK HAT SEO

C’est le 1er axe d’amélioration et sans doute le plus important; se débarrasser du “Black Hat” SEO. Mais qu’est-ce que c’est ?

Voici sa définition:



Or on retrouve sur le site du black hat à divers endroits:

* le texte fixé à 1px, presque invisible, avec des mots clés
* les images avec de faux “alt”
* l’”annuaire” en bas de page avec de faux liens ou des liens vers les concurrents: il s’agit de *backlinking* et il s’agit d’une pratique à éviter

Une seule solution pour ces lignes de black hat: les supprimer !

1. LE CONTRASTE ET L’ACCESSIBILITÉ

Pourquoi le contraste et en quoi cela gène-t-il l’accessibilité ? Voici un extrait de la [MDN](https://developer.mozilla.org/fr/docs/Web/Accessibility/Understanding_WCAG/Perceivable/Color_contrast):

Le contraste de la couleur entre l'arrière-plan et le contenu de premier-plan (qui est en général du texte) doit être assez prononcé pour assurer la lisibilité.

Lors de la conception d'interfaces lisibles pour différentes capacités de vision, les directives WCAG recommandent les rapports de contraste suivants: 4.5 : 1 pour le texte de taille normale, 3 : 1 pour les textes de grande taille.

Un bon contraste des couleurs sur votre site profite à tous vos utilisateurs, mais celà est particulièrement intéressant pour les utilisateurs souffrant de certains types de daltonisme et d'autres particularités similaires, qui rencontrent un faible contraste et ont du mal à faire la différence entre des couleurs proches. En effet, ces utilisateurs ne voient pas les zones claires et sombres aussi facilement que ceux sans ces particularités et ont donc du mal à voir les bords, les bordures et d'autres détails.

C'est bien d'avoir un design cool sur votre site, mais le design ne vaut rien si vos utilisateurs ne peuvent pas lire votre contenu.

Or, sur les rapports WAVE et aXe en Introduction, on peut voir qu’il y a énormément de contrastes pas assez suffisants pour une accessibilité maximale.

Solution: Utiliser un comparateur de contraste comme [Contrast Finder](https://app.contrast-finder.org/?lang=fr) par exemple, pour vérifier les contrastes et en trouver de nouveaux si nécessaires.

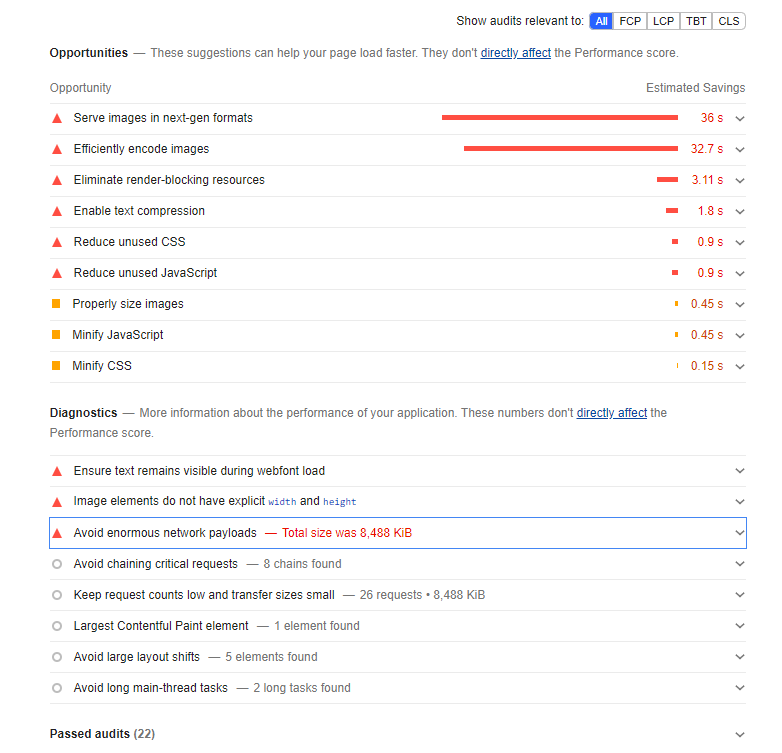
1. UN CODE VALIDÉ PAR LE W3C

Le World Wide Web Consortium, ou W3C, a créé des outils permettant d’avoir de bonnes pratiques dans son HTML ou son CSS et ainsi permettre l’affichage de son code sans erreur avec tous les navigateurs. Un passage

du code source au validateur est un passage obligé pour que son code soit “en règle”.

Vous pourrez noter qu’après les modifications que j’ai apportées au code source, le validateur du W3C n’affiche ni erreur ni avertissement.

1. RÉDUCTION DU POIDS DES IMAGES

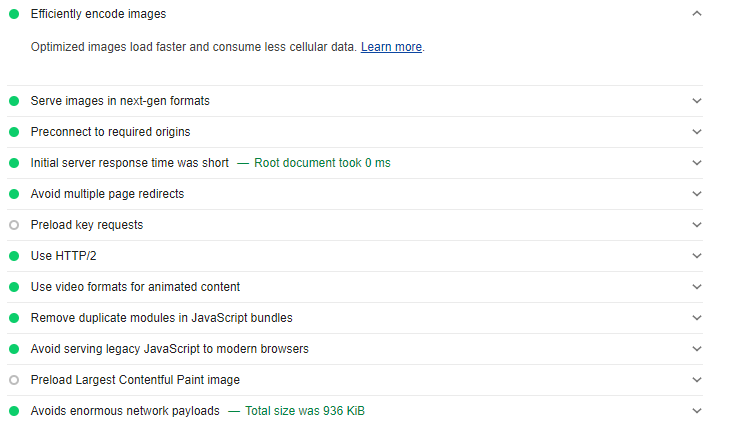


Sur la capture ci-dessus, une capture du rapport Lighthouse de la version mobile du site original, 3 avertissements peuvent être directement reliés aux images fournies et utilisées.

* Serve images in next-gen formats
* Efficiently encode images
* Avoid enormous network payloads

Par exemple, le premier avertissement permettrait donc de faire gagner 36s de temps de chargement, d’après Lighthouse.

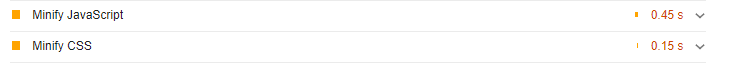
Intéressons-nous tout d’abord au deuxième avertissement: Efficiently encode images. Aujourd’hui, avec les outils de compression disponibles, il est possible de réduire de manière importante le poids d’une image tout en lui garantissant une qualité optimale, et ce gratuitement. C’est ce que j’ai fait. Une fois cette étape réussie, on ne se retrouve plus avec un fichier .bmp de 6Mo (!), qui demande énormément de ressources pour charger la page (3ème avertissement). Afin d’optimiser encore plus le chargement des images, le 1er avertissement préconise de convertir ses images au format .webp, un format léger et “next-gen”. Voici le rapport Lighthouse après compression et conversion.



1. RÉDUCTION DU POIDS DU JS ET CSS

À l’instar des images, on peut également compresser le JavaScript et le CSS, et gagner en performances !

Le CSS et le JavaScript peuvent être lus par le navigateur en format minifié, difficilement lisible pour nous à l'œil nu: il s’agit de retirer tous les espaces et retours à la ligne. Les fichiers sont plus légers, et plus rapidement lus par le navigateur.



Voici encore un gain de temps effectué.

1. LES BALISES MÉTA

Dans cette sous-partie, il y a beaucoup de sujets à aborder.

Pour commencer, la balise <lang> ne doit jamais être “defaut”; celà peut causer des erreurs navigateurs d’accessibilité. Ici, la langue doit être “fr”.

La balise “keywords” est mal utilisée: c’est elle qui servira pour le référencement avec des mots-clés. Dans le code original, on retrouve

plusieurs fois les mêmes mots-clés, et aucune trace du mot-clé “Entreprise Webdesign Lyon”, difficile donc d’apparaître parmi les résultats de recherche…

La meta description est vide: Il n’y a donc aucune description du site qui apparaît dans une recherche Google… Et la balise <title> est représentée par un point, alors qu’il faudrait un titre cohérent qui puisse apparaître dans l’onglet comme dans les résultats de recherche.

La favicon est appelée en format jpg alors qu’on écrit qu’elle est en format png, c’est une incohérence.

Pour la 2ème page, j’ai trouvé plus cohérent de la renommer en page “contact” ! Et l’appel aux stylesheets ne marchait pas, car les fichiers en question n’existaient pas.

Voici l’essentiel pour les balises meta, très importantes pour le référencement du site. Il reste quelques éléments dont je vais parler dans d’autres sections de ce rapport.

1. LA PRISE EN CHARGE DU RESPONSIVE

Aujourd’hui, tout site Internet se doit d’avoir une version mobile: depuis l’arrivée des smartphones les utilisateurs ont changé leurs habitudes de consommation d’Internet et les écrans de ces machines, beaucoup plus petits, impliquent des variations de la page Web, pour toujours une meilleure accessibilité et ergonomie. Il se trouve par ailleurs que le bot Google, qui s’occupe de la lecture des pages pour le référencement, lit les pages mobile avant la version Desktop, souvent plus lourde.

Le problème rencontré au niveau du responsive sur le site original est bien étrange; il s’agit d’images qui ne s’adaptent pas, oui mais des images contenant des images contenant du texte. Nous y reviendrons peu après dans un prochain axe mais il fallait de toutes façons procéder à la suppression de ces fichiers. Le site repose sur le framework Bootstrap; il s’agit d’un système de grilles CSS déjà codées permettant un gain de temps et une certaine uniformisation. Un des avantages de Bootstrap est la prise en charge du responsive, c’est pourquoi il n’y avait pas d’autres réels soucis au niveau du responsive. Cependant, pour éviter que ce souci réapparaisse, il ne faut pas oublier d’ajouter la classe .img-fluid à l’image concernée (*sur Bootstrap 5)*.

1. APPELS AUX SCRIPTS ET STYLESHEETS

Les stylesheets CSS sont appelées de manière asynchrones au chargement de la page Web, c’est-à-dire que le navigateur va lancer la lecture de ces feuilles de styles en même temps que l’HTML.

Le javascript, plus gourmand en ressources, lui est mal lu par les navigateurs, et surtout le bot google. De plus, son usage est souvent utile après le chargement des autres éléments de la page page. C’est pourquoi il est possible d’ajouter “*defer*” à une ligne script JavaScript, pour lancer un chargement en différé. Afin de gagner plus de temps, on peut également faire appel à ces scripts en fin de code source, ce que j’ai fait.

À un moment dans le CSS, on fait appel à une image pour le *background* d'un bloc. Le fichier CSS faisant 800 et quelques lignes, cette image n’est donc appelée qu’une fois la fin de la lecture de ce fichier. J’ai donc créé un autre fichier CSS avec les propriétés background afin d’assurer un chargement plus rapide.

1. ACCESSIBILITÉ POUR LES TA

Les TA (ou Technologies d’assistance) sont des logiciels et matériels utilisés par des personnes en situation de handicap pour améliorer leurs interactions avec le Web. Cela inclut les lecteurs d’écran qui vocalisent les pages web pour les personnes qui ne peuvent pas lire le texte, les loupes d’écran pour certains types de malvoyance, les logiciels de reconnaissance vocale et les contacteurs pour les personnes qui ne peuvent pas utiliser un clavier ou une souris.

Le WCAG, le rapport des normes d’accessibilité encourage de bonnes pratiques afin de pouvoir rendre le Web accessible à tous. On trouve par exemple les tags ARIA et la technologie WAI. Par exemple, un aria-hidden sur un lien fera que sur un lecteur d’écran, le texte alternatif du lien ne sera pas lu.

De la même manière, il est très important sur un site d’avoir des balises “alt” très concrètes sur les images utilisées: pour le logo du site, plutôt qu’un ensemble de mot-clés visant à créer du black hat SEO, la balise appropriée serait donc “logo site la Chouette”. C’est aussi pour cette raison que nous avons supprimé les images contenant du texte alors invisibles pour certaines personnes.

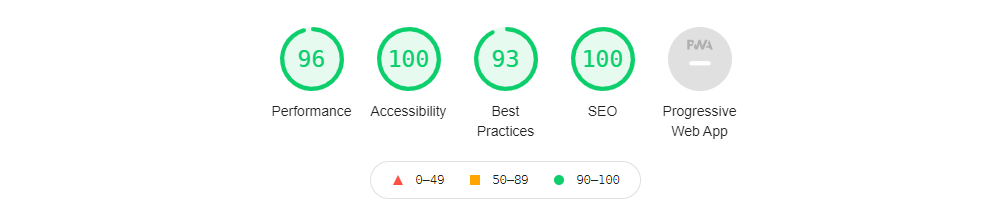
1. ALLÈGEMENT DES FICHIERS ET DU CODE

L’impression générale que renvoie ce site, quand on jette un coup d'œil au niveau de son code source, est assez mauvaise. En effet, ce n’est pas assez ordonné, et on a l’impression qu’il y a un surplus de fichiers et de caractères.

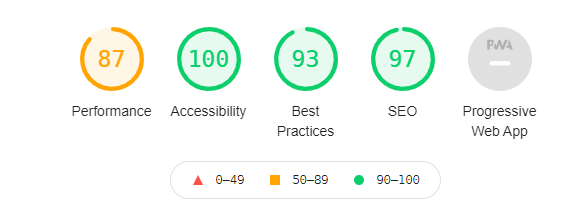
C’est vrai, en regardant le code et l’arborescence des fichiers, on peut rapidement se rendre compte qu’il y a des fichiers js comme gmaps.js qui ne sont utilisés nulle part mais sont quand même appelés, de même qu’un formulaire en php. On peut évidemment les supprimer. De même, sans le faire entièrement, on peut se rendre compte que de nombreuses classes css ne sont jamais utilisées et peuvent être supprimées. Inversement, dans l’HTML, certaines classes sont attribuées à des éléments alors qu’elles ne leur apportent aucune modification. J’ai procédé à un nettoyage de ce côté-là, mais pas entièrement, cela prendrait trop de temps. Cependant, il existe des outils techniques qui peuvent le faire. Cette technique s’appelle le *tree-shaking*. Il est possible de le faire avec le CSS et le JS et ainsi alléger le code.

1. CONCLUSION

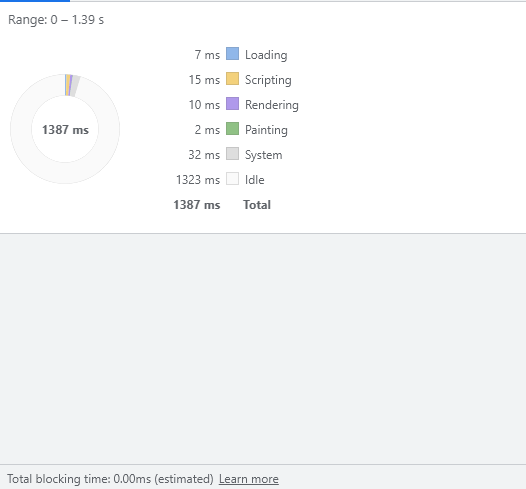
Il y aurait encore de nombreuses choses à rajouter au cas par cas, mais l’ensemble des axes abordés précédemment permettent d’aborder une vue générale des améliorations possibles. À titre de comparaison, voici des captures d’écran des rapports après application des conseils donnés précédemment.



Rapport Lighthouse version desktop APRÈS



Rapport Lighthouse version mobile APRÈS



*Rapport de performances au chargement de la page de l’inspecteur de la console de Google Chrome (on retire le Idle: 1387-1323=64ms(!))*

Comme on peut le voir, on peut noter une très nette amélioration des résultats après application des bonnes pratiques évoquées plus tôt. Il est très difficile d’obtenir un score parfait, mais voici quelques pistes pour obtenir un encore meilleur résultat:

* Faire du treeshaking pour les fichiers CSS et JS utilisés
* Utiliser une version plus récente de jQuery et Bootstrap: en plus de faire face à d’éventuelles failles de sécurité, les dernières versions ont un chargement plus rapide.

Pour aller encore plus loin, j’ai utilisé un CDN pour jQuery, Bootstrap et Font Awesome: le site est plus léger car rien n’est stocké dans les fichiers du site; et le CDN est l’alternative la plus rapide pour l’utilisateur.

De même, il y avait un commentaire pour les Analytics dans la balise <head>, j’ai inséré en commentaire le script qu’il faudrait compléter afin de pouvoir bénéficier des Analytics Google.