

Avant tout, la performance d'un site web est une étude cruciale qui impacte l'expérience utilisateur, le référencement SEO, la sécurité d'un site et surtout le chiffre d'affaires (si activité commerciale). Dans cet audit nous comparerons le site web <http://todolistme.net/> au projet todo-list-app.

En premier lieu, 1.3 secondes après ouverture du site, se chargent les fichiers css puis les images, et les fichiers javascript. Il est important de charger les styles en premier pour avoir un rendu visuel rapidement. Néanmoins le fichier css en charge du site est le dernier à être chargé (0.9s après ouverture du site). De même, le fichier principal chargé du fonctionnement front-end du site (javascript_e.js) se charge après les 15 autres fichiers javascript; au bout de 2.5sec après ouverture du site. Cela signifie que l'utilisateur ne peut pas utiliser le site pendant 2.5 secondes. Sachant qu'au bout d'une seconde l'utilisateur s'impatiente, c'est contre-productif pour l'expérience utilisateur. Entre temps des fichiers uniquement à l'affichage de la publicité sont chargés.

En second lieu, les images sont en majorité de petites icônes png en basse résolution. Vu qu'ils sont au nombre de 22, avec peu de détails et prenant en moyenne 200ms pour être chargés, ils auraient pu être facilement vectorisés voire stylisés dans un fichier css. Par exemple, 4 des icônes chargées (arrow_up.png, arrow_down.png, category_up.png, category_down.png) affichent un visuel pratiquement identique : des triangles équilatéraux. Ainsi, en css ils auraient gagnés en qualité (pas de pixellisation) mais également en vitesse de chargement car unifiés dans un même fichier le poids final serait similaire voire plus faible.

Ensuite en matière de sécurité les fichiers javascript présents en local sur <http://todolistme.net> présentent un risque de sécurité. En effet, leur code javascript n'est pas encapsulé. Ainsi on peut accéder à toutes les variables globales du site. C'est le cas de la variable productivityunit comme de la classe ProductivityUnit qui l'instancie qui peuvent être modifiées.

Contrairement au site <http://todolistme.net>, dans le projet todo-list-app grâce à l'encapsulation, il n'est pas possible d'accéder aux variables globales, aux classes et aux fonctions, dès le chargement du site. Par ailleurs, le chargement du site se fait en moins de 200ms, grâce à l'absence d'images png (1 fichier svg uniquement), et peu de dépendances en javascript comme l'absence de jQuery. De plus, les scripts sont chargés de manière séquentielle à la suite de 2 fichiers css. Cependant le fichier svg, chargé en dernier de manière asynchrone, devrait être chargé plus tôt. Au total 12 fichiers sont chargés contre 98 chez le concurrent ce qui explique la vitesse de chargement du site.

Malgré tout, le site web <http://todolistme.net>, présente 2 avantages sur le projet todo-list-app. D'une part les animations lors de l'ajout ou la suppression d'un item dans la todo list permettent de faire abstraction de certaines lenteurs pour le visiteur du site. D'autre part les fonctionnalités pour trier et imprimer la todo list voir l'ordonner dans le temps (demain, pour plus tard) permettront une meilleure expérience utilisateur car le champ des possibilités sera plus ouvert.