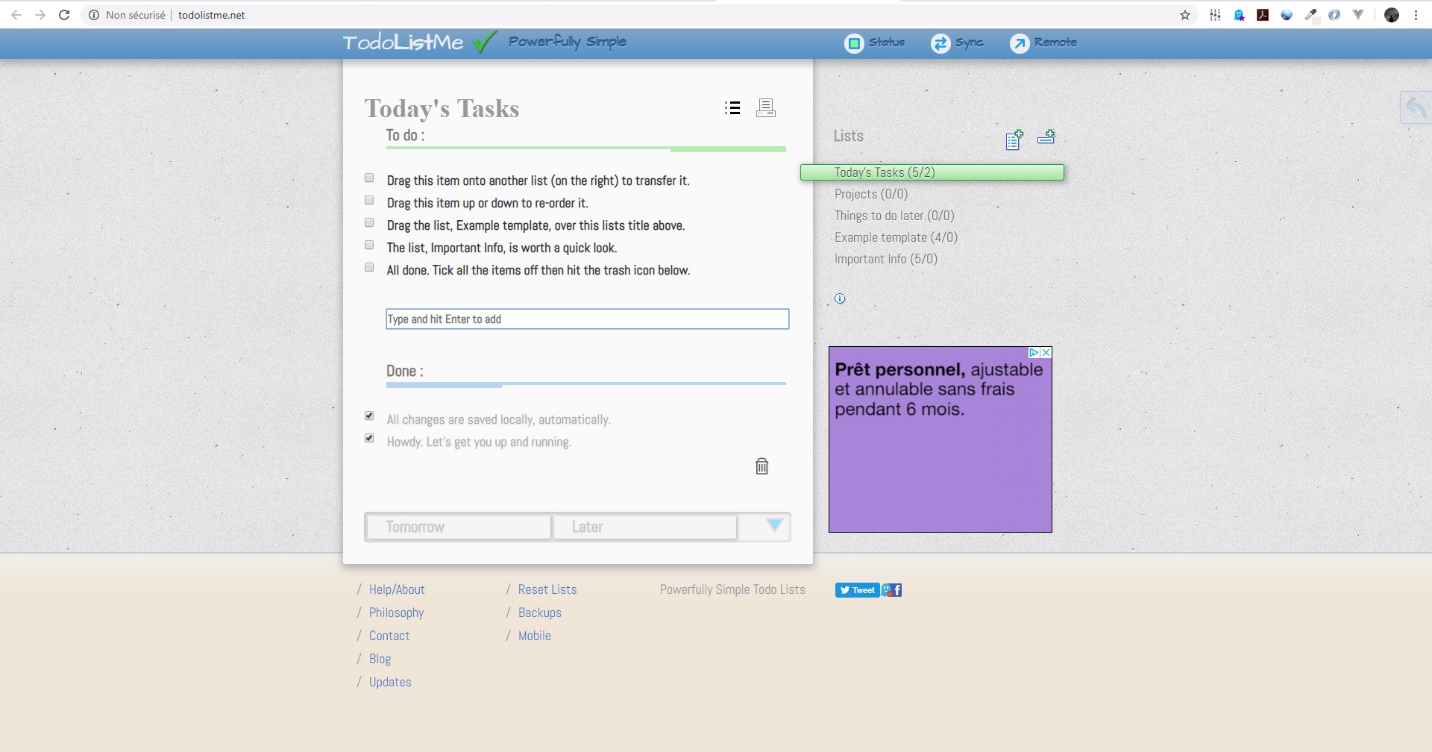
Audit Projet 8

Grégory Huyghe

# Application concurrente

L’application concurrente qui va être analysée dans ce document est TodoListMe (<http://todolistme.net/>). C’est aussi une application qui permet de lister des tâches à effectuer, de les ranger par catégories ou date, par complétion, de les supprimer, ainsi que de créer de nouvelles catégories de tri.



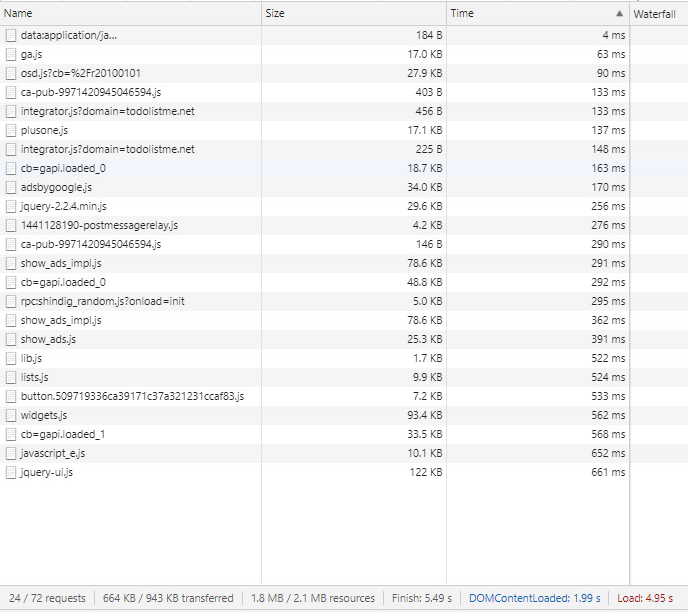
Ses possibilités sont donc plus nombreuses mais l’audit sur ses performances va nous révéler qu’il y a de nombreux écueils que l’on pourra éviter si nous souhaitons développer notre outil.

L’objectif de cet audit est d’avoir une vision d’ensemble des performances de l’application concurrente par rapport à la nôtre, afin d’avoir toutes les informations nécessaires en vue d’un éventuel scaling de notre application.

## Performance

La performance de l’application concurrente va être analysée aux outils de la console du navigateur Chrome.

Le chargement complet de l’application prend 4,95s, pour un total de 943kb et 72 fichiers transférés.

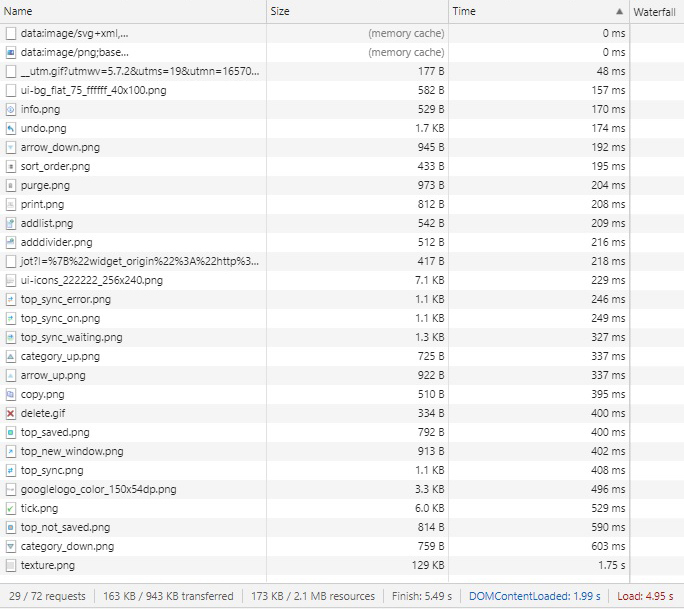


Au niveau des scripts, les fichiers nécessaires au bon fonctionnement de l’application sont lib.js, lists.js et javascript\_e.js. Leur poids est léger et pour les deux premiers le chargement rapide.

Au niveau des publicités, il y a deux fichiers show\_ads de 78.6 et 25.3kb chacun.

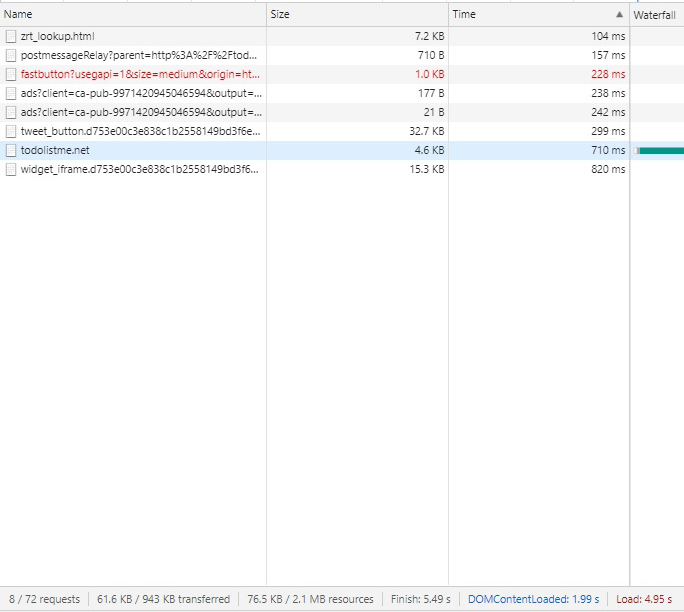
Ensuite, il y a widgets.js dont l’URL est : <http://platform.twitter.com/widgets.js>, donc le bouton Twitter pour un total de 93,4kb.

Enfin, jquery-ui.js qui charge en 661ms pour un total de 122kb.

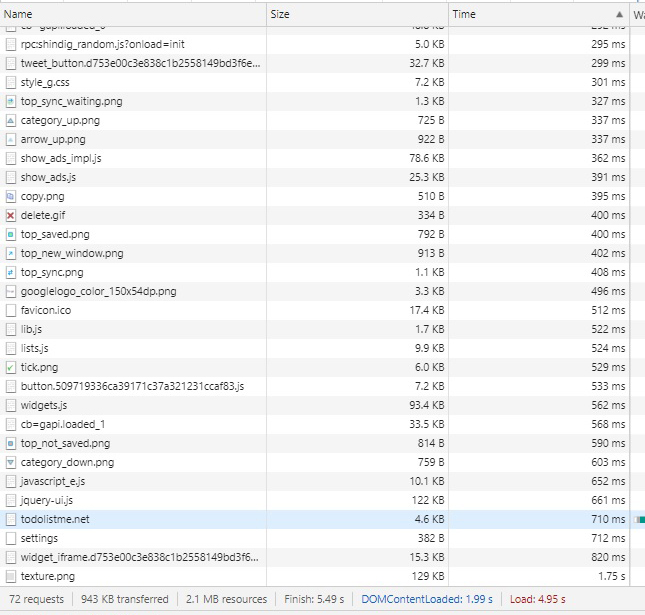


Les images impactent grandement le temps de chargement global de l’application.

Texture.png est l’image utilisée en background et pèse 129kb pour 1.75s de chargement.



En triant par Doc dans l’onglet Network de la console Chrome, on aperçoit des temps de chargement dédiés aux publicités (Google Ads) et à Twitter (tweet\_button et widget\_iframe).



En analysant le chargement au niveau global sans trier par type de fichiers, on aperçoit que parmi les éléments mettant plus de 600ms à charger, il y a deux images (category\_down et texture), deux scripts (javascript\_e et jquery-ui), la partie html du site (todolistme), une requête http pour Twitter (settings), et le widget de twitter (widget\_iframe).

* Bilan

Ces tests ont été effectués avec le cache désactivé. L’activation de celui-ci permet de diviser le temps de chargement par 2. Les points de vigilance semblent donc être l’implémentation des réseaux sociaux (Twitter), l’utilisation d’une image en background, l’utilisation de la bibliothèque jQuery (abandonnée dans Bootstrap 5), ainsi que l’ajout de publicités.

## Propositions en vue d’un scaling

Si nous souhaitions améliorer notre propre application, certains écueils peuvent être évités après analyse de l’application concurrente.

* Un background simple, une couleur. L’utilisation d’une image alourdie considérablement le chargement du site. Même si la mise en cache supprime ce problème quand l’utilisateur revient, la première fois est importante en termes d’image pour l’entreprise.
* Ne pas monétiser le site web ou de manière plus légère. Offrir des fonctionnalités premium contre abonnement peut permettre de générer des revenus sans alourdir le fonctionnement de l’application, surtout qu’il n’y a pas de mise en cache pour cet élément.
* Implémenter les réseaux sociaux mais laisser l’utilisateur libre de les activer ou non. Cela permet de respecter la RGPD et allégera le chargement de l’application. La mise en cache rendra la navigation plus facile pour les utilisateurs réguliers.
* Ne pas utiliser jQuery mais plutôt un framework tel que Vue, ou si les fonctionnalités sont simples, directement vanilla JS. Ceci évitera le chargement d’une librairie. Les navigateurs ont maintenant une compatibilité plus performante et couvrir IE n’est pas nécessairement utile.