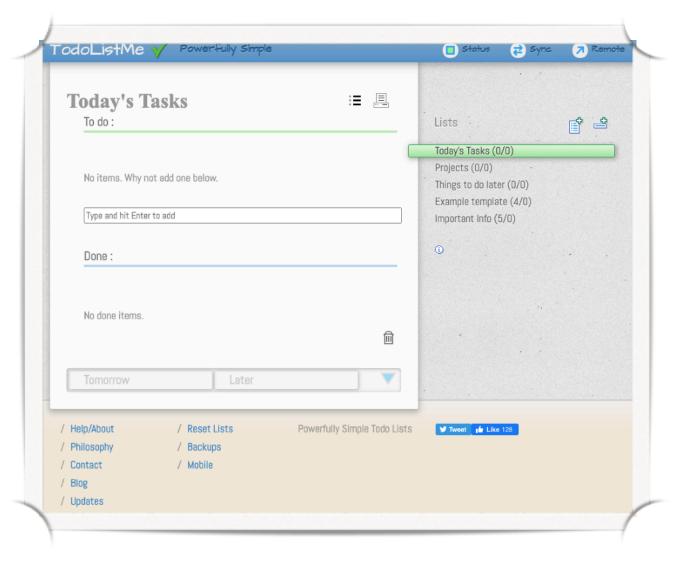
AUDIT PROJET P8

1. Zoom sur l'application concurrente

L'objectif de cet audit est donc d'avoir une vision d'ensemble des performances de cette application concurrente afin d'optimiser la notre en vue d'un éventuel scaling.

1.1 Présentation.

L'application que nous allons analyser lors de cet audit est TodoListMe. C'est une application qui permet également de lister des tâches. Par contre, elle va plus loin dans les fonctionnalités offertes à l'utilisateur comme par exemple de pouvoir ranger les différentes tâches par catégories.



1.1 Comparatif entre les deux applications.

	TODOLISTME	TODOS	
Rangement par catégorie	✓	X	
Edition de tâche	✓	✓	
Programmation de tâches	>	X	
Publicités	>	×	
Réseaux sociaux		×	
Stockage	Local storageserveur	Local storage	

Comme on peut le voir, l'application concurrente possède un nombre de fonctionnalités supérieur à la notre. L'application concurrente à choisit d'implanter de la publicité ainsi que des widget pour les réseaux sociaux. Pour finir, Todolistme propose aux utilisateurs possédant un compte de pouvoir stocker leur liste de tâche sur un serveur. De ce fait, une fois authentifié, l'utilisateur peut retrouver sa liste sur différents appareils.

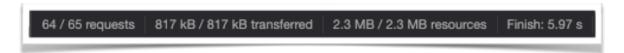
2. Résultats de l'audit Network (console Google Chrome).

Nous allons utiliser les outils de la console du navigateur Chrome pour analyser les performances.

Name	Size	Time ▼
texture.png	132 kB	537 ms
todolistme.net	4.7 kB	426 ms
✓ tick.png	6.1 kB	272 ms
javascript_e.js	10.3 kB	270 ms
top_not_saved.png	814 B	267 ms
lib.js	1.7 kB	213 ms
top_sync.png	1.1 kB	204 ms
cb=gapi.loaded_1	33.9 kB	175 ms
adddivider.png	512 B	164 ms
category_down.png	759 B	164 ms
purge.png	973 B	151 ms
■ info.png	529 B	150 ms
sort_order.png	433 B	149 ms
addlist.png	542 B	149 ms
print.png	812 B	147 ms
settings	167 B	147 ms
undo.png	1.7 kB	146 ms
lists.js	10.1 kB	146 ms
arrow_down.png	945 B	143 ms
style_g.css	7.4 kB	142 ms
top_sync_error.png	1.1 kB	141 ms
arrow_up.png	922 B	141 ms
x delete.gif	334 B	141 ms
a category_up.png	725 B	139 ms
top_new_window.png	887 B	138 ms
top_saved.png	792 B	138 ms
top_sync_on.png	1.1 kB	136 ms
copy.png	510 B	135 ms
top_sync_waiting.png	1.3 kB	134 ms
jquery-ui.js	125 kB	133 ms
show_ads.js		130 ms
plusone.js	0 B	102 ms
3aUMYd3yWUN.js?_nc_x=lj3Wp8lg5Kz	133 kB	91 ms

2.1 Le temps de chargement.

On constate que, sans le cache, la page aura mis **5,97s** à se charger pour un total de **65 fichiers** transférés. Par curiosité, j'ai également essayé avec le cache et cela a mis aux alentours de **4s**.



Si on s'intéresse plus en détail aux éléments, on peut voir que l'élément qui prend le plus de temps à se charger est le fichier « texture.png » en **537 ms**. Quand on clique dessus, on peut voir qu'il s'agit de l'image utilisée en background. D'ailleurs, de manière globale, les images utilisent énormément de temps de chargement.

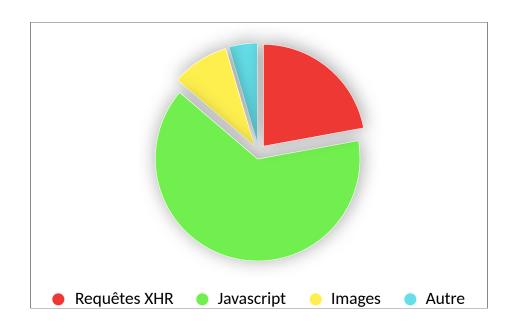
Les principaux fichiers scripts se chargent plutôt rapidement. On est à **270ms** pour le plus long (javascript_e.js) et **130ms** pour le plus rapide (plusone.js).

Les fichiers CSS ont également un délais de chargement assez rapide à 142 ms.

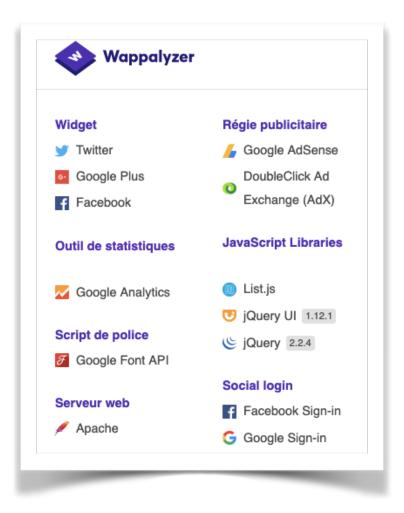
En regardant en détails les fichiers chargés, on aperçoit des temps de chargement dédiés aux publicités (show_ads.js) et au réseaux sociaux (Twitter et Facebook).

2.2 Le poids des fichiers.

Au niveau du poids des fichiers, on retrouve en première position les fichiers Javascript. Sachant que le poids total de l'application est de **2,2 MB**, on constate qu'ils prennent plus de la moitié du poids total de l'application. On retrouve ensuite des requêtes XHR pour l'implémentation des réseaux sociaux pesant **0,5MB** et pour finir, les images au format png qui représentent **0,2MB** sur le poids total de l'application.



2.3 Technologies utilisées.



2.4 Conclusions

Les performances du site http://todolistme.net/ sont loin d'être optimales en raison notamment des critères suivants :

- certains fichiers scripts qui peuvent peser énormément.
- l'implémentation des réseaux sociaux (Twitter et Facebook) ou, de manière plus générale, l'ajout de code tiers qui peuvent ralentir l'application.
- l'utilisation d'images qui peuvent ralentir les temps de chargement (ex. « texture.png »),
- l'ajout de publicités.

3. Le scaling

Après avoir analyser les performances de cette application concurrente, il va être important d'apporter une attention particulière au points suivants :

- Compresser nos fichiers HTML,CSS et JS en les minifiant. De nombreux outils gratuits permette d'éliminer les espaces, les sauts de ligne, les commentaires et les délimiteurs de bloc afin d'alléger le fichier au maximum.
- Eviter de mettre une image en arrière-plan. La remplacer par une simple couleur peut être une solution car on a pu voir que le fichier « texture.png » était lourd et prenait énormément de temps de chargement. De plus, il serait intéressant de mettre l'ensemble de nos images aux formats nouvelle génération tels que JPEG 2000 ou JPEG XR.
- Il faut garder en tête que notre application sera plus rapide sans publicités. SI une monétisation de notre application est envisagée à l'avenir, il va falloir trouver un juste milieu afin que la présence de pub n'ait pas d'impact trop négatif sur l'expérience client.
- Réduire les code tiers afin de limiter les requêtes. On a pu voir que l'implémentation des réseaux sociaux pouvait ralentir l'application. On pourrait, par exemple, ne pas les charger dès l'ouverture de la page mais ultérieurement de manière asynchrone ou à la demande du client.
- Et bien sûr, utiliser la mise en cache.

4. Bonnes pratiques.

Même si les points suivant n'influent pas sur la performance globale du site, il faudrait néanmoins y accorder une importance particulière.

Accessibilité :

- Certaines images n'ont pas d'attribut alt.
- Des champs de formulaires n'ont pas de <label>.
- Site web non responsive alors qu'aujourd'hui, un site se doit de l'être.

Sécurité:

Le site n'utilise pas le protocole https.

SEO:

- Le site ne contient pas de **tag <meta name="viewport">** avec attribut **width** ou **initial- scale**.
- Les tap-target (zone de toucher au doigt sur dispositif mobile) ont une taille inadaptée.
- La balise <html> n'a pas d'attribut linguistique (lang).