

# Audit de performance de

## <http://todolistme.net/>

### 1- Résumé du fonctionnement de l' application




Notre concurrent propose une application de todo-list perfectible.

Cette application permet de faire plusieurs choses :


- créer des catégories de liste de tâches à effectuer
- créer des liste de tâches à effectuer, les enregistrer, éditer, effacer
- donner une temporalité

Via Wappalyzer nous voyons la présence des éléments suivants sur le site :

#### Widget

-  Facebook
-  Google Plus
-  Twitter


#### Outil de statistiques

-  Google Analytics

#### Framework JavaScript

-  jQuery 2.0.3
-  jQuery UI 1.10.3
-  List.js

#### Script de police

-  Google Font API



#### Serveur web

-  Apache

#### Langage de programmation

-  PHP 5.3.28

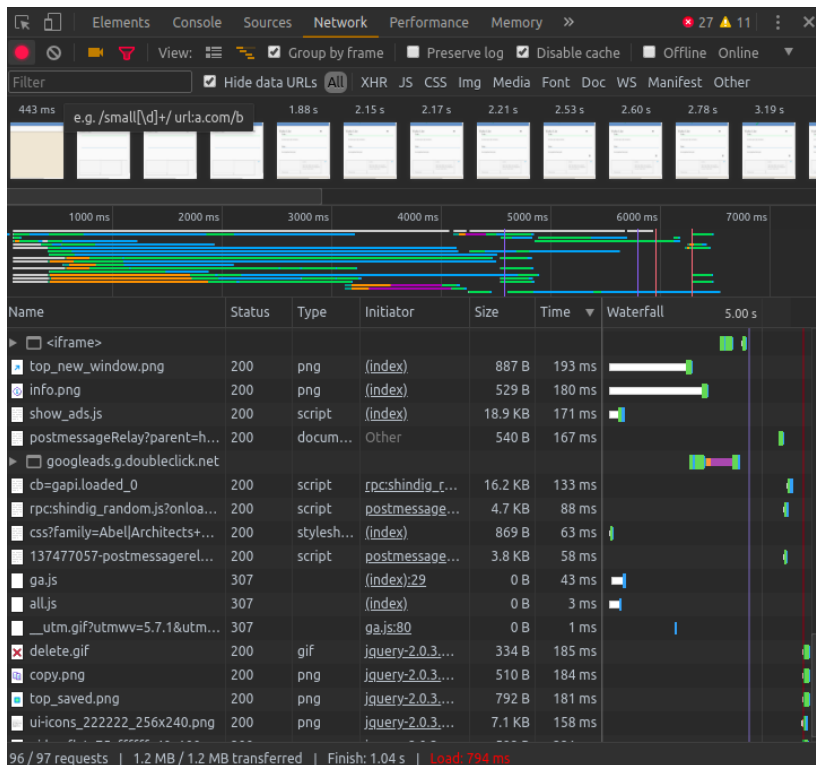
#### Régie publicitaire

-  DoubleClick Ad Exchange  
(AdX)
-  Google AdSense

### 2- Utilisation des outils du navigateur Chrome

L' audit à été réalisé avec les outils fournis par la console de notre navigateur, dans ce cas Chrome Version 64.0.3282.119 de Google.

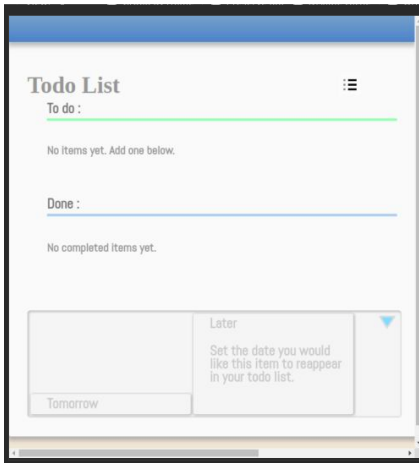
## 2-1- Outil « network »



L'affichage complet de la page prend 6,49 secs.

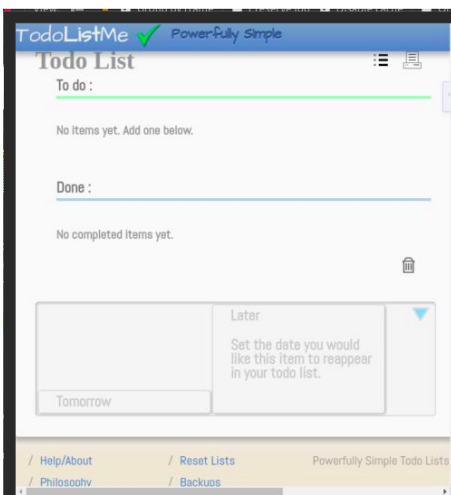
Les différentes image prennent un peu de temps à charger, amélioration possible ici.

1,88 secondes pour ce visuel :



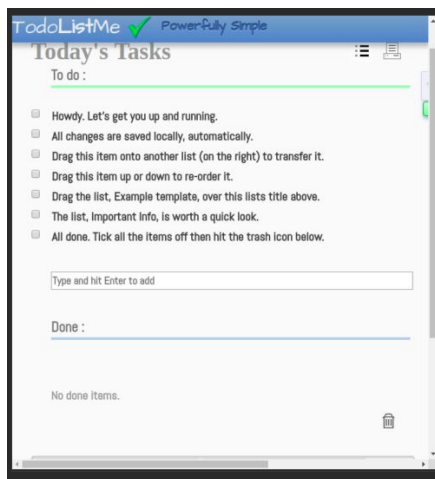
1,88 secondes pour afficher le squelette de l'application. Pas extrêmement rapide pour des éléments de base. La console montre que c'est vraiment les PNG qui posent le plus problème.

5,91 secondes pour ce visuel:



5,91 secondes pour l'affichage de l'entête et du footer : grosse marge d'amélioration en optimisant les images (format SVG à la place de PNG).

6,31 secondes pour ce visuel :



0,4 secondes pour l'affichage des éléments todo, c'est plutôt rapide.

On peut optimiser le temps de chargement en utilisant le cache pour les images.

Les 2 fichiers les plus longs à charger sont le background (texture.png) et jQuery (jquery-ui.js).

Le background peut-être optimisé (background repeat + fichier en SVG), jQuery intégré uniquement les fonctions utiles.

Name	Headers	Preview	Response	Cookies	Timing
texture.png					
all.js					
jquery-ui.js					
javascript_e.js					
lists.js					
integrator.js?domain=todolistme.net					
tick.png					
ga.js					
brdGGFwqYxjg2CD1E9o7g.woff2					
lib.js					
top_not_saved.png					
widgets.js					
undo.png					
aswift_0					
integrator.js?domain=todolistme.net					
statusbar_facebook.com					

Name	Headers	Preview	Response	Timing
texture.png				
all.js				
jquery-ui.js				
javascript_e.js				
lists.js				
integrator.js?domain=todolistme.net				
tick.png				
ga.js				
brdGGFwqYxjg2CD1E9o7g.woff2				
lib.js				
top_not_saved.png				
widgets.js				
undo.png				
aswift_0				

URL	Type	Total By...	Unused Bytes	
http://code.jquery.com/ui/1.10.3/jquery-ui.js	JS (coarse)	435 844	330 667 75.9 %	
http://code.jquery.com/jquery-2.0.3.min.js	JS (coarse)	83 612	33 667 40.3 %	
http://code.jquery.com/ui/1.10.3/.../jquery-ui.css	CSS	32 046	31 145 97.2 %	
http://todolistme.net/javascript/javascript_e.js	JS (coarse)	33 830	24 992 73.9 %	
http://todolistme.net/javascript/lists.js	JS (coarse)	30 111	19 862 66.0 %	
http://todolistme.net/css/style_g.css	CSS	22 374	10 367 46.3 %	
http://todolistme.net/javascript/lib.js	JS (coarse)	3 837	2 211 57.6 %	
chrome-extension://gppongmyhkpfnb.../inject.js	JS (coarse)	1 354	6 0.4 %	

442 KB of 628 KB bytes are not used. (70%)

Avec l'outil coverage on se rend compte par exemple que 75 % JQuery est non utilisés.

## 2-3- Outil « audit »

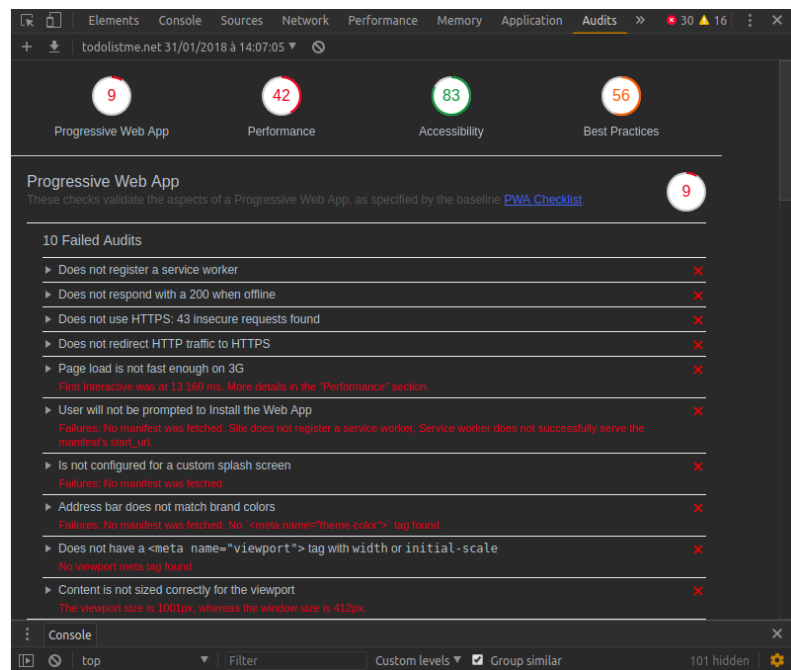
4 catégories :

- Progressive Web App
- Performance
- Accessibilité
- Bonnes pratiques

### 2-3-1- Progressive Web App

9 % de réussite, 10 tests ratés sur 11.

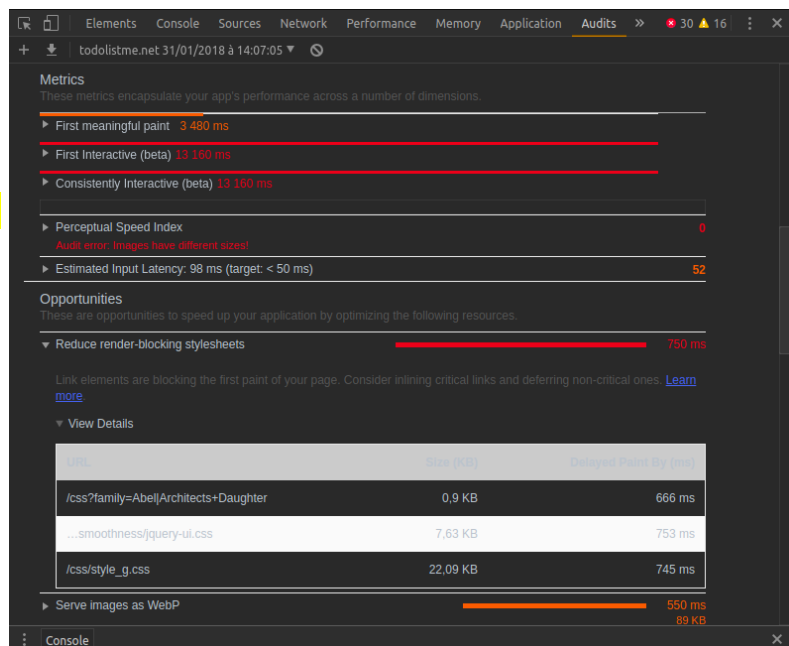
Le site de notre concurrent n'est pas du tout orienté PWA, un point sur lequel nous avons une énorme marge de manœuvre.



### 2-3-2- Performance

42 % de test validé, application trop lente.

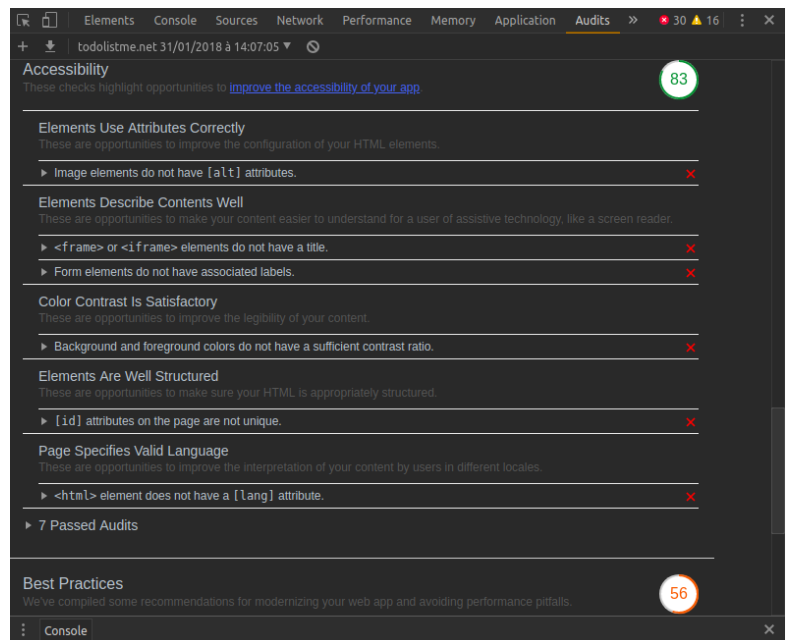
Les PNG peuvent être amélioré (SVG, background repeat) tout comme certains fichier JS (utilisation de jquery).



### 2-3-3- Accessibilité

83 % de réussite, quelques points d'amélioration comme :

- ajouter un **attribut alt** pour les images
- accentuer plus les **contrastes**
- ajouter un **attribut lang** dans le HTML

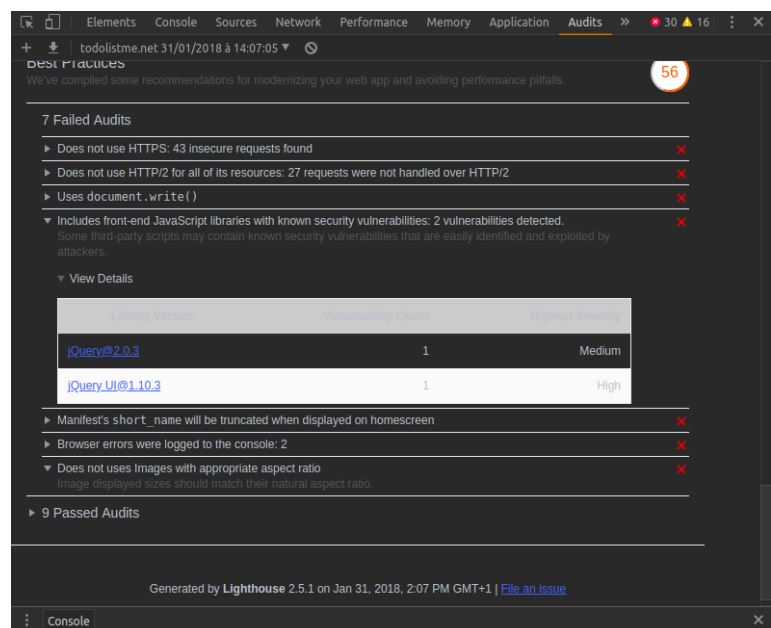


### 2-3-4- Bonnes pratiques

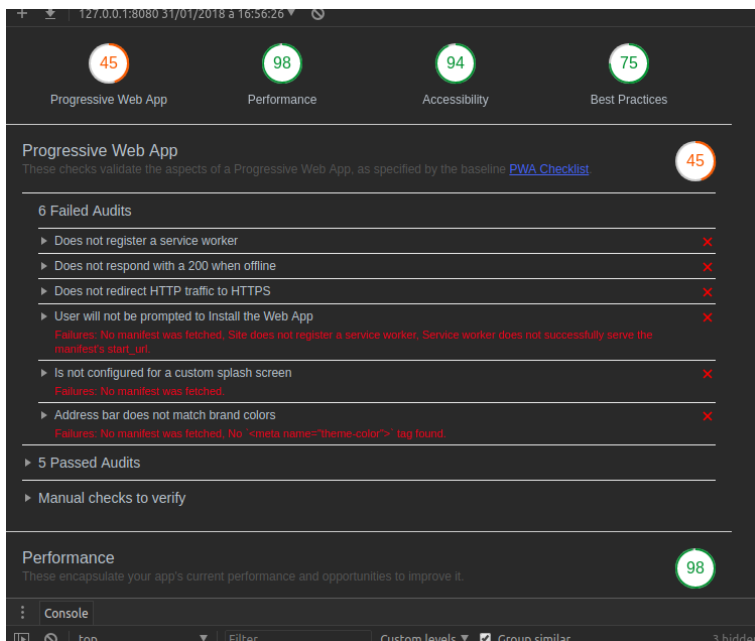
Recommandations pour améliorer les performances et moderniser l'application.

56 %, **grosse marge de progression** ici également.

- pas de HTTPS
- 2 failles de sécurités dans les librairies front utilisés
- présences de deux erreurs dans la console



### 3- Comparaison avec notre application



#### Amélioration possible du PWA :

- simple comme `<meta name= «theme-color »>`
- gestion du offline avec un splashscreen...

#### Performance : OK

#### Bonnes pratiques :

4 tests ratés sur 12. A vérifier.

#### Accessibilité :

- label : `<input class="new-todo" placeholder="What needs to be done?" autofocus>`
- amélioration du contraste entre le 1<sup>er</sup> plan et le fond d'écran.

### 4- Scalling :

Fonctionnalité que l'on pourrait éventuellement intégrer à notre application :

- temporalité
- catégories de liste

### 5- Résumé

Application de notre concurrent :	Notre application :
Lente, pas optimisé	Rapide, optimisé et performante
pas orienté PWA	PWA améliorable, axe d'amélioration évident
Accessibilité correcte +	Bonne accessibilité, amélioration via <code>&lt;label&gt;</code> et contrastes pour que ce soit parfait
Design discutable, quelques fonctionnalités intéressantes	La temporalité et les catégories de listes devraient être intégrés ?