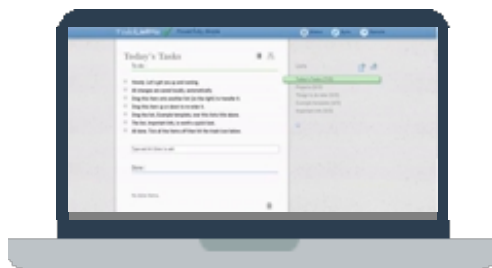




Rapport d'analyse performance et qualité



<http://todolistme.net/>

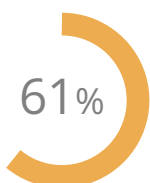
Ce rapport a été généré sur Dareboost.com, **service en ligne d'analyse et de surveillance de la performance et de la qualité des sites web.**

N'hésitez pas à découvrir [nos offres](#) ou à nous contacter : contact@dareboost.com.

Sommaire

| | |
|------------------------------|----|
| Résumé | 3 |
| Conseils et bonnes pratiques | 4 |
| Accessibilité | 4 |
| Compatibilité | 8 |
| Nombre de requêtes | 10 |
| Optimisation du rendu | 15 |
| Politique de cache | 19 |
| Qualité | 21 |
| Référencement (SEO) | 27 |
| Sécurité | 32 |
| Volume de données | 39 |
| Google Analytics | 46 |
| Google Font API | 47 |
| jQuery | 48 |

Résumé



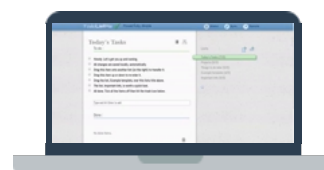
Problèmes



Améliorations



Succès



VISITEUR SIMULÉ :



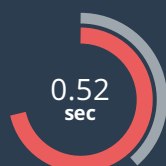
Chrome



Paris

8.0/1.5Mbps (Latence : 50 ms)

Premier octet



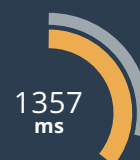
Début affichage



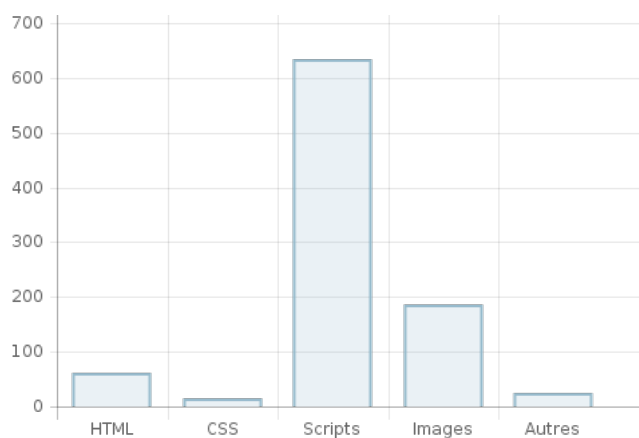
Fin chargement



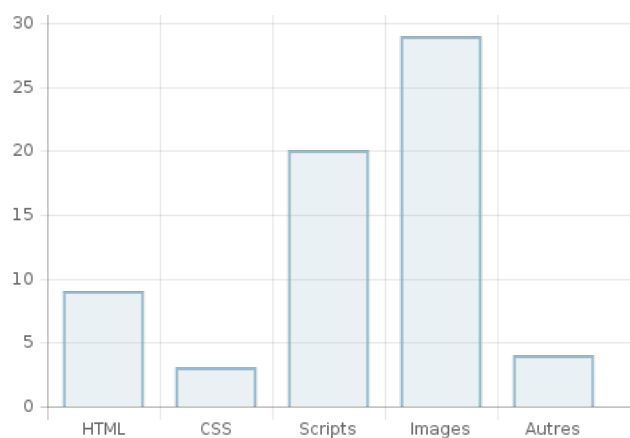
Speed Index



Répartition du poids



Répartition du nombre de requêtes



Technologies détectées



DoubleClick Ad Exchange (AdX)



Google AdSense



Google Analytics



Google Font API



Nginx



jQuery



jQuery UI

Conseils et bonnes pratiques

Accessibilité

Vos principaux axes d'optimisation

0/100

#2334

! La langue de la page n'est pas spécifiée

Votre page devrait posséder un attribut `lang` sur le nœud racine `html` : cela permettra aux liseuses d'écran de prendre correctement en charge votre site.

0/100

#2569

! 1 élément vide peut perturber les lecteurs d'écrans

Les éléments `<p>`, ``, `<button>`, `<legend>`, `<caption>`, `<figcaption>` and `<quote>` ne doivent pas être vides car s'ils le sont, certains lecteurs d'écran auront des difficultés à interpréter leur présence.

Supprimez ces éléments vides de votre code ou décochez-les avec l'attribut `aria-hidden` pour que les lecteurs d'écran les ignorent.

```
<p aria-hidden="true"></p>
```

Exemple

L'élément suivant est vide:

- `<p id="synctime">`

! Expliquez le but de chaque champ de formulaire

Améliorez l'expérience utilisateur sur votre site en spécifiant une étiquette descriptive pour chaque champ de formulaire.

Un formulaire est composé de plusieurs champs qui doivent être les plus explicites possibles pour que l'utilisateur comprenne rapidement leur fonction.

Définir un label

Pour indiquer l'objectif d'un champ, nous vous recommandons l'utilisation de la balise `label` :

```
<label for="nom">Renseignez votre nom :</label>
<input id="nom" type="text" name="nom">
```

Exemple

Si vous ne souhaitez pas afficher de label, vous pouvez utiliser les attributs `aria-label` ou `title` (ce dernier n'est cependant pas supporté par certaines liseuses d'écran). À noter qu'il n'est **pas conseillé** d'utiliser l'attribut `placeholder` seul.

En savoir plus sur [l'association de texte à un champ de formulaire](#).

Cette page contient **14 champs sans explications sur leur objectif** :

- `<input type="text" id="syncname">`
- `<input type="password" id="syncpassword">`
- `<input type="button" id="syncbutton" value="Sync" onclick="sync_login();">`
- `<input type="text" id="registersyncname">`
- `<input type="password" id="registersyncpassword">`
- `<input type="button" id="registerbutton" value="Create Account" onclick="create_account();">`
- `<input type="checkbox" onclick="finish_todo(0, 0)">`
- `<input type="checkbox" onclick="finish_todo(0, 1)">`
- `<input type="checkbox" onclick="finish_todo(0, 2)">`
- `<input type="checkbox" onclick="finish_todo(0, 3)">`
- `<input type="checkbox" onclick="finish_todo(0, 4)">`
- `<input type="checkbox" onclick="finish_todo(0, 5)">`
- `<input type="checkbox" onclick="finish_todo(0, 6)">`
- `<input id="newtodo" type="text" value="New Todo" class="newtodonormal" placeholder="Type and hit Enter to add">`

Saviez-vous que ?

#72

Aucune balise `<noscript>` n'est définie

Lorsque la page utilise des scripts, il est souvent conseillé de définir au moins une balise `<noscript>`. Celle-ci est nécessaire pour afficher un message lorsque le JavaScript est désactivé par l'utilisateur.

```
<script type="text/javascript">
  document.write('Hello World!')
</script>
<noscript>Your browser does not support JavaScript!</noscript>
```

Exemple

Bravo ! Vous respectez les bonnes pratiques suivantes

100/100

#2380

Vos tableaux contiennent tous des balises `<th>`

Un tableau représente un ensemble de données, sous plusieurs dimensions. Pour aider à la compréhension de ces données, il est nécessaire de fournir des titres pour chaque ligne/colonne.

La balise `<th>`

La représentation d'un titre d'une ligne/d'une colonne d'un tableau se fait par l'intermédiaire de la balise `<th>`. Les données associées sont quant à elles représentées dans des balises `<td>`.

Voici un exemple classique d'utilisation de la balise `<table>` :

```
<table>
<caption> <p>table 1. List of HTML elements</p> </caption>
<thead>
  <tr> <th>Number</th> <th>Element</th> </tr>
</thead>
<tbody>
  <tr> <td>4.1.1</td> <td>HTML</td> </tr>
  <tr> <td>4.2.1</td> <td>HEAD</td> </tr>
</tbody>
</table>
```

Exemple

[Voir les spécifications de la balise `<th>`.](#)

Tous les tableaux de cette page définissent des balises `<th>`. Il s'agit d'une bonne pratique.

100/100

#2395

Aucune `<meta>` 'refresh' détectée

Il n'est pas recommandé de forcer automatiquement le rafraîchissement d'une page. Ce comportement perturbe les utilisateurs, qui subissent l'action sans pouvoir la maîtriser.

Cette page n'utilise pas de balise `<meta refresh>`. Il s'agit d'une bonne pratique.

100/100

#2410

Aucun attribut `"src"` n'est vide

L'attribut `"src"` permet d'accéder à une autre ressource. Rien ne justifie son utilisation avec une valeur vide.

Tous les attributs `src` de la page définissent une cible. Il s'agit d'une bonne pratique.

100/100

#2423

✓ Vos liens ont des titres cohérents

Un lien est plus attractif si son texte décrit ce qui se cache derrière. Vous pouvez par ailleurs en profiter pour travailler ces textes avec des mots clés bien choisis, dans une démarche d'amélioration de votre référencement.

Tâchez de mettre en place une description du lien au sein de vos balises `<a>`, plutôt que d'indiquer le lien lui-même. Exemple : `Ma description`

Les contenus de vos balises `<a>` sont différents des liens eux-mêmes. Il s'agit d'une bonne pratique.

100/100

#2468

✓ Votre titre principal est bien le premier titre spécifié

Facilitez le travail des liseuses d'écran en indiquant votre titre principal (`<h1>`) avant tout autre titre.

Il est conseillé de respecter une hiérarchie cohérente entre vos titres. Le titre principal de la page doit ainsi être spécifié avant tout autre titre `<hX>`.

100/100

#2523

✓ Tous vos labels sont liés à un élément

L'attribut `for` permet d'associer le label à un élément de la page, et aidera les lecteurs d'écran à mieux interpréter votre contenu.

Label et attribut for

Un label a pour vocation de décrire un élément (texte à remplir, option à cocher, etc). Par exemple, lors du clic sur un label associé à un input de type radio, l'option associée au label sera directement sélectionnée, améliorant ainsi l'expérience utilisateur.

Comment se servir d'un label ?

Pour lier le label à un élément de la page, vous devez passer par un identifiant. Exemple :

```
<form action="/action">
  <label for="monIdentifiant">
    <input type="radio" name="mesOptions" id="monIdentifiant" value="1" >
```

Bravo ! Vous respectez les bonnes pratiques suivantes

100/100

#80

✓ **Aucune balise frameset, frame ou noframes détectée**

Ces balises sont aujourd'hui à proscrire, car elles présentent des problèmes pour la cohérence de navigation, le référencement, ou encore la sauvegarde de pages dans les favoris d'un navigateur.

Aucune de ces balises n'est détectée sur cette page.

L'utilisation de la balise `iframe` est préférée.

100/100

#82

✓ **Aucune applet Java détectée**

Les applets Java sont considérées comme obsolètes en 2015, et n'apportent rien par rapport à ce qui peut être fait nativement avec le HTML5. Leur utilisation vous expose à des soucis de compatibilité et peut renvoyer des signaux négatifs à vos utilisateurs (exemple : le navigateur qui indique qu'un contenu a été bloqué car susceptible d'être dangereux).

Félicitations, cette page ne contient pas d'applet Java.

100/100

#89

✓ **N'utilisez pas <bgsound> pour mettre un fond sonore**

Aucune balise `bgsound` détecté. Il s'agit d'une bonne pratique: cette balise n'est pas un standard HTML.

Cet élément ne fait pas partie des standards HTML. Son utilisation peut entraîner des problèmes de compatibilité. [Voir plus d'informations](#).

Utilisez la balise `audio` pour fournir du contenu audio sur votre page.

```
<audio src="my-audio-file.ogg" autoplay>
  Your browser doesn't support the audio element.
</audio>
```

Exemple

100/100

#99

✓ **Aucun lien vers un document Word détecté**

Les documents `.doc` et `.docx` ne garantissent pas la compatibilité avec les principaux systèmes d'exploitation. Il est recommandé d'utiliser des documents PDF.

Cette page contient uniquement des liens standards.

100/100

#2382

✓ **Aucun attribut déprécié n'est utilisé dans la balise <body>**

Certains attributs de présentation sont dépréciés en HTML 5 : `alink` / `background` / `bgcolor` / `link` / `text` / `vlink`

Félicitations, nous n'avons pas trouvé d'attributs dépréciés dans la balise `body`.

Préférez l'utilisation du CSS à la place.

100/100

#2424

✓ **Aucun BOM (Byte Order Mark) détecté**

Certains robots/analyseurs de textes ne sont pas capables d'interpréter une page contenant un BOM.

Qu'est ce que le BOM ?

Le BOM est un caractère invisible placé en début de page, dont le but est d'aider à déterminer quel encodage est utilisé par la page. Or, les bonnes pratiques du web encouragent l'utilisation de l'en-tête HTTP `Content-Type` pour définir l'encodage utilisé par la page. Le BOM ne trouve donc pas son intérêt dans ce contexte.

Outre le fait que ce type d'indicateur est inutile sur le web, il peut impliquer des problèmes. C'est par exemple le cas pour la validation W3C, qui va tenter d'interpréter le premier caractère, qui correspond au BOM. Le document ne sera alors pas valide.

Aucune ressource n'utilise un indicateur d'ordre des octets (BOM). Il s'agit d'une bonne pratique.

100/100

#2454

✓ **Vos caractères sont encodés en UTF-8**

Votre contenu est lisible par le plus grand nombre possible d'internautes.

Codage de caractères

Le codage de caractères permet d'indiquer au navigateur web comment interpréter les octets de la page web pour les convertir en caractère lisible par l'utilisateur. De nombreux codages existent afin de pouvoir représenter tous les caractères spécifiques à chaque langue. Notez que les panels de codages supportés sont différents d'une machine à une autre. Si vous utilisez un codage trop spécifique, de nombreux internautes ne pourront donc pas interpréter correctement la page.

Pourquoi choisir l'UTF-8 ?

L'UTF-8 est reconnu pour être un codage supporté par quasiment tous les internautes, et prenant en compte une grande variété de caractères. Son universalité est donc particulièrement bien adaptée au web.

100/100

#2488

✓ **Aucune ressource Flash détectée**

Flash est considéré comme obsolète en 2015, et n'apporte rien par rapport à ce qui peut être fait nativement avec le HTML5. Son utilisation vous expose à des soucis de compatibilité et peut renvoyer des signaux négatifs à vos utilisateurs (exemple : le navigateur qui indique qu'un contenu a été bloqué car susceptible d'être dangereux).

Par ailleurs, Google indique directement dans ses résultats de recherche sur mobile que [le site risque de pas fonctionner \(EN\)](#).

Félicitations, cette page ne contient pas de ressources utilisant Flash.

100/100

#2493

✓ **Le plugin Silverlight n'est pas utilisé**

Le plugin Silverlight est considéré comme obsolète en 2015, et n'apporte rien par rapport à ce qui peut être fait nativement avec le HTML5. Son utilisation vous expose à des soucis de compatibilité et peut renvoyer des signaux négatifs à vos utilisateurs (exemple : le navigateur qui indique qu'un contenu a été bloqué car susceptible d'être dangereux).

Félicitations, cette page ne nécessite pas le plugin Silverlight.

Vos principaux axes d'optimisation

0/100

#2344

! 1 ressource est inaccessible

Attention, certaines requêtes liées à cette page rencontrent des erreurs. Il est par exemple possible :

- que vous ayez commis une erreur dans votre code HTML, CSS ou JavaScript
- que votre serveur ait rencontré une erreur non gérée
- que certains services utilisés par votre site ne répondent pas

Les codes HTTP suivants ont été renvoyés :

- 404:
 - [plus.google.com/se/0/_/+1/fastbutton?us\[...\]w/m%3D__features__](https://plus.google.com/se/0/_/+1/fastbutton?us[...]w/m%3D__features__)

Ces erreurs nuisent probablement à certains contenus ou comportements de votre site, et provoquent un trafic réseau inutile, ce qui impacte donc le temps de chargement de votre site.

! Économisez 18 requêtes à l'aide de sprites CSS

Combiner les petites images dans une sprite CSS réduit le nombre de fichiers que le navigateur doit télécharger et accélère donc le chargement de votre page.

Les sprites CSS

Une sprite CSS est un fichier unique dans lequel plusieurs petites images sont regroupées, positionnées les unes à côté des autres. L'affichage de chaque petite image au sein de votre page se fait ensuite par l'application de styles CSS. Une seule requête est alors nécessaire pour récupérer l'ensemble des images. Cette technique est à utiliser uniquement pour les petites images, comme des icônes par exemple, afin que la sprite CSS ne soit pas trop lourde.

Exemple

Voici un exemple de sprite CSS, regroupant plusieurs icônes :



La page applique par ailleurs les styles associés à la sprite, du type :

```
.sprite {  
  background-image: url(img/sprite.png);  
  background-repeat: no-repeat;  
  display: block;  
}  
  
.sprite-browsers-firefox {  
  width: 31px;  
  height: 28px;  
  background-position: -74px 0;  
}
```

Exemple

Il reste à définir la bonne classe dans votre fichier HTML, et l'icône apparaît :

```
<span class="sprite sprite-browsers-firefox"></span>
```

Exemple

Comment créer une sprite CSS ?

Des outils facilitent leur création qui peut se révéler complexe. Par exemple :

- <https://draeton.github.io/stitches/>
- <http://spritegen.website-performance.org/>

Vous obtenez alors votre sprite CSS avec les styles à appliquer.

19 images peuvent être regroupées au sein d'une sprite CSS. Le nom de domaine todolistme devrait utiliser cette technique pour les ressources suivantes :

- http://todolistme.net/images/top_sync.png
- http://todolistme.net/images/sort_order.png
- http://todolistme.net/images/top_saved.png
- http://todolistme.net/images/top_sync_waiting.png
- http://todolistme.net/images/top_sync_error.png
- <http://todolistme.net/images/info.png>
- http://todolistme.net/images/arrow_down.png
- <http://todolistme.net/images/undo.png>
- <http://todolistme.net/images/purge.png>
- http://todolistme.net/images/category_up.png
- et 9 autres

92/100

#2418

✓ Regroupez 3 fichiers JavaScript

Chaque requête HTTP représente un coût en terme de performance (roundtrip time, utilisation de bande passante...).

Il est ainsi préférable de faire une requête vers un fichier de 50ko de données plutôt que 10 requêtes vers des fichiers de 5ko.

Quelle répartition adopter ?

Répartissez vos scripts en les intégrant directement dans votre code HTML ou en les regroupant dans des fichiers de taille plus conséquente. Nous vous conseillons l'emploi de cette dernière méthode pour profiter des mécanismes de cache.

Vous devriez reconsidérer les ressources suivantes :

- <http://todolistme.net/javascript/lists.js>
- <http://todolistme.net/javascript/lib.js>
- http://todolistme.net/javascript/javascript_e.js

Saviez-vous que ?

#2466

i 1 image utilise une seule couleur

Une requête vers une image à couleur unique augmente inutilement le volume de données qui transitent sur le réseau.

Il est inutile de récupérer une image composée uniquement d'une couleur de fond. Les styles CSS vous permettront de dessiner la même forme à moindre coût.

Par exemple, pour dessiner un rond, vous pouvez utiliser le code CSS suivant :

```
#myElement {
  background:#ff0000;
  border-radius:50%;
  width:160px;
  height:160px;
}
```

Exemple

Sur cette page 1 image utilise une seule couleur :

Les ressources suivantes sont hébergées par un parti tiers, il se peut donc qu'elles ne soient pas sous votre responsabilité. Vous devriez cependant considérer toute alternative possible à ces fichiers pour rester en accord avec la bonne pratique.

- [code.jquery.com/ui/1.10.3/themes/smo\[...\]ffffff_40x100.png](https://code.jquery.com/ui/1.10.3/themes/smo[...]ffffff_40x100.png) (208 octets)

#2543

i Répartition des ressources par domaine

Cette page charge ses ressources depuis 19 domaines différents. Nous avons extrait pour chaque domaine :

- Le temps de chargement (cumulé) : délai nécessaire au chargement de toutes les ressources du domaine
- Le temps de réponse serveur (cumulé) : délai des réponses du serveur (éventuelle connexion TCP + attente de la réception du premier octet)
- Le poids total des données qui ont été chargées
- Le nombre de requêtes

Ci-dessous les domaines utilisés lors du chargement de la page :

| Domaine | Temps total (ms) | Temps niveau serveur (ms) | Poids (ko) | Requêtes |
|-------------------------------|------------------|---------------------------|------------|----------|
| todolistme.net | 9043 | 6734 | 203 | 28 |
| code.jquery.com | 844 | 376 | 169 | 5 |
| pagead2.googlesyndication.com | 786 | 435 | 102 | 4 |
| platform.twitter.com | 769 | 527 | 51 | 4 |
| apis.google.com | 805 | 568 | 103 | 4 |
| www.facebook.com | 718 | 491 | 132 | 3 |
| fonts.gstatic.com | 268 | 218 | 24 | 2 |
| connect.facebook.net | 427 | 273 | 60 | 2 |
| syndication.twitter.com | 646 | 617 | 2 | 2 |
| googleads.g.doubleclick.net | 468 | 434 | 8 | 2 |
| www.google-analytics.com | 458 | 208 | 18 | 2 |
| fonts.googleapis.com | 163 | 137 | 1 | 1 |
| www.googletagmanager.com | 311 | 215 | 29 | 1 |
| plus.google.com | 266 | 244 | 1 | 1 |
| accounts.google.com | 82 | 78 | 1 | 1 |
| adservice.google.com | 271 | 222 | 1 | 1 |
| adservice.google.com.tw | 280 | 232 | 1 | 1 |
| www.google.com | 216 | 211 | 3 | 1 |
| staticxx.facebook.com | 194 | 166 | 12 | 1 |

Bravo ! Vous respectez les bonnes pratiques suivantes

100/100

#2339

✓ **Aucune redirection trop longue n'a été détectée**

Les redirections déclenchent des allers-retours évitables sur le réseau, et allongent le temps de chargement de la page.

Redirections HTTP

Une redirection HTTP permet d'indiquer que le contenu souhaité est accessible depuis une autre URL. Plusieurs requêtes sont alors envoyées successivement pour récupérer la ressource cible. Les redirections HTTP se distinguent grâce à leur code de retour, compris entre 300 et 399. [Vois les spécifications](#) des redirections HTTP.

Que faire ?

Permettez à l'utilisateur d'accéder directement à votre contenu, sans passer par les redirections, ou déterminez et améliorez ce qui provoque ces temps de chargement trop importants sur vos redirections. [Voir les recommandations](#) données par Google.

Cette page ne fait pas appel à des redirections avant d'accéder au bon contenu.

100/100

#2403

✓ **Vous n'utilisez pas trop de "prefetch" sur vos liens**

Aucune ressource n'est pré-chargée sur cette page.

Nous vous conseillons de limiter le nombre de prefetch sur une seule page afin de ne pas surcharger inutilement le réseau. Analysez les comportements de vos utilisateurs afin d'identifier les ressources ayant le plus de chances d'être réclamées à la suite d'une action.

Sachez par ailleurs que certains navigateurs limitent automatiquement le nombre de ressources pré-chargées. C'est par exemple le cas d'Internet Explorer 11 (10 ressources au maximum).

100/100

#2419

✓ **Vos fichiers CSS sont bien répartis**

Chaque requête HTTP représente un coût en terme de performance (roundtrip time, utilisation de bande passante...).

Il est ainsi préférable de faire une requête vers un fichier de 50ko de données plutôt que 10 requêtes vers des fichiers de 5ko.

Quelle répartition adopter ?

Répartissez vos styles en les intégrant directement dans votre code HTML ou en les regroupant dans des fichiers de taille plus conséquente. Nous vous conseillons l'emploi de cette dernière méthode pour profiter des mécanismes de cache.

Sur cette page, vous effectuez des requêtes vers des fichiers CSS à bon escient.

Vos principaux axes d'optimisation

0/100

#2531

! 2 dépendances critiques détectées

La défaillance d'un fournisseur de contenu externe peut rendre indisponible votre site.

Les Single Point Of Failure

Un Single Point Of Failure (SPOF) front-end se caractérise par une dépendance critique d'une page web à un contenu externe, susceptible d'empêcher totalement l'affichage de la page en cas de problème avec le fournisseur externe.

Par exemple, si votre page web appelle un script bloquant hébergé sur les serveurs de Google, votre page se trouve dépendante de toute défaillance provenant de l'appel à ce script. Voir [cet article sur les SPOF](#) pour en savoir plus.

Que faire pour éviter les SPOF ?

Éliminez ces dépendances autant que possible, même si elles concernent des fournisseurs renommés. Si vous devez faire appel à un contenu externe, assurez-vous que l'appel est effectué de manière asynchrone, et que vous disposez de fallbacks (alternatives en cas d'échec), afin de maîtriser le comportement de la page en cas de panne d'un service externe.

Nous nous assurons ici que la page analysée ne dépend pas de manière critique de ressources externes parmi les plus connues (googleapis, typekit, etc). Il s'agit en effet de cas de Single Point Of Failure front-end

Les ressources suivantes constituent un SPOF pour cette page :

- <http://fonts.googleapis.com/css?family=Abel|Architects+Daughter>
- <http://code.jquery.com/ui/1.10.3/themes/smoothness/jquery-ui.css>

0/100

#2353

! Différez l'utilisation du code JavaScript

Lorsque le navigateur web rencontre du code JavaScript en interprétant le code source d'une page web, cela peut ralentir considérablement l'affichage de la page, surtout s'il est nécessaire de télécharger un script externe.

Différez au maximum l'utilisation du JavaScript pour assurer un début rapide de l'affichage de la page.

Que faire ?

Avant toute chose, distinguez les portions de votre code JavaScript qui sont critiques au chargement de la page, et doivent être chargées le plus tôt possible. Placez-les dans un fichier externe JavaScript et chargez-le au plus tôt. Gardez ce fichier aussi petit que possible et différez les chargements ou l'exécution de tous les autres scripts JS ([en savoir plus](#)).

Utilisez si possible l'une des techniques suivantes pour des appels à des fichiers externes :

- utilisation de l'attribut [async](#)
- utilisation de l'attribut [defer](#)
- ajout du script dans le DOM en JavaScript, lors de l'évènement onload
- placer les scripts à la fin de votre code source (idéalement à la fin du <body>)

166.3 Ko du code JavaScript sont analysés lors du chargement initial de la page. Différez l'analyse de ce code pour éviter de bloquer l'affichage de la page.

- [platform.twitter.c\[...\].js](#) (82.9 Ko)
- [pagead2.googlesynd\[...\].js](#) (52.1 Ko)
- [staticxx.facebook.\[...\]n=44](#) (29.6 Ko de code JavaScript intégré)
- [connect.facebook.n\[...\].js](#) (1.2 Ko)
- <http://todolistme.net/> (551 o de code JavaScript intégré)

Bravo ! Vous respectez les bonnes pratiques suivantes

100/100

#2566

✓ **Votre réponse HTML n'est pas trop lourde**

Pourquoi réduire la quantité de code présente dans une page web ?

Avant de pouvoir être affichée, une page web doit, entre autres, être téléchargée, analysée et modélisée en un document compréhensible par le moteur de rendu. Si la quantité de code présenté dans la page est trop importante, ces étapes sont ralenties et le rendu, retardé.

Comment réduire la quantité de code ?

Votre réponse HTML ne devrait contenir que les ressources immédiatement nécessaires à l'affichage de la zone visible de la page. Déplacez les informations inline dans des fichiers externes (JS pour les scripts, CSS pour les styles, requêtes asynchrones pour des contenus additionnels) et simplifier la structure de votre page web.

100/100

#2355

✓ **Les en-têtes Keep-Alive sont activés**

Le keep-alive permet d'émettre et de recevoir plusieurs requêtes en utilisant la même connexion TCP (activé par défaut avec HTTP 1.1).

100/100

#2356

✓ **Vous spécifiez un jeu de caractères dans vos en-têtes HTTP de réponse**

Spécifier le jeu de caractères utilisé par la page dans l'en-tête HTTP `Content-Type` permet au navigateur de commencer son analyse tout de suite.

100/100

#2370

✓ **Vous évitez judicieusement les @import CSS**

L'utilisation de l'instruction CSS @import permet d'ajouter des feuilles de style. En réalité, les navigateurs internet ne peuvent pas les télécharger en parallèle, et cela peut ajouter un délai dans l'affichage de la page. Il faut plutôt utiliser la balise `link`. [En savoir plus.](#)

100/100

#2405

✓ **Aucune <meta> "http-equiv" inutile détectée**

Les en-têtes HTTP sont plus performants que les metas http-equiv.

Les balises <meta http-equiv=""> sont utiles ?

Les metas `http-equiv` permettent de communiquer au navigateur web des informations équivalentes à celles d'en-têtes HTTP. Par exemple, la meta `<meta http-equiv="content-type">` aura la même portée que l'en-tête HTTP `Content-Type`.

Deux points n'encouragent pas à l'utilisation des metas http-equiv :

- passer par la meta nécessite d'interpréter le début de la page HTML, ce qui en terme de performances est plus lent que de passer par les en-têtes HTTP
- si l'en-tête HTTP est déjà présent, la meta est ignorée

Dans quel cas les <meta http-equiv=""> sont utiles ?

Un seul cas peut justifier la présence de ces metas : si vous n'avez pas accès à la configuration de votre serveur, et donc des en-têtes HTTP.

Nous vous conseillons cependant d'utiliser un serveur configurable, pour établir un site le plus performant possible.

Cette page ne contient pas de balise meta `http-equiv`. C'est une bonne pratique.

100/100

#2416

✓ Le DOM de cette page ne contient pas trop d'éléments

Une structure complexe de la page implique plus d'octets à télécharger, et complexifie la recherche d'éléments précis.

Une page bien conçue peut présenter des contenus riches tout en conservant un nombre d'éléments raisonnable. [En savoir plus \(EN\)](#).

Nous vous conseillons de conserver une page avec moins de 1000 éléments dans le DOM.

Le nombre d'éléments du DOM est raisonnable (390 éléments).

100/100

#2445

✓ Vous interprétez et exécutez vos scripts une seule fois

Une librairie ou un script externe sont le plus souvent destinés à être appelé une seule fois par page. Cependant l'utilisation de widgets par exemple peut conduire à plusieurs exécutions inutiles si aucune précaution n'est apportée.

Duplication de scripts

Il est de plus en plus courant de voir des scripts utilisés plusieurs fois au sein d'une même page. Le cas le plus courant est l'intégration de widgets de réseaux sociaux, qu'il est parfois utile de retrouver à plusieurs endroits sur la page pour améliorer son taux d'engagement par exemple. Il ne s'agit pas d'une mauvaise pratique en soit, il faut cependant veiller à certains critères afin de ne pas voir le temps de chargement de la page s'allonger.

Que se passe-t-il si un script est inclus 2 fois dans le code ? Combien de fois est-il chargé ? Interprété ? Exécuté ?

La plupart des navigateurs web récents ne téléchargent qu'une seule fois un script inclus 2 fois. Une exception subsiste : Firefox, qui chargera le fichier autant de fois qu'indiqué si aucune politique de cache efficace n'est configurée.

Outre cette exception, les problèmes de performances vont se ressentir au moment de l'interprétation et l'exécution des scripts. En effet, si le script est mentionné 3 fois dans le code, il sera exécuté 3 fois, et ce sur tous les navigateurs.

Pour plus de détails, n'hésitez pas à [consulter cet article sur le sujet \(EN\)](#).

La solution

Il reste tout de même une solution pour pouvoir utiliser un script plusieurs fois sans l'interpréter et l'exécuter plus d'une fois. Vous devez écrire du code JavaScript, qui vérifie si le script est présent. Si oui, il l'utilise, sinon, il l'injecte afin de l'utiliser.

Prenons l'exemple du widget Facebook, toujours tiré de l'article ci-dessus. À chaque fois que vous voulez intégrer cette fonctionnalité dans votre page, il est nécessaire d'inclure le code suivant :

```
(function(d, s, id){  
  var js, fjs = d.getElementsByTagName(s)[0];  
  if (d.getElementById(id)) {return;}  
  js = d.createElement(s); js.id = id;  
  js.src = "//connect.facebook.net/en_US/sdk.js";  
  fjs.parentNode.insertBefore(js, fjs);  
}(document, 'script', 'facebook-jssdk'));
```

Exemple

Notez en gras la ligne de vérification de la présence du script dans le document. Le script est alors inclus, interprété et exécuté uniquement lors du premier appel dans la page. Les autres appels tomberont dans le cas de la ligne mise en valeur, et ne feront donc qu'utiliser le script déjà inclus et exécuté.

100/100

#2539

✓ Cette page n'utilise pas de redirection côté client

Aucune redirection n'a été effectuée côté client (window.location, meta refresh...) pour l'affichage de cette page. Les redirections côté client sont à éviter absolument.

100/100

#2550

✓ Vos scripts semblent être injectés efficacement

L'injection de scripts via l'instruction JavaScript `document.write` retarde l'affichage d'une partie au moins de votre page et/ou l'interactivité de votre page pour l'utilisateur.

Besoin d'injecter un script ?

`document.write` est parfois utilisé à tort pour injecter un script. Exemple :

```
document.write('<script src="' + src + '" type="text/javascript"></script>');
```

Exemple

Comme indiqué sur [cet article \(EN\)](#), vous devriez plutôt envisager l'utilisation du pattern "createElement-insertBefore" :

```
var sNew = document.createElement("script");
sNew.async = true;
sNew.src = "http://ajax.googleapis.com/ajax/libs/jquery/1.5.1/jquery.min.js";
var s0 = document.getElementsByTagName('script')[0];
s0.parentNode.insertBefore(sNew, s0);
```

Exemple

Sur cette page, aucun `document.write` n'a été détecté.

Vos principaux axes d'optimisation

0/100

#2430

! 24 de vos requêtes ne définissent pas de politique de cache

Les en-têtes **Expires** sont indispensables pour une politique de cache efficace, et vont grandement impacter le temps de chargement de vos pages lors des prochaines visites.

L'en-tête expires

Vous pouvez définir une date d'expiration pour chaque ressource : tant que la date n'est pas dépassée, l'utilisateur stocke et utilise la ressource en cache.

La date d'expiration des ressources se paramètre à l'aide de l'en-tête HTTP **Expires** :

```
Expires: Thu, 25 Dec 2014 20:00:00 GMT
```

Exemple

Vous pouvez indiquer une date d'expiration lointaine pour les ressources statiques (1 an maximum), et plus proche pour des ressources susceptibles d'être modifiées (48h minimum).

Lorsque vous mettez en production **une nouvelle version de votre site, pensez bien à renommer les ressources statiques** ayant été modifiées (versioning). En effet, si vous ne changez pas leur nom, vos utilisateurs garderont les ressources stockées en cache, correspondant à l'ancienne version. Ils risquent alors de se retrouver dans une version instable de votre page. Par exemple :

```
maresource.min.20140101.js
```

Exemple

Voir [les recommandations de Yahoo!](#) sur le sujet.

24 de vos ressources **n'ont pas de date d'expiration** :

- http://todolistme.net/css/style_g.css
- <http://todolistme.net/javascript/lists.js>
- <http://todolistme.net/javascript/lib.js>
- http://todolistme.net/javascript/javascript_e.js
- http://todolistme.net/images/top_new_window.png
- http://todolistme.net/images/top_sync.png
- <http://todolistme.net/images/info.png>
- et 17 autres

45/100

#2352

! Spécifiez un en-tête "Vary: Accept-Encoding"

Les ressources compressibles suivantes, qui peuvent être mises en cache, doivent être associées à un en-tête "Vary: Accept-Encoding" :

- [todolistme.net/css\[...\].g.css](http://todolistme.net/css[...].g.css)
- [todolistme.net/jav\[...\].e.js](http://todolistme.net/jav[...].e.js)
- [todolistme.net/jav\[...\].ib.js](http://todolistme.net/jav[...].ib.js)
- [todolistme.net/jav\[...\].ts.js](http://todolistme.net/jav[...].ts.js)

L'en-tête HTTP **Vary: Accept-Encoding** permet de mettre en cache deux versions d'une ressource sur les proxys parcourus: l'une est compressée, l'autre non. Ainsi, les clients ne pouvant pas décompresser les ressources peuvent utiliser la version décompressée, et les autres la version compressée.

✓ 6 requêtes provenant de partis tiers n'adoptent pas une politique de cache assez longue

En permettant la conservation de vos ressources statiques plusieurs jours en cache, vous diminuez la charge de votre serveur.

Certaines de vos ressources utilisent l'en-tête HTTP **Expires** pour disposer d'une politique de cache efficace, ce qui correspond à une bonne pratique. Cependant, vous devriez envisager d'améliorer sa configuration pour profiter au mieux des mécanismes de cache.

Quelle configuration adopter ?

Nous vous conseillons de paramétrer l'en-tête **Expires** afin que la date mentionnée soit comprise entre 2 jours et 1 an par rapport à aujourd'hui.

Cette page contient 6 ressources **n'ayant pas une date d'expiration assez lointaine** :

Les ressources suivantes sont hébergées par un parti tiers, il se peut donc qu'elles ne soient pas sous votre responsabilité. Vous devriez cependant considérer toute alternative possible à ces fichiers pour rester en accord avec la bonne pratique.

- http://connect.facebook.net/en_US/all.js
- http://pagead2.googlesyndication.com/pagead/show_ads.js
- <http://platform.twitter.com/widgets.js>
- <http://www.google-analytics.com/ga.js>
- <https://apis.google.com/js/plusone.js>
- [www.googletagmanager.com/activeview/js\[...\].js?cb=%2Fr20100101](http://www.googletagmanager.com/activeview/js[...].js?cb=%2Fr20100101)

Bravo ! Vous respectez les bonnes pratiques suivantes

✓ Cette page ne contient pas de scripts inline trop longs

Vous devriez permettre la mise en cache de tout script ayant une taille non négligeable, afin de réduire le temps de chargement de vos pages.

Scripts inline / mise en cache

Les scripts dits "inline" permettent d'intégrer facilement de petites portions de scripts directement dans le code HTML. Exemple :

```
<script type="text/javascript">
  (function(i,s,o,g,r,a,m){if('GoogleAnalyticsObject']...,'analytics.js','ga');
  ga('create', 'UA-11111111-1', 'monsite.com');
</script>
```

Exemple

En procédant ainsi, vous évitez d'effectuer une requête vers le serveur pour récupérer la ressource. Les scripts inline représentent alors un gain de performance pour intégrer de petits scripts.

Dès lors qu'un script dispose d'une taille assez conséquente, nous vous conseillons de l'externaliser et d'effectuer une requête pour le récupérer. Cela vous permettra de bénéficier des mécanismes de cache sur la ressource.

Que faire ?

Externalisez vos scripts de plus de 1 500 caractères dans un ou des fichiers distincts.

Vous n'avez pas de script inline trop long sur cette page.

Vos principaux axes d'optimisation

0/100

#2491

! Plusieurs éléments utilisent le même identifiant

Utiliser un même identifiant pour 2 éléments différents peut entraîner des effets de bord, notamment lors de l'exécution de JavaScript ou lors de l'application des règles CSS.

Identifiant d'éléments

Chaque élément d'une page web peut disposer d'un identifiant, défini par l'attribut `id` :

```
<p>
  <span id="mySpan1"></span>
</p>
```

Exemple

Ces identifiants vous permettent de manipuler les éléments de votre page avec des instructions CSS ou JavaScript.

Que faire ?

Vous devez faire en sorte qu'aucun élément de la page ne dispose du même identifiant. Si vous souhaitez partager des propriétés/comportements entre plusieurs éléments, il faudra utiliser l'attribut `class`, dédié à cet effet :

```
<p>
  <span class="mySpans"></span><span class="mySpans"></span>
</p>
```

Exemple

L'identifiant suivant est utilisé plusieurs fois dans la page :

- `later_number`, utilisé 2 fois

0/100

#2517

! Le mot clé !important est utilisé 17 fois

Le mot clé !important s'approche d'un hack permettant d'interdire toute surcharge d'une propriété définie. Si parfois il peut s'avérer utile, cela doit rester exceptionnel !

Si vous en abusez, songez à revoir votre structure de code CSS. Dans ce conseil, vous êtes pénalisé au delà de 10 utilisations de ce mot clef.

Voici la liste des !important détectés :

http://todolistme.net/css/style_g.css

- `.dropontotitle {color: #f7ba79 !important}` (ligne 6, col 5418)
- `.currentlist {border: 1px solid #279c4e !important}` (ligne 6, col 12473)
- `.currentlist {padding-left: 45px !important}` (ligne 6, col 12587)
- `.currentlist {color: #444 !important}` (ligne 6, col 12628)
- `.currentlist:hover .mergelist {visibility: hidden !important...` (ligne 6, col 13072)
- `.dropontotomorrow {background-color: #f7ba79 !important}` (ligne 6, col 14869)
- `#backbutton {width: 150px !important}` (ligne 6, col 15261)
- `body.remote .currentlist {padding: 0 !important}` (ligne 6, col 18333)
- et 9 autres

95/100

#2516

✓ 1 sélecteur CSS est trop complexe

Des sélecteurs CSS trop complexes nuisent à la lisibilité générale du code et constituent un gain possible en terme de performance.

Les règles CSS

Les règles CSS permettent de sélectionner un ensemble d'éléments du DOM, et de leur appliquer des propriétés de mise en forme.

Elles sont composées d'un ensemble de sélecteurs (identifiants, classes, noms d'éléments, etc) permettant d'adresser les éléments à cibler. Une page bien structurée permet l'écriture de règles CSS simples, rapidement lisibles et performantes.

Comment améliorer ce point ?

Simplifiez vos sélecteurs tant que possible, quitte à introduire de nouvelles classes dans votre code pour éviter des enchaînements trop longs.

Par exemple, la règle suivante :

```
body td .maClasse div .maClasse2 span .maClasse3 { }
```

Pourrait être améliorée, en passant éventuellement par un identifiant :

```
#monIdentifiant .maClasse3 { }
```

Les fichiers suivants définissent des sélecteurs CSS trop complexes :

http://todolistme.net/css/style_g.css

- `// This code is copyright Ryan Chadwick 2012 // Please do not use this code your...`

Les autres conseils

92/100

#2383

✓ Séparez vos styles de votre code HTML

Dissocier les balises HTML de leur style améliore la maintenabilité du code et favorise sa factorisation.

Définir les styles CSS

Les styles CSS permettent de mettre en forme la page. Pour les définir, vous disposez de 3 méthodes principales :

- la déclaration dans un fichier CSS
- la déclaration dans un bloc "inline" (balises <style>)
- la déclaration dans un attribut style d'un élément HTML

Que faire ?

Nous vous conseillons de regrouper vos styles CSS au sein de balises `<style>` ou dans des fichiers CSS dédiés. Ainsi, la partie HTML se charge uniquement de fournir la structure de la page, et sa mise en forme est externalisée. L'attribut `style` devrait uniquement être généré en cas de besoin par du code JavaScript (exemple : nécessité de connaître la taille de l'écran).

Cette page utilise 1 attribut(s) `style` :

- ``

Saviez-vous que ?

#71

i Attention aux commentaires au sein du code source

Ne transmettez à vos internautes que les données dont ils ont besoin.

Pages web et commentaires dans le code source

Les commentaires dans le code source sont indispensables aux développeurs : ils permettent de détailler une portion de code complexe, et aident à naviguer plus efficacement dans le DOM. Cependant, il n'y a aucun intérêt à communiquer ces informations à vos utilisateurs. Les seuls commentaires dont la présence est justifiée sur le navigateur de l'internaute sont ceux permettant la compatibilité de votre site sur toutes les plateformes (exemple : rendre le site compatible avec Internet Explorer 8).

Veillez donc à ne pas y exposer des données sensibles pouvant permettre à une personne malveillante de corrompre votre système. Par ailleurs, rien ne justifie que des blocs de code soient transmis à vos visiteurs.

#2512

i 1 propriété CSS est surchargée

Il est souvent inutile de définir 2 fois la même propriété au sein d'une même règle : la seconde surchargera la première. Cela risque de nuire à la lisibilité du code CSS. Supprimer les propriétés surchargées inutilement réduira également le poids du fichier.

Comment améliorer ce point ?

Supprimez l'une des occurrences de la propriété en cause. Par exemple, si votre fichier dispose des propriétés suivantes :

```
.maClasse {  
  margin: 20px;  
  ...  
  margin: 10px;  
}
```

Vous devriez envisager leur réécriture de la manière suivante :

```
.maClasse {  
  margin: 10px;  
}
```

Les fichiers suivants définissent plusieurs fois la même propriété dans une règle. La [gestion de fallbacks \(EN\)](#) peut justifier une duplication.

Assurez-vous que les propriétés remontées ici sont maîtrisées :

http://todolistme.net/css/style_g.css

- `#head {background: linear(...)t(to bottom,#80a6c8 0,#5491c8 100%)}` (ligne 6, col 1245)

#2520

i 45% des couleurs définies ne sont utilisées qu'une seule fois

Vous avez défini 53 couleurs, et 24 d'entre elles ne sont utilisées qu'une seule fois (certaines peuvent être très proches, exemple : #FF0000 and #FE0000).

Nous vous conseillons d'harmoniser au maximum les couleurs de votre site afin de simplifier votre CSS mais aussi d'éviter des incohérences dans votre charte graphique.

Les couleurs suivantes ne sont utilisées qu'une seule fois :

| | | | | | | | | |
|-----------|-----------|-----------|-------------------------|-------------------------|-------------------------|-----------|-----------|-----------|
| ■ #000000 | ■ #279c4e | ■ #296cae | ■ #2f76b9 | ■ #323232 | ■ #444444 | ■ #8bc7f6 | ■ #8ef4b1 | ■ #9fec99 |
| ■ #a6cbcd | ■ #b1b1b1 | ■ #bbbbbb | ■ #cc4433 | ■ #cfcfcf | ■ #dfdfdf | ■ #e4e4e4 | ■ #efefef | ■ #f3f6c5 |
| ■ #f5f5f5 | ■ #f7f7f7 | ■ #fcfcfc | ■ rgba(100,100,100,0.6) | ■ rgba(176,221,245,0.4) | ■ rgba(200,200,200,0.3) | | | |

Bravo ! Vous respectez les bonnes pratiques suivantes

Rapport généré
par **dareboost**

Rapport d'analyse performance web

10 mai 2019 - 23 / 48

100/100

#2379

✓ Une favicon est définie

Vous avez défini une favicon sur cette page. C'est une bonne pratique.

Une Favicon est une petite image fournissant une icône à un site web. Elle est située à la racine du serveur et le navigateur la demandera toujours. Il est préférable de ne pas répondre par un code HTTP 404 (ressource non trouvée).

De plus ce fichier sera demandé pour chaque nouvelle page: il faut le mettre en cache pour qu'il ne soit demandé qu'une seule fois. [En savoir plus](#).

100/100

#2398

✓ Vos images PNG ne sont pas compressées (gzip/deflate)

La compression a un coût, du côté du serveur et du navigateur client. Il ne faut donc l'activer que si elle est efficace.

PNG et compression

Les images au format png ne supportent pas la compression gzip. Il est donc inutile d'effectuer des opérations de compression au niveau de votre serveur, et de décompression au niveau client pour ce type de ressources.

Cette page n'essaie pas de compresser les images. C'est une bonne pratique.

100/100

#2448

✓ Les extensions de vos ressources sont bonnes

L'extension d'une ressource permet de repérer facilement le type de contenu qui y est associé. Vous n'avez aucun intérêt à indiquer une extension différente du type réel d'un fichier.

Ainsi, une ressource nommée `ressource.js` devra être associée à un en-tête HTTP `Content-Type` ayant pour valeur `application/javascript`. Vous pouvez vérifier la valeur de l'en-tête `Content-Type` à l'aide de l'outil développeur de votre navigateur préféré, par exemple DevTools sur Chrome.

100/100

#2449

✓ Toutes les ressources de la page définissent leur type

Chaque ressource peut définir le type de son contenu afin de faciliter leur interprétation par les navigateurs web.

Vous n'avez aucun intérêt à dissimuler le type d'une ressource.

Nous vous conseillons de définir un en-tête HTTP `Content-Type` sur toutes les ressources de la page.

100/100

#2515

✓ Aucune propriété CSS n'est surchargée par des shorthand

Le CSS comporte [quelques pièges](#) (EN) pouvant engendrer des effets de bords indésirables sur vos styles.

La surcharge de propriétés en CSS

Pour une même règle CSS, si 2 propriétés identiques sont spécifiées, la dernière surchargera toutes les autres. La possibilité de définir des propriétés "raccourcies" (shorthand) peut parfois prêter à confusion dans ce contexte. Par exemple, "border" se réfère à toutes les bordures d'un élément, et revient à définir "border-top", "border-right", "border-left" et "border-bottom". Ainsi, avec le code suivant :

```
.maClasse {  
  border-color:red;  
  border:5px solid; // border reprend la couleur par défaut  
}
```

La propriété "border-color" se retrouve surchargée par "border", la couleur reprend donc la valeur par défaut.

Évitez les effets de bord

Avoir de telles surcharges en place présente un risque, puisque la surcharge n'est pas explicite et peut donc être un effet de bord. Si le style obtenu est celui désiré, alors la propriété qui est ainsi surchargée est donc inutile, supprimez-la (cela réduira également le poids du CSS).

100/100

#2527

✓ Vos propriétés CSS ne sont pas dupliquées

Les propriétés dupliquées au sein d'une même règle CSS nuisent à la lisibilité générale du code CSS. C'est également une source d'optimisation, puisque supprimer les éléments dupliqués réduira le poids du fichier.

Les propriétés CSS

Les propriétés CSS permettent d'appliquer une mise en forme pour un ensemble d'éléments donnés. Il est inutile de définir 2 fois une même propriété sur une règle : la seconde surchargera automatiquement la première.

Comment améliorer ce point ?

Supprimez l'une des occurrences de la propriété en cause. Par exemple, si votre fichier dispose des propriétés suivantes :

```
.maClasse {  
  margin: 10px;  
  ...  
  margin: 10px;  
}
```

Vous devriez envisager leur réécriture de la manière suivante :

```
.maClasse {  
  margin: 10px;  
}
```

✓ Tous vos sélecteurs CSS sont uniques

Utiliser plusieurs fois le même sélecteur sur des blocs déclaratifs différents au sein d'un même fichier CSS peut nuire à la lisibilité et à la maintenabilité du code. C'est également une source d'optimisation, puisque regrouper ces éléments au sein d'une même règle réduit le poids du fichier et réduit le temps nécessaire avant le rendu.

Le modèle objet CSS (CSSOM)

Les règles CSS permettent d'appliquer des propriétés de mise en forme sur des éléments ciblés de votre code HTML.

Pour ce faire, le navigateur construit sa propre modélisation avec le code CSS qu'il a récupéré : le modèle objet CSS (CSSOM). Cette transformation se fait en plusieurs étapes: lecture du fichier, conversion des chaînes de texte en jetons, transformation de ces jetons en objets avec propriétés et règles et organisation des éléments dans un modèle arborescent. Ces opérations se dérouleront d'autant plus vite que le code CSS aura été rédigé de manière simple et non-redondante.

Comment améliorer mes règles CSS ?

Pour des raisons de maintenance autant que de performance, regroupez au sein d'une même règle les sélecteurs CSS dupliqués.

Par exemple, si votre fichier dispose des règles suivantes :

```
.maClasse {  
  margin: 0;  
}  
...  
.maClasse {  
  border: 1px solid black;  
}
```

Vous devriez envisager leur réécriture de cette manière :

```
.maClasse {  
  margin: 0;  
  border: 1px solid black;  
}
```

Si la duplication des sélecteurs est héritée d'un processus historique ou qu'elle est justifiée par un besoin de maintenance, vous pouvez envisager l'utilisation d'un outil de minification CSS capable de faire cette transformation automatiquement.

Vos principaux axes d'optimisation

0/100

#69

! Ajoutez l'attribut alt sur vos balises

L'attribut `alt` est un critère important en terme de référencement (SEO). En effet, les robots d'indexation des moteurs de recherche ne peuvent pas analyser de contenu graphique. Il se servent alors de leur texte alternatif pour répondre aux requêtes des internautes. [C'est par exemple le cas pour Google images](#).

```

```

Exemple

De plus, l'attribut `alt` s'avère utile dans d'autres cas:

- utilisation d'un lecteur d'écran (personnes malvoyantes)
- connexion trop lentes pour charger l'image
- affichage d'un contenu dans le cas d'une erreur dans l'attribut `src`

Vous avez 12 balises `img`, mais 12 d'entre elles ne définissent pas d'attribut `alt` :

- ``
- ``
- ``
- ``
- ``
- ``
- ``
- ``
- ``
- ``
- ``
- ``

Il reste possible de définir un texte alternatif vide si aucune description ne semble cohérente pour l'image, mais soyez vigilant. Il est préférable de veiller à ce que la majorité de vos images indiquent un texte cohérent. Voir [les recommandations du W3C \(EN\)](#).

0/100

#2503

! Le fichier robots.txt doit être défini

Facilitez au maximum la découverte de votre site par les robots en leur indiquant quelles URLs ne doivent pas être explorées.

Fichier robots.txt

Ce fichier est placé à la racine du site et est interprété par les robots en charge du référencement de votre site. Il délivre des instructions pour indiquer les pages à explorer par les robots.

Notez que ces instructions sont données à titre indicatives. Un robot quelconque ne sera pas bloqué par les restrictions du fichier.

Nous n'avons pas détecté de fichier robots.txt sur ce site, vous devriez le mettre en place à l'adresse suivante :

- <http://todolistme.net/robots.txt>

0/100

#2399

! Améliorez la lecture de votre site par les réseaux sociaux

Transmettre aux réseaux sociaux des informations liées à la page vous permet de renforcer votre référencement.

Les propriétés Open Graph

Plusieurs propriétés permettent aux réseaux sociaux d'en savoir plus sur le contenu de la page. Nous vous recommandons d'utiliser au moins les quatre propriétés suivantes :

- `<meta property="og:title" content="The title" />` Exemple
- `<meta property="og:type" content="The type" />` Exemple
- `<meta property="og:url" content="http://url.com/" />` Exemple
- `<meta property="og:image" content="http://image.jpg" />` Exemple

Ces informations permettent d'améliorer le lien entre votre page et les différents réseaux sociaux, notamment Facebook, ce qui constitue un plus en terme de référencement. [En savoir plus sur les propriétés Open Graph.](#)

Aucune information à destination des réseaux sociaux n'a été détectée.

Saviez-vous que ?

#2457

i Cette page comporte 11 liens

Il existe 2 grands types de liens :

- **les liens internes**, qui font référence aux pages du même nom de domaine
- **les liens externes**, qui pointent vers d'autres sites (doivent être pertinents et pointer vers des contenus de qualité)

Si vous faites référence à de nombreux liens, vous pouvez demander aux algorithmes de référencement de ne s'intéresser qu'à certains d'entre eux, en ajoutant un attribut `rel=nofollow` sur ceux qui ne sont pas pertinents (ex: publicités).

Voici la répartition des 11 liens contenus dans la page :

- 11 liens internes (100,00%)
- Aucun lien externe "follow" (0,00%)
- Aucun lien externe "nofollow" (0,00%)

Bravo ! Vous respectez les bonnes pratiques suivantes

100/100

#78

✓ Cette page définit des balises <h1> et <h2>

Il est recommandé de placer vos mots clés dans les balises de titres (h1 et h2 au minimum). Les moteurs de recherche utilisent les balises h1, h2 et h3 pour le référencement (SEO). Cette page contient:

- 1 balise(s) <h1>
- 2 balise(s) <h2>
- 5 balise(s) <h3>

100/100

#81

✓ Cette page utilise uniquement des formats d'images standards

Les images au format non standard risquent de ne pas être indexées par les moteurs de recherche.

Seuls certains formats d'images sont considérés comme standards sur le web : jpeg, jpg, png, gif, svg, ico, webp. Vous devriez considérer une alternative à tout autre format.

Par ailleurs, pensez à soigner le texte entourant vos images: certains moteurs de recherche analysent les 10 mots (environ) précédents et suivants l'image, pour restituer son contexte.

100/100

#83

✓ Un titre est défini sur cette page

La page doit disposer d'un titre unique.

Utilisation du titre par les moteurs de recherche

Lorsqu'il est bien configuré, le titre de la page se retrouve directement dans les résultats de recherche :

Amazon.fr : livres, DVD, jeux vidéo, MP3, high-tech, informatique ...
<https://www.amazon.fr/> ▼
Achat et vente en ligne parmi des millions de produits en stock. Livraison gratuite à partir de 25€. Vos articles à petits prix : culture, high-tech, mode, jouets, sport ...

L'utilisation d'un titre adapté est un critère important pour le référencement. De plus, il vous permet de maîtriser au maximum l'affichage sur les pages de résultats, et d'intégrer les mots clés sur lesquels vous souhaitez que votre site ressorte.

Comment définir le titre d'une page ?

Le titre de la page est renseigné grâce à la balise `<title>`, qui doit être placée dans la balise `<head>`, au début du code.

Cette page définit bien un titre via la balise `title`.

Voici le titre défini :

(7/0) Today's Tasks

100/100

#84

✓ Une description est définie sur cette page

La page doit disposer d'une description unique.

Utilisation de la description par les moteurs de recherche

Lorsqu'elle est bien configurée, la description de la page se retrouve directement dans les résultats de recherche:

Amazon.fr : livres, DVD, jeux vidéo, MP3, high-tech, informatique ...

<https://www.amazon.fr/>

Achat et vente en ligne parmi des millions de produits en stock. Livraison gratuite à partir de 25€. Vos articles à petits prix : culture, high-tech, mode, jouets, sport ...

Cette description vous permet de maîtriser au maximum l'affichage sur les résultats des moteurs de recherche, et d'améliorer le taux de clics vers votre page. [En savoir plus.](#)

Comment définir la description d'une page ?

La description de la page est renseignée grâce à la balise `<meta name="description" content="description de la page">`, qui doit être placée dans la balise `<head>`, au début du code.

Cette page définit une `<meta>` `description` :

On-line todo list manager with a minimum of fuss and a sprinkling of funky features. No account required.

100/100

#90

✓ Aucun paramètre de requête n'est détecté dans l'URL

Cette URL ne contient aucun paramètre.

Une URL doit être la plus lisible possible. Par exemple, `http://example.com?userId=332&group=MonGroupe` sera moins bien lisible que `http://example.com/mongroupe/moi`.

100/100

#2345

✓ Les mots de cette URL sont bien délimités

Les underscores dans l'URL freinent inutilement la remontée d'une page dans les résultats de Google.

URL et référencement

Les mots contenus dans l'URL de la page font partie des nombreux facteurs impactant son référencement. Ainsi, lors d'une recherche sur les mots `performance web`, les URLs contenant les mots `performance` et `web` se donnent plus de chances de remonter dans les résultats.

Or pour Google, les underscores ne sont pas des séparateurs de mots : si votre URL contient `performance_web`, elle n'aidera pas à ressortir la page sur la requête `performance web` (`performance_web` est considéré comme étant un seul mot).

Ce comportement n'est cependant pas propre à tous les moteurs de recherche. Par exemple, Bing ne différencie pas tirets et underscores : l'URL aidera à la remontée de la page sur vos mots clés dans les deux cas.

Notez que cette bonne pratique souligne un frein à la remontée de votre page, mais qu'en aucun cas Google ne vous pénalisera pour avoir utilisé un underscore dans l'URL. [En savoir plus.](#)

Comment résoudre le problème ?

Nous vous conseillons d'utiliser des `-` à la place des `_` sur vos nouvelles pages.

Le chantier à mener est plus complexe sur les pages déjà en place comme celle analysée ici : si vous changez l'URL, vous perdrez vos efforts de référencement effectués jusqu'à présent. Il faut alors mettre en place une redirection permanente (code HTTP 301) conservant l'ancienne URL, et redirigeant vers la nouvelle. Mais attention : un grand nombre de redirections sur votre site peut nuire à la visibilité de vos pages. N'utilisez donc des redirections que si le nombre des pages concernées reste restreint.

Cette URL est en accord avec la bonne pratique.

100/100

#2432

✓ La description de la page a une taille cohérente

La page doit disposer d'une description comportant entre 20 et 300 caractères.

Utilisation de la description par les moteurs de recherche

Lorsqu'elle est bien configurée, la description de la page se retrouve directement dans les résultats de recherche:

Amazon.fr : livres, DVD, jeux vidéo, MP3, high-tech, informatique ...
<https://www.amazon.fr/> ▼
Achat et vente en ligne parmi des millions de produits en stock. Livraison gratuite à partir de 25€. Vos articles à petits prix : culture, high-tech, mode, jouets, sport ...

Mais les moteurs de recherche ne se servent pas nécessairement de la description que vous leur fournissez. Si celle-ci ne semble pas pertinente, ils se baseront sur d'autres critères pour présenter votre page, et vous perdrez la main sur le contenu affiché. Notez cependant que vous ne serez pas pénalisé directement par Google par exemple si votre texte est trop grand ou trop court. [En savoir plus](#).

Comment définir la description d'une page ?

La description de la page est renseignée grâce à la balise `<meta name="description" content="description de la page">`, qui doit être placée dans la balise `<head>`, au début du code.

Cette page définit une `<meta>` `description` comportant entre 20 et 300 caractères:

On-line todo list manager with a minimum of fuss and a sprinkling of funky features. No account required.

100/100

#2444

✓ Le titre de la page a une taille adéquate

Indiquez un titre assez court pour maximiser les chances qu'il soit utilisé sur les SERPs (Search Engine Results Page).

Utilisation du titre par les moteurs de recherche

Lorsqu'il est bien configuré, le titre de la page se retrouve directement dans les résultats de recherche :

Amazon.fr : livres, DVD, jeux vidéo, MP3, high-tech, informatique ...
<https://www.amazon.fr/> ▼
Achat et vente en ligne parmi des millions de produits en stock. Livraison gratuite à partir de 25€. Vos articles à petits prix : culture, high-tech, mode, jouets, sport ...

L'utilisation d'un titre adapté est un critère important pour le référencement. De plus, il vous permet de maîtriser au maximum l'affichage sur les pages de résultats, et d'intégrer les mots clés sur lesquels vous souhaitez que votre site ressorte.

Définir un titre assez court

Le titre de la page est renseigné grâce à la balise `<title>`, qui doit être placée dans la balise `<head>`, au début du code.

Celui-ci doit rester assez court afin de pouvoir être affiché au sein d'une page de résultats. Nous vous conseillons d'**établir un titre en moins de 75 caractères** ([voir plus d'informations \(EN\)](#)).

Le titre de cette page comporte 19 caractères:

(7/0) Today's Tasks

100/100

#2467

✓ Vos titres ne sont pas vides

Les balises `<h1>`, `<h2>` et `<h3>` envoient des signaux aux robots destinés au référencement de votre page, et doivent donc contenir des mots-clés.

Les titres de cette page renseignent bien un contenu.

Vos principaux axes d'optimisation

0/100

#2549

❗ Utilisez HTTPS pour collecter des données sensibles

La page contient des champs pour récolter des mots de passe ou des informations bancaires, vous devez garantir à l'utilisateur une connexion sécurisée.

Google multiplie les actions en faveur du HTTPS. Après avoir ajouté ce critère dans ses algorithmes de référencement ([voir l'annonce](#) - en anglais), Google a fait progressivement évoluer Chrome pour [mettre en valeur l'absence d'un contexte sécurisé](#) dans différents cas de collecte d'informations provenant des utilisateurs. Les autres navigateurs suivent également cette tendance.

De plus, cette page récolte des données sensibles (mot de passes ou informations bancaires). Sur [Google Chrome](#), [vos internautes seront avertis \(EN\)](#) que le site est "Non Sécurisé" :



La mise en place du HTTPS sur un site provoque parfois quelques réticences (coût, impact sur les performances...). Mais le marché a évolué et nombre de ces craintes sont désormais injustifiées. Le passage à HTTPS doit être envisagé..

Comment mettre en place le protocole HTTPS ?

Vous devez mettre en place un certificat obtenu auprès d'une autorité de certification fiable. Renseignez vous notamment auprès de votre hébergeur qui peut vous aider à obtenir ce certificat. [La page suivante devrait par ailleurs vous aider dans votre démarche de migration vers le protocole HTTPS.](#)

Un certificat gratuit ? Let's Encrypt !

[Let's Encrypt](#) est une autorité de certification gratuite, automatisée et open source. De nombreux hébergeurs proposent d'activer la génération et le renouvellement automatique de certificats gratuits directement depuis l'interface d'administration de votre domaine. Contactez votre hébergeur pour plus d'informations.

! Il manque une politique de sécurité sur la provenance de vos ressources

Protégez votre site Web contre les attaques de type XSS (Cross-Site Scripting) en mettant en place une politique restrictive de sécurité du contenu.

Les attaques XSS

Les attaques XSS sont un type d'attaque dans laquelle des données malveillantes sont malicieusement ajoutées aux sites Web. Le nombre de vulnérabilités permettant ces attaques est assez important, c'est pourquoi il est aussi utile de les prévenir que de limiter leurs effets néfastes.

Vous pouvez protéger vos pages contre ces attaques en limitant l'exécution à des portions de code identifiées par le domaine auquel elles appartiennent, ou légitimées par un identifiant d'intégrité. Le code ne correspondant pas à cette politique de sécurité ne sera pas exécuté et l'utilisateur sera informé.

Configurer un en-tête HTTP "Content-Security-Policy" (CSP)

Pour prévenir ou limiter les dégâts potentiels d'une attaque XSS, vous devez configurer votre serveur afin que la réponse de la ressource principale contienne l'en-tête HTTP "Content-Security-Policy".

Voici un exemple d'utilisation de cet en-tête :

```
Content-Security-Policy: script-src 'self' https://apis.google.com
```

Exemple

Dans ce cas, seuls les scripts provenant de l'hôte courant ou de <https://apis.google.com> seront exécutés.

[Le lien suivant](#) vous aidera à en savoir plus sur cet en-tête HTTP. Vous serez alors amené à utiliser les directives exposées [sur ce lien](#).

Soyez vigilant, si l'en-tête est mal configuré, certains de vos contenus, scripts, ou encore styles pourront être bloqués, ce qui pourrait engendrer des effets de bords non souhaités. De plus, les restrictions s'appliquent à toutes les pages du site. Nous vous conseillons de tester les différentes pages de votre site avant de déployer l'instruction dans votre environnement de production.

Aucune CSP n'a été détectée sur cette page : elle est plus facilement exposée à des attaques de type XSS.

0/100

#2480

! Cette page est exposée à des attaques du type "clickjacking"

Ne permettez pas à des personnes malveillantes d'intégrer vos pages sur leur site.

Clickjacking

Ce type d'attaque consiste à intégrer votre page sur un site malveillant via des balises <frame> ou <iframe>. Ainsi il est possible de faire croire à un internaute qu'il est sur votre propre page. L'internaute peu averti sera en confiance et sera potentiellement amené à saisir des informations que le site malveillant sera à même d'intercepter.

Vous devez donc indiquer quels domaines sont autorisés à intégrer votre page.

Comment se prémunir du Clickjacking ?

Il y a deux méthodes permettant de se protéger contre cette attaque.

1/ Utilisez un en-tête HTTP "X-Frame-Options". Configurez votre serveur de telle sorte que la réponse de la ressource principale contienne l'en-tête HTTP "X-Frame-Options".

Trois types de valeurs peuvent être définies :

- **DENY** pour refuser toute frame ou iframe intégrant la page
- **SAMEORIGIN** pour n'autoriser que les frames provenant du même nom de domaine
- **ALLOW-FROM uri** pour préciser les domaines pouvant intégrer la page dans une frame ([non compatible avec tous les navigateurs \(EN\)](#))

2/ Définissez une directive `frame-ancestors` explicite dans votre politique de sécurité. Cette approche, plus récente et supportée par moins de navigateurs, vous permettra cependant d'autoriser plusieurs domaines plutôt qu'uniquement le domaine courant. Configurer cette directive avec la valeur 'none' revient à mettre en place l'en-tête `X-Frame-Options: DENY`.

Quelle approche choisir ? Si vous n'avez que le domaine courant à autoriser, mettez en place les deux politiques de sécurité pour une meilleure compatibilité avec les anciens navigateurs. Si vous souhaitez autoriser plusieurs domaines à intégrer le vôtre, privilégiez la directive de sécurité "frame-ancestors".

Ni l'en-tête "X-Frame-Options" ni la directive "frame-ancestors" ne sont définis sur cette page, vous êtes plus facilement exposé au clickjacking.

Les autres conseils

0/100

#2482

! Bloquez la totalité du contenu lorsqu'une attaque XSS est suspectée

Assurez-vous que le navigateur du client fasse son maximum pour prévenir d'une attaque de type XSS.

Les attaques XSS

Une attaque de type XSS (Cross-Site scripting) a pour objectif d'injecter un contenu sur la page.

Les navigateurs récents disposent de filtres intégrés pour détecter une possible attaque et prévenir l'internaute. Cette fonctionnalité peut cependant être désactivée par le client. Il est conseillé de forcer l'exécution de ces filtres. De plus, dans le cas d'une attaque XSS, il est préférable que l'utilisateur n'ait accès à aucun contenu de la page, afin de prévenir tout risque.

La solution : configurer un en-tête HTTP "X-XSS-Protection"

Ajoutez l'en-tête HTTP "X-XSS-Protection" avec "1; mode=block" pour valeur (1 pour indiquer l'activation, et mode=block pour préciser que toute la page doit être bloquée en cas de problème).

La protection XSS n'est pas activée sur cette page.

! 3 ressources tierces sont délivrées sans contrôle d'intégrité (SRI)

La page charge des contenus provenant de tiers, assurez-vous de l'intégrité de ces données.

Contrôle d'intégrité SRI

Pour vous assurer qu'une ressource provenant d'une tierce partie n'a pas été altérée, utilisez un contrôle d'intégrité SRI (SubResource Integrity). Pour cela, ajoutez l'attribut `integrity` sur les balises `<script>` et `<link>` qui chargent ces ressources. Exemple :

```
<script src="https://exemple.com/exemple-framework.js"
  integrity="sha384-oqVuAfXRKap7fdgcCY5uykM6+R9GqQ8K/uxy9rx7HNQIGY11kPzQho1wx4JwY8wC"
  crossorigin="anonymous">
</script>
```

Exemple

L'attribut `integrity` a pour valeur une empreinte, correspondant au SHA du contenu souhaité, encodé en base64. Le navigateur comparera alors ce SHA avec celui du contenu téléchargé, pour déterminer si la ressource correspond bien à celle attendue.

Plusieurs outils vous aideront dans la création de cette empreinte. En ligne de commande, vous pouvez utiliser openssl. Des outils en ligne existent par ailleurs, tels que srihash.org ou report-uri.io. [En savoir plus sur SubResource Integrity.](#)

Les ressources suivantes sont chargées par des tierces parties, ajoutez-leur un contrôle d'intégrité :

- <http://code.jquery.com/jquery-2.2.4.min.js>
- <http://code.jquery.com/ui/1.12.1/jquery-ui.js>
- <http://code.jquery.com/ui/1.10.3/themes/smoothness/jquery-ui.css>

! Désactivez la détection automatique du type des ressources

Protégez-vous de l'exploitation malveillante du MIME Sniffing.

Le MIME-Type sniffing

Les navigateurs Internet Explorer et Chrome disposent d'une fonctionnalité dite de "MIME-Type sniffing", qui consiste à détecter automatiquement le type d'une ressource web. Ainsi, une ressource déclarée comme étant une image pourra être interprétée comme un script si tel est son contenu.

Une personne malveillante pourrait profiter d'un envoi de fichier sur votre site par exemple pour injecter du code malveillant. Nous vous conseillons de désactiver le MIME-Type sniffing pour limiter les effets de l'envoi d'un tel fichier.

La solution : configurer un en-tête HTTP "X-Content-Type-Options"

Ajoutez **dans les réponses de chacune de vos ressources** l'en-tête HTTP "X-Content-Type-Options", associé à la valeur "nosniff". Cela vous permettra de vous prémunir du risque d'une mauvaise interprétation de vos ressources.

Sur cette page, **vous devriez configurer les ressources suivantes**, qui risquent d'être mal interprétées :

Ressources provenant de "todolistme"

- <http://todolistme.net/>
- http://todolistme.net/css/style_g.css
- <http://todolistme.net/javascript/lists.js>
- <http://todolistme.net/javascript/lib.js>
- http://todolistme.net/javascript/javascript_e.js

Ressources hébergées par un parti tiers

Les ressources suivantes sont hébergées par un parti tiers, il se peut donc qu'elles ne soient pas sous votre responsabilité. Vous devriez cependant considérer toute alternative possible à ces fichiers pour rester en accord avec la bonne pratique.

- <http://fonts.googleapis.com/css?family=Abel|Architects+Daughter>
- <http://code.jquery.com/ui/1.10.3/themes/smoothness/jquery-ui.css>
- <http://platform.twitter.com/widgets.js>
- <http://code.jquery.com/jquery-2.2.4.min.js>
- <http://code.jquery.com/ui/1.12.1/jquery-ui.js>
- [platform.twitter.com/widgets/widget_iframe.2e9f3\[...\].n.twitter.com%2Fsettings](http://platform.twitter.com/widgets/widget_iframe.2e9f3[...].n.twitter.com%2Fsettings)
- <https://platform.twitter.com/js/button.dd024c345fc26f7c7a8d9938b67e5d3d.js>
- [platform.twitter.com/widgets/tweet_button.2e9f36\[...\].eb8d923cba8c5b11.en.html](http://platform.twitter.com/widgets/tweet_button.2e9f36[...].eb8d923cba8c5b11.en.html)
- <https://syndication.twitter.com/settings>
- [accounts.google.com/o/oauth2/postmessageRelay?pa\[...\].OZwLw%2Fm%3D__features__](https://accounts.google.com/o/oauth2/postmessageRelay?pa[...].OZwLw%2Fm%3D__features__)

Saviez-vous que ?

i 17 ressources de cette page sont destinées à une utilisation publique

Par défaut, le navigateur n'accepte d'effectuer des requêtes AJAX, ou encore de récupérer des polices web, que sur le nom de domaine de la page. Une police de caractère définie sur toto.com ne pourra donc être utilisée que par les pages de toto.com. Cela évite une mauvaise utilisation de vos ressources par n'importe quel site.

Certaines ressources sont publiques, et veulent explicitement être mise à disposition de chacun (ex: les Google Fonts). Dans ce cas, l'en-tête HTTP `Access-Control-Allow-Origin` peut-être utilisé, avec la valeur `"*"`. Vous ne devez cependant utiliser cette propriété que si votre ressource a bien pour objectif d'être utilisée par le plus grand nombre. Sinon, nous vous conseillons de garder la valeur par défaut, ou de définir un nom de domaine spécifique dans l'en-tête `Access-Control-Allow-Origin`.

Vous devriez porter votre vigilance sur les ressources suivantes, qui utilisent un en-tête HTTP `Access-Control-Allow-Origin:*`. Assurez vous qu'elles sont effectivement destinées à être utilisées par des pages de tous horizons :

Les ressources suivantes sont hébergées par un parti tiers, il se peut donc qu'elles ne soient pas sous votre responsabilité. Vous devriez cependant considérer toute alternative possible à ces fichiers pour rester en accord avec la bonne pratique.

- <http://fonts.googleapis.com/css?family=Abel|Architects+Daughter>
- <http://code.jquery.com/ui/1.10.3/themes/smoothness/jquery-ui.css>
- http://connect.facebook.net/en_US/all.js
- <http://platform.twitter.com/widgets.js>
- <http://code.jquery.com/jquery-2.2.4.min.js>
- <http://code.jquery.com/ui/1.12.1/jquery-ui.js>
- <http://fonts.gstatic.com/s/abel/v9/MwQ5bhbm2POE2V9BPQ.woff2>
- [fonts.gstatic.com/s/architectsdaughter/v9/KtkxAK\[...\]xHZHDtgO_Y-bvTYlg4w.woff2](http://fonts.gstatic.com/s/architectsdaughter/v9/KtkxAK[...]xHZHDtgO_Y-bvTYlg4w.woff2)
- [connect.facebook.net/en_US/all.js?hash=b8b06faed\[...\]db5a9097fc&ua=modern_es6](http://connect.facebook.net/en_US/all.js?hash=b8b06faed[...]db5a9097fc&ua=modern_es6)
- [www.google-analytics.com/r/_utm.gif?utmwv=5.7.2\[...\]qBAAAAAAAAAAAAAAAAAAAAAE~](http://www.google-analytics.com/r/_utm.gif?utmwv=5.7.2[...]qBAAAAAAAAAAAAAAAAAAAAAE~)
- [platform.twitter.com/widgets/widget_iframe.2e9f3\[...\]n.twitter.com%2Fsettings](http://platform.twitter.com/widgets/widget_iframe.2e9f3[...]n.twitter.com%2Fsettings)
- <https://platform.twitter.com/js/button.dd024c345fc26f7c7a8d9938b67e5d3d.js>
- [platform.twitter.com/widgets/tweet_button.2e9f36\[...\]eb8d923cba8c5b11.en.html](http://platform.twitter.com/widgets/tweet_button.2e9f36[...]eb8d923cba8c5b11.en.html)
- [code.jquery.com/ui/1.10.3/themes/smoothness/imag\[...\]flat_75_ffffff_40x100.png](http://code.jquery.com/ui/1.10.3/themes/smoothness/imag[...]flat_75_ffffff_40x100.png)
- [code.jquery.com/ui/1.10.3/themes/smoothness/imag\[...\]icons_22222_256x240.png](http://code.jquery.com/ui/1.10.3/themes/smoothness/imag[...]icons_22222_256x240.png)
- <https://www.facebook.com/rsrsrc.php/v3/y5/r/OqOE21UvWe3.png>
- https://www.facebook.com/rsrsrc.php/v3iEpO4/y0/l/en_US/Bfavi1pHNTf.js

i 3 cookies ne sont pas sécurisés

Un cookie transmis au navigateur via le protocole HTTP doit pouvoir transiter uniquement sur une connexion sécurisée (sauf cas particulier).

Les cookies

Les cookies sont placés par le serveur web sur le navigateur web via l'en-tête HTTP **Set-Cookie**. Le navigateur retransmet ensuite ces informations lors des prochaines requêtes au serveur via l'en-tête HTTP **Cookie**. Si votre serveur fournit un cookie sur une connexion sécurisée (HTTPS), c'est probablement que ce cookie contient des informations sensibles : vous devez alors garantir qu'il ne pourra pas être exploité sur une connexion non sécurisée.

L'instruction Secure

En ajoutant l'instruction **Secure** au niveau de l'en-tête HTTP **Set-Cookie**, le serveur informe au navigateur qu'il n'est autorisé à retransmettre le cookie que sur des requêtes sécurisées. [Lire l'article dédié à la sécurisation de cookies](#) pour en savoir plus.

Attention : vérifiez que la redirection HTTP vers HTTPS est bien en place sur votre site. Dans le cas contraire, le cookie Secure pourrait ne pas être envoyé lors d'une requête HTTP.

Les cookies suivants ne sont pas sécurisés :

<https://syndication.twitter.com/settings>

- set-cookie: tfw_exp=0; Max-Age=86400; Expires=Sat, 11 May 2019 06:58:45 GMT; Path=/; Domain=.twitter.com

<https://apis.google.com/js/plusone.js>

- set-cookie: NID=183=E6rDgYWSYj1LVMrV5s_RJ1yH1uv4I3YW8xdBVFEQlhBZGwOAnANLDkD4qMOI68uBXeXisG_cAFP4yjqZOXQ9d_rFD3GIsA9QfrjQG6f8nP9yFvY73Rhyxvd4VC7Rnw1f57dULLzwWYkAVsIJ96PgYg_7Z5AjpLXpW-jodJYJE;Domain=.google.com;Path=/;Expires=Sat, 09-Nov-2019 06:58:45 GMT;HttpOnly

[googleads.g.doubleclick.net/pagead/ads?client=ca\[...\]/todolistme.net&dtd=511](googleads.g.doubleclick.net/pagead/ads?client=ca[...]/todolistme.net&dtd=511)

- set-cookie: test_cookie=CheckForPermission; expires=Fri, 10-May-2019 07:13:44 GMT; path=/; domain=.doubleclick.net

i 2 cookies peuvent être altérés par le client

Un cookie doit uniquement être manipulé par le serveur. Il n'est pas recommandé de l'utiliser au niveau du navigateur web.

Les cookies

Les cookies sont placés par le serveur web sur le navigateur web via l'en-tête HTTP **Set-Cookie**. Le navigateur retransmet ensuite ces informations lors des prochaines requêtes au serveur via l'en-tête HTTP **Cookie**.

Vous devez garantir que ces cookies ne peuvent être altérés que par le serveur.

L'instruction HttpOnly

En ajoutant l'instruction **HttpOnly** au niveau de l'en-tête HTTP **Set-Cookie**, le serveur informe le navigateur que le cookie ne peut être modifié. Le client se contente de récupérer et retourner le cookie envoyé par le serveur : il transite donc sur le protocole HTTP mais ne peut pas être modifié via du Javascript par exemple. [Lire l'article dédié à la sécurisation de cookies](#) pour en savoir plus.

Les cookies pourraient être exploités en cas de faille XSS, vous devriez considérer l'ajout de l'instruction **HttpOnly** lors du **Set-Cookie** :

<https://syndication.twitter.com/settings>

- set-cookie: tfw_exp=0; Max-Age=86400; Expires=Sat, 11 May 2019 06:58:45 GMT; Path=/; Domain=.twitter.com

[googleads.g.doubleclick.net/pagead/ads?client=ca\[...\]/todolistme.net&dtd=511](googleads.g.doubleclick.net/pagead/ads?client=ca[...]/todolistme.net&dtd=511)

- set-cookie: test_cookie=CheckForPermission; expires=Fri, 10-May-2019 07:13:44 GMT; path=/; domain=.doubleclick.net

Vos principaux axes d'optimisation

60/100

#2388

! Vous pouvez réduire certains fichiers JavaScript (minification)

En compressant votre code JavaScript, vous pouvez libérer de nombreux octets de données et réduire les délais de téléchargement, d'analyse et d'exécution.

Réduisez la taille des ressources JavaScript suivantes afin de gagner 46.0 Ko (réduction de 38%).

Les ressources suivantes sont hébergées par un parti tiers, il se peut donc qu'elles ne soient pas sous votre responsabilité. Vous devriez cependant considérer toute alternative possible à ces fichiers pour rester en accord avec la bonne pratique.

- Une réduction de la taille de code.jquery.com/ui/1.12.1/jquery-ui.js pourrait libérer 46.0 Ko (réduction de 38%).

De nombreux outils existent pour minifier des fichiers JavaScript. C'est le cas de [YUI Compressor](#) ou [JSTMin](#), recommandés par Google.

0/100

#2490

! Allégez la favicon

La favicon se doit d'être la plus légère possible.

La favicon sur le web

Cette petite image permet de représenter votre page via une petite icône, sur l'onglet du navigateur par exemple.

Que faire ?

Une favicon doit idéalement être < 10ko.

La favicon (<http://todolistme.net/favicon.ico>) dispose d'un poids trop important (17ko).

90/100

#2389

✓ Optimisez vos images sans perte de qualité

En choisissant un format approprié pour vos images et en les compressant, vous pouvez libérer de nombreux octets de données.

Optimisez les images suivantes afin de réduire leur taille de 2.4 Ko (réduction de 36%).

Les ressources suivantes sont hébergées par un parti tiers, il se peut donc qu'elles ne soient pas sous votre responsabilité. Vous devriez cependant considérer toute alternative possible à ces fichiers pour rester en accord avec la bonne pratique.

- La compression sans perte de code.jquery.com/ui/1.12.1/jquery-ui.png pourrait libérer 2.4 Ko (réduction de 36%).

Les images peuvent comporter des données inutiles à leur utilisation sur le web, ce qui peut augmenter considérablement leur poids. Des outils existent pour supprimer automatiquement ces données, sans perte de qualité, et ainsi réduire le poids de vos images.

Nous vous conseillons donc de supprimer les données inutiles grâce par exemple à [jpegtran](#) (format JPEG), [OptiPNG](#) (format PNG) ou [ImageRecycle](#).

Les autres conseils

90/100

#2436

✓ 1 image est redimensionnée côté navigateur

Une page récupérant une image plus grande que celle réellement affichée charge inutilement un volume de données trop important.

Redimensionnement d'images

Le redimensionnement d'une image côté navigateur pour en réduire la taille d'affichage n'est pas recommandé.

Fournissez directement vos images aux dimensions d'affichage utilisées sur votre site. Vous évitez ainsi l'envoi de données inutiles sur le réseau, ce qui diminue le temps de chargement de la page.

Responsive design ? Images Retina ?

Le responsive design ou encore l'utilisation d'images retina ne justifient pas un redimensionnement d'images. Même dans ces cas, différentes méthodes existent afin de délivrer vos images aux bonnes tailles. Nous vous conseillons de lire les ressources suivantes :

- [Introduction sur les images responsive](#)
- [Picturefill, pour supporter l'élément <picture> dès maintenant \(EN\)](#)
- [RICG, groupe de travail sur les images responsive \(EN\)](#)

Ne redimensionnez pas l'image suivante :

- <http://todolistme.net/images/tick.png> (taille d'affichage : 41x35)

Saviez-vous que ?

#2415

i 1 domaine envoie 2 ko de cookies

Les cookies HTTP sont utilisés pour le suivi d'un utilisateur pour personnaliser sa visite. Les cookies HTTP sont envoyés via des en-têtes HTTP par le serveur Web au navigateur. Ensuite, à chaque accès à la page par le navigateur, le cookie reçu lors de la première requête est renvoyé au serveur. Le client est alors reconnu. [En savoir plus](#).

Sur cette page, 1 nom de domaine envoie 2 ko de cookies :

Nom de domaine : todolistme

- **_utmz** : 897 octets répartis sur 13 requête(s)
- **_utma** : 689 octets répartis sur 13 requête(s)
- **_utmb** : 312 octets répartis sur 13 requête(s)
- **_utmc** : 104 octets répartis sur 13 requête(s)
- **_utmt** : 12 octets répartis sur 12 requête(s)

#2443

i Cette page charge 918 ko de données

Une page chargeant trop de données ralentit sa vitesse d'affichage, d'autant plus sur les connexions bas débit. Cela peut également générer de la frustration pour vos internautes qui sont limités par des forfaits data (voir [whatdoesmysitecost.com](#)).

Situer le poids de ma page web

En février 2016, [le poids moyen des 100 pages les plus visitées dans le monde était de 1,38Mo](#).

Comment réduire le poids de ma page ?

Vous pouvez vous reporter à notre catégorie "Volume de données" pour voir quelles sont les optimisations possibles dans votre cas. De manière générale, les images sont souvent à pointer du doigt.

Veillez aussi à construire vos pages web de manière à charger d'abord ce qui est essentiel à l'internaute en priorité (optimisation du chemin critique de rendu).

Pour tout autre contenu (plugins de réseaux sociaux, publicités, contenus situés en bas de page...), il est préférable de différer le chargement (asynchrone, lazy-loading...), pour ne pas pénaliser les contenus prioritaires.

Afin de vous aider à établir vos priorités d'optimisation, voici la répartition du poids de la page par type de ressources :

- **JavaScript** : 69,05% du poids total
- **Images** : 20,18% du poids total
- **Textes** : 6,58% du poids total
- **Polices de caractères** : 2,56% du poids total
- **CSS** : 1,47% du poids total
- **Autres** : 0,10% du poids total
- **JSON** : 0,06% du poids total

Par ailleurs, voici les 10 ressources les plus lourdes qui sont téléchargées lors du chargement de votre page :

- <http://todolistme.net/images/texture.png> (132 ko)
- <http://code.jquery.com/ui/1.12.1/jquery-ui.js> (125 ko)
- [www.facebook.com/rsrc.php/v3iEpO4/y0\[...\]/S/Bfav1pHNTf.js](http://www.facebook.com/rsrc.php/v3iEpO4/y0[...]/S/Bfav1pHNTf.js) (115 ko)
- [pagead2.googlesyndication.com/pagead\[...\]/show_ads_impl.js](http://pagead2.googlesyndication.com/pagead[...]/show_ads_impl.js) (78 ko)
- [connect.facebook.net/en_US/all.js?ha\[...\]/fc&ua=modern_es6](http://connect.facebook.net/en_US/all.js?ha[...]/fc&ua=modern_es6) (57 ko)
- [apis.google.com/_scs/apps-static/_/\[...\]/cb=gapi.loaded_0](http://apis.google.com/_scs/apps-static/_/[...]/cb=gapi.loaded_0) (50 ko)
- [apis.google.com/_scs/apps-static/_/\[...\]/cb=gapi.loaded_1](http://apis.google.com/_scs/apps-static/_/[...]/cb=gapi.loaded_1) (34 ko)
- <http://code.jquery.com/jquery-2.2.4.min.js> (30 ko)
- [www.googletagservices.com/activeview\[...\]?cb=%2Fr20100101](http://www.googletagservices.com/activeview[...]?cb=%2Fr20100101) (29 ko)
- <http://platform.twitter.com/widgets.js> (29 ko)

Bravo ! Vous respectez les bonnes pratiques suivantes

100/100

#2325

✓ **Vos contenus sont bien appelés par des URLs uniques**

Si un même contenu est référencé plusieurs fois sur la page, il est préférable d'utiliser une seule URL. Cela permet d'éviter de télécharger plusieurs fois ce contenu.

Servir le même contenu via plusieurs URLs

Si vous utilisez différentes URLs pour servir le même contenu, les navigateurs de vos visiteurs téléchargeront le contenu plusieurs fois. Ce qui implique des requêtes inutiles, du contenu dupliqué et donc un temps de chargement de votre page plus long. Si vous utilisez une seule URL, les navigateurs demanderont une seule fois le contenu au serveur. Il est également conseillé d'utiliser une seule URL pour un même contenu sur les différentes pages de votre site pour bénéficier du mécanisme de cache.

Comment résoudre le problème ?

Utilisez une seule URL pour un même contenu. Si vous avez besoin d'un contenu identique plusieurs fois, assurez vous qu'il est toujours référencé par la même URL.

100/100

#2364

✓ **Vous n'avez pas besoin de minifier vos ressources HTML**

Vous avez réduit la taille de vos ressources HTML. En savoir plus sur la [réduction de la taille des ressources HTML](#).

Vous obtenez 100/100 alors que vos ressources HTML ne sont pas minifiées ici ? Nous avons considéré que les gains apportés par la minification étaient trop faibles pour être remontés dans le rapport.

100/100

#2384

✓ **La compression des ressources est activée**

Vous avez activé la compression. En savoir plus sur l'[activation de la compression](#).

100/100

#2387

✓ **Vos ressources CSS sont minifiées**

En compressant votre code CSS, vous pouvez libérer de nombreux octets de données et réduire les délais de téléchargement et d'analyse.

[Réduisez la taille des ressources CSS](#) suivantes afin de gagner 1.4 Ko (réduction de 23%).

Les ressources suivantes sont hébergées par un parti tiers, il se peut donc qu'elles ne soient pas sous votre responsabilité. Vous devriez cependant considérer toute alternative possible à ces fichiers pour rester en accord avec la bonne pratique.

- Une réduction de la taille de [code.jquery.com/ui\[...\].css](#) pourrait libérer 1.4 Ko (réduction de 23%).

De nombreux outils existent pour minifier des fichiers CSS. C'est le cas de [YUI Compressor](#) ou [cssmin.js](#), recommandés par Google.

Si vous souhaitez en savoir plus sur [la minification CSS](#), n'hésitez pas à [consulter cet article sur le sujet](#) !

100/100

#2414

✓ **12 ressources statiques contiennent 333 octets de cookies**

Vous envoyez certaines données inutilement sur le réseau, à travers des cookies sur des ressources statiques.

Cookies et ressources statiques

Les cookies sont utilisés pour personnaliser le contenu d'une page selon le profil de l'internaute. Il est inutile d'en envoyer sur des ressources renvoyant tout le temps le même contenu (ressources statiques).

Les requêtes vers des images, des ressources JavaScript ou encore sur des styles CSS ne doivent donc pas faire appel à des cookies.

Que faire ?

N'utilisez plus de cookies sur les requêtes vers des ressources statiques. Cependant, certains noms de domaines nécessitent l'envoi de cookies sur toutes les ressources. Si possible, faites appel à d'autres domaines n'appliquant pas cette restriction. Vous pouvez par exemple configurer un nom de domaine spécifique pour servir vos ressources statiques

: static.mondomaine.com

Vous devriez arrêter de servir les ressources suivantes avec des cookies :

- [todolistme.net/images/ad\[...\]vider.png](http://todolistme.net/images/ad[...]vider.png) (69 octets de cookies)
- <http://todolistme.net/images/addlist.png> (24 octets de cookies)
- <http://todolistme.net/images/undo.png> (24 octets de cookies)
- [todolistme.net/images/to\[...\]nc_on.png](http://todolistme.net/images/to[...]nc_on.png) (24 octets de cookies)
- et 8 autres

100/100

#2421

✓ Toutes vos ressources ont un poids inférieur à 1Mo

Les fichiers trop lourds sont à proscrire sur une page web. Assurez vous que tous les éléments du fichier sont nécessaires au chargement de la page. Vous pouvez mettre en oeuvre plusieurs types d'actions :

100/100

#2446

✓ Le Lazyloading est en place (ou non requis)

Chargez en priorité les images au dessus de la ligne de flottaison, c'est à dire celles qui sont visibles sans défilement (scroll) de l'utilisateur.

Page web et images

Les images représentent en moyenne plus de 60% du poids d'une page web. En chargeant uniquement les images visibles sans scroll de l'utilisateur au chargement initial de la page, vous limitez la bande passante consommée, pour votre serveur comme pour les internautes et mobinautes. Si une image n'est visible que suite à une action de l'utilisateur comme le scroll, il est alors préférable de la charger à la demande, uniquement si elle devient nécessaire pour l'internaute (lazyloading).

Comment mettre en place le lazyloading ?

Vous pouvez vous informer sur les fonctionnalités et plugins proposés par les frameworks/CMS utilisés par votre page, qui vous proposeront probablement une solution clef en main. Il existe également des librairies Javascript qui pourront vous aider à mettre le lazyloading en oeuvre.

🌀 Par exemple, avec jQuery vous pouvez disposer d'un plugin dédié au lazyloading d'images.

100/100

#2450

✓ Cette page n'envoie pas de cookies > 100 ko

N'utilisez pas de cookies trop longs afin de minimiser leur impact sur le temps de chargement.

Les cookies HTTP sont utilisés pour le suivi d'un utilisateur pour personnaliser sa visite. Les cookies HTTP sont envoyés via des en-têtes HTTP par le serveur Web au navigateur. Ensuite, à chaque accès à la page par le navigateur, le cookie reçu lors de la première requête est renvoyé au serveur. Le client est alors reconnu. [En savoir plus.](#)

100/100

#2453

✓ Cette page n'envoie pas trop de cookies

N'utilisez pas trop de cookies pour que leur impact sur le temps de chargement soit le plus faible possible.

Les cookies HTTP sont utilisés pour le suivi d'un utilisateur pour personnaliser sa visite. Les cookies HTTP sont envoyés via des en-têtes HTTP par le serveur Web au navigateur. Ensuite, à chaque accès à la page par le navigateur, le cookie reçu lors de la première requête est renvoyé au serveur. Le client est alors reconnu. [En savoir plus](#).

Ci-dessous la liste des 3 ko de cookies utilisés sur cette page :

- **NID** : 1 ko répartis sur 6 requête(s) (nom de domaine : google)
- **_utmz** : 897 octets répartis sur 13 requête(s) (nom de domaine : todolistme)
- **_utma** : 689 octets répartis sur 13 requête(s) (nom de domaine : todolistme)
- **_utmb** : 312 octets répartis sur 13 requête(s) (nom de domaine : todolistme)
- **_utmc** : 104 octets répartis sur 13 requête(s) (nom de domaine : todolistme)
- **_utmt** : 12 octets répartis sur 12 requête(s) (nom de domaine : todolistme)

100/100

#2461

✓ 25 images utilisent le format PNG de façon adaptée

Le choix d'un format adapté pour une image permet de réduire son poids.

Le format PNG

Le format d'images PNG est destiné aux images nécessitant la fonctionnalité de transparence, ou encore aux petites images, disposant de peu de détails et de couleurs.

Préférez le format JPEG...

Le principal défaut du format PNG est de ne pas supporter la perte de qualité. En effet, un format tel que JPEG propose de "dégrader" la qualité de l'image sans que cela ne soit perceptible par l'utilisateur. Vous pouvez ainsi diminuer la qualité de l'image d'[environ 25%](#) sans que vos utilisateur ne s'en rendent compte.

Une image avec un poids important sera donc sans doute mieux compressée en utilisant le format JPEG.

...ou PNG-8

Dans le cas où votre image nécessite obligatoirement d'utiliser le mécanisme de transparence, non supporté par le format JPEG, vous devriez convertir votre image PNG "classique", en PNG-8. Ce format, basé sur une palette de 256 couleurs maximum permet de diminuer le poids de l'image sans altérer significativement sa qualité en général. Des outils tels [pngquant](#) ou encore [tinypng](#) vous aideront dans cette démarche.

Enfin, si la qualité proposée par le format PNG-8 n'est pas satisfaisante pour votre image, vous pouvez vous renseigner sur [les techniques avancées possibles \(EN\)](#) pour obtenir ce comportement sans délivrer une image au format PNG. Par exemple, il est possible de séparer votre image en 2 images JPEG, l'une contenant les données de transparences et l'autre contenant les données liées aux couleurs, et de regrouper l'image côté client à l'aide d'un élément CANVAS.

Pour en savoir plus, n'hésitez pas à consulter [cet article sur la compression des images \(EN\)](#).

Le choix du format PNG sur les images de cette page n'a pas d'impact négatif.

✓ Les réponses de redirection sont légères

Le contenu d'une redirection ne sera en aucun cas utilisé par les navigateurs web. C'est donc poids inutile qu'il convient d'éliminer.

Les redirections sur le web

Les redirections permettent d'indiquer que la ressource demandée est disponible à une autre URL. Le navigateur web va suivre automatiquement la redirection.

Les redirections peuvent être temporaire (code HTTP 302) ou permanente (301).

Ce mécanisme est par exemple utilisé sur des domaines secondaires (.fr vers .com) ou encore des mécanismes de détection de langue.

Que faire ?

Tout d'abord s'assurer que la redirection est inévitable. Si c'est le cas il est nécessaire d'alléger, voire de supprimer totalement le contenu des réponses serveur qui correspondent à des redirections.

Une redirection doit idéalement être < 1ko.

Aucune de vos redirections ne dispose d'un poids trop important.

Vos principaux axes d'optimisation

0/100

#2348

Vous devriez utiliser Universal Analytics

Depuis avril 2014, Google a mis à disposition de ses utilisateurs une nouvelle version de Google Analytics, nommée Universal Analytics. Cet outil permet entre autre de mesurer les interactions des clients à travers les différents périphériques et plates-formes utilisés.

Vous n'utilisez pas encore ces [nouvelles propriétés](#). Suivez [les instructions suivantes](#) afin de mettre à jour Google Analytics.

Notez que si besoin, vous pouvez mettre en place à la fois l'ancienne (ga.js/dc.js) et la nouvelle version (analytics.js) de Google Analytics, celles-ci pouvant très bien cohabiter.

Bravo ! Vous respectez les bonnes pratiques suivantes

100/100

#2359

✓ **Les polices de caractères de Google sont chargées en une seule fois**

L'API de Google Font permet de minimiser le nombre de requêtes pour récupérer les polices de caractères que vous souhaitez.

Les polices de caractères avec Google Font

Cette page utilise des polices de caractères mises à disposition par Google afin d'améliorer son rendu.

Comment les récupérer ?

Il est possible de charger les polices caractères de Google en une seule requête. Par exemple, si votre code ressemble à ceci :

```
<link rel="stylesheet" type="text/css" href="http://fonts.googleapis.com/css?family=Droid+Sans:normal,italic">
<link rel="stylesheet" type="text/css" href="http://fonts.googleapis.com/css?family=Lobster:bold,bolditalic">
```

Exemple

Il est préférable de le remplacer par la forme suivante, qui récupèrera le même contenu en une seule fois :

```
<link rel="stylesheet" type="text/css" href="http://fonts.googleapis.com/css?family=Droid+Sans:normal,italic|Lobster:bold,bolditalic">
```

Exemple

Dans votre cas, la bonne pratique est respectée :

- <http://fonts.googleapis.com/css?family=Abel|Architects+Daughter>

100/100

#2360

✓ **Vos polices de caractères sont chargées rapidement**

Internet Explorer empêche l'affichage de la page tant que la police de caractères n'est pas chargée. Si une balise `<script>` est placée avant la police de caractères, la page n'affichera rien tant que les scripts et les polices ne sont pas chargées.

Félicitations, l'appel à vos polices de caractères est placé avant vos scripts.

Vos principaux axes d'optimisation

0/100

#2555

! Envisagez d'utiliser jQuery 1.12

Votre page web utilise jQuery 1.9.1. Vous devriez migrer vers la dernière version de la branche 1.x : jQuery 1.12, qui contient plusieurs correctifs de bugs et de sécurité.

Dois-je migrer vers la dernière version de jQuery ?

La migration d'une version 1. x de jQuery vers la dernière version (3.x) peut avoir de nombreux impacts inattendus et implique une perte de compatibilité avec les navigateurs plus anciens. Vous ne devriez envisager d'abandonner jQuery 1.x que dans le cadre d'une refonte complète de votre infrastructure Front-End. Lors de la réflexion préalable à cette refonte, vous découvrirez peut-être que [vous n'avez pas besoin de jQuery](#).

Saviez-vous que ?

#2303

i En savoir plus sur la performance avec jQuery

jQuery est la librairie JavaScript la plus utilisée aujourd'hui. Rendez votre site plus performant en respectant ses bonnes pratiques. Pour connaître les fondamentaux de la performance jQuery, nous vous conseillons de lire le lien suivant : <http://learn.jquery.com/performance/>.

Bravo ! Vous respectez les bonnes pratiques suivantes

100/100

#2299

✓ Aucune manipulation de DOM dans des boucles détectée

Travailler avec le DOM a un coût. Si vous avez besoin d'ajouter des éléments à un nœud, il est préférable de les ajouter tous d'un coup plutôt qu'un par un.

Aucune de vos boucles ne contient de méthode jQuery de modification du DOM.

100/100

#2302

✓ Vos sélecteurs jQuery ne sont pas trop spécifiques

L'utilisation de sélecteurs jQuery trop spécifiques diminue les performances de votre site web : [en savoir plus](#). Voici un exemple de bonne utilisation de jQuery:

```
$( ".data table.firstClass td.secondClass" );  
// Plus performant: supprimer le sélecteur  
// intermédiaire si possible  
$( ".data td.secondClass" );
```

Exemple

Nous n'avons pas trouvé de sélecteurs trop spécifiques sur cette page.

100/100

#2470

✓ Une seule version de jQuery est chargée

Charger plusieurs fois jQuery revient à alourdir le poids de la page et complexifie sa maintenabilité.

Une seule version de jQuery a été détectée, la page est en accord avec la bonne pratique.