**Author:** Eric Nicosia

**Version:** 21

**Date:** 2014.09.02

**Pôle Architecture technique**

**Audit Ginger Web (Septembre 2014)**

0

Table of ContenTs

[1 Introduction 3](#_Toc256000000)

[2 Points relevés 4](#_Toc256000001)

[2.1 Librairies 4](#_Toc256000002)

[2.2 Utilisation de la compression Gzip 4](#_Toc256000003)

[2.2.1 Comment activer la compression Gzip avec Apache 5](#_Toc256000004)

[2.3 Pages HTML mal formées 6](#_Toc256000005)

[2.4 Gestion des cookies 6](#_Toc256000006)

[2.5 Directory listing 7](#_Toc256000007)

[2.5.1 Désactiver le directory listing sous Apache 7](#_Toc256000008)

[2.6 Malformed URL's 7](#_Toc256000009)

[2.7 Taille des ressources de l'application 8](#_Toc256000010)

[2.8 Cache de requêtes JSON interne à l'application 8](#_Toc256000011)

[2.9 Comportement des écrans 9](#_Toc256000012)

[2.10 Mieux gérer le cache navigateur (A partir de la config Apache) 9](#_Toc256000013)

[2.10.1 Configurer la gestion du cache client dans Apache 9](#_Toc256000014)

[2.11 Améliorer le rendu visuel des cartes à l'écran 11](#_Toc256000015)

* [Introduction](#scroll-bookmark-1)
* [Points relevés](#scroll-bookmark-2)
  + [Librairies](#scroll-bookmark-3)
  + [Utilisation de la compression Gzip](#scroll-bookmark-4)
    - [Comment activer la compression Gzip avec Apache](#scroll-bookmark-5)
  + [Pages HTML mal formées](#scroll-bookmark-6)
  + [Gestion des cookies](#scroll-bookmark-7)
  + [Directory listing](#scroll-bookmark-8)
    - [Désactiver le directory listing sous Apache](#scroll-bookmark-9)
  + [Malformed URL's](#scroll-bookmark-10)
  + [Taille des ressources de l'application](#scroll-bookmark-11)
  + [Cache de requêtes JSON interne à l'application](#scroll-bookmark-12)
  + [Comportement des écrans](#scroll-bookmark-13)
  + [Mieux gérer le cache navigateur (A partir de la config Apache)](#scroll-bookmark-14)
    - [Configurer la gestion du cache client dans Apache](#scroll-bookmark-15)
      * [Explications :](#scroll-bookmark-16)
  + [Améliorer le rendu visuel des cartes à l'écran](#scroll-bookmark-17)

Introduction

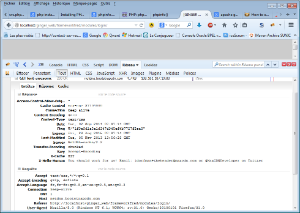
L'objectif de cet audit est de s'assurer de l'utilisation des bonnes pratiques en matière de développement Web, en terme de performance, sécurité et bonnes pratiques sur l'application Ginger Web

Points relevés

Librairies

Une des libraires utilisées par l'application "Téléphone à la maison".

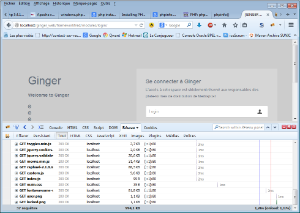
La copie d'écran suivante montre qu'une des librairies utilisée par Ginger va chercher une ressource (en l’occurrence un CSS) sur un site web externe en lui fournissant toutes sortes de données. Si cela n'est pas gênant en développement, ce n'est pas acceptable en production et il faudra soit désactiver ce comportement ou remplacer la librairie si la désactivation n'est pas possible.



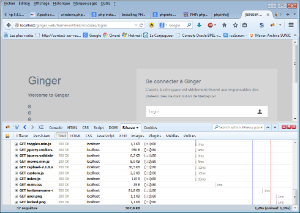
Utilisation de la compression Gzip

Par défaut sur Apache, aucune compression n'est appliquée aux fichiers et pages transmises sur le réseau. Sur le terrain, la bande passante réseau est limitée et il est souhaitable de réduire le trafic réseau aussi bien pour éviter la saturation que pour améliorer le temps d'affichage des pages. Ci-après deux copies d'écran Ginger, la première sans compression, la deuxième avec. On constate que le volume de données qui a transité sur le réseau a été divisé par 3 !  
Le temps d'affichage de la page ne varie quasiment pas entre les deux options car le test est réalisée sur un réseau Gigabit. Sur le terrain avec un réseau à 512 Ko, l'amélioration du temps d'affichage de la page sera importante.

Sans compression



Avec compression



|  |
| --- |
| Ginger fait une utilisation intensive du JSON pour échanger des données avec le serveur. Ces données sont volumineuses et peuvent dépasser le Méga. Il est donc important de s'assurer que la compression est active pour le type mime application/json |

|  |
| --- |
| La compression des données peut aussi être activée au niveau du routeur NAT / Firewall par l'administration réseau, ce qui déchargera le serveur Apache de cette tâche. Voir avec D.T. la solution à adopter |

### Comment activer la compression Gzip avec Apache

Il faut éditer le fichier httpd.conf d'apache et réaliser la configuration suivante :

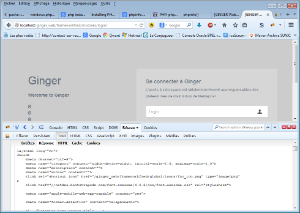
* Activer les modules mod\_deflate, mod\_filter et mod\_setenvif
* Ajouter la configuration suivante (par exemple après la section des modules)

|  |
| --- |
| <IfModule mod\_mime.c>  AddType application/x-javascript .js  AddType text/css .css </IfModule> <IfModule mod\_deflate.c>  AddOutputFilterByType DEFLATE text/css application/x-javascript text/x-component text/html text/richtext image/svg+xml text/plain text/xsd text/xsl text/xml image/x-icon application/javascript application/json  <IfModule mod\_setenvif.c>  BrowserMatch ^Mozilla/4 gzip-only-text/html  BrowserMatch ^Mozilla/4\.0[678] no-gzip  BrowserMatch \bMSIE !no-gzip !gzip-only-text/html  </IfModule>  <IfModule mod\_headers.c>  Header append Vary User-Agent env=!dont-vary  </IfModule> </IfModule> |

Pages HTML mal formées

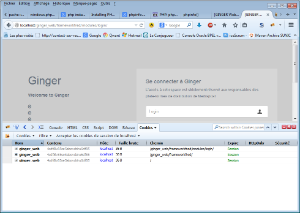
Certaines pages html de Ginger sont mal formées et ne respectent pas le standard W3C pour les pages HTML, ce qui peut poser problème pour le chargement et et l'affichage de ces pages.

Par exemple, la page de connexion de Ginger contient des caractères spéciaux en tout début de page, avant même le tag HTML



Gestion des cookies

L'application Ginger ne devrait avoir qu'un seul cookie de session actif pour un client donné. Hors, le simple fait d'afficher la page de login créé 3 cookies identiques, pour des chemins différents, comme le montre la copie d'écran ci-après.



Directory listing

Il est possible par défaut d'afficher le contenu des répertoires avec Apache. Si cela peut être utile en développement, il faut désactiver cette fonctionnalité en production pour des raisons évidentes de sécurité.

### Désactiver le directory listing sous Apache

Il faut éditer le fichier de configuration httpd.conf et supprimer l'option Indexes pour le répertoire racine du serveur

Options **Indexes** FollowSymLinks

Malformed URL's

Certaines URL de l'application sont malformées. Par exemple, lorsqu'on passe de la page d'accueil à la représentation d'une distribution, on obtient l'URL suivante :

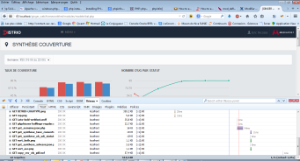
|  |
| --- |
| http://localhost/ginger\_web/frameworkfred/modules/modele/ponctuels.php?&device=338508&sem\_id=26&day=2 |

La séquence ?&device n'est pas correcte et il ne devrait pas y avoir de & devant le premier paramètre

Taille des ressources de l'application

Certaines ressource de Ginger ne sont pas adaptées.

Par exemple, le logo Distrio utilisé a un poids de 350 Ko et une taille réelle de 8970x2620 pixel ! A lui seul, c'est plus de 50% du poids de la page, alors qu'il est représenté avec une hauteur d'à peine 25 pixel.



|  |
| --- |
| Pour les ressources graphiques et les images, il est préférable de pré-générer les images à la bonne échelle pour que le navigateur puisse les afficher directement, plutôt que d'utiliser des images en HD qui seront ensuite redimensionnées par le navigateur, ce qui est couteux en bande passante réseau et en CPU sur le poste client. |

Cache de requêtes JSON interne à l'application

Ginger embarque un mécanisme de cache pour les requêtes retournant des données JSON afin d'améliorer les temps de réponse. Les fichiers de données générés sont mis dans le répertoire ..\ginger\_web\frameworkfred\cache  
Problème : ce cache n'est jamais purgé et ne fait que grossir, ce qui posera rapidement des problèmes de place en production. D'autre part, si les données en central sont mises à jours et que les données de cache n'expirent jamais, les nouvelles données ne seront jamais visibles.

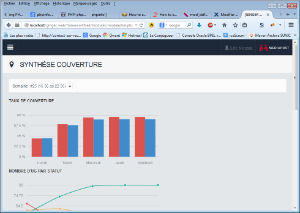
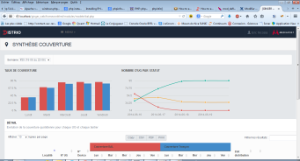
Un mécanisme de purge doit être mis en place, de préférence interne à l'application Ginger pour éviter une maintenance supplémentaire à la charge de l'exploitation.

Je préconise comme solution les points suivants :

* Au démarrage du cache, purge de tous les fichiers existants
* Utilisation d'un mécanisme d’obsolescence des fichiers basés sur l'ancienneté des fichiers et leurs nombres, pour purger les fichier les plus ancien lorsque leur nombre (ou leur volume) devient trop important.

Comportement des écrans

Un même écran Ginger, en fonction de la résolution d'écran choisie, n'affiche pas les mêmes infos et n'a pas le même comportement. Exemple avec l'écran de restitution des semaines, qui fait soit apparaitre le logo Distrio, soit un menu de navigation dans le coin supérieur gauche de l'écran, comme le montre les 2 copies d'écran ci-après :



Mieux gérer le cache navigateur (A partir de la config Apache)

Par défaut, Apache utilise le Tag "ETag" pour gérer la mise en cache des objets et réponses HTTP. Cette méthode n'est pas optimale et est connue pour poser des problèmes de performances. La plupart des optimiseurs de page Web comme Yslow préconisent de le remplacer par le tag "Expires". Cela permet d'avoir le contrôle sur la durée de mise en cache des éléments et sur les types d'éléments mis en cache.

### Configurer la gestion du cache client dans Apache

La configuration sera faite dans le fichier httpd.conf

|  |
| --- |
| Le module mod\_headers doit avoir été activé au préalable |

Ajouter la configuration suivante :

|  |
| --- |
| # config de la gestion du cache navigateur client Header unset Pragma FileETag None Header unset ETag <FilesMatch "\.(ico|jpg|jpeg|png|gif|js|css|swf|pdf|flv|mp3)$"> <IfModule mod\_expires.c> ExpiresActive on ExpiresDefault "access plus 14 days" Header set Cache-Control "public" </IfModule> </FilesMatch> <FilesMatch "\.(html|htm|xml|txt|xsl)$"> Header set Cache-Control "max-age=7200, must-revalidate" </FilesMatch> |

#### Explications :

Les trois premières lignes suppriment les en-tête HTML ETag et pragma, puisque nous allons les remplacer par le tag expires

Pour les fichiers dont l’extension est .ico, .jpg, ..., .mp3

* on active le le tag Expires
* On positionne la date d'expiration à la première date d'acces plus 14 jours
* On positionne le cache-control à "Public"

Pour les fichiers dont l'extention est html, htm, ..., xsl

* le tag Cache-Control est positionné à "max-age=7200, must-revalidate", c'est à dire que le cache expirera au bout de 2 heures ou si la page à été modifiée entre temps (=must-revalidate)

|  |
| --- |
| Il faut tester cette configartion, et completer les types de fichier à cacher en fonction des ressources spécifiques de Ginger, comme par exemple les polices de caractères dont l'extention est .woff |

}

Améliorer le rendu visuel des cartes à l'écran

OpenLayers est entièrement configurable par css. Par défaut, la librairie va charger un fichier de style dans le répertoire relatif (à celui ou elle est chargée...) **theme/default/style.css**

Pour améliorer le rendu visuel lors du chargement des tuiles des fonds cartographiques, on peut ajouter le paramétrage css suivant dans le fichier :

|  |
| --- |
| .olLayerGrid .olTileImage {  -webkit-transition: opacity 0.2s linear;  -moz-transition: opacity 0.2s linear;  -o-transition: opacity 0.2s linear;  transition: opacity 0.2s linear; } |

Cette configuration indique à OpenLayers d'utiliser un effet de fondu / transparence sur les tuiles déjà présentes durant le chargement des nouvelles tuiles. Le changement de niveau de zoom est de fait beaucoup plus agréable.

|  |
| --- |
| Cet effet ne sera actif que si la couche a comme attribut transitionEffect : 'resize' |