

## Momo - Bug #10840

### mymeaux : non prise en compte d'une modif à un fichier css

04 mai 2016 17:37 - Frédéric Péters

<b>Statut:</b>	Rejeté	<b>Début:</b>	04 mai 2016
<b>Priorité:</b>	Normal	<b>Echéance:</b>	
<b>Assigné à:</b>		<b>% réalisé:</b>	0%
<b>Catégorie:</b>		<b>Temps estimé:</b>	0:00 heure
<b>Version cible:</b>		<b>Planning:</b>	
<b>Patch proposed:</b>	Oui		
<b>Description</b>			
il y a assets/meaux.css, il est dans un nouvel assets.zip qui est bien téléchargé, mais dans l'application, on n'en voit pas d'effet.			
<b>Demandes liées:</b>			
Lié à Momo - Bug #9972: non-prise en compte de la CSS distante lors de la mis...		<b>Fermé</b>	<b>12 février 2016</b>

### Historique

#### #1 - 04 mai 2016 18:03 - Frédéric Péters

Très simplement les assets sont bien téléchargés mais à l'URL assets/ c'est les données "internes" qui se trouvent, pas les données téléchargées; il faudrait faire un truc équivalent au patchResponse (modif des contenus pour pointer les media vers le filesystem).

Il me semble :

- modifier le refreshAssets pour s'exécuter dans le callback d'un window.requestFileSystem, pour avoir accès au fs
- (en passant je note qu'on fait LocalFileSystem.TEMPORARY, il me semble qu'on devrait plutôt faire LocalFileSystem.PERSISTENT)
- dans la boucle "Refresh stylesheets", pour les différents css, vérifier s'il existe en local (<http://stackoverflow.com/questions/10294166/how-to-check-a-files-existence-in-phone-directory-with-phonegap>), dans un callback (toujours des callbacks...), si ça a marché prendre la version du filesystem, sinon celle assets/foobar.css.
- il y a une vérification faite sur el.href === file, qui ne va plus marcher, il faudra taper sur le <link> un attribut supplémentaire, qui ne contiendra pas tout le path.

#### #2 - 05 mai 2016 10:48 - Frédéric Péters

Non, c'est encore plus simple ce qu'il faut faire.

À l'initialisation de l'application on récupère un handle sur le file system (c'est déjà le cas et on s'en sert juste pour faire app.rootPath = fileSystem.root.toURL(); on pourrait plutôt simplement conserver l'objet fileSystem). Si sur le filesystem il n'y a pas déjà de assets.zip décompressé, on l'y décompresse (ça veut dire qu'on shippe dans l'application un fichier assets.zip plutôt que le répertoire assets/). À partir de là, tout le temps, partout, pour les css mais aussi pour les fichiers js, on fait référence aux fichiers décompressés; on n'a pas à chaque fois se poser la question d'où regarder.

Sans doute a-t-on besoin d'ajouter un timestamp formel quelque part, pour déterminer si à l'installation d'une nouvelle version de l'application il faut écraser les assets existants par ceux distribués dans l'application. (je trouve que de toute façon ça aiderait beaucoup de ne pas juste après l'installation de l'application afficher un message disant que des mises à jour sont dispo).

Pour continuer à fonctionner hors device, si window.requestFileSystem est vide, le window.appRoot il continue à être initialisé à DEBUG\_WWW\_URL mais plutôt qu'être mise à <http://localhost/~ghis/momo/www/>, elle est initialisée sur window.location.origin + window.location.pathname.

@@ -21,7 +21,7 @@

```
// Constants
var DEBUG = false;
-var DEBUG_WWW_URL = 'http://localhost/~ghis/momo/www/';
+var DEBUG_WWW_URL = window.location.origin + window.location.pathname;
var ANIMATION_ENABLED = true;
var ANIMATION_OUT_CLASS = 'pt-page-moveToLeftEasing pt-page-ontop';
var ANIMATION_IN_CLASS = 'pt-page-moveFromRight';
@@ -622,7 +623,7 @@ var app = {
```

```
// Refresh scripts
for(i = 0, l = page.javascripts.length; i < l; i++){
-   file = page.javascripts[i];
+   file = app.rootPath + page.javascripts[i];
   found = false;
   els = document.getElementsByTagName("script");
   for (j = 0, m = els.length; j < m; j++) {
```

```
@@ -646,7 +647,7 @@ var app = {
    // Refresh stylesheets
    for(i = 0, l = page.stylesheets.length; i < l; i++){
-     file = page.stylesheets[i];
+     file = app.rootPath + page.stylesheets[i];
        found = false;
        els = document.getElementsByTagName("link");
        for (j = 0, m = els.length; j < m; j++) {
```

### #3 - 05 mai 2016 18:54 - Frédéric Péters

- Fichier 0001-always-load-assets-from-cache-10840.patch ajouté
- Statut changé de Nouveau à En cours
- Patch proposed changé de Non à Oui

Voilà une version du patch, sans timestamp supplémentaire (i.e. une mise à jour de l'application ne va pas copier ses nouveaux assets). (testée sur Android)

À noter qu'il y aurait à nettoyer le répertoire www/ qui se trouve inclus dans l'application (particulièrement virer le assets.zip, qui n'est pas à jour) et améliorer le système de build pour que cet assets.zip soit construit automatiquement. Idéalement il faudrait aussi pouvoir exclure des fichiers de l'application générée (pour ne pas inclure le répertoire assets/).

### #4 - 05 mai 2016 20:17 - Frédéric Péters

- Lié à Bug #9972: non-prise en compte de la CSS distante lors de la mise à jour ajouté

### #5 - 05 mai 2016 20:24 - Frédéric Péters

- Fichier 0001-always-load-assets-from-cache-10840.patch ajouté

Patch mis à jour pour retirer un alert() de debug.

### #6 - 04 mars 2019 10:43 - Frédéric Péters

- Statut changé de En cours à Rejeté

## Fichiers

0001-always-load-assets-from-cache-10840.patch	6,58 ko	05 mai 2016	Frédéric Péters
0001-always-load-assets-from-cache-10840.patch	6,5 ko	05 mai 2016	Frédéric Péters