

Référenciel de format de données pour FX.php

L'intention de ce document est de présenter de quelle manière FX.php retourne les données. Le Web Companion de FileMaker fait transiter une grande variété de données et FX.php est conçu pour restituer toute cette diversité. Cependant, cette large gamme de données transitées rend parfois difficile à placer exactement ce qu'il faudrait pour le résultat escompté.

Pour comprendre plus facilement ce document, vous pouvez consulter ceux nommés "DatasetDiagram.pdf" et "displayFXDataset.php": ils sont inclus au dossier de la dernière version de FX.php.

(trois autres dossiers de displayFX sont aussi inclus dans cette version, pour traiter de domaines particuliers: une liste de base de données (un ensemble de fichiers filemaker), une liste de modèles dans une base de données (dans un fichier uniquement), et une présentation d'un modèle spécifique. Il est possible que vous ayez à modifier l'adresse IP et le numéro du port indiqués dans les dossiers pour les adapter à votre configuration individuelle de fonctionnement FileMaker Unlimited.)

Si vous n'avez pas ces dossiers, ils peuvent être téléchargés par <http://www.iviking.org/FX.php> ou demandés par email à l'auteur de ce document, Chris Hansen, à l'adresse FX@iviking.org.

Important: La première ligne en caractères gras sous chaque titre contient l'échantillon des données retournées associées étudié (uses/syntax) dans le paragraphe.

Conventions utilisées:

- + \$ReturnedData est employé pour indiquer la rangée contenant les données retournées par une requête. Des exemples de syntaxe pour construire \$ReturnedData, se trouvent dans "dernières fonctions" ("Last Functions") au sein du document nommé "explication_fonction.txt".
- + Lines, là où les descriptions de paramètre commencent et sont indiquées par un signe du genre ">>".
- + Les signes // indiquent le début de commentaires pour les pages php.

```
_____ $ReturnedData['linkNext']  
if (strlen($ReturnedData['linkNext']) < 1) {  
echo "<span class=\"medgraytext\">Next</span>\n";  
} else {  
echo '<a href="'. $ReturnedData['linkNext']. "\" class=\"tbody\">Next</a>\n'; }  
_____
```

La section '**linkNext**' des données retournées contient un lien de préformage pour une sélection de fiches suivantes, dans un ensemble FileMaker assemblé selon la requête, le compte de saut, et la taille courants de la sélection. Si toutes les fiches ou le dernier groupe de fiches de la sélection courante trouvée sont actuellement affichées, '**linkNext**' sera vide. Le code ci-dessus montre le lien suivant s'il est présent, ou le terme grisé **Next** dans le cas contraire.

En programmation avancée, les paramètres spéciaux '- foundSetParams_end' et '- foundSetParams_begin' peuvent être employés pour empêcher que des parties de la requête courante soient incluses dans les liens stockés en 'linkNext' et 'linkPrevious'.

Syntaxe d'utilisation et exemples de ces paramètres peuvent être trouvés dans FXExamples.php et le dernier document de fonctions de FX.

```
_____ $ReturnedData['linkPrevious']  
if (strlen($ReturnedData['linkPrevious']) < 1) {  
echo "<span class=\"medgraytext\">Prev</span>\n";  
} else {  
echo '<a href="'. $ReturnedData['linkPrevious']. "\" class=\"tbody\">Prev</a>\n'; }  
_____
```

La section '**linkPrevious**' des données retournées contient un lien de préformage pour une sélection de fiches précédentes, dans un ensemble FileMaker assemblé selon la requête, le compte de saut, et la taille courants de la sélection. Si toutes les fiches ou le premier groupe de fiches de la sélection courante trouvée sont actuellement affichées, '**linkPrevious**' sera vide. Le code ci-dessus montre le lien précédent s'il est présent, ou le terme grisé **Prev** dans le cas contraire.

En programmation avancée, les paramètres spéciaux '- foundSetParams_end' et '- foundSetParams_begin' peuvent être employés pour empêcher que des parties de la requête courante soient incluses dans les liens stockés en 'linkNext' et 'linkPrevious'.

Syntaxe d'utilisation et exemples de ces paramètres peuvent être trouvés dans FXExamples.php et le dernier document de fonctions de FX.

```
____$ReturnedData['foundCount']_____  
echo $ReturnedData['foundCount'] . " Fiches répondant à vos critères de recherche.";
```

Le **'foundCount'** contient le nombre total de fiches dans FileMaker répondant aux critères de recherche de la dernière requête (courante).

```
____$ReturnedData['fields']_____  
$counter = 0;  
foreach ($ReturnedData['fields'] as $key => $value) {  
if ($counter % 2 == 0) {  
$rowColor = '#99CC99';  
} else {  
$rowColor = '#99CCCC';  
}  
echo "<tr bgcolor=\"\$rowColor\">\n";  
echo "<td align=\"left\" class=\"body\">' . $value['name'] . "</td>\n";  
echo "<td align=\"center\" class=\"body\">' . $value['type'] . "</td>\n";  
echo "<td align=\"center\" class=\"body\">' . $value['emptyok'] . "</td>\n";  
echo "<td align=\"center\" class=\"body\">' . $value['maxrepeat'] . "</td>\n";  
echo "</tr>\n";  
++$counter;  
}
```

Dans la section **'fields'** sont stockées des informations sur les attributs des champs présents sur le modèle courant. (de l'information complémentaire est présente dans la section **'data'** des requêtes de **'-view'**). FileMaker Pro fournit de l'information sur quatre attributs de chaque champ: son nom, son type (par exemple multimédia, texte, nombre, etc.), s'il est vide, et le nombre maximum de valeurs pour un champ multivalué.

L'attribut **'name'** est simplement le nom qui a été donné au champ (rubrique) quand il a été créé dans FileMaker. Si le champ présent sur le modèle courant appartient à un fichier lié, le nom du lien le précédera comme ceci: `nom_du_lien::nom_du_champs`.

Attention: vous rencontrerez moins de problème si vous n'utilisez pas d'espaces dans le nommage des champs (rubriques). L'underscore (caractère de soulignement) – ex: `nom_de_rubrique --` ou la technique "camel caps" – `- NomDeMaRubrique --` sont des alternatives préférables à l'emploi des espaces.

Connaître le **'type'** d'un champ peut grandement faciliter le retour correct des données.

Par exemple, dans *FXExamples.php*, des contrôles de type de rubriques sont employés de sorte que des rubriques multimédia (qui produisent des URLs, et non l'image réelle) peuvent être correctement utilisées.

La valeur **'emptyok'** identifie simplement si un champ est vide, et s'emploie pour insérer dynamiquement un contrôle Javascript pour les champs qui ne peuvent pas être laissés vides.

Le dernier attribut, **'maxrepeat'**, a pour valeur "1" pour tous les champs, sauf pour les multivalués.

Bienqu'un portail puisse contenir de multiples instances d'un champ, l'attribut **'maxrepeat'** sera à "1", sauf dans le cas d'un champ multivalué. (en cette matière, je ne suis pas tout à fait sûr que FX.php, ou même le Web Companion, puisse utiliser correctement un champ multivalué dans un portail).

Dans le cas de champ multivalué, l'attribut **'maxrepeat'** contiendra le nombre de valeurs définies pour un champ spécifique, et non le nombre de valeurs affichées sur le modèle courant.

Malheureusement, au moment de cette rédaction, FileMaker Pro ne renvoie pas de données indiquant si un champ spécifique est réellement modifiable. Ceci ne devrait pas être un problème pour une opération sur une base dont les données sont bien spécifiées et connues, mais *FXExamples.php* est donc contraint de lancer une inscription sur chaque champ individuel pour le tester.

_____ \$ReturnedData['data'] _____

'data' est l'élément de la rangée retournée de données qui effectue la majeure partie du travail. Cette section se divise en trois niveaux secondaires de rangées, et contient toutes les données stockées dans les fiches actuellement retournées. Pour une aide complémentaire à visualiser comment les données sont stockées, vous pouvez consulter les documents référencés dans le premier paragraphe ci-dessus. *displayFXDataset.php* devrait être particulièrement utile.

```
foreach ($ReturnedData['data'] as $key => $value) {
if ($counter % 2 == 0) {
$tempBGColor = '#FFFFFF';
} else {
$tempBGColor = '#CCCCCC';
}
$counter2 = 0;
$IDsArray = explode('.', $key);
```

La première des rangées secondaires contient un élément pour chaque fiche retournée. Dans la plupart des cas, l'index pour cette section est l'identifiant (interne à FileMaker) de la fiche, une virgule, et ensuite l'identifiant de la modification (interne à FileMaker).

Il est nécessaire d'extraire l'identifiant d'une fiche particulière afin de pouvoir éditer cette fiche. Dans l'exemple ci-dessus, j'emploie la fonction **explode()** de php pour séparer les éléments de l'index. On peut alors accéder à l'identifiant de la fiche, par exemple, en employant **\$IDsArray[0]**

Les cas spéciaux comme les listes de base de données, les listes de modèles, ou l'information sur un modèle utilisent un zéro, une virgule, et en index, un nombre simple.

```
print_r($value['Object_Components']);
```

Dans la seconde sous-rangée de 'data' (celle du milieu), ce sont les noms de champs qui sont employés comme index. L'exemple précédent -- qui est aussi commenté dans *FXExamples.php* -- montrera tout le contenu du champ Object_Components, qui a quatre répétitions, pour la fiche courante.

(employer **print_r()** plutôt que **echo()** est important puisque la production de données par *FX.php* pour un champ entier -- même un champ non-multivalué -- est toujours une rangée.

Employer **echo()** montrerait simplement le mot "rangée" dans ce cas-ci).

Évidemment, montrer le contenu d'un champ de cette façon est d'utilité limitée. Le niveau le plus bas d'une sous rangée est habituellement la meilleure manière de montrer le contenu d'un champ.

```
echo($value['Object_Components'][0]);
```

Au niveau le plus bas, les données pour chaque itération d'un champ emploient des nombres (à partir de zéro) en index. Ceci signifie que, excepté dans le traitement des champs multivalués ou des portails, ces champs sont consultés avec un post-marquage "[0]" comme ci-dessus.

(comme dans l'exemple précédent, cette ligne de code est aussi commentée dans *FXExamples.php*).

Dans le cas de champs multivalués et de portails, je constate que la construction du **foreach()** de php est la manière la plus facile de parcourir les valeurs; veuillez consulter *FXExamples.php* pour un exemple détaillé sur la façon dont ceci est réalisé.

_____ \$ReturnedData['URL'] _____

```
echo $ReturnedData['URL'];
```

Dans la plupart des cas, l'URL stocké dans cette section peut être employé comme voie d'accès alternative au même FileMaker Pro XML actuellement parsé (analysé syntaxiquement) par *FX.php*.

Puisqu'une requête de type GET a une longueur maximum limitée (selon le logiciel de serveur et le browser en service), des questions complexes peuvent en effet être malencontreusement tronquées.

De plus, cet URL peut être très utile pour le débogage.

(astuce: accéder à XML par cette voie est la seule tâche pour laquelle je préfère employer Internet Explorer)..

```
____$ReturnedData['errorCode']_____  
echo $ReturnedData['errorCode'];
```

L'élément 'errorCode' de la rangée retournée de données contient tout code d'erreur qui a pu être retourné par FileMaker Pro. Un zéro indique "aucune erreur"; pour la signification des autres codes d'erreur, consultez la documentation de FileMaker Pro, ou consultez FMErrors.php -- un fichier utile inclus dans la dernière version de FX.php (merci à Gjermund Thorsen pour ce travail).

(*FXExamples.php* montre comment intégrer ce fichier dans une solution).

```
____$ReturnedData['valueLists']_____  
print_r($ReturnedData['valueLists']);  
print_r($ReturnedData['valueLists']['myValueList']);  
echo($ReturnedData['valueLists']['myValueList'][0]);
```

Lorsqu'une action de FMView() est appelée, toutes listes de valeurs sur le modèle indiqué seront retournées dans cet élément de la rangée retournée de données. Veuillez noter que cet élément est seulement retourné dans une action de type FMView(); vous pourriez souhaiter exécuter un FMView() séparé pour extraire les listes de valeurs et en disposer dans votre modèle courant de recherche.

La section 'valueLists' contient deux sous-rangées (rangées secondaires). La première sous-rangée utilise le nom de la liste de valeur en tant qu'index, alors que la seconde sous-rangée est numérotée à partir de zéro. Une utilisation typique serait de vouloir se servir de la fonction php: foreach(). Reportez-vous à *FXExamples.php* pour une démonstration complète de l'utilisation du contenu de cet élément.

____Informations complémentaires _____

Pour plus d'informations sur FX.php (incluant la dernière version anglaise):

<http://www.iviking.org/>

Pour plus d'informations sur FileMaker et le XML:

http://www.filemaker.com/downloads/pdf/xml_Chapter_7.pdf

Pour plus d'informations sur les paramètres acceptés par FileMaker et le Web Companion :

http://www.filemaker.com/downloads/pdf/xml_Appendix_B.pdf

Pour plus d'informations sur les codes d'erreurs renvoyés par FileMaker et le Web Companion:

http://www.filemaker.com/downloads/pdf/xml_Appendix_C.pdf

En conclusion, il est impossible de couvrir ici toutes les possibilités qu'offre PHP.

Heureusement, Internet est une source riche d'informations, entre autres, le manuel en ligne de PHP qui est excellent. On peut le trouver à: <http://www.php.net/manual/fr/>

____Notes complémentaires _____

Les documents relatifs à FX.php (version française) sont disponibles à:

<http://le-nomade.com/francais2/fx.php>

<http://www.iviking.org/>

<http://www.file-making.com/files/>